# Redistricting - Can the Minority Get the Upper Hand?

Write-up

Antonio Jimenez, Kamilla Madera, Dieu-Quyen Nguyen

EID: aoj268, kjm4223, dcn558

TACC username: aoj19037@isp.tacc.utexas.edu, kjm24@tisp.tacc.utexas.edu, dcn558@isp.tacc.utexas.edu

Email: antjim19037@utexas.edu, kjmadera24@utexas.edu, dieuquyennguyen558@utexas.edu

# Table of Contents

## Introduction

The fight for representation and power continues to concern many state districts and their respective representatives. In order to cement their power, governing parties often try to tilt the political map to work in their favor, known as Gerrymandering. However, the goal of redistricting contains drawing the boundaries of districts that accurately reflect the population changes and racial diversity. This allows an equitable representation in Congress and state legislatures (Aclu, 2021).

Currently, redistricting remains a problem in the current political climate with representatives, "attempts to tailor district lines for political gain" (Levitt, 2021). The goal for these governing parties remains in drawing boundaries of legislative districts to claim as many seats as possible for the party's candidates (Wines, 2021). The act of Gerrymandering often leads citizens to question the need to redistrict. Not only does the Constitution and the federal courts require the legislature to do it, but redistricting also allows fair and equitable access for minority populations.

Within the scope of the program, the main goal of districting maintains minority rule. The program dives into the need to divide the population into multiple districts and then make the remaining population its own district. This allows for the minority to gain leverage within its district. The incremental approach will allow proper partitions to be analyzed rather than looking at all possible partitions. Through further analysis of the code, the results will help to simulate whether the minority can gain leverage in the political field. Along with the simulation of the code, the results will also help the comprehension of the ethics of redistricting and the need for further representation.

## Background & Problem

As populations grow and communities change, the migration of political ideals and philosophy follows. Such changes require redistricting, “the process of drawing the lines of the district from which public officials are elected” (Aclu, 2021). The idea of redistricting first started when Elbridge Gerry, “ a wild-eyed eccentric” man, felt that the central government, aristocracy, and citizens must remain balanced for any republic (Trickey, 2017). Historically, when states did not redistrict to reflect their population growth, many of the voter results did not reflect the population growth of each state. For example, in the time period between 1901 and 1961, the Tennessee legislature ignored the need to redistrict which cause disproportionate growth in several districts and further underrepresent the minority population. On the contrary, in the 1960s, When the Los Angeles County did not redistrict, the largest district in California, contained 422 times as many people as California’s smallest district. Due to the disproportionate sizes, “each person in the smallest district enjoyed 422 times the Senate representation of each Los Angeles resident” (LLS, 2021). In the 1960s, the Supreme Court upheld the Fourteenth Amendment that guaranteed equality for minority communities and ensured the “one person, one vote” promise. The electoral systems in the states that failed to allocate any voting power to these communities were punished based on the basis of unconstitutional grounds. The Supreme Court also emphasized the need for each district to have roughly the same population to prevent population disparity in the United States.

Redistricting provides states an opportunity to ensure districts can reflect proper diversity and representation. However, when abused, can cause many issues for minority rule. Although Gerry’s idea to increase Congress’ power to override presidential vetoes did not receive well from his delegates, he won over the country and Congress at the time (Trickey, 2017). The newfound power Congress received caused many political figures and representatives to fight for a seat in politics. Districting allowed these political parties to gain more voters based on voter dilution. The abused use of redistricting coined the term Gerrymandering. The main tactics of Gerrymandering often include the practice of “packing” and “cracking” (Wines, 2021). Packing consists of diluting the power of the opposition by drawing district lines to create a packed district made up of the opposing party’s voters. The dilution of voters helps to strengthen the governing party’s win to gain votes in the surrounding districts. An application of this includes packing large minority groups into a district to create “minority voter dilution.” For example, Virginia’s state legislature “drew its majority-minority districts to be 55 percent black” (Soffen, 2021). The state’s legislature hoped to create a disproportionate district to limit minority influence in certain prominent districts. Whereas cracking splits up the clusters of the opposition voters in multiple districts so the votes become outnumbered.

