

Eastern Kentucky University

Encompass

---

Online Theses and Dissertations

Student Scholarship

---

January 2019

## Examining Partisan Advantage In Congressional Maps Using Simulations Based On Election Data

Zachary James Morgan  
*Eastern Kentucky University*

Follow this and additional works at: <https://encompass.eku.edu/etd>



Part of the [American Politics Commons](#), and the [Statistics and Probability Commons](#)

---

### Recommended Citation

Morgan, Zachary James, "Examining Partisan Advantage In Congressional Maps Using Simulations Based On Election Data" (2019). *Online Theses and Dissertations*. 578.  
<https://encompass.eku.edu/etd/578>

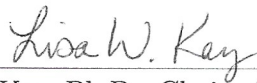
This Open Access Thesis is brought to you for free and open access by the Student Scholarship at Encompass. It has been accepted for inclusion in Online Theses and Dissertations by an authorized administrator of Encompass. For more information, please contact [Linda.Sizemore@eku.edu](mailto:Linda.Sizemore@eku.edu).

EXAMINING PARTISAN ADVANTAGE IN CONGRESSIONAL MAPS  
USING SIMULATIONS BASED ON ELECTION DATA

BY

ZACHARY MORGAN

THESIS APPROVED:



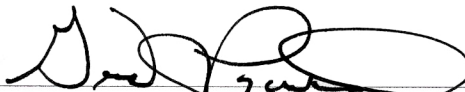
\_\_\_\_\_  
Lisa Kay, Ph.D., Chair, Advisory Committee



\_\_\_\_\_  
Shane Redmond, Ph.D., Member, Advisory Committee



\_\_\_\_\_  
Michelle Smith, Ph.D., Member, Advisory Committee



\_\_\_\_\_  
Jerry Pogatshnik, Ph.D., Dean, Graduate School

## STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a M.S. degree at Eastern Kentucky University, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made. Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or, in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Zachary J. Morgan

Date 1/16/19

EXAMINING PARTISAN ADVANTAGE IN CONGRESSIONAL MAPS  
USING SIMULATIONS BASED ON ELECTION DATA

BY

ZACHARY MORGAN

B.S. Statistics  
Eastern Kentucky University  
Richmond, KY  
2016

Submitted to the Faculty of the Graduate School of  
Eastern Kentucky University  
in partial fulfillment of the requirements  
for the degree of  
MASTER OF SCIENCE  
2019

Copyright ©ZACHARY MORGAN, 2019  
All rights reserved

## DEDICATION

I dedicate this thesis to my wonderful wife, Bridgit Morgan, who always supports me in everything I do.

## ACKNOWLEDGEMENTS

First, I would like to thank my thesis and academic advisor, Dr. Lisa Kay. Her guidance has proved invaluable during my time in the graduate program and throughout the thesis process. I would also like to thank the members of my thesis committee, Dr. Shane Redmond and Dr. Michelle Smith who have spent a great deal of time sitting down with me and providing me with excellent feedback, without which this thesis would not have been possible. Additionally, I want to thank Dr. Jason Gibson and Dr. Steve Szabo who assisted me in selecting a thesis topic. Finally, I want to thank the entire Department of Mathematics and Statistics at ECU for providing me a great learning environment for over five years.

## ABSTRACT

Partisan gerrymandering has been and will continue to be a topic of interest in the coming years. States will soon begin their redistricting process following the 2020 Census. We introduce a method of simulating Congressional elections which provides a new way of examining and visualizing the votes-to-seats relationship for a state Congressional map using past election data. We are able to build upon Mira Bernstein's method of uniformly simulating elections by injecting a data-driven component of variation into the simulations. Additionally, we are able to directly evaluate the accuracy of our simulations using a type of cross-validation. We compare our results from a handful of notable states to other measures of partisan gerrymandering, such as the efficiency gap, and do so in light of recent court cases and other important contexts.



## Table of Contents

Chapter	Page
1 Introduction . . . . .	1
2 Existing Methods . . . . .	5
2.1 Efficiency Gap . . . . .	5
2.2 Three Tests . . . . .	6
3 Congressional Election Data . . . . .	9
3.1 Source and Import Process . . . . .	9
3.2 Exclusion Criteria . . . . .	9
3.3 Percentiles of Discrete Data . . . . .	10
4 Simulations . . . . .	11
4.1 Main Objective . . . . .	11
4.2 Simulation Process . . . . .	12
4.2.1 Shifts and Residuals . . . . .	12
4.2.2 Performing and Visualizing the Simulation . . . . .	16
4.2.3 Seed Values . . . . .	22
4.2.4 Effect of Using Nationwide Residuals . . . . .	22
4.2.5 Accuracy of Simulation Process . . . . .	24
4.3 Simulation of Certain States . . . . .	26
4.3.1 Pennsylvania . . . . .	26
4.3.2 Wisconsin . . . . .	30
4.3.3 Maryland . . . . .	33
4.3.4 North Carolina . . . . .	36
4.3.5 Arizona . . . . .	39
4.3.6 Ohio . . . . .	42
5 Conclusions . . . . .	45

6	Ideas for Future Research . . . . .	46
6.1	Possible Test for Partisan Asymmetry . . . . .	46
6.2	State Legislatures . . . . .	46
6.3	Analysis of Residuals . . . . .	46
6.4	Effect of Fewer Data Restrictions . . . . .	47
	References . . . . .	48
	APPENDICES . . . . .	50
A	Additional Tables . . . . .	51
B	R Scripts . . . . .	54

List of Tables

Table	Page
2.1 Efficiency Gap Scenario . . . . .	6
2.2 Modified Efficiency Gap Scenario . . . . .	6
4.1 Result of Kentucky’s 2014 Congressional Elections . . . . .	11
4.2 Shifts Between Kentucky’s 2012 and 2014 Congressional Elections . . .	13
4.3 Results of Simulation . . . . .	17
4.4 Cross-Validation Results for Least Likely Outcomes . . . . .	26
A.1 Full Cross-Validation Results . . . . .	52

## List of Figures

Figure	Page
4.1 Bernstein Shift of Congressional Elections in Kentucky . . . . .	12
4.2 Histogram and Normal Probability Plot of Residual Shifts . . . . .	14
4.3 Boxplots of Residual Republican Shift by Year . . . . .	15
4.4 Boxplots of Residual Republican Shift by State . . . . .	15
4.5 Partial Scatter Plot of Statewide Vote and Seats Won in Kentucky . .	17
4.6 Scatter Plot of Statewide Vote and Seats Won in Kentucky . . . . .	18
4.7 Step Chart of Statewide Vote and Seats Won in Kentucky . . . . .	19
4.8 Scatter Plot of Statewide Vote and Seats Won by Party in Kentucky .	20
4.9 Step Chart of Statewide Vote and Seats Won by Party in Kentucky . .	21
4.10 Bar Chart of Districts Won and Efficiency Gap in Kentucky . . . . .	21
4.11 Plot of Residual Variance Versus Percent Mean Disagreement . . . . .	23
4.12 Step Chart of Statewide Vote and Seats Won in Nebraska . . . . .	24
4.13 Scatter Plot of Statewide Vote and Seats Won in Pennsylvania . . . . .	27
4.14 Bernstein-Style Simulations of Elections in Pennsylvania . . . . .	28
4.15 Step Chart of Statewide Vote and Seats Won by Party in Pennsylvania	29
4.16 Bar Chart of Districts Won and Efficiency Gap in Pennsylvania . . . . .	29
4.17 Scatter Plot of Statewide Vote and Seats Won in Wisconsin . . . . .	31
4.18 Bernstein-Style Simulations of Elections in Wisconsin . . . . .	31
4.19 Step Chart of Statewide Vote and Seats Won by Party in Wisconsin . .	32
4.20 Bar Chart of Districts Won and Efficiency Gap in Wisconsin . . . . .	32
4.21 Scatter Plot of Statewide Vote and Seats Won in Maryland . . . . .	33
4.22 Bernstein-Style Simulations of Elections in Maryland . . . . .	34
4.23 Step Chart of Statewide Vote and Seats Won by Party in Maryland . .	34
4.24 Bar Chart of Districts Won and Efficiency Gap in Maryland . . . . .	35
4.25 Scatter Plot of Statewide Vote and Seats Won in North Carolina . . . . .	36
4.26 Bernstein-Style Simulations of Elections in North Carolina . . . . .	37

4.27	Step Chart of Statewide Vote and Seats Won by Party in North Carolina	37
4.28	Bar Chart of Districts Won and Efficiency Gap in North Carolina . . .	38
4.29	Scatter Plot of Statewide Vote and Seats Won in Arizona . . . . .	40
4.30	Bernstein-Style Simulations of Elections in Arizona . . . . .	40
4.31	Step Chart of Statewide Vote and Seats Won by Party in Arizona . . .	41
4.32	Bar Chart of Districts Won and Efficiency Gap in Arizona . . . . .	41
4.33	Scatter Plot of Statewide Vote and Seats Won in Ohio . . . . .	42
4.34	Bernstein-Style Simulations of Elections in Ohio . . . . .	43
4.35	Step Chart of Statewide Vote and Seats Won by Party in Ohio . . . . .	43
4.36	Bar Chart of Districts Won and Efficiency Gap in Ohio . . . . .	44

## 1. Introduction

The United States Constitution requires that federal elections be held every two years to elect members to the U.S. House of Representatives. Each of the 435 voting members of the House is elected by voters from predefined, distinct Congressional districts. Each district elects only a single representative, making the elections winner-take-all. This means, for example, that a district with 60% of the vote going to the Republican candidate would yield the same number of seats, 1, as a district where the Republican won 90% of the vote. Every state is allocated a specific number of districts according to their population as enumerated by the United States Census which is conducted every 10 years. State governments are given the power to draw their own district boundaries with the following conditions required by federal law:

1. Districts must have roughly equal populations as recorded by the latest census.
2. The Congressional map must conform to standards defined by the Voting Rights Act of 1965.

