COUNTY TO CONGRESSIONAL DISTRICT MAPPING: VISUALIZATION OF LYME DISEASE IN THE U.S.

ELLA KIM

2022 Civic Digital Fellowship- U.S. Department of Health and Human Services (HHS)
jk925@uw.edu

Abstract. This report addresses the issue of converting Lyme disease case reporting from county-level to Congressional district-level. The purpose of the report is to provide and explain in more detail the various code/algorithms and methodologies that are explored by the author during their time as a Summer 2022 Civic Digital Fellow. Several methods are applied, but the majority of the algorithm focuses on solving the conversion of area and case counts to Congressional district form and demonstrating the many visualizing methods for the results. Additional and alternate methods are encouraged to be applied by the reader to use in their own presentation or analysis, as emphasized in Section 4 of this report. Throughout this report, *cases* and *count* are used interchangeably, as well as *algorithm* and *code*. When otherwise cited, methods and assumptions in this report are of the author's own words and do not represent other organizations.

1. **Introduction and Background**

As per the HHS Health Equity DataJam page for Lyme disease, the purpose of the author's code solution is to "address Lyme disease, and improve health equity for all tick-borne diseases, using emerging technologies that couple the power of the crowd and patient insights with data." [3]

One of the ideas for exploration (and is this report's focus) is State and Local Data Visualization: "Lyme disease cases are reported by county and states, yet policy and budget decisions often happen at the level of U.S. Congressional districts, so what innovative maps and extrapolation methods can map case counts to Congressional districts?" [3]

Organizing open data in the format that is most useful for another department is a crucial middle step that must be taken so that the open data is actually utilized. In this case, the author has no direct communication to the potential stakeholders. Enabling the reader to choose which visual format is most easily digestible is also necessary.

The author uses these two files as resources to address this issue (URL of pages retrieved from):

"County-level Lyme Disease Cases/Dataset (200-2019, .csv file):" `https://www.cdc.gov/lyme/stats/survfaq.html`

NOTE There is a caveat to the cases reported in this dataset (refer to Caveats in Section 2.

"Counties within Congressional Districts (.zip files):" `https://www.census.gov/geographies/mapping-files/time-series/geo/cartographic-boundary.html`

Date: August 30, 2022.

NOTE The author only uses the available .zip files that contain both county and Congressional district areas (2013-2019 files).

As for the programming solution, the author's algorithm uses Google Colab's (`https://colab.research.google.com/`) .ipynb (Python notebook) file for more versatile paths for files and markdown visualization (for readers who are less familiar with programming). The resulting visualizations are choropleth maps of the U.S.' or its states' Lyme disease case counts at the Congressional district level (as explained and shown in Section 3).

2. **Algorithm Implementation and Development**

Packages

The following packages were used to perform these functions/methods in our algorithm:

Matplotlib[4]: was used for all static plots (colored intensity of counts, etc.)

Pandas[6]: was used to create and wrangle data frames in the algorithm

GeoPandas[5]: was used to plot and locate areas on the U.S. Map, covert county boundaries to Congressional district boundaries on plots, and plotting interact maps.

Methods

The next sub-sections cover the different methods and their corresponding approaches used in the .ipynb code (reader can select them in the algorithm in the 'Select Map Type' section), where Approach 1, etc. correspond to 'approach_1' in the code. All methods with 'approach'-es have 'average' options, which takes the mean of all the available approaches. By averaging different approaches, case counts will potentially be more reliable and accurate than any single approach.

Method: Splitting County Counts

Splitting County Counts refers to the methods of how to reasonably allocate the original county cases into Congressional district cases. The author only explores one approach for this, but the algorithm holds an example of how to add more approaches and include it in the 'average' calculations.

Approach 1: Area Ratio

The main issue with county areas is that their areas include more than one Congressional district. This first approach makes the assumption that the cases are evenly distributed over the county land area. Figure 1 is an example of Baltimore City, a county in Maryland that has some parts of Congressional districts 2, 3, and 7. We will use this example for all other methods below.