The difference between redistricting and Gerrymandering ought to be noted for the clarification of the program itself. The very ethics involved with the simulation of code helps to emphasize the result of minority rule. The main issue that will be expanded within the program will discuss not only the possibilities of minority rule but also the ethical implications for the population as a whole. Throughout the program it will help to answer:

> Can the minority truly receive the upper hand? If so, will it properly reflect the population growth of a state, or cause a further disparity in how legislatures will define districts?

## Strategies & Method

In order to set up the minority rule, the scope must start with each individual voter and their respective affiliation. The overall program first starts with the voter class, where each voter receives identification and their associated affiliation. Within the code, there are two affiliations: -1 and +1. By setting up the foundation of the simulation with the individual voters, the random generation of the population can generate more organic results. This also allows for better analysis and can provide a more detailed answer to the main question.

The scope then widens to the whole population where the size, the majority, and the lean (skew) of the population are analyzed. The population class takes in a vector of voters with their affiliations and identifications. With the class, the affiliations help to define the majority and the lean of the population. The majority helps to define the ratio of each affiliation to each other along with the relationship between the population as a whole. For example, if the population organizes into +1; +1; +1; -1; -1, the +1 affiliation takes up three-fifths of the population; the majority would calculate the ratio to 0.6. The affiliations also help with defining the lean, or the skew of the population. For example, if the population has the following sequence of voters: +1; +1; +1; -1; -1, the lean then becomes +1, since it maintains the majority of the population's affiliations.

In order for the program to accurately present any analysis, the creation of a simulated population becomes the next priority of the code. For the main population program, there are three defined ways to make a simulated population. The first method to create the population requires an input of strings of the affiliation. This method first determines the size of the population based on the length of the string and then initializes voters with different affiliations based on the characters in the string. The new vector of voters created then helps to determine the lean and the majority calculation.

The next method entails a random population generator which can create larger population sizes. This function has the arguments; population size, lean, and majority. The user should input these three parameters to properly communicate with the computer for the wanted simulation. The function begins by setting the lean of the population to what the user inputted. A loop is then used to create a vector of voters whose size matches the argument population size. Then the random number generator gets called to retrieve a number between zero and one. That number then gets put into a specific range based on its relativity to the majority. For example, if there is a population size of five, and would like sixty percent of the population to identify with a +1 affiliation, the loop will run five times until the population reaches the set majority. With smaller population sizes the desired lean is sometimes not achieved. To solve this issue, if the lean gathered through the random number generator does not match the inputted lean, the function will discard the old population and use the loop to create a new population. It will continue this cycle until a population is created with the desired lean. For the population to organize, the majority obtains the range of 0 to 0.6 and the minority obtains the range of 0.6 to 1. Based on the random number generated, the number then becomes part of the respected range. Through this method, larger population sizes are created with a majority much closer to the inputted majority.