Several states have additional conditions, such as compactness or contiguity, which they require of their own maps. These conditions will not be the focus here, however. Instead, we will focus primarily on the partisan fairness of state Congressional maps. There are many different ways one could define fairness in this context, many of which revolve around the relationship between votes and seats. Should a map be considered fair if this relationship is proportional? If party A wins  $X\%$  of the vote, should it be expected to win  $X\%$  of the seats? Evidence suggests this expectation is likely unreasonable. Tufte (1973) noted that a lack of proportionality occurs quite often, and naturally, in winner-take-all elections. More specifically, a party who wins a majority of the vote will usually win an even larger majority of seats. We refer to this phenomenon as a “winner’s bonus.”

Definition 1.1 (Winner’s Bonus). The “extra” proportion of seats won by a party who won a majority of the statewide vote. We can compute this explicitly as

$$\text{Winner’s Bonus} = \text{Seats Won (\%)} - \text{Votes (\%)}. \quad (1.1)$$

Since the states are in charge of drawing their own Congressional maps, they each have their own protocol for doing so. Most have their maps drawn and voted upon by partisan bodies (*Who Draws the Maps? Legislative and Congressional Redistricting*, 2018), meaning a controlling party can draw maps which afford them a higher winner’s bonus. The difficulty lies in determining when this process goes too far and produces an unfair map, that is, determining when gerrymandering has taken place.

Definition 1.2 (Gerrymandering). The process of dividing a geographical area into political districts with the intent of providing a political advantage to a certain group or party.

There are two main types of gerrymandering often focused upon in the United States. Racial gerrymandering is the drawing of districts with the intent to diminish the voting power of one or more racial minorities. The aforementioned Voting Rights Act of 1965 exists, in part, as an effort to curb this kind of gerrymandering. We will focus on the other type, partisan gerrymandering, which is the drawing of districts with the intent to benefit one political party. In the United States Supreme Court case, *Davis v. Bandemer* (1986), Indiana Democrats argued that the state’s legislature was unfairly apportioned to weaken the voting power of Democratic voters. While the Court ultimately disagreed with the plaintiffs, they did maintain that partisan gerrymandering is justiciable under the Equal Protection Clause of the 14th Amendment to the United States Constitution. In later cases, the court has held that some political gerrymandering is acceptable and, as Justice Scalia wrote in *Vieth v. Jubelirer* (2004), the difficulty is “determining when political gerrymandering has gone too far.”

The court has established that any test for partisan gerrymandering should be able to show both intent and effects. That is, it must be demonstrated that a Congressional map was drawn with the intent to favor one political party over another and that the Congressional map creates an unfair distribution of voting power for a party not in control of redistricting. One of the reasons the intent component is important is the existence of natural gerrymandering. Due largely to patterns in human geography, certain parties can be placed at an inherent disadvantage. This phenomenon was examined thoroughly by Chen, Rodden, et al. (2013).

The Court holds that partisan gerrymandering can be identified by a partisan asymmetry. In other words, if the statewide vote were flipped to favor the other party instead, the map would be considered asymmetric if the other party were not expected to win the same number of seats as the first party did. For example, if Republicans win an average of 70% of seats in a state's Congressional election with 60% of the vote, Democrats should be expected to win an average of 70% of the seats if they, instead, had received 60% of the vote. If this were not the case, the map would be considered asymmetric.

Initially our hope was to derive a statistical test for partisan asymmetry of a Congressional map. Such a test would have examined the relationship between statewide vote and seats won for both major parties, returning a  $p$ -value which could be used to determine if there was a significant difference between the numbers of seats won for the two parties given any given statewide vote. Due to the limited number of elections which take place under each Congressional map, this ultimately proved too difficult a task.

Instead, we chose to re-focus our efforts on a new way of simulating statewide Congressional elections using historical election data. We use the simulations to examine and visualize the relationship between statewide vote and seats won, the same relationship we were interested in statistically testing. We use a type of cross-validation to evaluate the accuracy of the simulations. Finally, we compare our



results from a handful of notable states to other measures of partisan asymmetry, such as the efficiency gap, and do so in light of recent court cases and other important contexts.

## 2. Existing Methods

### 2.1. Efficiency Gap

The efficiency gap, first proposed in an article by researchers Nicholas Stephanopoulos and Eric McGhee (2015), is centered around the idea of wasted votes, that is, which party wastes more votes in an effort to elect their candidates in a certain state. A vote is considered wasted if it is cast for a losing candidate, or if it is cast for a winning candidate who would have won without it. The efficiency gap with respect to party A is the advantage in wasted seats as a percentage of statewide vote. Let  $T_A$  and  $S_A$  be the percentage of votes and seats respectively won by party A. If we assume that the districts have the same population size, a condition already required by the Constitution, and that there are only two parties, the simplified version of the formula with respect to party A is

$$\text{Efficiency Gap} = (S_A - 50\%) - 2(T_A - 50\%). \quad (2.1)$$

Positive values would indicate an electoral advantage for party A while negative values would suggest a disadvantage.

Example 2.1. Consider the hypothetical scenario shown in Table 2.1. The Republicans won five out of 10, or 50% of the seats and 476 out of 1000, or 47.6% of the votes, meaning the efficiency gap for this election is  $(50\% - 50\%) - 2(47.6\% - 50\%) = 4.8\%$

Example 2.2. Now consider the slightly altered hypothetical scenario shown in Table 2.2. Note that the only difference from the previous example is that Democrats won the 5th district 51-49 instead of a Republican victory by the same margin. Here, the Republicans won 40% of the seats and 47.4% of the vote. Thus, the efficiency gap will be  $(40\% - 50\%) - 2(47.4\% - 50\%) = -4.8\%$

These examples illustrate how susceptible the efficiency gap, like any other measure, can be to variation. A 2% shift in a single district produced a 9.6%

Table 2.1: Efficiency Gap Scenario

	Republican Votes	Democratic Votes	Total Votes
1st District	75	25	100
2nd District	75	25	100
3rd District	70	30	100
4th District	55	45	100
5th District	51	49	100
6th District	45	55	100
7th District	35	65	100
8th District	25	75	100
9th District	25	75	100
10th District	20	80	100
Total	476	524	1000

Table 2.2: Modified Efficiency Gap Scenario

	Republican Votes	Democratic Votes	Total Votes
1st District	75	25	100
2nd District	75	25	100
3rd District	70	30	100
4th District	55	45	100
5th District	49	51	100
6th District	45	55	100
7th District	35	65	100
8th District	25	75	100
9th District	25	75	100
10th District	20	80	100
Total	474	526	1000

difference in the efficiency gap. As mentioned earlier, the court has held that some partisan gerrymandering is acceptable and the main difficulty is determining how much is too much. With respect to the efficiency gap, the authors suggest thresholds of  $\pm 2$  Congressional seats be used for identifying states whose maps deviate from the norm, roughly corresponding to  $\pm 1.5$  standard deviations.

## 2.2. Three Tests

In Samuel Wang's (2016) Stanford Law Review article, he proposes three additional tests for measuring partisan gerrymandering. We will not be employing any of these tests in our work; however, they are worth mentioning for their statistical nature. A key component of the article was to establish what Wang

referred to as a “zone of chance,” or an arbitrarily wide interval of possible outcomes which could have occurred due to random variation alone. He established a zone of chance for the number of seats won given a statewide vote by simulating delegations. For any given state with  $N$  districts, Wang simulated delegations by randomly selecting  $N$  results from the 435 nationwide Congressional elections which added to the same vote totals (within 0.5%). The way in which these simulations were conducted would indicate data which are reflective of nationwide district characteristics rather than the districts within the state of interest. All three of the tests Wang proposed, which are described below, are statistical in nature in that they take random variation into account.

1. Excess seats test: As the name would suggest, this test focuses on the proportion of seats won by a party in excess of the statewide proportion vote for that party or, in essence, the winner’s bonus for that party. The test statistic in this case is calculated by taking the winner’s bonus and dividing it by the estimate of the standard deviation extracted from the aforementioned simulations. In context, this test statistic would represent a standardized measure of departure from the nationwide vote-to-seat relationship. As Wang mentions, one disadvantage of this test is that it is not self-contained; i.e., it requires nationwide election data for all 435 districts to test a single state.
2. Lopsided outcomes test: This is by far the simplest of the three tests. It involves using a grouped  $t$ -test to compare the share of Democratic votes in Democratic districts to the share of Republican votes in Republican districts. Higher shares of votes in districts won by a party would be an indication of “packing” voters of that party together, thus weakening their electoral power. One obvious advantage of this test is its simplicity: it is self-contained, requires no simulations, and uses only elementary statistical techniques. One disadvantage Wang describes is that it does a poor job of detecting bipartisan gerrymandering, which is redistricting with the intent to protect incumbents for both parties.

3. Reliable wins test: The third and final proposed test considers the number of reliable wins, or protection, for the party in charge of redistricting. This test is performed one of two different ways, depending on how competitive the state is. If the state is closely divided, then a statistical test is performed to determine if the mean vote across districts is higher than the median district vote. If the state is dominated by the party in charge of redistricting, then a statistical test is performed to see if the variances are different between the winning vote shares of that party at the state and national levels.

### 3. Congressional Election Data

#### 3.1. Source and Import Process

Historical Congressional Election data from the years 2004 through 2016 were acquired from the Federal Elections Commission website. The data for these years were available for download in .xls or .xlsx format. The downloaded data files were opened in Microsoft Excel and saved as .csv files.

