Gerrymandermaxxing

CSDS 133

Trevor Nichols, Aidan Bugayong

2024-12-15

Abstract

This research paper looks at the likelihood certain states are subject to gerrymandering. Gerrymandering refers to when politicians change county lines in order for their political party to have the majority representation in the total number of counties. To determine if a state is gerrymandered, we need to compare it to a population of non gerrymandered county distributions for that state. To do this, we generated random counties, not under the influence of gerrymandering, and then used the 2020 US election voter data to determine if each of those counties were democratic or republican, thus creating a population of non gerrymandered county distributions. Comparing our simulation county data to the real world county distributions during the 2020 election, comparing with the z-test, we found that Texas, Ohio, Pennsylvania, and California are under the influence of gerrymandering with 95% confidence level, while Illinois, Arizona, Kansas, and Wisconsin are not under the influence of gerrymandering.

1. Introduction

This project aims to determine if gerrymandering has occurred, or is possible with the current voting distribution of various US states. Additionally, we would like to be able to compare various states of different political preference in order to see if anything correlates with possible gerrymandering, or susceptibility to gerrymandering.

Gerrymandering is when election district lines are drawn in a way that artificially advantages one person, party, or group over another – a contributor to unfair elections. Politicians gerrymander by packing and cracking the political parties in order for them to win more areas for their party.

1.1 Data

We utilized the New York Time's compilation of neighborhood-level 2020 election data. This dataset contains a geospatial outline of voting areas, along with real-world vote counts for that area. These areas can vary in size from a couple blocks to many acres of land, but they tend to represent each area serviced by one particular polling location.

Each area included in the dataset contains information including its GeoID, vote counts for each major political party, vote density, total votes, and the outline of the physical area covered.

1.2 Approach

We attempt to simulate this via pseudo random groupings of polling neighborhoods into larger groupings which we may approximate as counties. Then, we simulate what a vote would result in given these county lines in order to determine the winner of each election. After running this simulation several times, we used a normal distribution to approximate the distribution of county affiliation per state. With this normal distribution, we can compare the real-world distribution to our expected ranges with a z-test in order to determine how likely a state has been affected by gerrymandering.

With this approach, we hope to simulate a wide range of possible country line distributions to see if the current distribution is fair or representative of the people voting.

1.3 Summary and Insights

We randomly distribute counties for various states in the US and calculate what an election would look like for such counties in order to gain insight into what range of party distributions are likely. Our findings found with 95% confidence that Ohio and Texas are gerrymandered in favor of Republicans while California and Pennsylvania are gerrymandered for Democrats. However, our data suggested that Wisconsin, Illinois, Kansas and Ohio are not under the influence of gerrymandering. Thus both parties use gerrymandering for their advantage, however more work is needed to be done in order to determine impact of these political actions.

2. Methods

One way to determine if a state is gerrymandered or not is to compare the county lines to a population of non gerrymandered county lines and check if the differences between them are

statistically significant. However, in order to compare the real life county lines to non gerrymandered counties we needed to create a population of counties that we know are not under the influence of gerrymandering.

We achieved this by pulling the US voter data of individual neighbourhoods from the 2020 election and then generating random counties from these neighborhoods. We generated 90,000 county sets for each of our selected states then tallied the election results for our new counties to determine the number of counties that are democratic or republican. Then we compared our simulated non gerrymandered county distributions to the real life counties to determine if the difference was statistically significant.

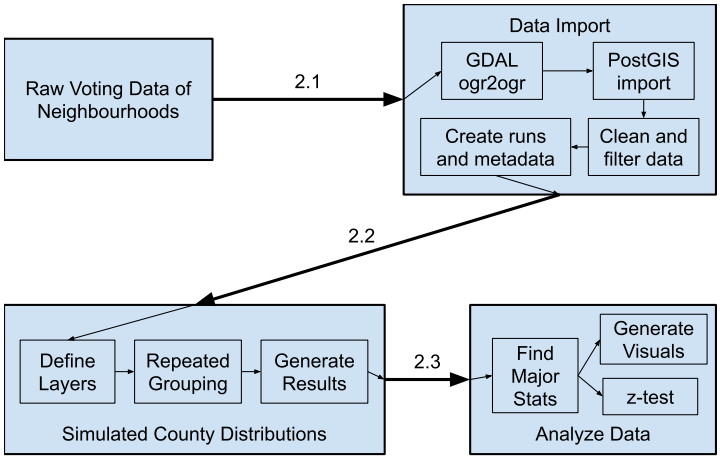

Figure 1: Workflow

Figure 1 is a high level overview of the explained process in the paragraph above. Specifically each step of the workflow is covered in further detail in their respective sections. Section 2.1 covers the process of converting the raw voting data into clean data ready for simulations. 2.2 covers the simulation process for how we generated 9,000 different county distributions for our selected states [California, Ohio, Wisconsin, Illinois, Kansas, Arizona, Texas, Pennsylvania]. 2.3 covers what methods we used to draw conclusions from our data.