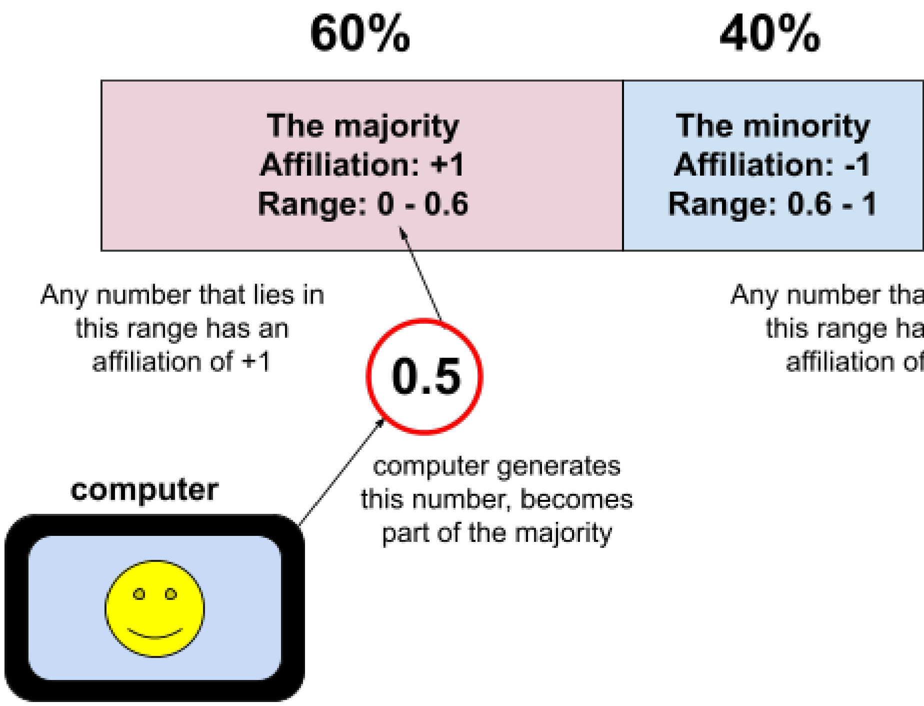


*Figure 1. Visual representation of how a random number becomes categorized*

The last method constructs a population out of a vector of voters. Through this method, a for loop iterates through the vector adding each element to the population. While adding the voters to the population, the constructor keeps track of the positive voters and the negative voters. Those value then get compared to determine the lean and majority of the population. This method, along with the first method, is especially effective when testing and simulating very small populations.

The next step to help with the districting process includes creating subpopulations. The goal of this function is to take a range of a district and make a smaller or sub-population to further categorize the districts. In the program, the function takes two integer inputs and uses them to create a vector of "new voters" from the first value to the last value. The function then returns the new sub-population.

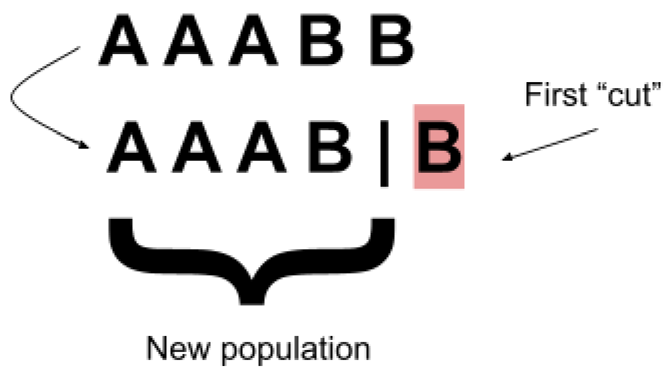


*Figure 2. Visual representation of sub-population cutting*

After the districts are created, the program will use the district class to properly identify the number of voters, the lean, the overall affiliation, and the majority. Then the program uses the districting code in order to determine if the population will obtain minority rule.

The minority rule function helps achieve the goal of giving the minority the upper hand, so in order to implement the function the user inputs a set amount of desired districts. To calculate the number of cuts, the program takes the number of desired districts and subtracts one. For example, if the user wants five districts the program will make four cuts.