The data were then imported using two R scripts, the first of which took the data and combined them into a single raw data set with the objective of matching our desired row structure. Our smallest units of interest are district-level general election results by party, and the FEC data files were structured so that there was a single row of data for each candidate in every Congressional primary and general election. Our first script modified the FEC data structure to yield that desired structure in the raw data set. The second R script modified our raw data set to produce an analysis data set with the desired column structure for our ensuing simulation and analysis. This primarily involved the creation and modification of variables of interest. Both of the R scripts are included in the Appendix.

#### 3.2. Exclusion Criteria

In the context of gerrymandering, it does not make sense to examine states with an at-large, or single, Congressional district since there would be no Congressional map to draw in those cases. Instead we will focus only on those states with 2 or more districts. Additionally, due to the unusual nature of Louisiana's election system, it will be excluded from our analysis. This leaves 42 states to be examined.

We are primarily interested in the votes-to-seats relationship for the two major political parties in the U.S., Republicans and Democrats. Thus, we will only examine elections where the third-party vote is less than 5%. In order to correct for the presence of some third-party vote in our data, the Republican and Democratic vote percentages are adjusted by calculating each as

$$\text{Republican Vote (\%)} = 100 \times \frac{\text{Republican Votes}}{\text{Republican Votes} + \text{Democratic Votes}} \text{ and (3.1)}$$

$$\text{Democratic Vote (\%)} = 100 \times \frac{\text{Democratic Votes}}{\text{Republican Votes} + \text{Democratic Votes}}. \quad (3.2)$$

### 3.3. Percentiles of Discrete Data

There are several algorithms which can be used to compute the percentiles of data. There are nine algorithms which can be used with the *quantile* command in R. These algorithms were discussed and evaluated by Hyndman and Fan (1996). By default, the command uses the seventh definition discussed which involves the linear interpolation of the modes of order statistics. We will use this type for any percentile calculations. Since some of our data, such as the number of districts won, is discrete, we will round the percentiles whenever applicable to mirror the discrete structure of our data.

## 4. Simulations

### 4.1. Main Objective

In order to assess the asymmetry of a Congressional map, it is helpful to examine the relationship between votes and seats. The efficiency gap provides a measure of this relationship, but we can also visualize it directly by plotting the elections with votes on the horizontal axis and seats on the vertical axis. Consider the outcome of Kentucky’s 2014 Congressional elections, shown in Table 4.1. Republicans won 64% of the vote and  $5/6 = 83.3\%$  of the seats. This election would get plotted as the point (64, 83.3). The difficulty with this approach is the small number of data points we would be able to plot. With new Congressional maps being drawn every 10 years at most and elections being held every two years, a single Congressional map will yield at most 5 points on the plot. Having so few data points makes it difficult to extract any meaningful information regarding the votes-to-seats relationship. One way of compensating is to fill in the gaps using simulations.

Table 4.1: Result of Kentucky’s 2014 Congressional Elections. Source: Federal Election Commission. (n.d.). Election Results. Retrieved from <https://transition.fec.gov/pubrec/electionresults.shtml>

	Republican Vote	Democratic Vote	Winning Party
1st District	73.1%	26.9%	Republican
2nd District	69.2%	30.8%	Republican
3rd District	35.6%	63.5%	Democratic
4th District	67.7%	32.3%	Republican
5th District	78.3%	21.7%	Republican
6th District	60.0%	40.0%	Republican
Total	64.0%	35.9%	

There are many ways one could simulate the data. One simple way, introduced by Mira Bernstein (2017), would be to uniformly shift the individual district totals from a given year, which we will refer to as our “seed election,” incrementally by a fixed value and observe how this impacts the number of seats



won. By connecting these simulated points, we have a step function modeling the votes-to-seats relationship for that map. Figure 4.1 shows the results of this method using Kentucky’s 2014 Congressional election results.

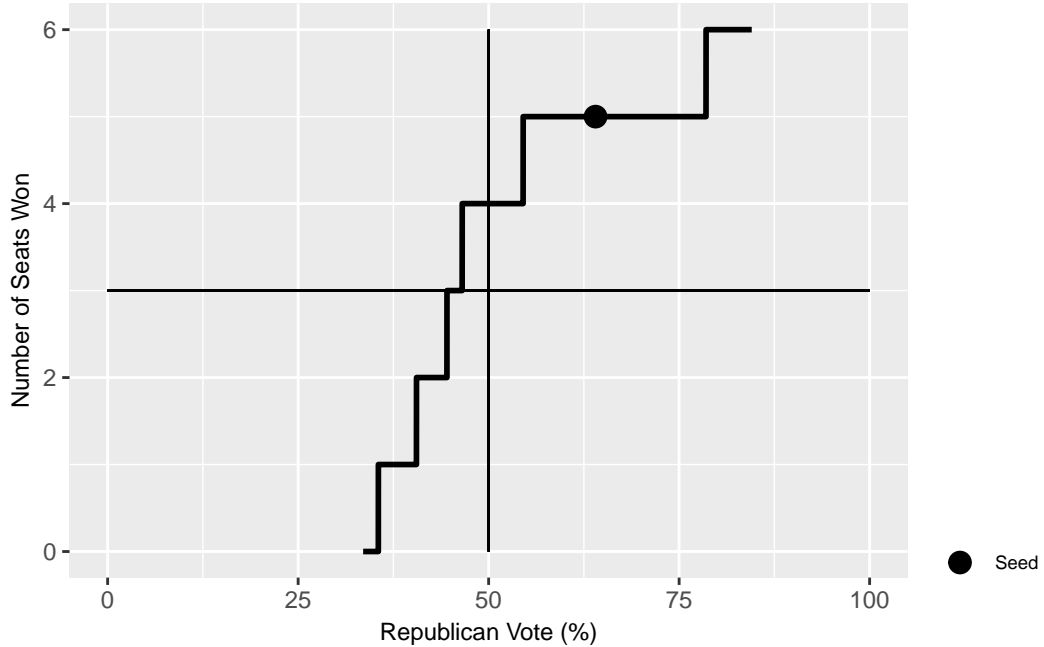


Figure 4.1: Bernstein Shift of Congressional Elections in Kentucky. Constructed using 2014 as a seed value with an increment of 1.

Any part of the line which crosses through the upper left or lower right quadrant of this plot represents an undesirable result since that would indicate that a party wins a majority of the vote without winning the majority of the seats. This plot, however, is generated from a single election and is therefore subject to variation. Our objective is to simulate the data points in a way which accounts for the random variation in elections.

## 4.2. Simulation Process

### 4.2.1. Shifts and Residuals

We are simulating shifts in statewide vote for a party from some seed value. In order to better account for the variation at play, we want to be able to effectively simulate the noise in how individual districts react about said shift. We can examine this noise by looking at the “residual,” or leftover shifts in district vote from previous elections in our data. In so doing, we standardize for the year-to-

year shifts in our data and isolate the noise. For a Congressional district  $X$  in state  $S$  which voted  $x_i$  percent for party A in the  $i^{\text{th}}$  election and whose state voted  $s_i$  percent for party A overall in the same election, we calculate the residual shift as

$$\text{Residual Shift} = \text{District Shift} - \text{State Shift} = (x_i - x_{i-1}) - (s_i - s_{i-1}). \quad (4.1)$$

Since we are looking at shifts between two consecutive elections, we only calculate the value for shifts where both years shared the same Congressional map. Additionally, both years must also not satisfy any of our exclusion criteria. Consider the results for Kentucky's 2012 and 2014 Congressional elections, shown in Table 4.2. Each of the residual shifts was calculated by taking the district shift and subtracting 3.1%, the statewide shift.

Table 4.2: Shifts Between Kentucky's 2012 and 2014 Congressional Elections. Source: Federal Election Commission. (n.d.). Election Results. Retrieved from <https://transition.fec.gov/pubrec/electionresults.shtml>

	Republican Vote		District Shift	Residual Shift
	2012	2014		
1st District	69.6%	73.1%	3.5%	0.4 %
2nd District	67.0%	69.2%	2.2%	-0.9%
3rd District	35.1%	35.9%	0.8%	-2.3%
4th District	64.0%	67.7%	3.7%	0.6 %
5th District	77.9%	78.3%	0.4%	-2.7%
6th District	52.0%	60.0%	8.0%	4.9 %
Total	60.9	64.0%	Statewide Shift: 3.1%	

We can now begin to examine the distribution of our residuals. The histogram and normal probability plot, shown in Figure 4.2, seem to present a decent case for normality of our nationwide residuals, but we will still avoid making any assumptions regarding the underlying distribution. Instead, we can simulate district-level shifts by sampling with replacement from the distribution of residuals. Our simulation process will rely on the assumption that the residual variables are homoskedastic across election years. The boxplots of our residual values by

year (Figure 4.3) suggest this assumption is reasonable. Unfortunately, there does not appear to be homoskedacity across the states, though, as evident in Figure 4.4.

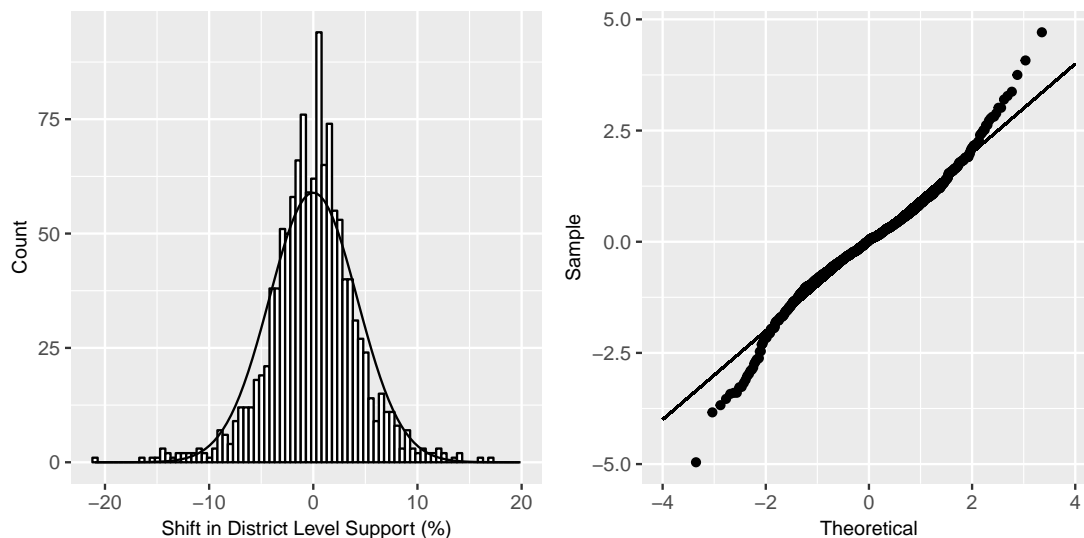


Figure 4.2: Histogram and Normal Probability Plot of Residual Shifts. Normal probability plot compares the theoretical percentiles of the normal distribution with the percentiles of our standardized residuals, providing another means of assessing normality.

