Gerrymandering: A Sampling Approach

Nicholas Tomlin Timothy Sudijono Shivam Nadimpalli

BMCM 2018

Contents

1. To The Readers of The North Squarolina Tribune

Gerrymandering is the act of unfairly dividing a region into districts so as to favor one political party over another. While manipulation of the redistricting process has been used as a key tool in partisan battles for power, it also severely undermines the very idea of democracy.

As a country that vests electoral power into the hands of representatives chosen by its people, to say that gerrymandering has caused damage to our political system would be a gross understatement.

For example, as you may all be aware, in the 2012 North Carolina Congressional elections, 51% of the electorate voted for Democratic candidates, while 49% voted for Republicans. In spite of this, nine Republican and four Democratic representatives were elected to the House. Very recently, the United States Court of Appeals deemed North Carolina's districting to be unconstitutionally gerrymandered and ordered a re-districting of the state.

How do we detect such districting manipulations, and how can we confirm that a district was purposefully gerrymandered to favor one party over another?

We present a mathematical model that attempts to detect partisan gerrymandering in a potential districting scheme. We attempt to capture several natural notions of gerrymandering in our model, and use pre-existing gerrymandering-detection schemes to calibrate our approach. Its performance is analyzed against several test-scenarios.

Finally, we verify that our model does indeed detect gerrymandering by successfully replicating the results of the well-documented 2012 North Carolina Congressional Election on a simplified representation of the same dataset.

2. Introduction

Voting systems and elections are an essential part of most democracies. Given the importance of these events in the political process, it is often the case that political parties or members will try to skew the outcome of an election through various means. We focus on one method in particular: that of affecting the election scheme by how countries or states are split up into smaller *districts* for purposes of voting.

A natural question to ask, then, is whether it is possible to detect manipulative attempts to district in ways that unfairly favor a political party? These so-called *gerrymanders* have been a long-standing issue within the U.S. political system, and have received considerable media attention in recent times. In a well-known example, North Carolina had 9 out of 13 Republicans from its district vote, although the popular vote favored the Democrats. [Her+18] Yet, there appears to exist no standard approach to identify and evaluate gerrymanders.

Formally, Black's Law Dictionary defines a gerrymander as *a name given to the process of dividing a state or other territory into the authorized civil or political divisions, but with such a geographical arrangement as to accomplish a sinister or unlawful purpose, as, for instance, to secure a majority for a given political party in districts where the result would be otherwise if they were divided according to obvious natural lines.*

Several novel attempts to quantify gerrymandering have been made in recent years, making use of tools and techniques from discrete geometry, stochastic processes, and statistics.

In this report, we present a model that attempts to systematically detect partisan gerrymandering on arbitrary planar graphs. Specifically, given a partition of a graph into several districts, we define a *gerrymandering score* to determine the likelihood of manipulative districting. Our score incorporates several natural aspects of partisan districting, and we analyze its performance in several different political landscapes as well as settings for partisan districting. Given a partisan districting and its score, we can analyze how unlikely this score is relative to a distribution we fix on the set of all districtings. We also discuss other approaches to gerrymandering, use them to tune our method, and place our method in the context of these other approaches.

2.1. Outline

- In Section 2.2, we briefly discuss existing approaches to detecting and quantifying a partisan gerrymander.
- In Section 3, we discuss our model and the assumptions we make. In particular, we assume two-party dominance in the government and the (weak) Pareto principle.
- In Sections 4–5, we present our approach to quantify gerrymandering, and analyze its performance on several carefully-chosen and some randomly-generated districting scenarios.
- In Section 6, we evaluate the performance of our model on a real-world instance of gerrymandering, namely that of Congressional elections in North Carolina in 2012.
- In Section 7, we discuss the drawbacks of and potential improvements to our model.

Our model utilizes the GerryChain package (which can be found here), and our code is included in Appendix B.

2.2. Previous Work

Given that gerrymandering is a topic of national interest, several research groups led by scientists and mathematicians have worked on recent solutions to this problem, especially at centers at Duke University, Tufts (Metric Geometry and Gerrymandering Group), and at Princeton University. But even prior to this work, several metrics based on efficiency, symmetry, and even geometry have been used to quantify gerrymandering.

The first and most well known method—the *efficiency gap*—is a way to quantify a common way of gerrymandering an election, namely "packing" and "cracking". This is the idea that most of the counties of the victim party are "packed" into a small number of districts, forcing only a small number of large wins for this party; the remaining counties are then "cracked" and spread out across many districts so that they are just shy of winning [Wan]. This notion can be quantified by the number of *wasted votes* for either party, which are defined to be the number of votes beyond what is needed to win a district, as well as the votes that that spent in losing districts. Formally, we can define the number of wasted votes for Party 1, and analogously for Party 2, in district $d_i, i = 1, \ldots, k$ as

$$W_i^1 = \begin{cases} T_i^1 - T_i/2, \text{ if } d_i \text{ won by Party 1} \\ T_i^1, \text{ if } d_i \text{ won by Party 2} \end{cases}$$

where T_i^1 is the number of votes cast in district i for Party 1, and T_i is the total number of votes in the district. Given this formula for the wasted votes, we define the Efficiency Gap EG to be

$$EG = \sum_{i=1}^{k} \frac{W_i^1 - W_i^2}{T}$$

with $T = \sum_{j=1}^{k} T_j$. The efficiency gap then gives a signed measure for how much one party is favored over the other. If large and positive, the districting plan would be unfair to A and conversely with B; in this case we can take the absolute value as the metric to assess gerrymandering against any party. We then consider a districting plan to be gerrymandered if the efficiency gap is greater than the threshold 0.08 [BD17b]. However, the efficiency gap is known to be not good enough in many cases: see [BD17b] for several of shortcomings of the approach. As one example, the efficiency gap actually tends to penalize the intuitively desirable notion of proportionality, that the proportion of districts voting for Party One should be equal to that of the population proportion. However, we will use this metric as a standard of comparison for our method later on in our analysis, due to its widespread usage in the literature.

A simpler idea, championed by Garry King, Andrew Gelman, and other researchers, enforces a symmetry or fairness between two parties in an election. This so called *partisan symmetry*, reviewed in [BD17b], requires that how one party performs given a certain share of the vote should be identical how the other party performs in the same situation. One way to quantify this is to use the notion of a *uniform partisan swing* [Duc18], which returns a curve of how many seats of power a party obtains as a function of the proportion of votes. The idea is that this curve should be symmetric around the midpoint $(0.5, 0.5)$, and this deviation can be measure by simple metrics, such as vertical distance from this point.

Another distinct body of work done in quantifying gerrymandering comes from a geometric/ topological view, which characterize how unnatural some gerrymandered districting plans appear to be. Approaches in this direction tend to use methods such as isoperimetry, curvature, hyperbolicity, and compactness quantification [BS18]; [DT18b]; [DeF+18]. However, an argument against the use of geometric and topological measures for gerrymandering can be found in [AM17]. In this paper, Alexeev and Mixon argue the statement that "a small efficiency gap is only possible with bizarrely shaped districts", indicating that geometric/topological approaches to quantifying gerrymandering may be ill suited. Indeed, we can imagine that in some states, the political and geographical landscape may enforce districting plans that have odd shapes. Therefore, we exclude the use of these methods in our model, although we note that the extensions of our model can employ geometric information if desired.