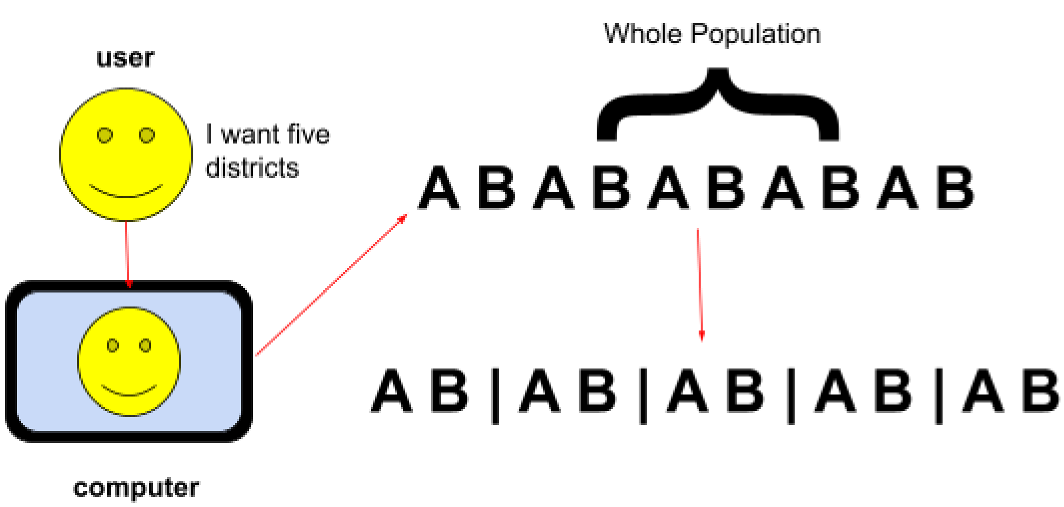


*Figure 3. Visual representation of the flow of cutting*

The function also uses a for loop to cut the population in all the possible ways. In order to keep track of the “best slice” it must keep track of the loop iteration that results in the largest difference between minority leaning districts and majority leaning districts. The program begins by slicing the right most voter and adding voters to the right slice as the loop counter increases. The right and left side of the cut are then made into districts that are added to a districting object called gerry. Within are then able to find the difference between minority leaning districts and majority leaning districts. Sometimes, two different cuts will result in the same difference of minority and majority districts. When this occurs it chooses the slicing that will result in the least number of minority votes wasted. Having decided the best way to slice the population, it will take the smaller slice, make it a district, and add it to the population’s attribute gerryFinal. The other side of the slice is then turned into a subpopulation that calls the function minority_rules with the argument number of districts - 1. The recursive function will continue slicing until the minority_rules function is called with number of districts = 1, since no more slices can be done on the population.