Our assumption allows us to sample from residuals across all election years. Due to the lack of homoskedacity across states, however, we will restrict our residual distribution to only our state of interest. Unless otherwise noted, residuals will be sampled only from that state.

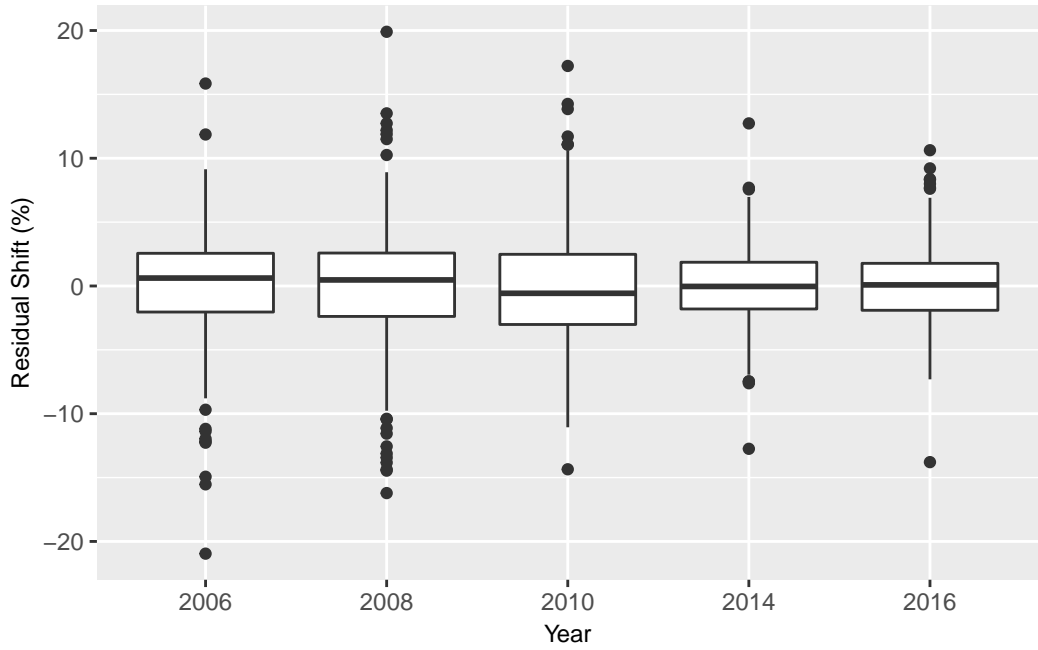


Figure 4.3: Boxplots of Residual Republican Shift by Year.

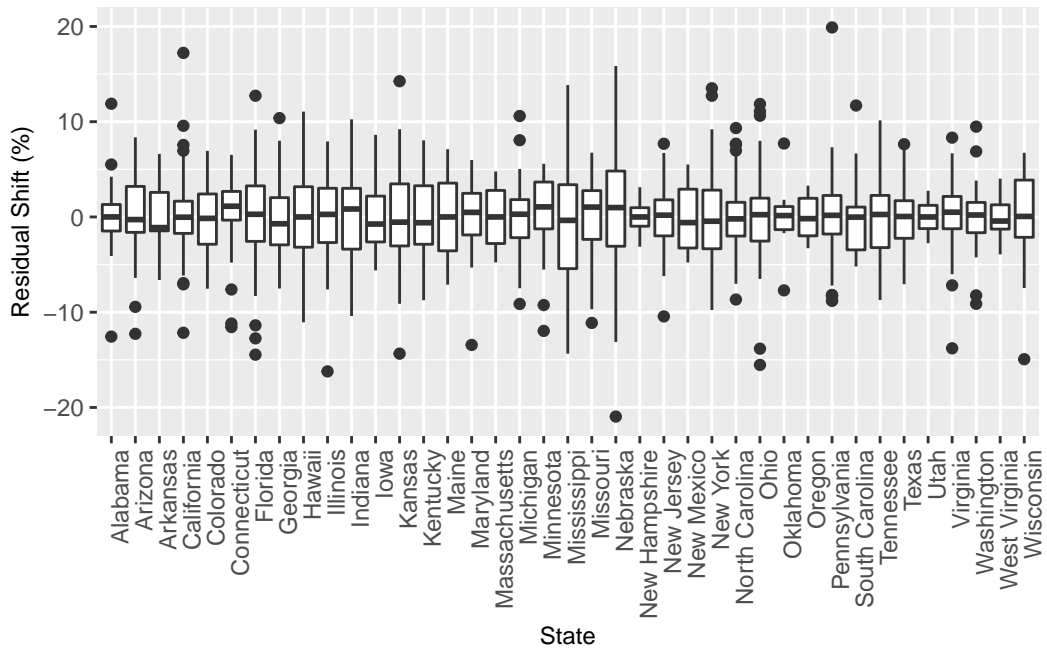


Figure 4.4: Boxplots of Residual Republican Shift by State.

### 4.2.2. Performing and Visualizing the Simulation

Let  $s_1, s_2, \dots, s_k$  be the district-level vote percentages for party A in a state with  $k$  districts for some election year; these will be our seed values. Suppose we want to simulate future district level elections,  $t_1, t_2, \dots, t_k$ , where a statewide shift of  $p\%$  in statewide party A vote has taken place. Let  $r_1, r_2, \dots, r_k$  be a random sample, with replacement, from the residual distribution. Let  $\bar{r}$  be the mean of the sampled residuals. Then we simulate our districts as

$$\begin{aligned}t_1 &= s_1 + p + r_1 - \bar{r} \\t_2 &= s_2 + p + r_2 - \bar{r} \\&\vdots \\t_k &= s_k + p + r_k - \bar{r}.\end{aligned}\tag{4.2}$$

Each simulated district result is calculated by taking the seed result for the district, adding the arbitrarily selected statewide shift, adding the randomly selected residual for the district, and subtracting the mean of the sampled residuals to ensure the simulated statewide shift is equal to our selected statewide shift.

Example 4.1. Suppose we want to simulate a future election in Kentucky using the 2014 Congressional elections as seed values. If we want to simulate a statewide shift of  $-10\%$  in statewide Republican vote, we begin by taking a random sample with replacement of 6 residuals, one for each of Kentucky's 6 Congressional districts, from the state's distribution of residuals. Using R, we get the sample 5.9, 0.3, 2.2,  $-9.8$ , 4.0,  $-7.7$ . The sampled residuals have a mean of  $-0.85$ . Table 4.3 provides the results of this simulation.

Note that the simulated result column is computed by taking the sum of the 2nd, 3rd, and 4th columns and subtracting the 5th column. Under this simulation, Republicans won  $54\%$  of the vote and  $4/6 = 66.7\%$  of the seats. We would repeat this process programatically many times and observe the frequencies of the number of seats won. We can do this for any statewide shift within reason to produce a

Table 4.3: Results of Simulation. Source: Federal Election Commission. (n.d.). Election Results. Retrieved from <https://transition.fec.gov/pubrec/electionresults.shtml>

	Republican Vote	Shift	Residual	Mean Residual	Simulated Result
1st District	67.7%	-10%	5.9%	-0.85%	64.5%
2nd District	78.3%	-10%	0.3%	-0.85%	69.5%
3rd District	60%	-10%	2.2%	-0.85%	53.1%
4th District	69.2%	-10%	-9.8%	-0.85%	50.3%
5th District	35.6%	-10%	4%	-0.85%	30.5%
6th District	73.1%	-10%	-7.7%	-0.85%	56.3%
Statewide	64.0%				54%

large data set of simulated frequencies. Using R, we simulated 2000 such elections where a -10% shift has taken place. Of those 2000 simulations, Republicans won 3 seats 20 times for 1%, 4 seats 1,155 times for 57.75%, and 5 seats 825 times for 41.25%. We can plot these results on a votes-seats scatter plot with the relative frequency being represented by the transparency of the point as it is shown in Figure 4.5.