3. Our Model and Assumptions

For our model, we consider a rather strong simplification of the political map into a planar graph $G = (V, E)$ where every node represents a county in a state, and edges enforce adjacency of two counties. We call a *districting plan* D to be a partition of V with the following conditions that are required by Black's Law Dictionary:

1. **Contiguity**: Districting plans should have connected districts.
2. **One Person, One Vote**: Voting districts should have mostly equal populations.
3. **Voting Rights Act**: Voting districts must not dilute the votes of protected minorities.

Given this, we enforce the following further assumptions on our model.

4. **Bipartisanism**: We assume that there are only two political parties to vote for, which we denote Party 1 and Party 2, or Blumocrats and Redublicans. We assume that every $v \in V$ prefers to vote for one party over the other, which is that county's political model. Such a mapping of nodes onto Party 1 or Party 2 is called a *political leaning*. In this model, we do not consider quantify how much each each county prefers towards one party over the other. This assumption is motivated by Arrow's impossibility theorem from social choice theory, which proves the non-existence of an ordinal voting scheme satisfying some natural criteria in the event of three or more candidates. [MAS+14]
5. **Uniformity**: We assume every node in the graph represents the same population, and so the districting plans should each have districts that are roughly of the same size, i.e., they have the same number of nodes in each district. We also assume populations of protected minorities are homogeneous across each node, which allows us to disregard the condition of the Voting Rights Act in our model.
6. **The (Weak) Pareto Principle**: Our analysis is predicated on the interpretation that political elections should reflect the will of the popular vote. That is, if the majority of people prefer Party 1 to Party 2, then the election outcome should reflect this. Our gerrymandering definition then reflects this principle.

The above conditions allow us to restrict to districting plans which are partitions of V into districts, or components, which are connected and roughly of the same size; we'll say that each size of each district will be within two of one another.

4. Detecting Gerrymandering

Given the setup described in Section 3, we come up with a method for detecting whether a given districting plan D is considered gerrymandered. The method proceeds by defining a score $S(D)$ describing how "gerrymandered" the districting plan D is, then analyzes how extreme this score is relative to the distribution of districting plans with the above conditions enforced. We model this distribution with a Gibbs measure and sample from it using a Metropolis-Hastings-type sampler, which allows us to get Monte Carlo estimates of the variance of the scoring function S on a typical districting plan. We then express how extreme the observed score was using a metric analogous to a z-score. The advantage of this model over the methods we introduced previously is the generality of this model, which we will detail in later sections.

4.1. Defining a Gerrymander Score

We recall that our definition for gerrymandering is motivated by the Pareto principle: namely, that the outcome of the election based on the bipartisan proportion in a districting plan should reflect the outcome dictated by the general population proportion. We can capture this using a simple metric, namely the difference between the true population proportion of Party 1 to Party 2.

Letting $D = \{D_1, \ldots, D_k\}$ be a districting plan with D_i denoting individual districts, we will denote the population proportion p_T to be the number of Party 1 counties to the total number of counties, and the proportion of districts won by Party 1, denoted

$$p_D = \frac{\#\{\text{Districts won by Party 1}\}}{|D|}$$

from which we can obtain our first scoring function:

$$S_1(D) = |p_T - p_D|.$$

We first note that lower scores of the function S_1 should be interpreted as lower indication of gerrymandering. As an example of when this metric captures a perfect districting plan, consider the following example, reproduced from [Ing15]:

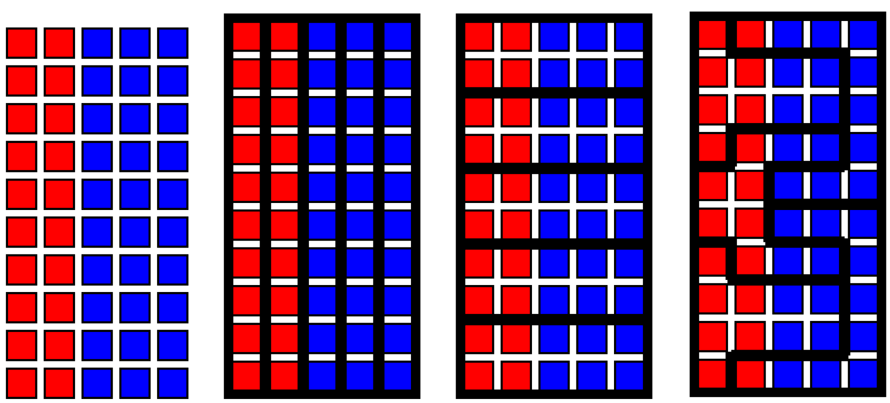


Figure 4.1: Possible districtings in a 40-60 population. Note that red wins the rightmost election, despite having a lower popular vote. We consider the rightmost diagram to be gerrymandered. The score metric S_1 will assign a perfect score of 0 to this districting plan.

The second image in the figure above gives a districting plan which most would agree as a fair plan that is representative of the true population. The score S_1 finds that the proportion of districts belonging to Party 1 is equal to the total number of counties that vote for Party 1, indicating $S_1 = 0$. However, consider the third image in Figure 4.1 above. Although we consider this districting plan not the ideal, we argue that this districting plan is a common occurrence that should not be considered gerrymandering. In this case, we observe that every within every partition, the proportion of Party 1 counties is close to the total proportion, but the proportion of Party 1 districts differs from the true population proportion significantly.

In order to extend our metric to account for this case as well, we consider a scoring function on the set of districts in our districting plan: recalling that $D = \{D_1, \ldots, D_k\}$, define the proportion of Party 1 counties in district D_i to be p_{D_i}. As the above example shows, we would like most of the values p_{D_i} to be roughly equal to p_T. Given this, we define a weighted $\mathcal{L}^2$ norm S_2 of the vector $(p_{D_i})_{i\in\{1,\ldots,k\}}$ to be

$$S_2(D) = \left(\sum_{i=1}^{k}\left(\frac{p_{D_i} - p_T}{k}\right)^2\right)^{\frac{1}{2}}.$$

Finally, to account for the fact that both examples above should not be considered gerrymandered, we define our scoring function

$$\boxed{S(D) = \min(S_1(D), S_2(D))}.$$

4.2. Our Sampling Approach

Given this scoring scheme, we want to take a statistical approach to analyzing whether a given districting plan is gerrymandered. Throughout this section, fix a districting plan $D = \{D_1, \ldots, D_k\}$, which we would like to test for gerrymandering. To do this, we analyze how extreme the observed value of $S(D)$ relative to the distribution of $S(\mathcal{D})$ where $\mathcal{D}$ is a random districting plan with the underlying Blumocrat/Redpublican preferences of the nodes fixed.