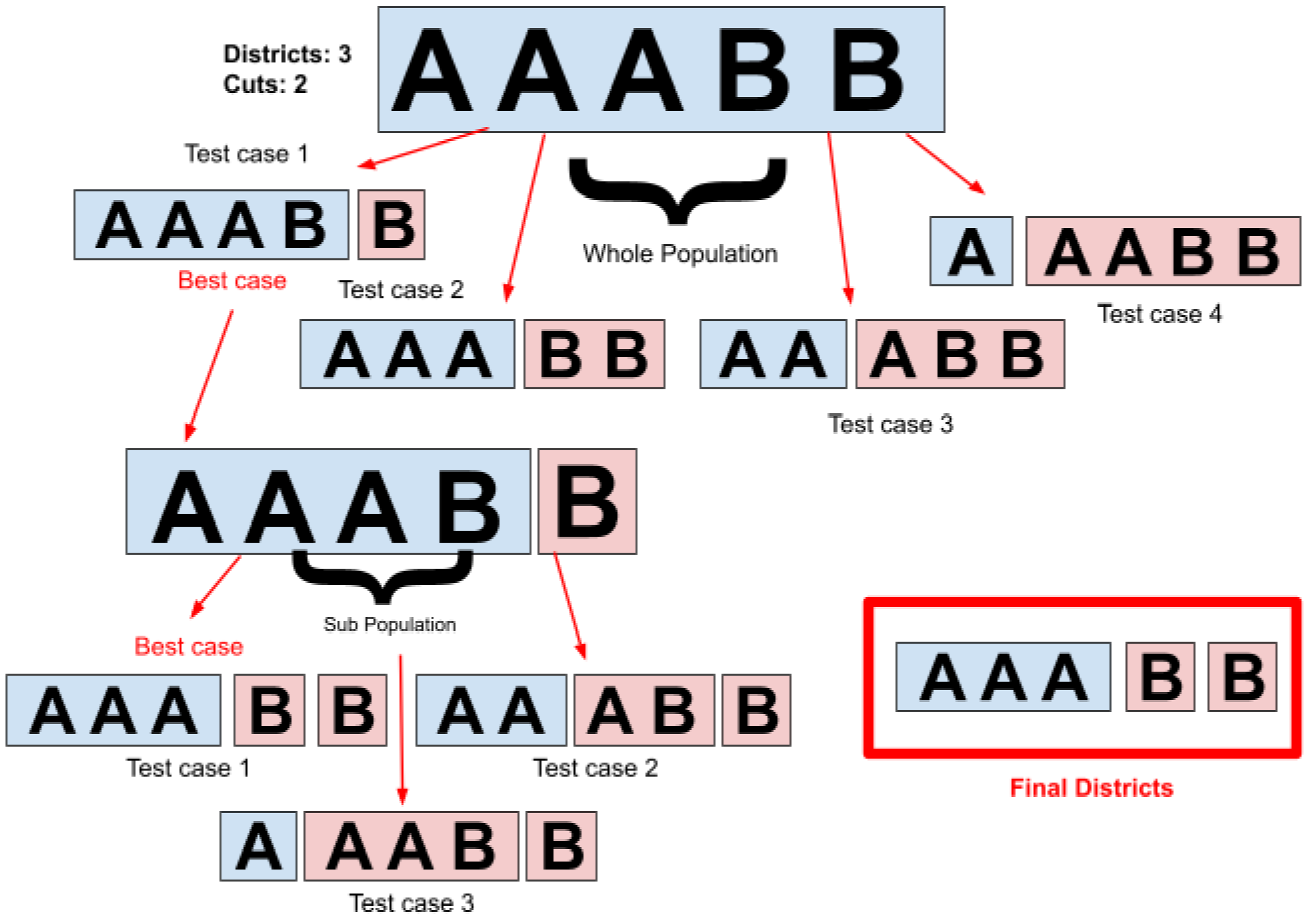


*Figure 4. Visualization of how the program finds the optimized distribution*

When legislatures set districting zones, they need to properly test if the current districts are constitutional. State legislatures calculate efficiency gaps, “a standard for measuring partisan gerrymandering,”, to see if the current district remains fair and gives enough representation to all communities (Petry, n.d.). The “wasted votes”, any vote that does not help elect a candidate, for each party help calculate the efficiency gap. The equation for the efficiency gap is:

$$efficiency\ gap\ = \frac{[Party\ A\ wasted\ votes] - [Party\ B\ wasted\ votes]}{total\ of\ statewide\ votes}$$

*Equation 1. Efficiency Gap (Handley, n.d.)*

The number of required votes of each district for a candidate to win helps to obtain the wasted votes for each party. For example, if each district contains 100 voters, then a candidate would need 51 votes to win a district. If a party were to lose in a district, all of their votes would become wasted votes because it did not help them win within the district. If a party were to win a district, then their wasted votes equal the total votes the party won, minus half of the total votes plus one vote, to claim the minimum majority votes. Consider the following example:

*Table 1. Example of a state and its districts*

| District | Party A Votes | Party B Votes | Total Amount of votes | P. A wasted Votes | P. B wasted Votes | Net wasted votes |
|---|---|---|---|---|---|---|
| 1 | 73 | 27 | 100 | 22 | 27 | 5 B |
| 2 | 42 | 58 | 100 | 42 | 7 | 35 A |
| 3 | 69 | 31 | 100 | 18 | 31 | 13 B |

Each district contains 100 voters and each party needs 51 votes in order to win within each district. In district one, Party A won with 73 votes, so their wasted votes would equal 22 votes. Since Party B lost, all of their votes contributed as wasted votes. To calculate the efficiency gap, the equation would look like:

$$efficiency\ gap = \frac{[82] - [65]}{300} = 0.0566667 = 5.66667\%$$

In other words, Party A converts their votes into legislative seats better than Party B. As a result, they won about 5.667 percent more seats than they would have if both parties had wasted an equal number of votes. In reality, Party A won a seat but did not win a larger margin of seats and therefore the districts remained constitutional. According to Eric Petry from the Brennan Center for Justice, “[... An] efficiency gap of two or more seats indicates a constitutional problem (for congressional plans). For state legislative plans, the threshold is an efficiency gap of 8 percent or greater.”