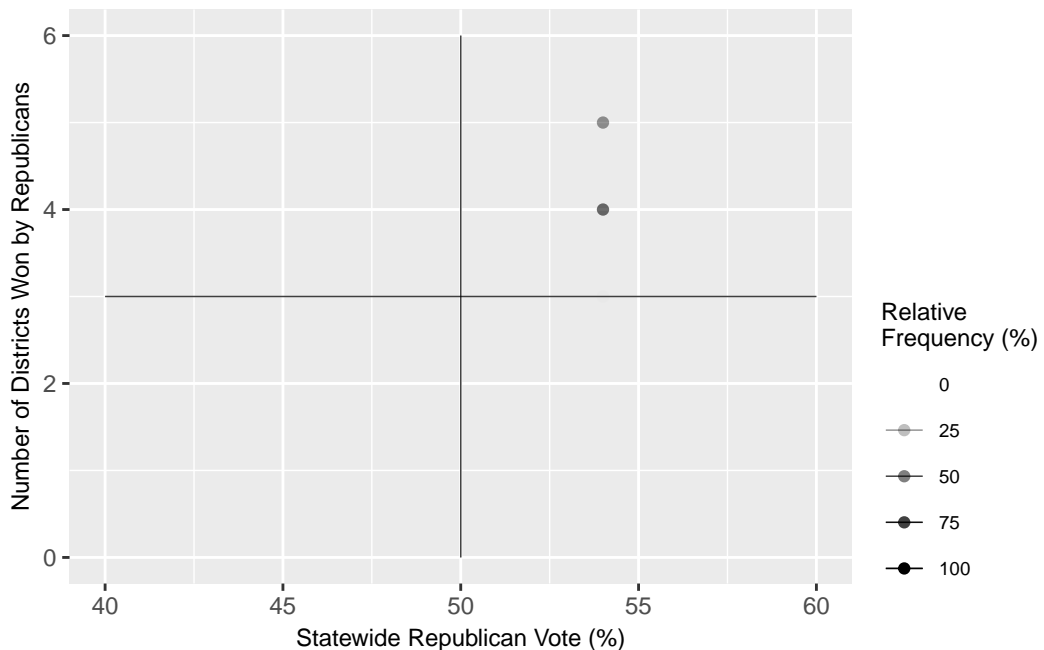


Figure 4.5: Partial Scatter Plot of Statewide Vote and Seats Won in Kentucky. Simulation of elections in Kentucky with a statewide Republican vote of 54 percent using 2014 as a seed.

We can perform the same simulations for a range of statewide shifts, each a certain distance apart, to help assess the overall relationship between votes and seats in a state of interest. Still using Kentucky's 2014 Congressional elections as our seed values, we performed 2000 simulations for every statewide mean between 35 and 78, incrementing by 1. These means were chosen by taking the means which would produce impossible results (district level percentages less than 0% or greater than 100%) fewer than 5% of the time. When sampling the residuals for a certain district, the distribution was truncated to avoid impossible results. We then plotted all the simulation results similarly to how we did so above to produce Figure 4.6. In this plot, the color of the point indicates which party received a higher percentage of seats than its statewide vote percentage, and the seed value is denoted by a larger point on the plot.

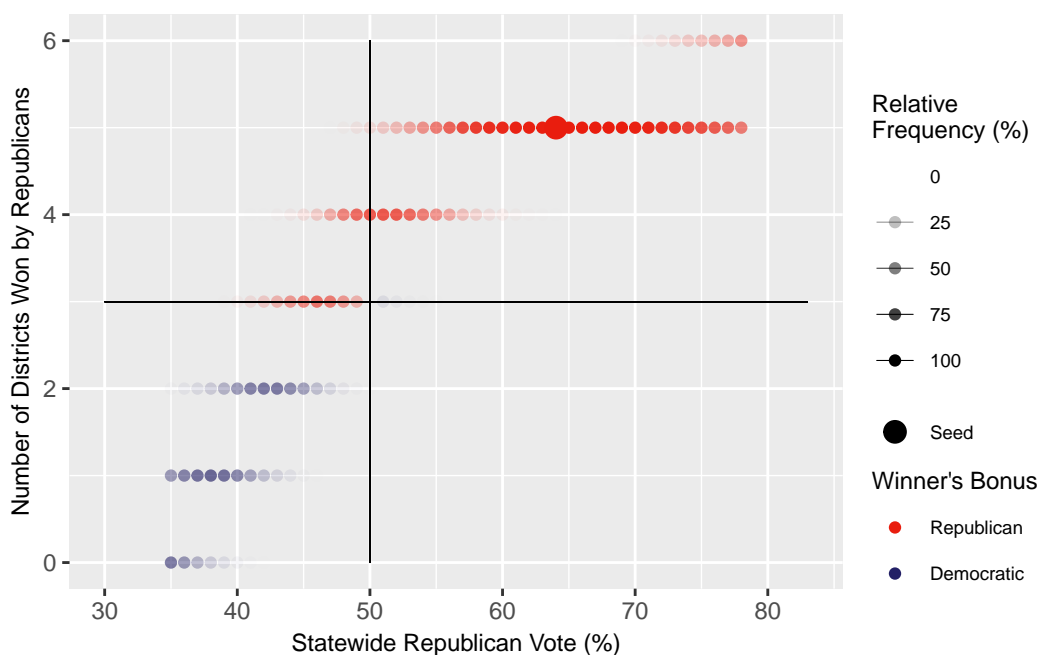


Figure 4.6: Scatter Plot of Statewide Vote and Seats Won in Kentucky. Simulation of elections in Kentucky using 2014 as a seed.

In addition to producing a scatter plot, there are a variety of other charts we can also use to visualize the results of a simulation. In Figure 4.7, we plot the 2.5th, 25th, 75th, and 97.5th percentiles as well as the mean number of districts won for the range of statewide means in our example. We do so by constructing

a step function for each based on the simulated frequencies. The solid black line represents the mean number of districts won, the dashed yellow lines represent the 2.5th and 97.5th percentiles, and the shaded yellow region represents the interquartile range (IQR).

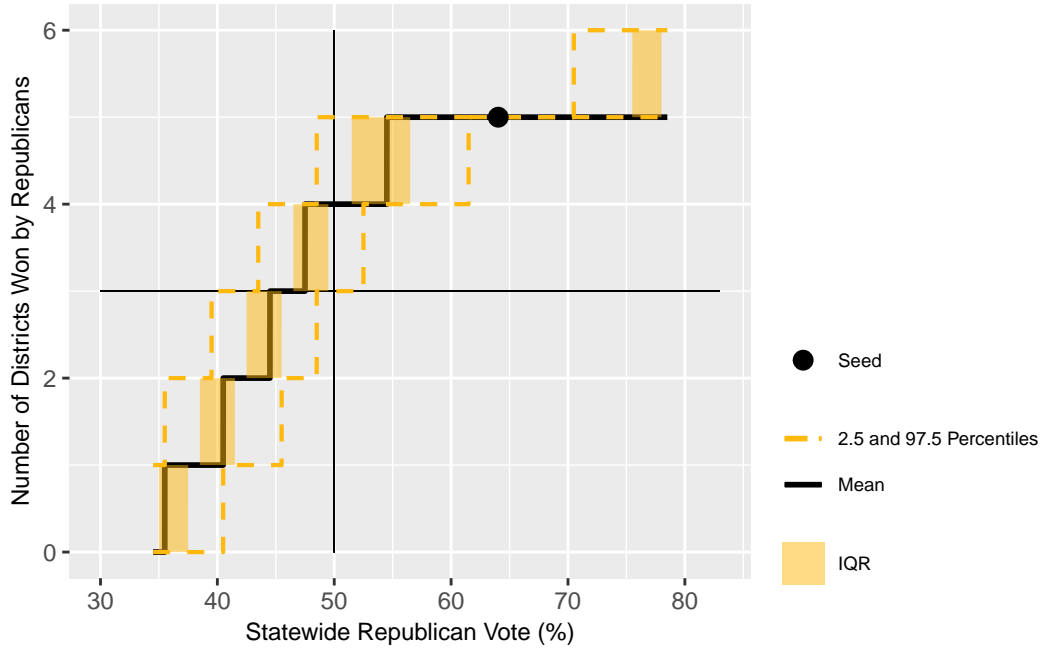


Figure 4.7: Step Chart of Statewide Vote and Seats Won in Kentucky. Simulation of elections in Kentucky using 2014 as a seed.

One of the main goals for our simulation is to be able to examine the level of partisan asymmetry present in a Congressional map. In any map which is perfectly symmetrical, we would expect both parties to convert votes-to-seats at the same rate. That is, if Republicans are expected to win 75% of seats with 70% of the vote, Democrats should also be expected to win 75% of the seats if they win 70% of the vote. We can visually compare this votes-to-seats relationship for each party using variations of the two previous plots. Figure 4.8 is constructed similarly to Figure 4.6 with the main difference being that the color now indicates which party the statewide vote and districts won is calculated with respect to. For example, if Republicans won 40% of the vote and 1 seat in Kentucky, this would be plotted as Democrats winning 60% of the vote and 5 seats.



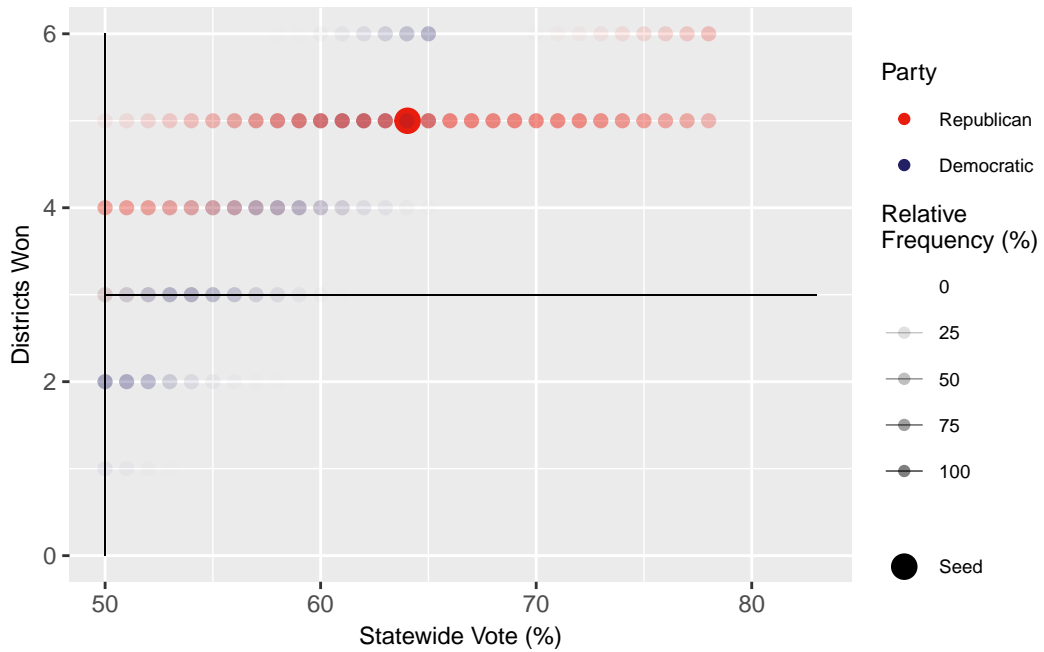


Figure 4.8: Scatter Plot of Statewide Vote and Seats Won by Party in Kentucky. Simulation of elections in Kentucky using 2014 as a seed.