From Figure 1, let the total count of cases in in Baltimore City county be TC, the Congressional district 2 area (the lightest shaded area) within the county as 2A, and the total area of the county as TA (the known values). Let Baltimore City county's count allocated to our 2A area be 2C (the value we want to find). Then, using the assumption mentioned above, we can solve for 2C:

$$2C = TC * (2A/TA) \tag{1}$$

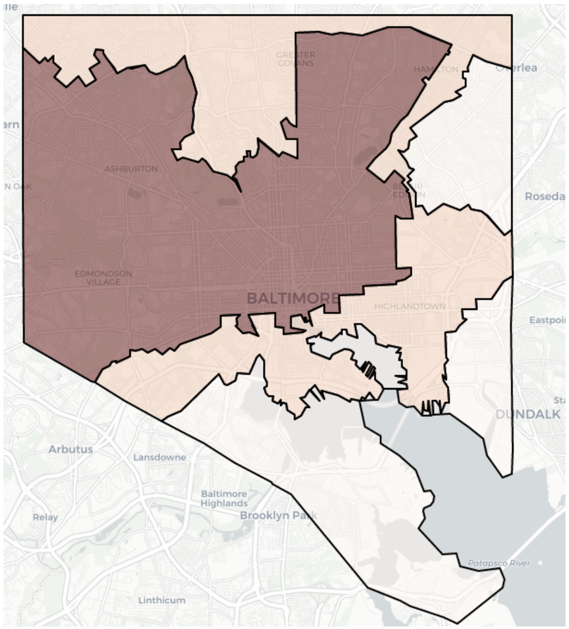


Figure 1. Plot of Baltimore City county with choropleth indicators of Congressional districts 2 (lightest color), 3 (medium), and 7 (darkest color)

After this is done for all counties in Maryland, all the sub-areas (like 2A) and their corresponding allocated counts (2C) will be added together within each Congressional district areas to get the total case count for each of Maryland's eight Congressional districts.

Average

This approach would hypothetically take the mean between the different Splitting County Counts approaches. In other words, take 2C and other values solved for and average them for a new case count for 2A's area.

Method: County 999 Counts

Similar to Splitting County Counts, County 999 Counts refers to cases in the Lyme disease dataset cases that fall under the row label for county FIPS code 999. In other words, county 999 holds any reported/probable cases within each state whose patient residency is unknown. Due this unknown, the author acknowledges that adding them could be debatable, and therefore includes it as a optional selection in the algorithm (in the 'Select Map Type' section). Any value calculated with this method would be added only the counts from a Splitting County Counts method (such as 2C from Equation 1) to get a total case count for portions of Congressional Districts (such as 2A from Equation 1).

Approach 1: Congressional District Populations

This first approach takes the assumption from how Congressional districts are mapped [1], where each district is divided into equal populations of 761,179 individuals within each state. We assume that cases have nothing to do with the Congressional district or population density and only total population counts, so unknown county counts (county 999) are evenly split between districts.

Let's again refer to Maryland state as an example. As mentioned in the last approach, the total number of Congressional districts in Maryland is eight (let this be 8D). Let the total number of cases in Maryland's county 999 be 999T. To solve for the number of cases in any Congressional district from county 999 cases (999D), we can solve:

$$999D = 999T/8D \tag{2}$$

Approach 2: Congressional District Case Ratio

Adding onto the Congressional district population assumption (each holds 761,179 people), we now assume that the count from a state's county 999 cases follows a similar ratio distribution as Approach 1 in the Splitting County Counts method (Equation 1), where county 999 cases is split among Congressional districts based on the ratio of total cases each district has compared to the state total.

Using Maryland state, let the total cases for Maryland state excluding county 999 be TS, the total cases for Maryland's Congressional district 2 excluding county 999 be 2C, and the total cases in Maryland's county 999 be 999T. To solve for the number of cases in district 2 from county 999 cases (999D2), we can solve:

$$999D2 = 999D * (2C/TS) \tag{3}$$

Average

This method will add the average of the approaches above and add it onto each Congressional district's counts from the 'Splitting County Counts' approaches. In other words, for our example county and our current approaches (2 total), if 'Average' is 999DAvg, we can solve:

$$999DAvg = 999T/2 + 999D2/2 = (999T + 999D2)/2 \tag{4}$$

Reporting Methods

Reporting method in the algorithm's 'Select Map Type' section refers to the reader's selections for the visualization of Congressional district total case counts in the choropleth maps: either by the raw individual cases (how it was solved in all of the Equations above) or cases per 100,000 individuals.

Individual Count

As this is the raw individual count, there is no additional equation to apply to this reporting method. We simply take the already calculated 'Splitting County Counts' approach cases and, optionally, the 'County 999 Counts' approach cases and use those values in the map plotting/visualizations.