2.1 Cleaning and Importing Data

Our data was originally in the GeoJSON format, which is a specific format of JSON for connecting and representing spatial and non-spatial data. Our data had granular voting areas for each polling station across multiple states in the US for the 2020 presidential election along with the non-spatial data of number of votes for each party and the location's GeoID.

We utilized the `ogr2ogr` processor from `GDAL` in order to import our GeoJSON data into PostgreSQL for ease of processing and spatial data manipulation tools. The `ogr2ogr` tool also converts the GeoJSON text geometry into the WKB format of geospatial geometry for faster processing later on.

Once the data has been imported into Postgres, we created a new trimmed table containing only the non-spatial attributes that we need, including parsing the GeoID into state and county columns for easier grouping later. While creating the table, we also utilize PostGIS to instantiate the WKB geometry into PostGIS's native closed shape object, while removing broken non-closed objects from the geometry.

2.2 Generating Simulation Data

The simulation process is broken down into several larger steps for the ease of understanding and parallelization. First, we define a group as any closed area within the US that strictly remains within one US state. Second, we define a run as any group of groups such that they do not intersect and span the area of a US state. Some of the runs will have a result, which is a summary of the total number of groups that have a majority of the vote for each party in the run. Third, we define a result group as a group of results with random varying runs to get a distribution of possible party totals with different country distributions. Four, we define a layer as a combination of a number of runs that are all based on the same previous run, a number defining how many runs to generate from the previous run, and a number defining the desired number of groups within each generated run. Last, we define a simulation as a group of sequential layers, where the initial layer is based on the raw data we obtained and each subsequent layer is based on each run of the previous layer. A simulation also has a result set for the final runs of the last layer.

We begin our simulation with grouping all neighborhood data of a state into a run. We then define layers such as 880 groups with 30 samples then 88 groups with 30 samples. We utilize a pseudo random, seeded k-means clustering algorithm to group a prior run into the specified number of groups. This k-means clustering algorithm randomly assigns the effective center of each group and weights each group randomly in order to increase entropy when clustering. Each child run will then have the next layer generated upon it. For this example run, we end up with 900 final runs of the state, which we can tabulate into a result group for later analysis.

In order to accomplish this, after importing all the data into Postgres, we define two functions. One creates our clusters, which take in a previous run, and desired groups among other metadata and updates the metadata of any results or statistics then utilizes PostGIS's ST_ClusterKMeans function in order to produce new groupings. The second function takes in simulation arguments and instanciates layers by repeatedly calling our grouping function for each sample needed, then calculates statistics for the result group.