Implementing an efficiency gap into the program, the main use would be to test the constitutionality, or the fairness, of each redistricting test case. When the final redistricting case gets chosen by the program, the efficiency gap will determine if the minority ruling maintains constitutionality. Since the program tests if the minority can gain the upper hand, the program does not consider the constitutionality of the redistribution. So, the efficiency gap will make-up for it and provide mathematical proof for the fairness of the districting lines.

## Results

When the current function runs with the example given in the textbook, the program successfully separates the proper districts with the two minority voters as their own districts.

```
-------------------   Final Slicing   -------------------
 P: 0
PLD Negative count: 1
PLD Positive count: 0
------------   Districting    ---------------
[3[0:-1,],[0:-1,],[0:1,1:1,2:1,],]
minority rule: 1

------------   Districting    ---------------
[3[0:1,1:1,2:1,],[0:-1,],[0:-1,],]
minority rule: 1
```

*Figure 5. Example result*

The population got defined in the main redistricting code as a string and the program converted the character into the respective +1 and -1 affiliations. Once the program properly converted the values, the program then separates the population into the best combination of cuts to favor the minority rule.

Although the program succeeded in creating the desired result, further analysis of the program must be discovered in order to properly achieve the main purpose. Three random populations with varying sizes but with the same desired amount of districts will test the capabilities of the program. The first test made a random population with ten voters, and -1 as the majority party by 60 percent.

```
Size: 10
Majority: 0.6
Positive: 4
Negative: 6
[ -1 , 1 , -1 , -1 , -1 , -1 , 1 , 1 , 1 , -1 ,
]

------------   Districting   ---------------
[3[0:-1,1:1,2:-1,3:-1,4:-1,5:-1,],[0:1,],[0:1,1
:1,2:-1,],]
minority rule: 1
```

*Figure 6. First test case output*

The program creates another successful districting plan that allows for minority rule. Due to the smaller population sample, the program can easily navigate through the code and properly district the voters. Since the majority needed for the population was 60 percent, the random number generator was able to generate through the range for the needed voters.

The next test case made a random population with 50 voters instead with the same parameters above.

```
---------------     District   -----------------
Size: 50
Majority: 0.52
Positive: 24
Negative: 26
[ -1 , 1 , -1 , -1 , -1 , -1 , 1 , 1 , 1 , -1 ,
-1 , 1 , -1 , -1 , -1 , 1 , -1 , -1 , -1 , -1 , 1
, -1 , 1 , 1 , -1 , -1 , 1 , -1 , 1 , 1 , 1 , -1
, -1 , 1 , -1 , 1 , 1 , 1 , -1 , -1 , 1 , 1 , 1 ,
1 , -1 , 1 , 1 , -1 , -1 , 1 , ]
------------   Districting   ---------------
[3[0:-1,1:1,2:-1,3:-1,4:-1,5:-1,6:1,7:1,8:1,9:-1,
10:-1,11:1,12:-1,13:-1,14:-1,15:1,16:-1,17:-1,18:
-1,19:-1,20:1,21:-1,22:1,23:1,24:-1,25:-1,26:1,27
:-1,28:1,29:1,30:1,31:-1,32:-1,33:1,34:-1,35:1,36
:1,37:1,38:-1,39:-1,40:1,41:1,],[0:1,1:1,2:-1,3:1
,4:1,5:-1,6:-1,],[0:1,],]
minority rule: 1
```

*Figure 7. Second test case output*

The program outputs another districting plan that favors the minority rule. As figure 7 shows, separated this population of 50 into 3 districts resulting in the middle district having 42 votes and the other two districts having 6 and 1 vote in each. The minority rule is +1 since there is one district with a single positive voter and the other two districts have a majority of +1 as well.