Per 100,000 Count

This method converts individual cases (let this be IC) to cases per 100,000 count. This is done under the assumption of all state Congressional districts having 761,179 individual residents. Using this population count, we can solve for cases per 100,000 people with in each Congressional district (let this be PER):

$$PER = (IC/761,179) * 100,000 \tag{5}$$

3. Computational Results

The 'Select Map Type' section in the algorithm additionally provides the reader the option to select the year of case reports to map (2013-2019), static (Figure 2) or interactive (Figure 3) plots, as well as an option to save the .png (static map) or .html (interactive map) file to the Google Colab environment. The files will appear in the same file paths as the 'data' and 'county' folders the reader will create in Colab and can be manually downloaded.

Examples of possible combinations of methods and their approaches, types of graphs, etc. the reader can explore, analyze, and/or test in the algorithm 'Select Map Type' section are shown in Figure 2 and 3 below:

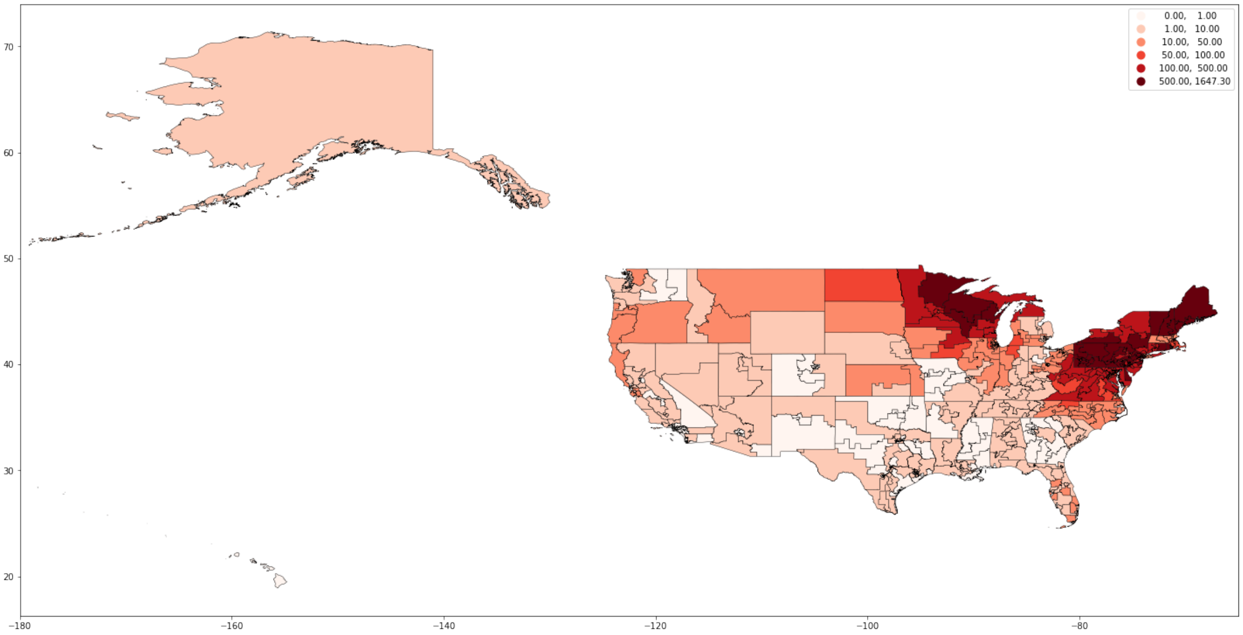
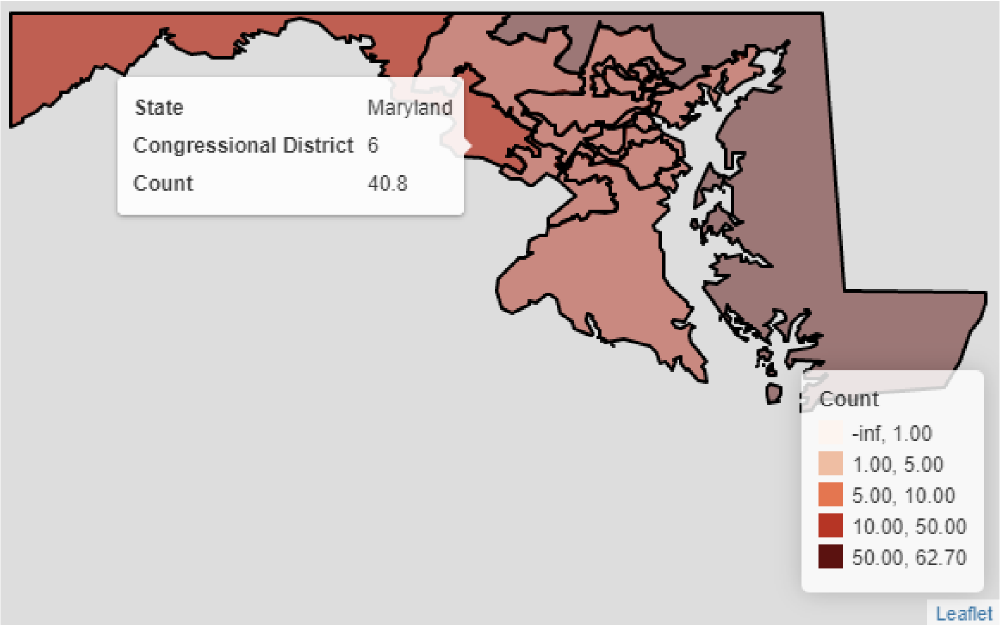