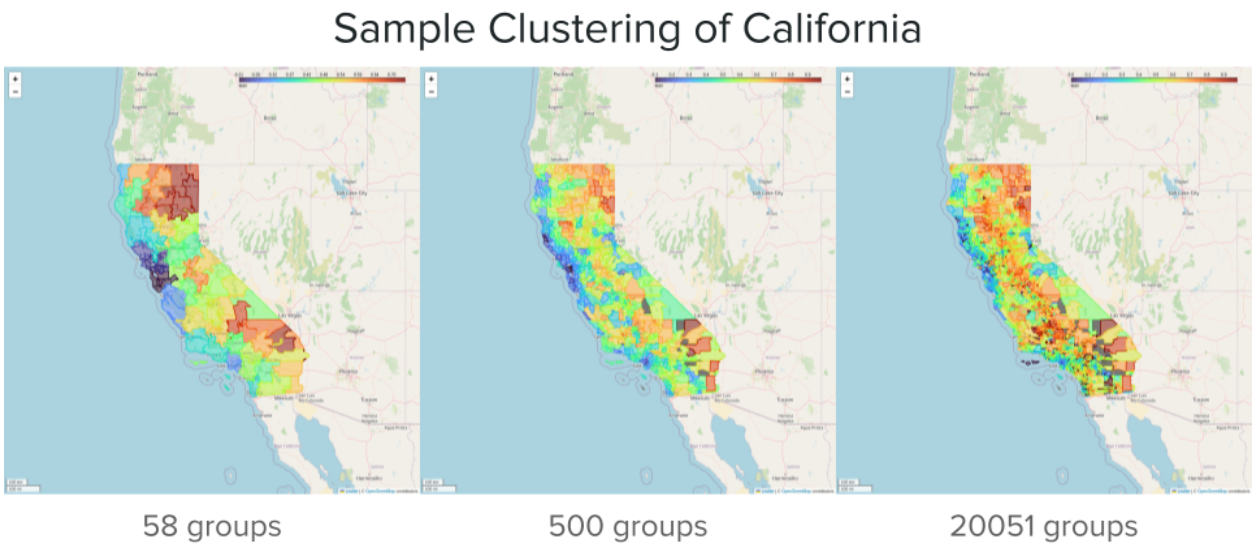

Figure 2: Example Clustering Process of California

Figure 2 demonstrates this process for California. California has 58 counties but comprises over 20,000 neighborhoods in our dataset. To generate counties quicker and with more randomness, we group the 20,000 neighborhoods into groups of 500. This is one of 2 layers. The second layer

would be the final groupings of 58 counties. To generate a large number of county distributions, we generate 30 different layer 1s (groups of 500) then generate 30 county distributions from each of our layer 1s. Thus totalling in 9,000 different counties for California. This process was then repeated for our selected states: [California, Wisconsin, Ohio, Illinois, Kansas, Pennsylvania, Arizona, Texas].

2.3 Analyzing Data

After running a total of nine thousand simulations across each of our chosen states, we can find the average and standard deviation for the number of democratic counties in our random simulations for each of the states. To find these statistics from our data we used the python library SciPy and we used the library Pandas to store our data. We use this to create a normal distribution for our population of randomized counties.

Using the normal distribution, average and standard deviation for our democratic counties, we performed z-tests on our states using the actual number of democratic counties in each state as our sample.

$$z = \frac{\bar{x} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

Equation 1: Z-test

Since our data only had one data point for the actual number of democratic counties per state, i.e. the election results of 2020, we performed our z-test with a sample size (n) of 1. In the z-test equation, $\bar{x}$ and σ represent the population mean and standard deviation respectively. In our case the population mean is the average number of democratic counties a given state had across our randomized simulations and σ is the standard deviation for those democratic counties. Next μ represents the sample mean, which is the actual number of democratic counties in real life.

In our report, we used a confidence level of 95%, thus the critical z-value is 1.96. Meaning any z-score above 1.96 or below -1.96 is considered gerrymandered with 95% certainty. To visualize

our data we used the python library Matplot, utilizing its histogram and normal distribution functions.

3- Results and Discussion

After running our simulation for our selected states, we get the following results displayed in figure 3. Remember, x and σ represent the average and standard deviation for the number of democratic counties in our simulation and μ is the actual number of counties for the 2020 election.

State	χ	σ	μ	z
Texas	34.9778	2.613	22	-4.967
Ohio	10.3433	1.406	7	-2.378
Illinois	15.1711	1.567	14	-0.747
Arizona	5.4700	0.857	5	-0.548
Wisconsin	14.3089	1.429	14	-0.216
California	3.67667	0.894	5	1.479
Kansas	31.0511	1.934	35	2.041
Pennsylvania	10.1567	1.226	13	2.319

Figure 3: Z Scores of likelihood to be Gerrymandered Left

Since we are analyzing the results for the number of democratic counties, a negative z-score indicates that the actual number of democratic counties is lower than the randomized county averages. Thus negative z-score means gerrymandered right, and a positive z-score is gerrymandered left.

As stated in section 2.3, the critical z-value is 1.96 so the states that are gerrymandered with 95% certainty are Texas, Ohio, California, and Pennsylvania. Texas and Ohio have negative z-scores,

indicating they are gerrymandered in favor of Republicans. Whereas California and Pennsylvania have positive z-scores, meaning they are gerrymandered in favor of Democrats.

Furthermore, according to the table the states with the largest standard deviation were California and Texas. Larger standard deviations indicate a larger variance in county distributions during our random simulations. Thus, a larger standard deviation signals that a state is more susceptible to gerrymandering. California and Texas are the largest states that we analyzed, however more analysis would need to be done in order to conclude if there is a correlation between state size and susceptibility to gerrymandering.

To better visualize our data, we can use the average and standard deviation for the number of democratic counties in our simulations to create a normal distribution. We can additionally mark where the real life county numbers are on the graph. First are the visualizations for the states that were considered gerrymandered according to the z-test.