The last test case made a random population with 100 voters instead with the same parameters above.

```
Size: 100
Majority: 0.57
Positive: 43
Negative: 57
[ -1 , 1 , -1 , -1 , -1 , -1 , 1 , 1 , 1 , -1 , -1 , 1 , -1 , -1
, -1 , 1 , -1 , -1 , -1 , -1 , 1 , -1 , 1 , 1 , -1 , -1 , 1 , -1
, 1 , 1 , 1 , -1 , -1 , 1 , -1 , 1 , 1 , 1 , -1 , -1 , 1 , 1 , 1
, 1 , -1 , 1 , 1 , -1 , -1 , 1 , -1 , -1 , -1 , -1 , -1 , -1 , 1
, 1 , -1 , 1 , -1 , -1 , -1 , 1 , -1 , -1 , -1 , 1 , -1 , -1 ,
-1 , -1 , 1 , -1 , -1 , 1 , -1 , 1 , 1 , 1 , -1 , -1 , 1 , 1 , 1
, 1 , 1 , 1 , 1 , -1 , -1 , -1 , -1 , -1 , 1 , -1 , 1 , -1 , -1
, -1 , ]

[3[0:-1,1:1,2:-1,3:-1,4:-1,5:-1,6:1,7:1,8:1,9:-1,10:-1,11:1,12:-
1,13:-1,14:-1,15:1,16:-1,17:-1,18:-1,19:-1,20:1,21:-1,22:1,23:1,
24:-1,25:-1,26:1,27:-1,28:1,29:1,30:1,31:-1,32:-1,33:1,34:-1,35:
1,36:1,37:1,38:-1,39:-1,40:1,41:1,42:1,43:1,44:-1,45:1,46:1,47:-
1,48:-1,49:1,50:-1,51:-1,52:-1,53:-1,54:-1,55:-1,56:1,57:1,58:-1
,59:1,60:-1,61:-1,62:-1,63:1,64:-1,65:-1,66:-1,67:1,68:-1,69:-1,
70:-1,71:-1,72:1,73:-1,74:-1,],[0:1,1:-1,],[0:1,1:1,2:1,3:-1,4:-
1,5:1,6:1,7:1,8:1,9:1,10:1,11:1,12:-1,13:-1,14:-1,15:-1,16:-1,17
:1,18:-1,19:1,20:-1,21:-1,22:-1,],]
minority rule: 0
```

*Figure 8. Third test case output*

This time however, with an even larger population, minority rule was not given. This brings an interesting perspective for the program. In this example, although there is more of a minority percentage in the population the districts do not represent a lean towards the minority. This flaw in the program underlines the need to update methods for greater population sizes, similarly how legislatures often need to change when the population growth causes clear disparities for the voters. There could be multiple reasons for why the minority rule is zero but the main reason is the order of the voters because the user cannot control the random number generator. Although the program chooses the best case scenario for the minority rule but that does not guarantee a benefit towards the minority every time.

```
---------------     District   -----------------
Size: 25
Majority: 0.84
Positive: 21
Negative: 4
Efficiency Gap: 0.16
[ 1 , 1 , 1 , 1 , 1 , -1 , 1 , 1 , -1 , 1 , 1 , 1
, 1 , 1 , 1 , 1 , 1 , 1 , 1 , -1 , -1 , 1 , 1 , 1
, 1 , ]
```

*Figure 9. Implementation of efficiency gap*

The efficiency gap was calculated with different districts within the program. This specific district had 25 voters with the lean of +1. So, the wasted votes for Party +1 equals to 8 and the wasted votes for Party -1 equals to 4. So the total efficiency gap equals to 0.16 following the EG equation. Therefore, Party +1 was able to claim a legislative seat around 16 percent more effectively than Party -1.

## Conclusion

Through the tribulations of coding, testing, and research the comprehension for the impact of districting increased. By experimenting with different methods of districting and testing with different groups of populations allowed for further education on; what the basis of districting is, how an effective distribution impacts votes, the consequences of Gerrymandering along with its loopholes, and how representation can often become convoluted with the political parties greed for more legislative power. One of the most interesting aspects of districting explored in this program was the continuous testing for the best district with the minority rule. Through mathematical processes like efficiency gap, the program was also able to test the constitutionality of each districting result.