Figure 4.9 also provides a means of comparing the votes-to-seats relationships of the two parties. In this variation of Figure 4.7, we look only at the 2.5th and 97.5th percentiles in addition to the mean. The color indicates which party the statewide vote and districts won is calculated with respect to. The shading indicates the area between the 2.5th and 97.5th percentiles.

A simple way of using our simulation process to examine the fairness of a Congressional map is to look at the behavior for a statewide vote of 50%. If no asymmetry were present, then each party would be expected to win half of the state's seats when they received 50% of the vote. Moreover, we would also expect the mean efficiency gap to be zero percent in the same situation. Our last plot, Figure 4.10, visualizes this information, tabulating the relative frequencies of districts won, the respective efficiency gaps, and displaying the mean efficiency gap in the corner. The values and mean efficiency gap are calculated with respect to the Republican party. The color indicates which party, if any, has an electoral advantage for each value.

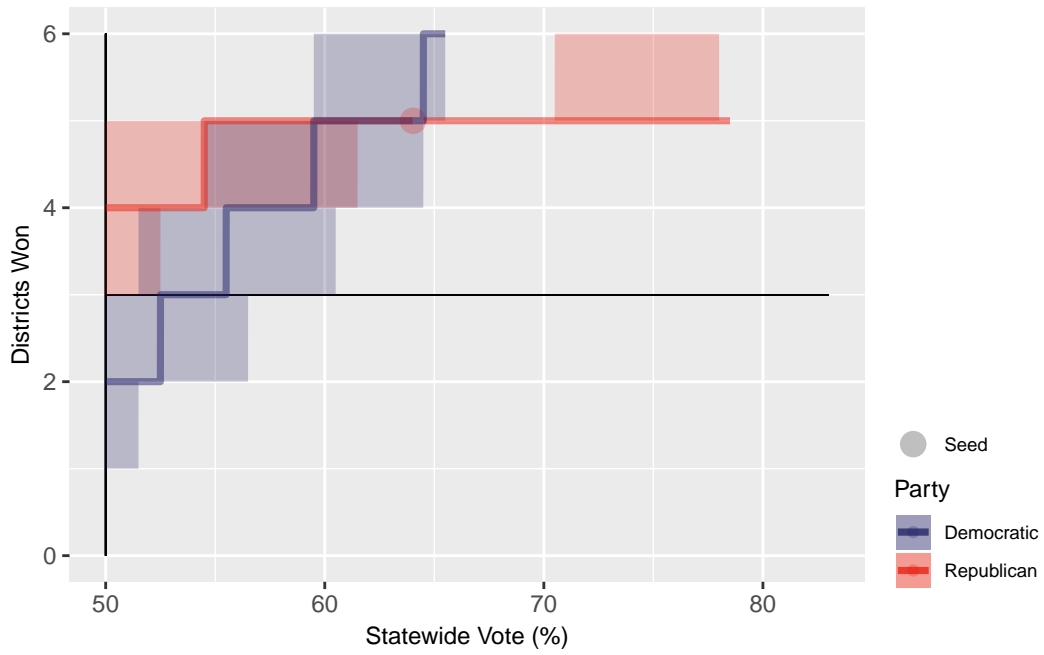


Figure 4.9: Step Chart of Statewide Vote and Seats Won by Party in Kentucky. Simulation of elections in Kentucky using 2014 as a seed.

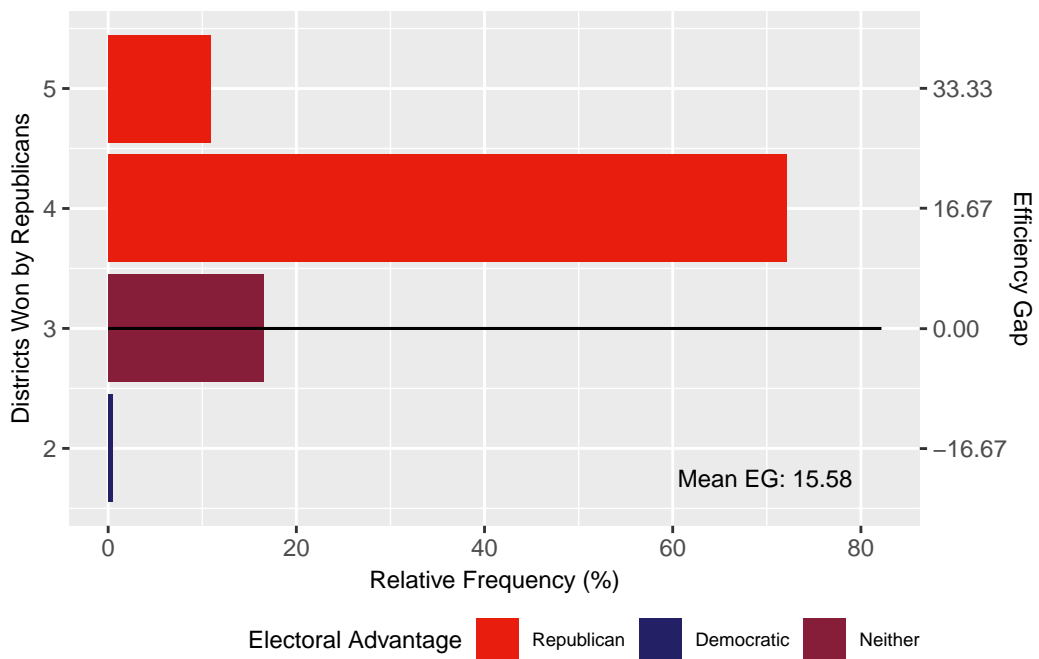


Figure 4.10: Bar Chart of Districts Won and Efficiency Gap in Kentucky. Simulation of elections with 50 percent vote in Kentucky using 2014 as a seed.

### 4.2.3. Seed Values

This process relies on having a complete set of seed values. If any district satisfies the exclusion criteria for a given year, that year cannot be used as a seed value. One option could be to select a different year for the Congressional map of interest to use for a seed value but it could be that there is not a complete year, especially for larger states. We can instead use the data across several years to help us complete our seeds. Suppose we are interested in simulating a state's elections for a Congressional map which was active for the years  $y_1, y_2, \dots, y_n$ . Let  $s_{ij}$  be the  $i^{\text{th}}$  district vote percentage for party A in the  $y_j^{\text{th}}$  election year. It would not suffice to simply take the means,  $\bar{s}_{i\bullet}$ , as our seed values since that would introduce confounding from the year-to-year shifts we are trying to control for. Instead let

$$p_j = \frac{1}{k} \sum_{i=1}^k (s_{ij} - s_{i1}) \quad (4.3)$$

and form  $s'_{ij} = s_{ij} - p_j$ . The  $p_j$  values are the mean district shifts from the first year of interest, and we subtract these from the  $s_{ij}$  values. We can then calculate our seed values as  $s_i = \bar{s}'_{i\bullet}$ . This does not do a perfect job of standardizing for the year-to-year shifts, but it does provide us with the means to compute a seed value for which we otherwise would not have had one. Unfortunately, there are still some states for which a seed value cannot be completed. Due to California's top-two primary format, there has not been a Republican running in the 40th or 44th district general election for its current Congressional map.

### 4.2.4. Effect of Using Nationwide Residuals

As mentioned earlier, we could not justify an assumption of homoskedacity across states for our residual distribution. We more closely examined the effect of using statewide versus nationwide residuals by running simulations for every possible statewide election and observing the percentage of steps where the means were not equal. We will refer to this value as the "percent mean disagreement." If we look at the relationship between variance of the statewide residuals and



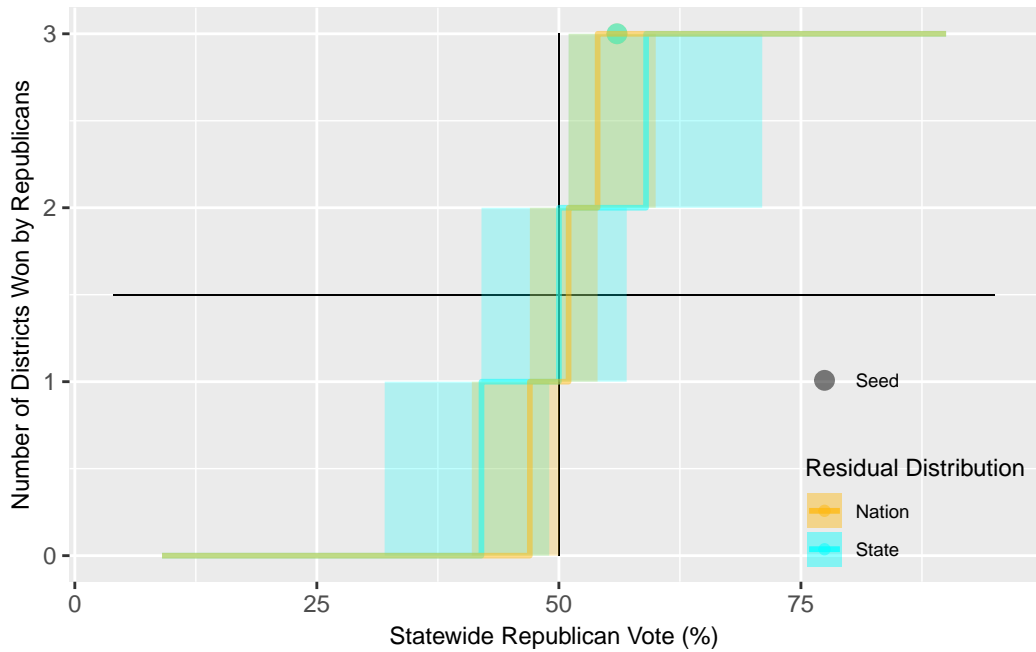


Figure 4.12: Step Chart of Statewide Vote and Seats Won in Nebraska. Simulations of Elections in Nebraska Using 2006 as the Seed. Shading represents values between 2.5th and 97.5th percentiles.