For clarity, we can interpret a districting plan as a partition of our graph G. We then sample $\mathcal{D}$ from a Gibbs measure on the set of partitions of our graph G which satisfy the conditions enforced in Section 3, which we can write generally as the distribution:

$$\mathcal{G}(D) = \frac{e^{-\beta J(D)}}{\mathcal{Z}_\beta}$$

where J is the Hamiltonian characterized by our sampling algorithm given below. This approach is similar to the one adopted by Herschlag et al. [Her+18] and other papers in literature, except we specialize this approach to the case of our planar graph model with equal populations at each node, and adopt an approach predicted on p-values. An informal characterization of this Gibbs distribution is given by the sampling procedure below:

The Markov Chain

The Markov chain we use has as its state space Σ the space of all valid districtings of the given graph $G = (V, E)$. The transition function is given as follows:

1. From a given state, determine the set S of all pairs (D, D') where D refers to the current districting, and D' refers to the districting obtained by:
   (a) Picking an arbitrary node that is on the boundary of some district, and swap it into one of the neighboring districts
   (b) Rejecting if any two districts vary in size (or area) by greater than a chosen threshold (in this case we pick our threshold to be 2).
2. From S, choose one pair (D, D') uniformly at random.
3. Transition $D \to D'$ if D' satisfies other desired criteria (if any; in this case, contiguity of districts).

Note that the update rule forces the outputted partitions to satisfy the conditions we impose. Additionally, the condition that the districts' size have to be within two of each other allows us to move between the states in the entire space of districting plans - we thus formally have irreducibility of our chain.

We use a burn-in of $10,000$ runs and sample every $10,000$ iterations, producing N samples $\{\tilde{D}_{i\in 1,\ldots,N}\}$ from our Metropolis Hastings algorithm. Using the samples of $S(\mathcal{G})$, we define the *gerrymandering severity* G_S of the redistricting plan D as a quantity similar to a z-score:

$$G_S = \frac{|S(D) - \mathbb{E}[\hat{S(\mathcal{G})}]|}{\operatorname{std}\hat{S}(\mathcal{G})}$$

which we estimate using simple Monte Carlo:

$$\mathbb{E}[\hat{S(\mathcal{G})}] = \frac{1}{N}\sum_{i=1}^{N} S(\tilde{D}_i)$$

$$\operatorname{std}\hat{S}(\mathcal{G}) = \frac{1}{N}\sqrt{\sum_{i=1}^{N}(S(\tilde{D}_i) - \mathbb{E}[\hat{S(\mathcal{G})}])^2}$$

Given the statistic G_S, we declare D to be gerrymandered if G_S exceeds some threshold T. We detail experiments below giving ideas for how to choose this threshold T, in the analysis section.

4.3. Possible Extensions to Our Scoring Scheme

Note also that we can adopt a non-parametric approach by approximating the distribution $S(\mathcal{G})$, the pushforward measure of this Gibbs measure by the functional S (our score function), using Kernel Density Estimation (KDE)—this is because as we increase the number of samples from our chain, the Glivenko-Cantelli theorem tells us this empirical measure should converge to the desired one. We would then take G_S as above to be the percentile of $S(D)$ among the values of $S(\tilde{D}_i)$ for all $i = 1, \ldots, N$. This is similar to a p-value approach.

The issue with the KDE approach, is, of course, computational limitations. The advantage of this approach over the z-score, however, is we utilize the entire distribution of $S(\mathcal{G})$. The p-value approach above utilizes only the first and second moments of the distribution and doesn't take into account the fact that $S(\mathcal{G})$ may not be unimodal, or at the very least, skewed. We check some of these assumptions in the appendix later on, but it would be convenient if results from the literature on functionals of Gibbs measures could be applied here to understand this distribution.

Other ideas were proposed in the literature to test whether given samples from this chain are extreme, an alternate to the z-score test we detail above. In [CFP17], Chikina et al. consider a test for assessing whether a sample from an MCMC sampler is sampled according to π, the stationary distribution of the MCMC sample. The so called $\sqrt{\epsilon}$ test starts with the districting plan D under question and evolves it according to the chain above to obtain samples $D = D_0, D_1, \ldots, D_k$. Then [CFP17] proves that given a functional f, the probability that $f(D)$ is an outlier of $\{f(D_i)_{i \le k}\}$ (as some function of ϵ) is greater than $\sqrt{2\epsilon}$ under the null hypothesis that $D \in \pi$.

5. Evaluation/Results

Throughout the section, we consider the performance of our model on two distinct simulation cases, with differing political leanings for each node in the graph. We consider the political map given by the square grid graph of side length 15.

- For the first simulation procedure, we consider the case where each node's political leaning is sampled independently from Party 1 or Party 2 with a proportion of 0.45 for Party 1.
- The second simulation more closely reflects the true political landscape of the US, where we have c small population clusters dominated by Party 1, while the rest of the political landscape is mostly dominated by the other party. To simulate a political leaning of this type, we work again on a 15×15 grid, and we generate clusters of Party 2 such that the total proportion of Party one to two is still 0.45. The simulations below consider the case where the number of clusters is specified for a range of values.

5.1. Choosing the Threshold

To choose the threshold parameter at which we declare a districting plan to be gerrymandered, we do a cross validation procedure with this algorithm, where the true labels will be given to us by the output of the efficiency gap score with threshold 0.08, a common number used in the literature as mentioned in the previous work section. To set up the procedure, we first fix a political leaning on the grid. Given this, we sample using our MCMC sampler above - using a burning of 50,000 with 50,000 iterations per sample to generate $N = 100$

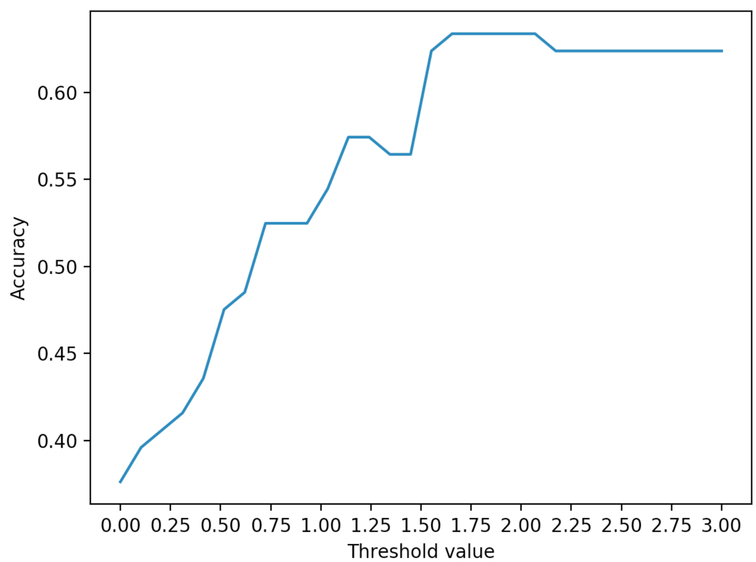
sample districting plans $\{D_i\}_{i=1}^N$. On each of these, we test whether the efficiency gap is greater than 0.08, which returns to us a 'training set' of gerrymandered districting plans. We then pick the appropriate value of T that maximizes the accuracy of the labels given by our procedure above. We implement this using a confusion matrix in our code.