In the future, to further expand the efficiency and accuracy to real world applications the program would implement memoization, dynamic programming, and 2-d coding. Without these concepts, the code is a bit inefficient since it repeatedly analyzes the same range of voters when making new sub-populations with the best district lines. By including memoization and dynamic programming it would allow us to save time due to having already analyzed ranges of the population and repeating the district lines stored in the memory. By being able to compare the new sub-populations to the previously analyzed sub-populations allowing the program to save space assigning new sub-districts for potential best case districts. This requires a proper recursive function with the ability to assign previously analyzed best district lines to an outer variable that can be compared within and out of the function itself.

As for the ability to redistrict and our 2D application, another conditional would need to be implemented that tests how the newly inserted voter would affect the right side of the population. Based on its impact there would be its own district created, if affiliated with -1, or use of the memoization previously programmed to apply the districting lines with the inclusion of the new voter. There is the ability to insert a new voter in the districting class and it would be used to test and represent the redistricting process and effects. Our 2D application would allow us to create some sort of visual representation of what an actual state with district lines would look like numerically. The different individual elements in the matrix would represent the most efficient sub-districts made but would give us the ability to sort the districts in greater groups based on their leaning affiliations.

Finally, can the program prove that the minority receive the upper-hand? Based on the given results, yes the program can but with limited population sizes. No matter the algorithm, the realization that the bigger the population size and the smaller the districts wanted the less likely the minority will rule due to the dispersion of votes and lack of control of who and where people are placed. The best the program could do was to separate the population with districts having as many minority votes with as little majority votes as possible. Just like the program, legislatures redistrict to the best of their abilities but have to go through many methods in order to find the best representation for the growth of their population. Despite finding the “optimal” distribution, the planned lines could be unconstitutional and cause voter dilution and accidental color-blindness districting. Overall, the program was able to explore the different cases of population sizes and further answer that the minority can have the upper hand but with the cost of constitutionalism.

## References & Bibliography

Aclu, A. (2021, September 29). *What is redistricting and why should we care?: News & commentary*. American Civil Liberties Union. Retrieved December 4, 2022, from https://www.aclu.org/news/voting-rights/what-is-redistricting-and-why-should-we-care

Apfelbaum, E. P., Norton, M. I., & Sommers, S. R. (2012). Racial Color Blindness: Energence, Practice, and Implications. Association For Psychological Science. https://doi.org/10.1177/0963721411434980

Handley, L. (n.d.). Some mathematical measures for determining if a ... - michigan. Some Mathematical Measures for Determining if a Redistricting Plan Disproportionally Advantages a Political Party . Retrieved December 9, 2022, from https://www.michigan.gov/-/media/Project/Websites/MiCRC/ExpertSubmission/MICRC_Dr_L_Handley_Memo_on_Three_Partisan_Fairness_Measures.pdf?rev=848faddda0a94acf8e0c0945072fcb03

Levitt, J. (2021, April 2). *Why should we care?* All About Redistricting. https://redistricting.lls.edu/redistricting-101/why-should-we-care

LLS. (2021, October 16). What is redistricting? All About Redistricting. Retrieved December 4, 2022, from https://redistricting.lls.edu/redistricting-101/what-is-redistricting/

Petry, E. (n.d.). *How the efficiency gap works - brennan center for justice*. Brennan Center for Justice. Retrieved December 9, 2022, from https://www.brennancenter.org/sites/default/files/legal-work/How_the_Efficiency_Gap_Standard_Works.pdf

Soffen, K. (2021, November 24). How racial gerrymandering deprives black people of political power. The Washington Post. Retrieved December 4, 2022, from https://www.washingtonpost.com/news/wonk/wp/2016/06/09/how-a-widespread-practice-to-politically-empower-african-americans-might-actually-harm-them/

Trickey, E. (2017, July 20). *Where Did the Term "Gerrymander" Come From?* Smithsonian Magazine. https://www.smithsonianmag.com/history/where-did-term-gerrymander-come-180964118/

Wines, M. (2021, November 21). *What Is Gerrymandering? And How Does It Work?* The New York Times. https://www.nytimes.com/2019/06/27/us/gerrymander-explainer.html