#### 4.2.5. Accuracy of Simulation Process

In order to assess how accurate our simulation process is, we employed a type of cross-validation to test our simulation process against the actual statewide results in our data. We did so by taking each complete statewide election and simulating an election with the same statewide vote total using the other elections from the same Congressional map to derive our seed values and sampling from residuals which are not affected by the test election. For each of these test elections, we record whether or not the actual result is between the 25th and 75th percentiles of the simulations and whether it is between the 2.5th and 97.5th percentiles. In a perfectly accurate simulation, we would expect the actual results to be in these ranges at least 50% and 95% of the time, respectively. Using R, we were able to calculate these values for 73 statewide elections. In the other cases, either the election satisfied the exclusion criteria, a completed seed could not be formed, or there were no residuals available from which to sample. The values, 72.6% and 93.2% respectively, suggest that our simulation may simulate slightly fewer

extreme outcomes than what is realistic but beyond that, the numbers are not especially problematic.

Table 4.4 shows the cross-validation results for all state Congressional elections whose outcome occurred fewer than 40% of the time in the respective simulations. The Districts Won column lists the actual result for each election and the Percentage column indicates how often the outcome occurred in the simulations. The Inner 50% column indicates whether or not the outcome was between the 25th and 75th percentiles in the simulation, and the Inner 95% column similarly notes whether it was between the 2.5th and 97.5th percentiles. Of the five elections outside the inner 95% of simulated outcomes, four were in states with fewer than five Congressional seats. This suggests that the simulation process is possibly less accurate for smaller states, most likely due to the small number of residuals which are available for those states. The entire table of cross-validation outcomes is included in the Appendix A.

Table 4.4: Cross-Validation Results for Least Likely Outcomes. Districts won column contains actual results for each election. Inner 50% and Inner 95% column denote whether or not the actual result was between the 25th/75th and 2.5th/97.5th percentiles, respectively.

State	Year	Districts Won	Percentage	Inner 50%	Inner 95%
Maryland	2008	1 (12.5%)	0%	No	No
Nebraska	2006	3 (100%)	0%	No	No
New Hampshire	2014	1 (50%)	0%	No	No
Kansas	2010	4 (100%)	1.05%	No	No
New Mexico	2010	1 (33.3%)	1.6%	No	No
North Carolina	2004	7 (53.8%)	2.95%	No	Yes
Kentucky	2010	4 (66.7%)	3.6%	No	Yes
New Hampshire	2012	0 (0%)	5.85%	No	Yes
Iowa	2010	2 (40%)	7%	No	Yes
Mississippi	2010	3 (75%)	10.4%	No	Yes
Colorado	2004	4 (57.1%)	10.5%	No	Yes
Indiana	2004	7 (77.8%)	12.3%	No	Yes
Virginia	2012	8 (72.7%)	13.4%	No	Yes
Ohio	2006	11 (61.1%)	15.95%	No	Yes
Connecticut	2004	3 (60%)	18.7%	No	Yes
Wisconsin	2010	5 (62.5%)	18.9%	No	Yes
North Carolina	2010	6 (46.2%)	19%	No	Yes
Mississippi	2008	1 (25%)	20.7%	No	Yes
New Mexico	2006	2 (66.7%)	21%	No	Yes
West Virginia	2010	2 (66.7%)	22.95%	No	Yes
Washington	2006	3 (33.3%)	26.25%	Yes	Yes
Iowa	2004	4 (80%)	30.25%	Yes	Yes
Kansas	2006	2 (50%)	38.65%	Yes	Yes

### 4.3. Simulation of Certain States

In this section, we use our simulation process to examine six state Congressional maps of interest. For each state, we provide some brief context behind each and discuss the results of the simulation. There are four plots included with each, one being a plot of a Bernstein-style shift using the same seed as our simulation for comparison.

#### 4.3.1. Pennsylvania

In *Vieth v. Jubelirer* (2004), three Pennsylvania citizens argued that the state's Congressional map, drawn after the 2000 census, was gerrymandered to favor Republicans. This, they argued, violated the one-person one-vote requirement

in the Constitution. A plurality of the court upheld the lower court ruling that partisan gerrymandering claims were unconstitutional. Justice Kennedy, while agreeing with the judgment that claims of partisan gerrymandering were not justiciable, stated that judicially manageable standards could be developed and used in future cases brought before the court.

In the 2002 Congressional elections, Republicans won 58.4% of the statewide vote and 12 out of 19, or 63.2% of the seats. This does not appear to be an overly disproportionate result. In 2004, however, Republicans won 49.8% of the vote and the same number of seats as they had won in the previous election. This represents a far more asymmetrical result.

Figures 4.13 through 4.16 show the simulation results for Pennsylvania's 2002-2010 Congressional map. The map appears to give an advantage to Republicans who, as figure 4.15 suggests, are able to more effectively convert votes-to-seats. Of the simulated elections with a statewide total of 50% (Figure 4.16), Republicans won 10 or more of the 19 seats over 95% of the time. This results in a mean efficiency gap of 9.27%.

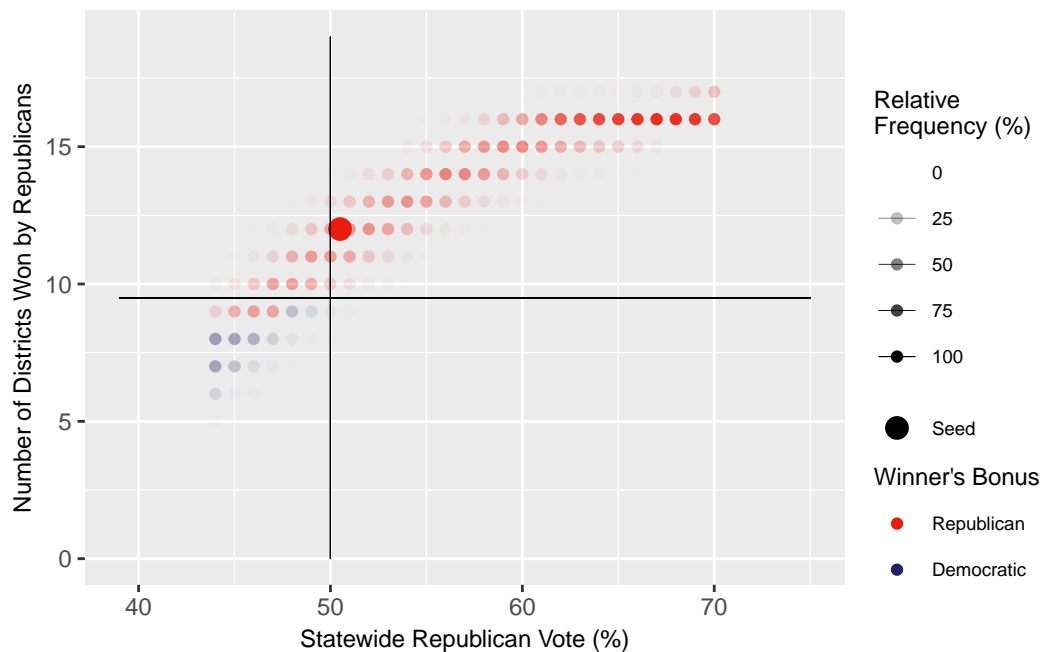


Figure 4.13: Scatter Plot of Statewide Vote and Seats Won in Pennsylvania. Simulations of elections in Pennsylvania using years 2004-2010 as the seed.  $n = 2000$  per step.



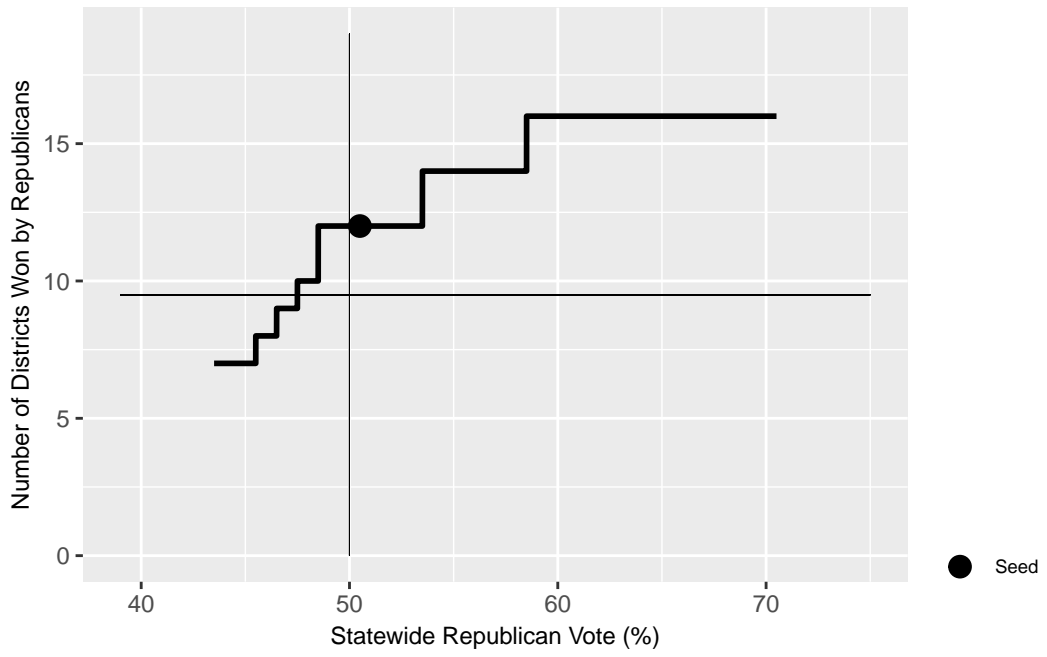


Figure 4.14: Bernstein-Style Simulations of Elections in Pennsylvania. Using years 2004-2010 as the seed and an increment of 1.