Figure 2. Static plot of cases in Congressional Districts in the US in 2017 (with 999 county data (Approach 1), reported as individual case counts). X and y-axis refer to global latitude and longitude measurements



State	Maryland
Congressional District	6
Count	40.8

FIGURE 3. Interactive plot of cases in Congressional Districts in Maryland in 2017 (without 999 county data, reported as cases per 100,000 people). Refer to Caveat 2 addressing $-\infty$ in the legend.

Caveats of results:

The author will address some caveats in the final mapped Congressional district case counts and visualization methods:

(1) The largest caveat is with the reported cases in the dataset the author used. As mentioned by CDC [2], due to lack of reporting made by individual health care providers, the actual number of Lyme disease cases estimated by CDC is estimated to be at least nine times larger in count than the dataset yearly totals (over 300,000 total cases, versus the little over 30,000 reported dataset cases).

(2) Due to occasional lack in the amount of counties per state, interactive map's legends can have a minimum interval of '-inf' ($-\infty$), but this does not mean the minimum count in the dataset is $-\infty$: the minimum count is always 0 at the least.

(3) While exploring different interactive plots, the reader may notice that there are a couple islands to very far right and very separate from the rest of the national/U.S. plot. These are the trailing islands on Alaska state. The visualization uses a map that has set longitudes on the edges of the plot, so if the edges were wrapped around to mimic the earth's roundness, the Alaskan islands will be in the correct location. In the static plot, the axis cuts off those few islands to prevent the plot becoming too zoomed out and unreadable. The argument behind this is that Alaska has only one Congressional District, so the reader should sufficiently see the total count from the rest of Alaska's territory.

4. Conclusions and Future Plans

The purpose of the methods and algorithm is not to require the usage of such count-splitting methods and averages, but rather the focus is to provide the algorithms to transform county areas to congressional. Methods applied are up to the reader's discretion and the author encourages application of statistical methods of splitting counts to produce potentially more accurate and reliable results.

The algorithm as well as this documentary of methods will potentially be released in public GitHub repositories on both HHS's and the author's accounts, to be open for analysis and usage by academic or health organizations who share the same goal to improve health equity for Lyme disease patients.

5. Acknowledgements

The author is thankful to HHS's InnovationX department for the inspiration to address this issue and HHS's Dr. Kristen Honey for her review and suggestions for possible applicability of the author's work. The author is also thankful for Johns Hopkins' Lyme disease dashboards (`https://www.hopkinslymetracker.org/overview/`), who were an enormous help in understanding what potential stakeholders could be during the author's Civic Digital Fellowship and in providing examples of map visualization and case reporting.

References

[1] About congressional districts. `https://www.census.gov/programs-surveys/geography/guidance/geo-areas/congressional-dist.html#:~:text=Each%20congressional%20district%20is%20to,the%20districts%20within%20each%20state`. Accessed: 2022-08-02.

[2] How many people get lyme disease? `https://www.cdc.gov/lyme/stats/humancases.html`. Accessed: 2022-08-02.

[3] Lyme innovation and health equity. `https://healthdata.gov/stories/s/9q39-eeqb`. Accessed: 2022-08-02.

[4] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[5] K. Jordahl. Geopandas: Python tools for geographic data. *URL: https://github. com/geopandas/geopandas*, 2014.

[6] W. McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.