We then perform the simulation above on two settings:

- A random political landscape described above. In this case, we take the number of districts to be 9.
- A realistic political landscape case, which consists of a mostly Redpublican landscape together with small "pockets" of Blumocrats. We take the number of districts to be 8.

Note: We will informally refer to these Blumocrat pockets as *islands* in the discussion that follows.

This produces the plots that follow. We can then see from the results that in the random political landscape case, we should take the threshold to be roughly $T = 1.75$. To reiterate the significance of this threshold on our pipeline, given a districting plan D on this political landscape, and if we have that $G_S(D) > T$, we consider G to be a gerrymander. The plot for the second simulation, the realistic political landscape case, suggests that we can take our threshold value for significance to be 0.75. If we had more time to sample, we would conduct this analysis on a given political landscape to find the best threshold T with which to classify districting plans on this political landscape.



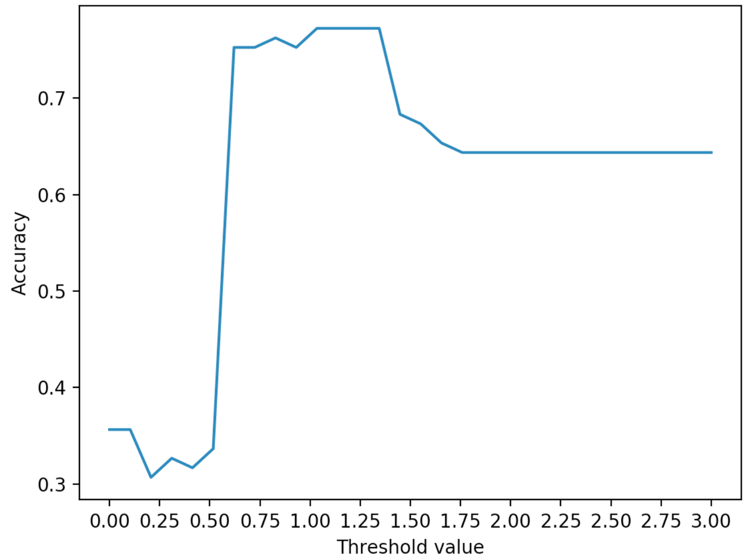


Figure 5.1: The first simulation (top image) considers the case where the political leanings are drawn uniformly at random, where the second simulation (bottom image) considers the realistic political landscape case.

5.2. Score Robustness Testing

We now tackle a different question. As a check of the robustness of our scoring function S, we want to answer if it is possible to flip an election while keeping our metric in question arbitrarily low. Call an election *flipped* if the outcome is opposite to whichever party has the majority population vote. We show in eight simulation examples that there do exist such districting plans, although there are differences in the simulation cases considered above. In particular, we show that in all cases, the mean value of S is lower for elections with the popular vote as opposed to flipped elections, which falls in line with the fact that flipped elections should be considered more gerrymandered than elections which are in line with the popular vote. However, it is clear from the plots below that there are counterexamples to this claim: we see examples of flipped elections with low scores.

In the following simulation examples, we again consider the random simulation case where we generate political leanings independently at random for each county, with proportion 0.45. We then consider the case with realistic political landscape, for the values $c = 1, 5, 15$, where recall c denotes the number of islands of Party 2. For each of these political leanings, we consider the case of different numbers of districts d which we take to be equal to 5 and 9.

In particular, consider Figures 5.2, 5.3 below: we see that in the case of $d = 5$, the cases where $c = 1, 5$, and 15 where the political landscape is city-like show the most difference in the scores of flipped vote elections versus popular vote elections. It is interesting to note that below some thresholds such as $0.02, 0.03$ we have no flipped vote elections; hence taking one of these thresholds would serve as a conservative measure of gerrymandering. Only in the random case is it difficult to distinguish the difference between these two distributions. Similar observations can be made about Figures 5.4, 5.5, with the additional remark that the variance of the scores seems to be smaller, making it slightly easier to separate the distributions of flipped vote elections and popular vote elections.

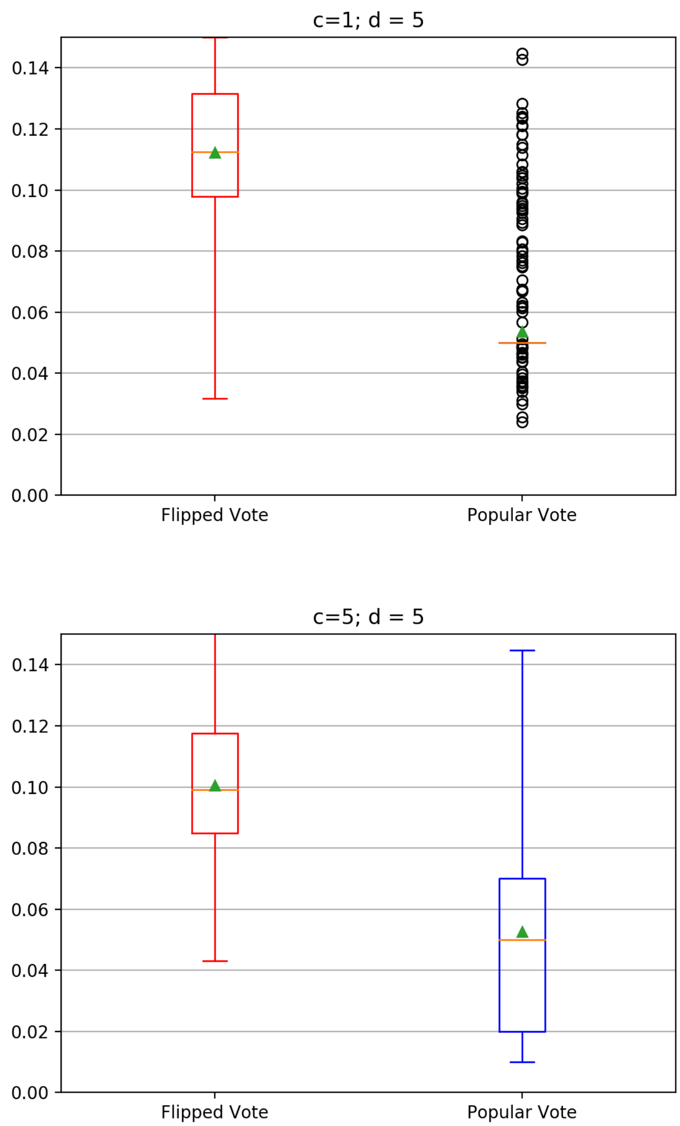


Figure 5.2: Quartile plot of gerrymander score in politically homogeneous regions in the 5-district setting, with one island (i.e., $c = 1$; top) and five islands (i.e., $c = 5$; bottom).