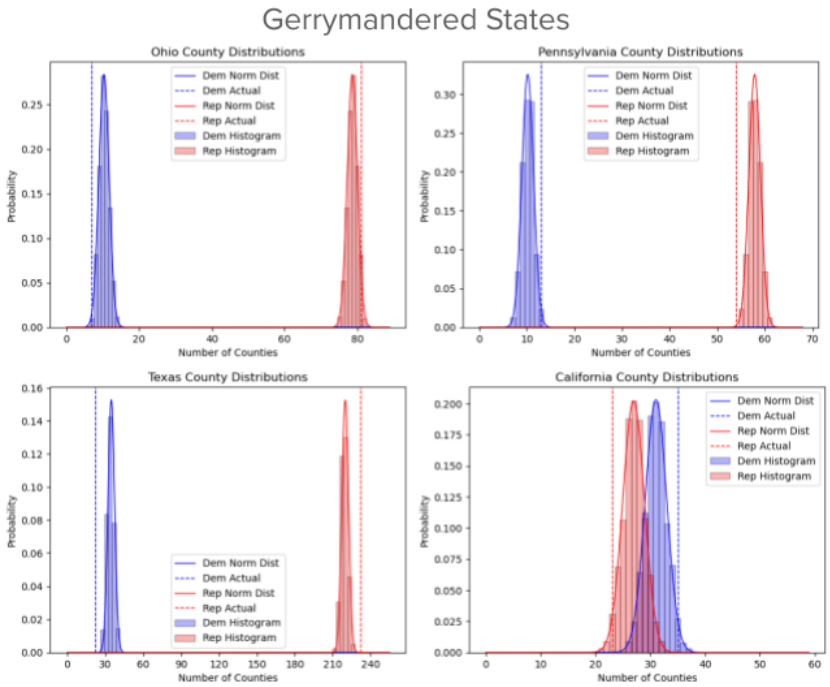

Figure 4: Normal Distributions of Gerrymandered States

In figure 4, the histogram inside each plot represents the distribution of counties for the respective states during our simulation process. To better visualize the distribution, the histogram is outlined with a normal distribution that uses the average and standard deviation from the histogram. Additionally, the dotted line represents the actual number of democratic vs republican counties during the 2020 election. Of course, red represents the data for republican counties and blue represents the data for democratic counties.

The normal distributions for each of the plots are centered about the average number of counties for the respective state during our simulations. As such, the peak of the normal distributions represent the average number of counties for randomized counties. States are more likely to be under some influence of gerrymandering if the dotted line is further from the peak, and less likely if the dotted line is closer to the peak.

In the above graphs for the states considered gerrymandered by the z-test, it is clear that the dotted line is not centered with the normal distribution. In fact, for some of the states the dotted line is almost not even on the normal distribution. Most notably is Texas (bottom left) where the dotted line is almost entirely off of the normal distribution. This indicates that the actual number of democratic/republican counties in the real world were so vastly different from the counties generated in our simulations.

Next is the visualization for the states which were not under the influence of gerrymandering according to the z-test.

