West Coast Group assignment 3: Algorithmic Restricting

Hannah Graham, Matt Miller, Morgan McCoy, Aishanee Wijeratna

MSDS 460 Fall 2024

GitHub:
https://github.com/hannah-r-graham/Assignment3_AlgorithmicRedistricting_MSDS460.git

Table of Contents

Data Sources

This assignment used various public data sources. To understand which counties were in Indiana, data from district.org/Indiana was used. Then, using census 2020 data, total population and white only population were pulled through census.gov federal site. The counties for Indiana is rather straightforward and not much room for error. However, there are a few concerns around the census data. First, the data is from the 2020 census which is over four years ago. While in some eras that is not a large opportunity for change, the majority of COVID happened and many, many people relocated from cities to rural areas during that time frame. Indiana which has many rural counties, could be heavily affected by this migration. Second, the race selections are optional, so the count of white alone individuals could be a vast undercount of the reality.

Specification (Objective Function and Constraints)

The objective of this project is to minimize the distance between counties within a district. In other words, we wanted to maximize the compactness of counties per district. Our constraints were mostly population based. The first constraint was that each county must be in exactly one district. The second constraint is that the total population of each district must be within 5% of the “ideal” population to keep the total populations within each district at a similar magnitude.. The ideal population was calculated by taking the sum of the total population for all the counties and dividing it by 9 districts. Each county can only belong to one district.

Programming

We began by collecting and centralizing the necessary data for the problem. Data included the list of counties, their populations, the counties nearby, and the latitude and longitude of their location. Each district was named a numerical value between 1 and 9.

The data was then split up into different dictionaries with the county ID as the key. One dictionary, "dict", included the adjacent counties and their distance from the county key. A second dictionary was created where each county was also the key and the population was listed for each county. We originally tried a simple data frame table where both dictionaries were combined, however, we were having a hard time keeping the structure straight and the counties were being assigned to more than one district. Pre-processing was mainly handled by Pandas and Regex. Crafting the linear programming solution relied heavily on the pulp package.

Solution

The initial result using a 5% deviation for population yielded an infeasible solution and left out roughly 10 counties for assigning to a district. When we expanded the percentage variance for population to 15%, we were still missing 2 counties. We had to expand to 30% in our final result to have every county assigned to a district. However, we realized after doing parts of our map, we found that districts were not adjacent, and we realized we messed up the constraints of having counties that were adjacent. Once we added in the constraint of having counties adjacent, the solution became infeasible.

The table below shows our results of not having adjacent counties as a constraint:

	County_Code	District	County
0	18001	5	Adams County
1	18003	8	Allen County
2	18005	6	Bartholomew County
3	18007	8	Benton County
4	18009	7	Blackford County
5	18011	7	Boone County
6	18013	1	Brown County
7	18015	1	Carroll County
8	18017	4	Cass County
9	18019	4	Clark County
10	18021	8	Clay County
11	18023	9	Clinton County
12	18025	3	Crawford County
13	18027	9	Daviess County
14	18029	6	Dearborn County
15	18031	8	Decatur County
16	18033	8	DeKalb County
17	18035	5	Delaware County
18	18037	6	Dubois County
19	18039	3	Elkhart County
20	18041	6	Fayette County
21	18043	1	Floyd County
22	18045	3	Fountain County
23	18047	4	Franklin County
24	18049	8	Fulton County

25	18051	4	Gibson County
26	18053	5	Grant County
27	18055	4	Greene County
28	18057	6	Hamilton County
29	18059	4	Hancock County
30	18061	9	Harrison County
31	18063	7	Hendricks County
32	18065	5	Henry County
33	18067	9	Howard County
34	18069	9	Huntington County
35	18071	7	Jackson County
36	18073	4	Jasper County
37	18075	3	Jay County
38	18077	4	Jefferson County
39	18079	5	Jennings County
40	18081	4	Johnson County
41	18083	4	Knox County
42	18085	9	Kosciusko County
43	18087	7	LaGrange County
44	18089	9	Lake County
45	18091	5	LaPorte County
46	18093	6	Lawrence County
47	18095	7	Madison County
48	18097	2	Marion County
49	18099	8	Marshall County
50	18101	3	Martin County

51	18103	9	Miami County
52	18105	1	Monroe County
53	18107	3	Montgomery County
54	18109	3	Morgan County
55	18111	5	Newton County
56	18113	5	Noble County
57	18115	1	Ohio County
58	18117	4	Orange County
59	18119	6	Owen County
60	18121	9	Parke County
61	18123	4	Perry County
62	18125	5	Pike County
63	18127	3	Porter County
64	18129	9	Posey County
65	18131	3	Pulaski County
66	18133	1	Putnam County
67	18135	4	Randolph County
68	18137	8	Ripley County
69	18139	4	Rush County
70	18141	7	St. Joseph County
71	18143	8	Scott County
72	18145	8	Shelby County
73	18147	5	Spencer County
74	18149	4	Starke County
75	18151	7	Steuben County
76	18153	3	Sullivan County

77	18155	3	Switzerland County
78	18157	8	Tippecanoe County
79	18159	6	Tipton County
80	18161	7	Union County
81	18163	7	Vanderburgh County
82	18165	4	Vermillion County
83	18167	1	Vigo County
84	18169	5	Wabash County
85	18171	8	Warren County
86	18173	4	Warrick County
87	18175	9	Washington County
88	18177	1	Wayne County
89	18179	8	Wells County
90	18181	1	White County
91	18183	1	Whitley County

Maps and Discussion

Our solution is not sufficient to present to the government for redistricting. It does not currently satisfy the need for counties to be adjacent in each district, for maximum compactness. Additionally, we would want to do further analysis around the demographic conditions in each district that our algorithm created. If the algorithm created skewed districts to one specific demographic, the solution would not sufficiently represent demographics equally and might give one demographic more power in elections. When compared to other districting solutions, such as the existing district lines for Indiana or potential alternative solutions developed in DistrictR, this solution as it stands currently is not an appropriate replacement. Further analysis needs to be conducted into where our linear programming model failed. Once we are able to revise the model to be able to accurately define districts using adjacent counties, and also take into account demographic balance across districts, we will then need to revisit this analysis to compare our functional model to other existing districting strategies.

We would recommend staying with the current Congressional district lines for now, because they are the most balanced in terms of population size. This allows the districts to most accurately adhere to the "one person, one vote" philosophy. However, further analysis is required to determine whether the current district lines are equitable in terms of demographic distribution.