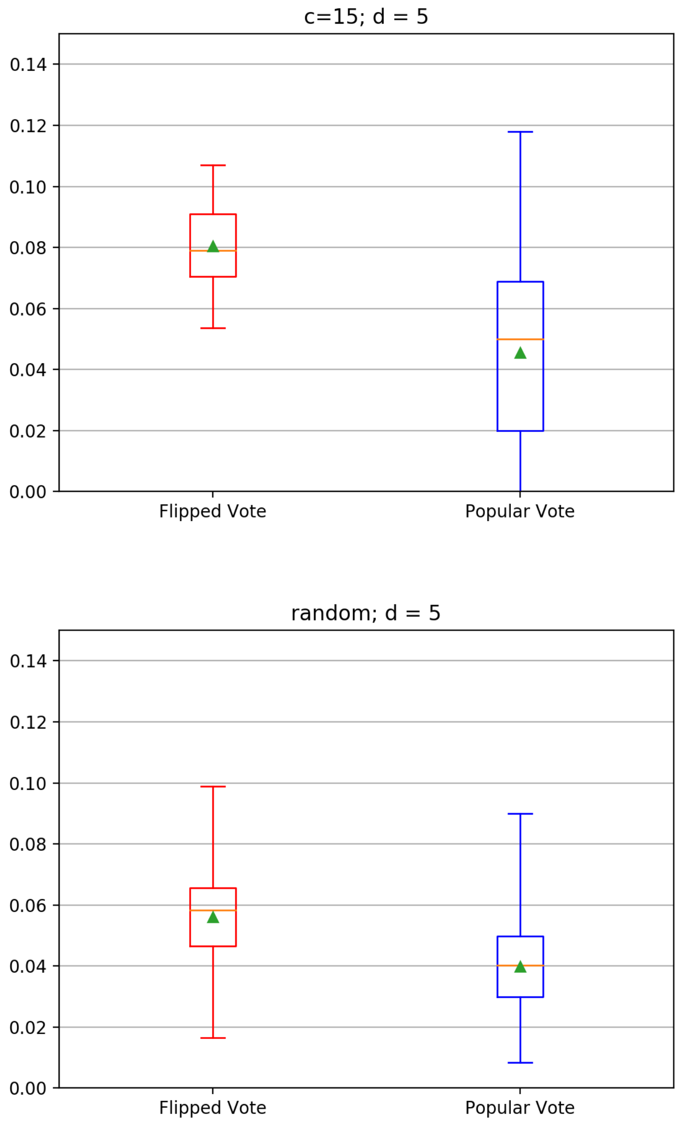


Figure 5.3: Quartile plot of gerrymander score in politically heterogeneous regions in the 5-district setting, with 15 islands (i.e., $c = 15$; top) and random distribution (bottom).

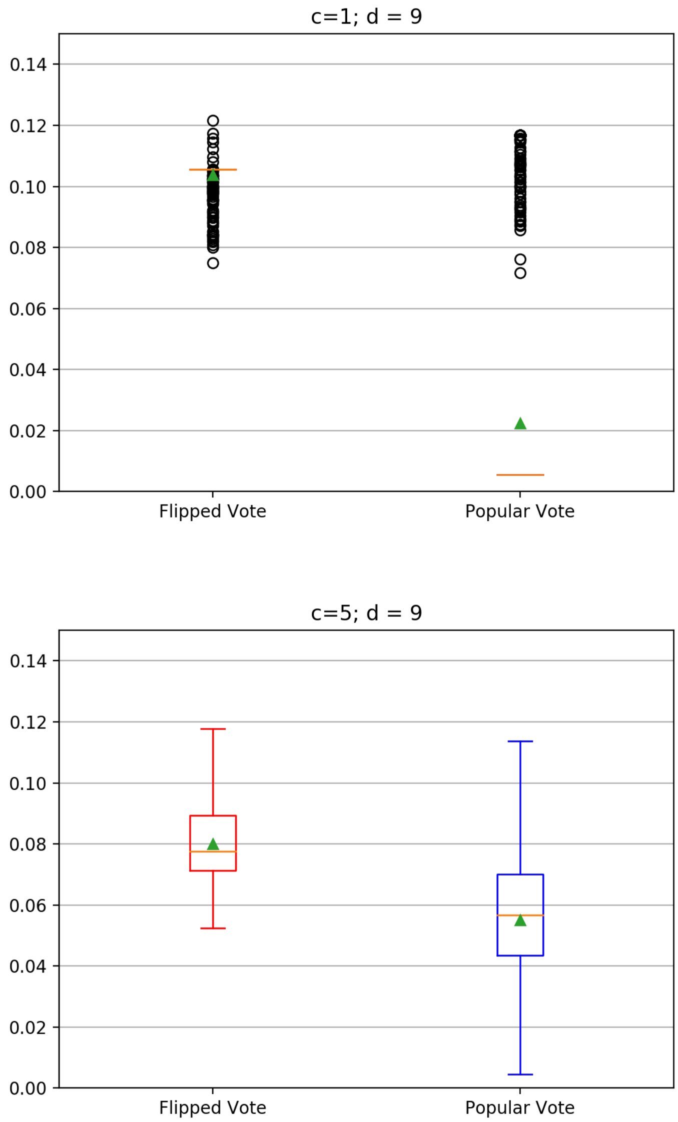


Figure 5.4: Quartile plot of gerrymander score in politically homogeneous regions in the 9-district setting, with one island (i.e., $c = 1$; top) and five islands (i.e., $c = 5$; bottom).