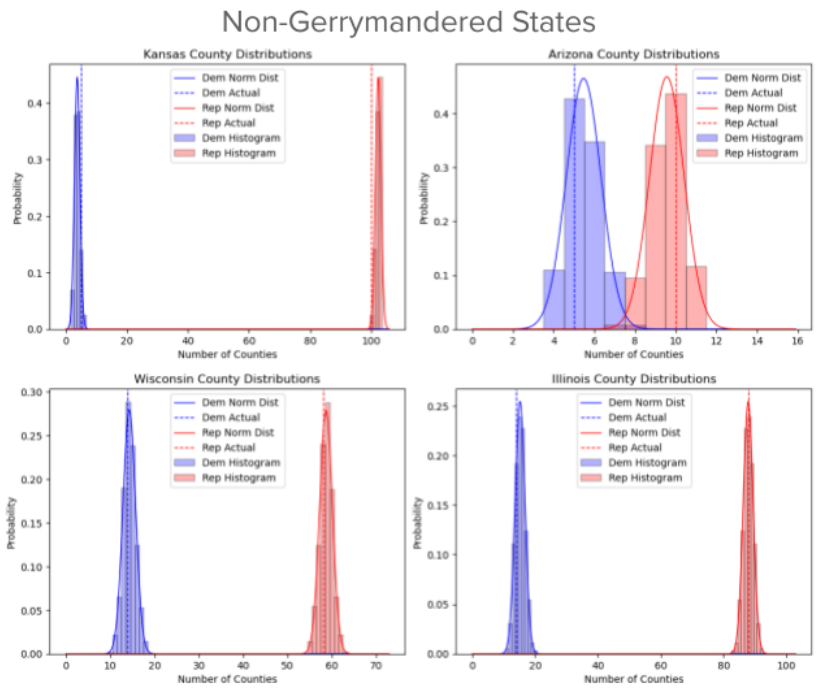

Figure 5: Normal Distributions of Non-Gerrymandered States

Figure 5 follows the same visualization methods described for figure 4, where the plot contains a histogram, normal distribution, and dotted line to represent the actual number of counties for the states. Additionally, the peak of the normal distributions are the most likely county distribution if the states are not under the influence of gerrymandering.

According to figure 5, these 4 states (Kansas, Arizona, Wisconsin, and Illinois) have similar democratic/republican county distributions as our randomized simulations. This means that these states are likely to not be under the influence of gerrymandering because our simulation is designed to mimic unbiased county lines.

The conclusions gathered from looking at figures 4 and 5 match conclusions derived from figure 3. Thus we can conclude that with 95% confidence that Texas, Ohio, California and Pennsylvania are all subject to gerrymandering.

4. Conclusion

Gerrymandering has serious ramifications regarding the stripping of power from citizens and instead giving it to politicians. Through our simulations of random county lines, we have created a population of counties for states that are truly random and not under the influence of gerrymandering. Using this population, we have concluded that California, Texas, Pennsylvania, and Ohio are under the influence of gerrymandering with 95% certainty. Specifically, California and Pennsylvania are gerrymandered in favor of democrats while Texas and Ohio are gerrymandered in favor of Republicans. However, our data suggested that Illinois, Wisconsin, Arizona, and Kansas are not under the influence of gerrymandering. While both political parties use gerrymandering to their advantage, more research needs to be done in order to fully understand the impact gerrymandering has had on citizens. More research needs to be done in order to find correlations factors that determine the likelihood of a state being gerrymandered as well as factors that contribute to the susceptibility of a state to be gerrymandered.

5. Roles

Trevor Nichols	Aidan Bugayong
Transforming Data Running Simulations	Visualization for Data Statistical Analysis

6. Source Code

All source code is available here: github.com/tnichols217/CSDS133-final.
The 2020 voter data by neighbourhood is available for download here: github.com/TheUpshot/presidential-precinct-map-2020.
An interactive visualization of this data is available thanks to New York Times here: www.nytimes.com/interactive/2021/upshot/2020-election-map.html

References

Park, Alice, et al. “An Extremely Detailed Map of the 2020 Election.” *The New York Times*, The New York Times, 2 Feb. 2021, www.nytimes.com/interactive/2021/upshot/2020-election-map.html

TheUpshot. “TheUpshot/Presidential-Precinct-Map-2020: The Geojson Dataset behind Our Nationwide Precinct Map of the 2020 Presidential General Election.” *GitHub*, 13 Mar. 2021, github.com/TheUpshot/presidential-precinct-map-2020

“Documentation.” *PostgreSQL*, The PostgreSQL Global Development Group, 21 Nov. 2024, www.postgresql.org/docs/

“Documentation.” *PostGIS*, PostGIS PSC & OSGeo, 26 Sept. 2024, https://postgis.net/documentation

“GDAL Documentation.” *GDAL Documentation*, Frank Warmerdam, Even Rouault, https://gdal.org/en/stable/programs/index.html#general

“Matplotlib 3.9.3 Documentation.” *Matplotlib Documentation - Matplotlib 3.9.3 Documentation*, The Matplotlib development team, 2012, matplotlib.org/stable/

“Introduction to Geopandas.” *Introduction to GeoPandas - GeoPandas 1.0.1+0.G747d66e.Dirty Documentation*, GeoPandas developers, geopandas.org/en/stable/getting_started/introduction.html

“Pandas Documentation.” *Pandas Documentation - Pandas 2.2.3 Documentation*, pandas via NumFOCUS, Inc., pandas.pydata.org/pandas-docs/stable/index.html

“Scipy.” *SciPy*, https://scipy.org/

“Well-Known Binary (WKB).” *GEOS*, GEOS, 4 Oct. 2021, libgeos.org/specifications/wkb/

Nichols, Trevor, and Aidan Bugayong. “CSDS133-Final.” *GitHub*, Nov. 2024, github.com/tnichols217/CSDS133-final