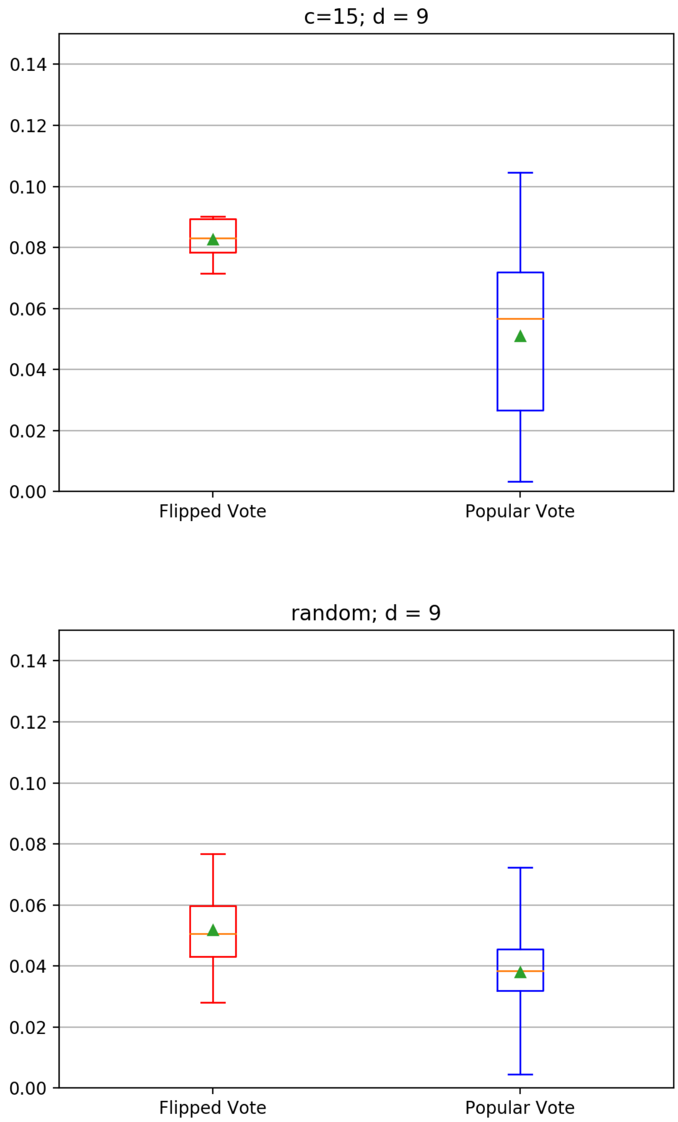


Figure 5.5: Quartile graph of gerrymander score in politically heterogeneous regions (c=15,∞) in the 9-district setting, where $c = 1$ (top) and $c = 5$ (bottom).

In conclusion, we note that in all of our simulations above, the score of a flipped vote (i.e. an instance in which the vote obtained under a districting scheme differs from the majority vote of the population) is on average higher than the score of a popular vote (i.e. an instance in which the vote obtained under a districting scheme is identical to the majority vote of the population).

While there do exist counterexamples to this claim as discussed earlier, we have strong evidence that our scoring scheme is, *on average*, robust with regard to classifying instances of flipped votes and instances of popular votes.

6. The Case of North Carolina

In this section, we evaluate the performance of our model on real-world data; specifically, we attempt to detect gerrymandering in the well-documented case of manipulative districting in the Congressional elections of North Carolina in 2012.

In Section 6.1, we describe our discretization of the geography of North Carolina, and our labelling of the political preferences on this discretized map. We then discuss the performance of our model on this graph in Section 6.2.

All the data used in this section is taken from [Str12].

6.1. Basic Setup

North Carolina has 100 counties, each of which we will represent as a node in a 10×10 grid. We account for uneven distribution in the population across counties in our discretization: 51% of the voter turnout was Blumocratic, which is represented by the 51 blue nodes in out 10×10 grid.

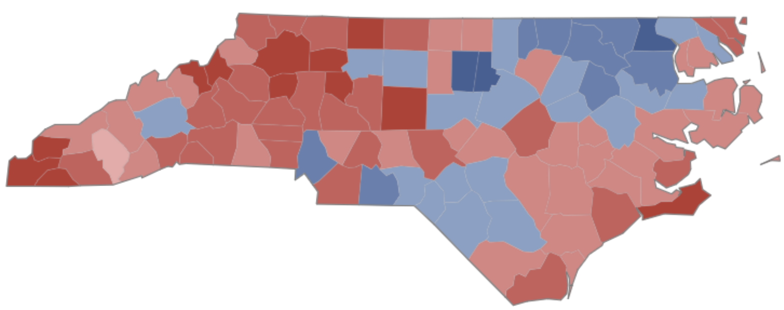


Figure 6.1: Counties in North Carolina with political leaning (2012)

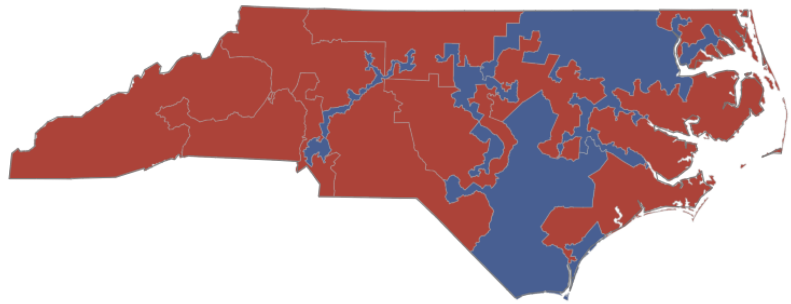


Figure 6.2: Congressional Districting of North Carolina (2012)

Our discretization of North Carolina, which we will dub "North Squarolina", is presented below, together with a skewed map of North Carolina for motivating our grid graph. We district it according to the 2012 Republican districting scheme (illustrated above).

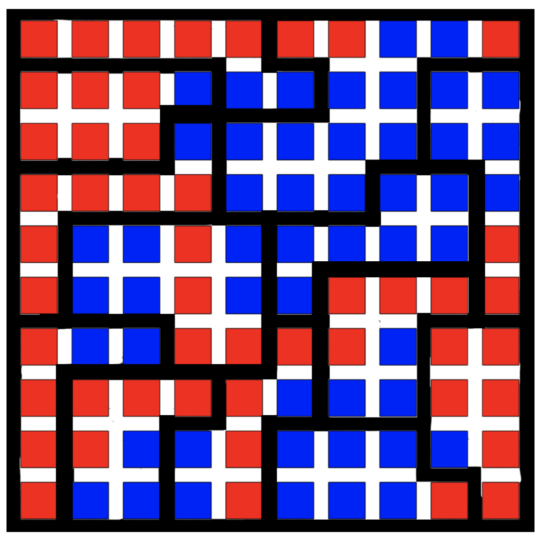


Figure 6.3: North Squarolina

The above encoding and representation of North Squarolina in electronic format for use with the code in Appendix B is available upon request.

6.2. Our Model's Performance

Our model returns $S_1(D) = 0.078$ and $S_2(D) = 0.088$, resulting in a gerrymander score of 0.078. To evaluate this result, we run our Markov chain on North Squarolina, sampling according to the parameters defined above. With over 5000 samples, we measure a standard deviation of $\sigma = 0.024$ and mean $\mathbb{E}[S(\mathcal{G})] = 0.057$. We use this to calculate Gerrymandering severity:

$$G_S = \frac{|0.078 - 0.057|}{0.024} = 0.875$$

which is above the threshold we previously defined for gerrymandering in realistically defined political districts. **Therefore, our model correctly predicts the effects of gerrymandering in the 2012 North Carolina Congressional elections.**

Note, however, that this is a weak effect relative to the threshold of $G_S = 0.75$. While there are several possible causes for the weakness of this effect, including the complicated translation from North Carolina electoral districts onto a square grid, we consider one possibility to be particularly likely: *the voting districts of North Carolina do not have equal populations*. This violates the "One Person, One Vote" assumption stated in Section 3.

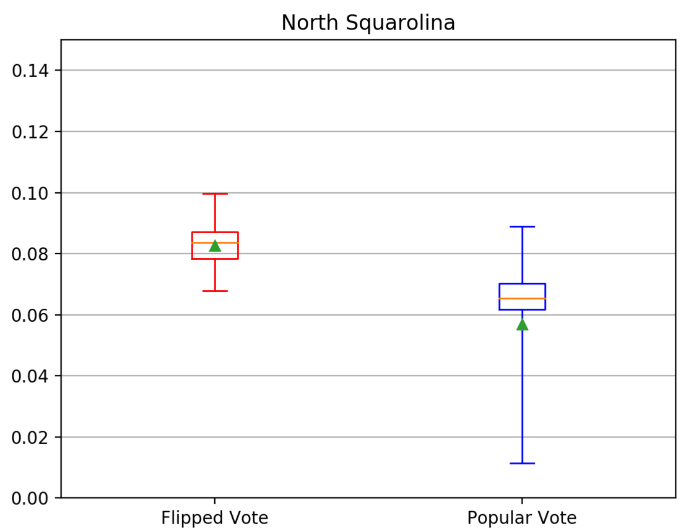


Figure 6.4: Gerrymander quartile scores for North Squarolina

7. Drawbacks and Improvements

As mentioned throughout the report, the sampling approach to the gerrymandering problem is a very general approach that can incorporate detailed information about the districting plan. Although we restricted our model to a planar graph with the same population at each node, we can extend this by directly placing more information in the Hamiltonian of the Gibbs measure of the set of districtings. Herschlag et al. in [Her+18] took the Hamiltonian to be a weighted sum information relevant to protecting minority voters, as well as geometry of the district shape, and differing district populations. This allowed the model to weight districting plans differently by these characteristics. However, the theoretical properties are still not yet well characterized, so further work into this would be illuminating [CFP17].

Given this assumption of having a Gibbs measure on the set of partitions of our graph, we looked at a certain functional of this distribution given by the scoring function we designed, although this can be swapped for any functional that measures other properties related to the gerrymandering problem.

Some further extensions of our approach could include modeling the extent to which each county leans towards Party 1 or Party 2. This model was approached by the Princeton group on gerrymandering [Wan]. Another interesting extension is to add stochastic elements to each county, with the possibilities of ties, or perhaps, low voter turnout—an issue which is extremely relevant to our modern political times.

8. Conclusion

Gerrymandering is an old problem with a complicated definition, intricately tied up with the law, as well as the political and geographic landscape. While this report gave an overview of the numerous methods that quantify and analyze gerrymandering, we chose to focus on the extremely general sampling approach, which can be recast as special cases of other models in the literature. As a departure from these methods however, we chose a functional approach to the problem, which summarized the samples from our Monte Carlo sampler using a scoring function and an associated z-score characterizing the extremity of a given score. We chose the scoring function in order to capture two different cases of districting plans which we deemed were tolerable, but may have been considered gerrymandering according to other metrics.

We then analyzed some aspects of this model, showing how to sample from a Gibbs measure on the space of districting plans, how to calibrate our model against existing metrics, and whether our scoring function was robust or not. Although we saw examples for which an election flipped while having a low score, examples of these districting plans are not common in the simulation case that mirrors the true American political landscape, which suggests that our scoring function is robust in application.

As a successful application of our model, we were able to show that the 2012 North Carolina House Election was indeed gerrymandered. We could make these results even more powerful if we had more resources: we could have considered the full KDE approach described in the model section, instead of defining the z-score of our quantity S, which may not be representative in the case where this distribution is not unimodal or is heavily skewed. This would likely have improved the significance of our results. But even with this first approximation, our result still demonstrates the efficacy of our model.

A. Acknowledgements

The authors would like to thank Moon Duchin and the Metric Geometry and Gerrymandering Group at Tufts University for promptly responding to a question about the GerryChain package, and also the BMCM 2018 organizers and graders for their time and effort.

B. Code

B.1. Generating and Scoring Districtings

```
import networkx

import gerrychain.partition

from gerrychain.partition import Partition
from gerrychain.partition import GeographicPartition
from gerrychain.proposals import propose_random_flip
from gerrychain.defaults import Grid

from gerrychain import MarkovChain
from gerrychain.constraints import Validator, single_flip_contiguous,
                                   new_constraint
from gerrychain.proposals import propose_random_flip
from gerrychain.accept import always_accept

import numpy as np
from math import sqrt
import matplotlib.pyplot as plt
import math
from termcolor import colored
import random

# partition n into k distinct numbers
def partitionfunc(n, k, l = 1):
    if k < 1:
        raise StopIteration
    if k == 1:
        if n >= l:
            yield (n,)
        raise StopIteration
    for i in range(l,n+1):
        for result in partitionfunc(n-i,k-1,i):
            yield (i,)+result

# return a random element from an iterable
def iter_sample_fast(iterable, samplesize):
    results = []
    for i, v in enumerate(iterable):
        r = random.randint(0, i)
        if r < samplesize:
            if i < samplesize:
                results.insert(r, v) # add first samplesize items in random
                                       order
            else:
                results[r] = v # at a decreasing rate, replace random items

    if len(results) < samplesize:
        raise ValueError("Sample larger than population.")

    return results

# generate political leanings that reflect cities
# and real-world political leanings
def bubble_leanings(size, proportion, number_bubbles):
```

```
    side_1 = size[0]
    side_2 = size[1]
    total_blue = int(math.floor(side_1 * side_2 * proportion))

    city_sizes = iter_sample_fast(partitionfunc(total_blue, number_bubbles),
                                       1)[0]

    biggest_bubble = max(city_sizes)

    leaning_matrix = np.zeros([side_1, side_2]).flatten() # zero = red

    positions = np.random.choice(side_1*side_2 - biggest_bubble,
                                       number_bubbles)

    # one = blue
    for i in range(number_bubbles):
        size = city_sizes[i]
        start = positions[i]

        for j in range(size):
            leaning_matrix[start + j] = 1

    leaning_matrix = np.reshape(leaning_matrix[:side_1*side_2], (side_1,
                                       side_2))
    leaning_matrix[1::2, :] = leaning_matrix[1::2, ::-1]

    return leaning_matrix.astype(int)

def get_political_leanings(size, num_classes=2):
    # return np.random.randint(0, num_classes, size)
    return np.random.choice([0, 1], size=size, p=[.45, .55])

def get_polarity(grid, num_classes=2):
    areas = np.zeros(num_classes)
    for row in grid:
        for value in row:
            areas[value] += 1
    if (num_classes == 2):
        return areas[0]/(np.sum(areas))

# partition = districts
# grid = political leanings
def get_district_vote(assignment, leanings_grid, num_classes=2, num_districts
                                    =4):
    district_votes = np.zeros((num_districts, num_classes))
    for node in assignment:
        district = assignment[node]
        leaning = leanings_grid[node]
        district_votes[district][leaning] += 1
    return district_votes

def get_district_polarities(district_votes):
    polarities = [areas[0]/np.sum(areas) for areas in district_votes]
    return polarities

def delta_dvec(polarities, polarity):
    running_sum = 0
    for p in polarities:
        running_sum += abs(p - polarity) ** 2
    running_sum = sqrt(running_sum)
```

```
    return running_sum/len(polarities)

def get_vote(district_votes):
    return [areas[0] > areas[1] for areas in district_votes]

def delta_d(votes, polarity):
    return abs(sum(votes)/len(votes) - polarity)

def flipped_vote(votes, polarity):
    # Returns true if district and popular votes are different:
    if ((polarity < 0.5) and (sum(votes)/len(votes) > 0.5)) or ((polarity > 0
                                        .5) and (sum(votes)/len(votes) < 0
                                        .5)):
        return True
    else:
        return False

def district_assignment(size, districts):

    side_1 = size[0]
    side_2 = size[1]

    population = side_1 * side_2
    district_size = math.floor(population/districts)

    assignment_matrix = np.zeros([side_1, side_2]).flatten()
    bonus = 1

    for i in range(population):
        assignment = math.floor(i/district_size) + 1

        if assignment > districts:

            assignment_matrix = np.insert(assignment_matrix, (bonus)*(
                                                district_size + 1) - 1,
                                                bonus)
            bonus = (bonus + 1) % districts

        else:
            assignment_matrix[i] = assignment

    assignment_matrix = np.reshape(assignment_matrix[:side_1*side_2], (side_1
                                    , side_2))
    assignment_matrix[1::2, :] = assignment_matrix[1::2, ::-1]
    assignment_matrix -= np.ones(size)

    assignment_dict = dict()

    for i in range(side_1):
        for j in range(side_2):
            assignment_dict[(i, j)] = int(assignment_matrix[(i, j)])

    return assignment_dict

def print_partition(partition, leanings, size):
    matrix = np.zeros(size)
    for key in partition.assignment:
        matrix[key] = partition.assignment[key]
    output = ""
    for i, row in enumerate(matrix):
```

```
        for j, value in enumerate(row):
            leaning = leanings[i, j]
            if leaning == 0:
                output += colored(int(value), 'red')
            else:
                output += colored(int(value), 'blue')
        output += "\n"
    print(output)
```

B.2. Calculating the Efficiency Gap

```
# political_leanings is dictionary with tuples of coordinates as keys, value
                                       is the political leaning, zero or one
# districting_plan is dictionary with tuples of coordinates as keys, value is
                                        the region.

def efficiency_gap(districting_plan, political_leanings):

  total_votes = len(political_leanings)

  # convert the districting plan into a dict
  districts = {}
  for (key, value) in districting_plan.items():
    if value in districts.keys():
      districts[value] = districts[value] + [key]
    else:
      districts[value] = [key]


  efficiency_gap = []

  # for each district calculate the wasted votes
  for district, counties in districts.items():
    district_size = len(counties)

    votes = [political_leanings[county] for county in counties]

    party_1_votes =  (district_size - sum(votes))
    party_2_votes =  sum(votes)


    # if party 1 wins
    if party_1_votes >= party_2_votes:
      party_1_wasted_votes = party_1_votes - district_size/2
      party_2_wasted_votes = party_2_votes
    else:
      party_1_wasted_votes = party_1_votes
      party_2_wasted_votes = party_2_votes - district_size/2

    efficiency_gap = efficiency_gap + [party_1_wasted_votes -
                                       party_2_wasted_votes]


  return abs(sum(efficiency_gap))/total_votes


def compute_confusion_matrix(labels, values ,threshold):
  thresholded_values = [1 if value > threshold else 0 for value in values  ]
```

```
  return confusion_matrix(labels, thresholded_values)

def compute_performance_curve(labels,values):
  curve = []
  for threshold in np.linspace(0,3,60):
    cf = compute_confusion_matrix(labels,values,threshold)
    performance = (cf[0,0] + cf[1,1])/sum(sum(cf))
    curve = curve + [performance]

  return np.array([np.linspace(0,3,60),np.array(curve)])
```

C. References

[AM17] Boris Alexeev and Dustin G Mixon. "An impossibility theorem for gerrymandering". In: *arXiv preprint arXiv:1710.04193* (2017).

[BD17a] M. Bernstein and M. Duchin. "A formula goes to court: Partisan gerrymandering and the efficiency gap". In: *ArXiv e-prints* (May 2017). arXiv: 1705.10812 [physics.soc-ph].

[BD17b] Mira Bernstein and Moon Duchin. "A formula goes to court: Partisan gerrymandering and the efficiency gap". In: *Notices of the AMS* 64.9 (2017), pp. 1020–1024.

[BS18] R. Barnes and J. Solomon. "Gerrymandering and Compactness: Implementation Flexibility and Abuse". In: *ArXiv e-prints* (Mar. 2018). arXiv: 1803.02857 [cs.CY].

[CFP17] Maria Chikina, Alan Frieze, and Wesley Pegden. "Assessing significance in a Markov chain without mixing". In: *Proceedings of the National Academy of Sciences* 114.11 (2017), pp. 2860–2864.

[CLR18] Amittai Cohen-Zemach, Yoad Lewenberg, and Jeffrey S Rosenschein. "Gerrymandering Over Graphs". In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2018, pp. 274–282.

[DeF+18] D. DeFord et al. "Total Variation Isoperimetric Profiles". In: *ArXiv e-prints* (Sept. 2018). arXiv: 1809.07943 [math.OC].

[DT18a] M. Duchin and B. E. Tenner. "Discrete geometry for electoral geography". In: *ArXiv e-prints* (Aug. 2018). arXiv: 1808.05860 [physics.soc-ph].

[DT18b] Moon Duchin and Bridget Eileen Tenner. "Discrete geometry for electoral geography". In: *arXiv preprint arXiv:1808.05860* (2018).

[Duc18] M. Duchin. "Gerrymandering metrics: How to measure? What's the baseline?" In: *ArXiv e-prints* (Jan. 2018). arXiv: 1801.02064 [physics.soc-ph].

[Her+18] G. Herschlag et al. "Quantifying Gerrymandering in North Carolina". In: *ArXiv e-prints* (Jan. 2018). arXiv: 1801.03783 [physics.soc-ph].

[Ing15] Christopher Ingraham. *This is the best explanation of gerrymandering you will ever see*. Mar. 2015. URL: https://www.washingtonpost.com/news/wonk/wp/2015/03/01/this-is-the-best-explanation-of-gerrymandering-you-will-ever-see/?utm_term=.c0911b04d7aa.

[MAS+14] ERIC MASKIN et al. *The Arrow Impossibility Theorem*. Columbia University Press, 2014. URL: http://www.jstor.org/stable/10.7312/mask15328.

[PT09] Clemens Puppe and Attila Tasnádi. "Optimal redistricting under geographical constraints: Why "pack and crack" does not work". In: *Economics Letters* 105.1 (2009), pp. 93–96.

[Str12] Stephanie Strom. *North Carolina–Election 2012*. https://www.nytimes.com/elections/2012/results/states/north-carolina.html. Accessed: 2018-11-11. 2012.

[Wan] S. Wang. *Princeton Gerrymandering Project*. URL: http://gerrymander.princeton.edu/info/.