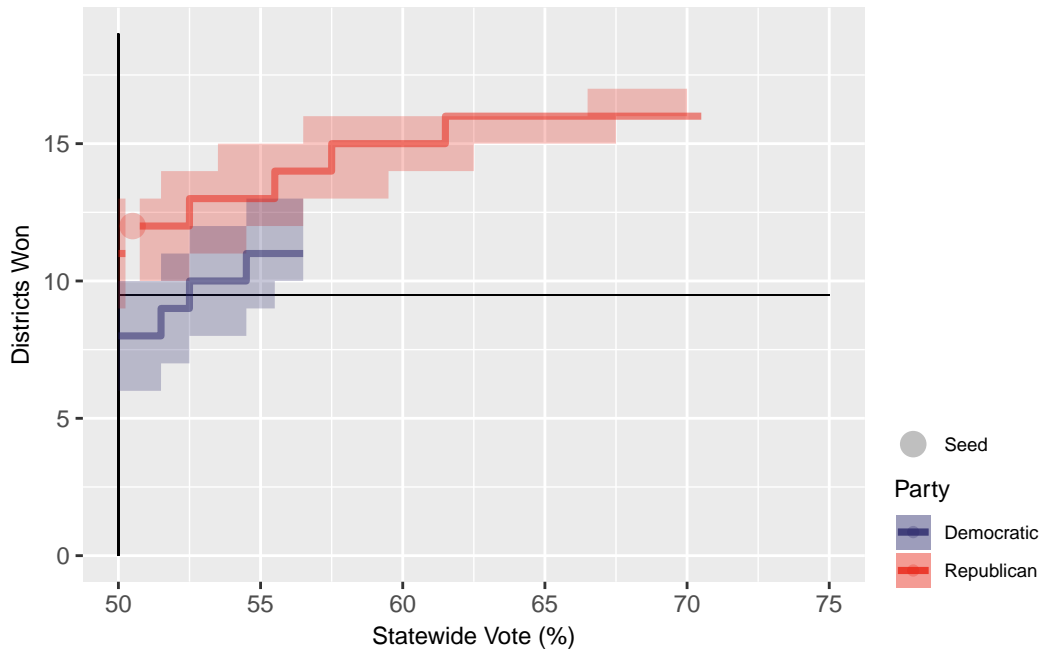


Figure 4.15: Step Chart of Statewide Vote and Seats Won by Party in Pennsylvania. Simulations of elections in Pennsylvania using years 2004-2010 as the seed. Shading represents values between 2.5th and 97.5th percentiles.  $n = 2000$  per step.

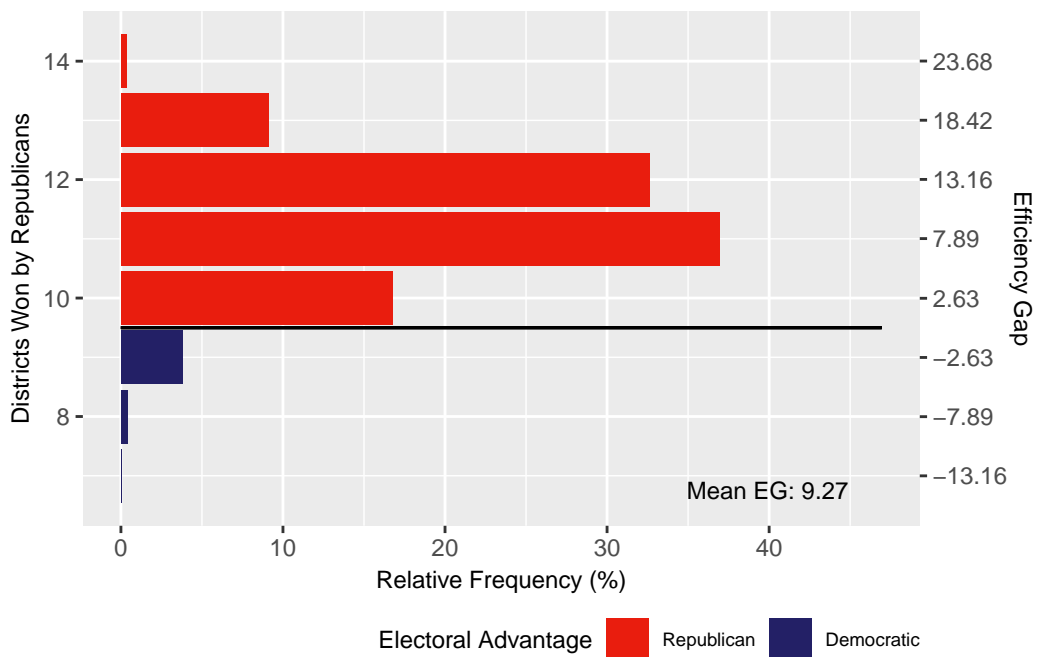


Figure 4.16: Bar Chart of Districts Won and Efficiency Gap in Pennsylvania. Simulations of elections with 50 percent vote in Pennsylvania using years 2004-2010 as the seed. Line segment indicates half of total districts.  $n = 2000$ .

### 4.3.2. Wisconsin

Wisconsin's State legislature map was the subject of *Gill v. Whitford* (2018), a Supreme Court case in which the plaintiffs argued their votes were wasted because of the map. It was the first case brought before the court which suggested use of the efficiency gap would meet Justice Kennedy's criteria for a judicially manageable standard laid out in *Vieth v. Jubelirer*. The court ultimately remanded the case back to lower courts, stating that the plaintiffs had failed to demonstrate standing.

The Redistricting Majority Project (REDMAP) was involved in the process of drawing the map in 2011 with the goal of ensuring Republican majorities in the U.S. House and State legislature. In the 2012 elections, Republicans won 48.6% of the statewide vote but 60.6% of the seats in the State Assembly. A similar lack of asymmetry was observed in the U.S. Congressional elections where Republicans won 48.9% of the statewide vote and five out eight, or 62.5% of the seats.

Figures 4.17 through 4.20 seem to suggest a Republican advantage for Wisconsin's current Congressional map. In over 85% of the simulated elections for a statewide vote of 50% (Figure 4.20), Republicans won five seats, which is more than half of the state's eight seats. Conversely, Democrats won five seats less than one percent of the time. For the 2000 simulations with a statewide vote of 50%, the mean efficiency gap was 10.69% in favor of Republicans.

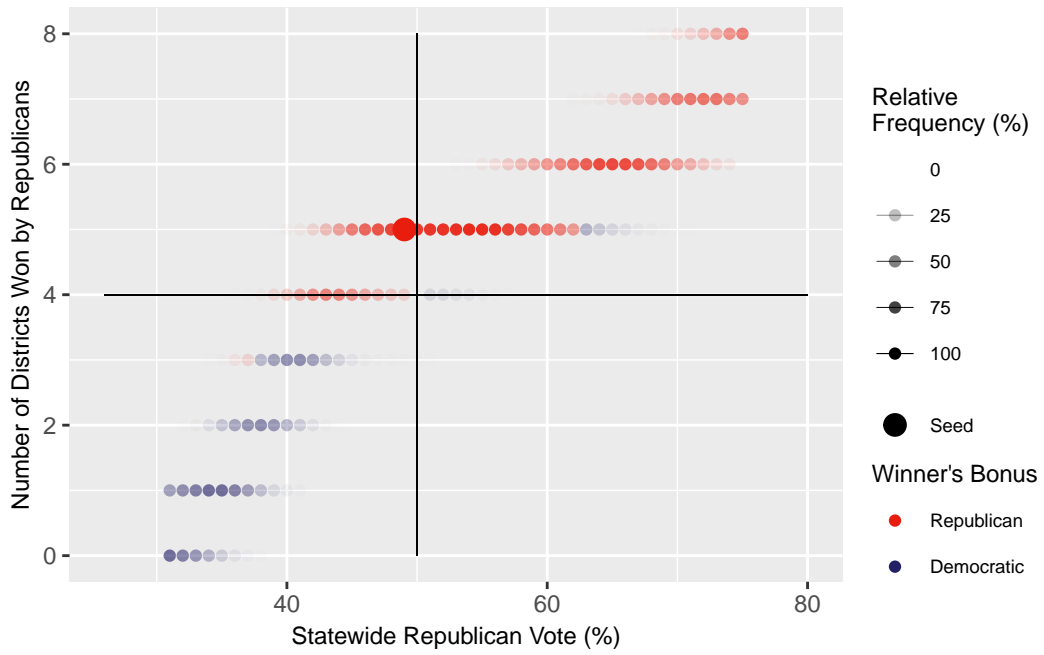


Figure 4.17: Scatter Plot of Statewide Vote and Seats Won in Wisconsin. Simulations of elections in Wisconsin using years 2012-2016 as the seed.  $n = 2000$  per step.

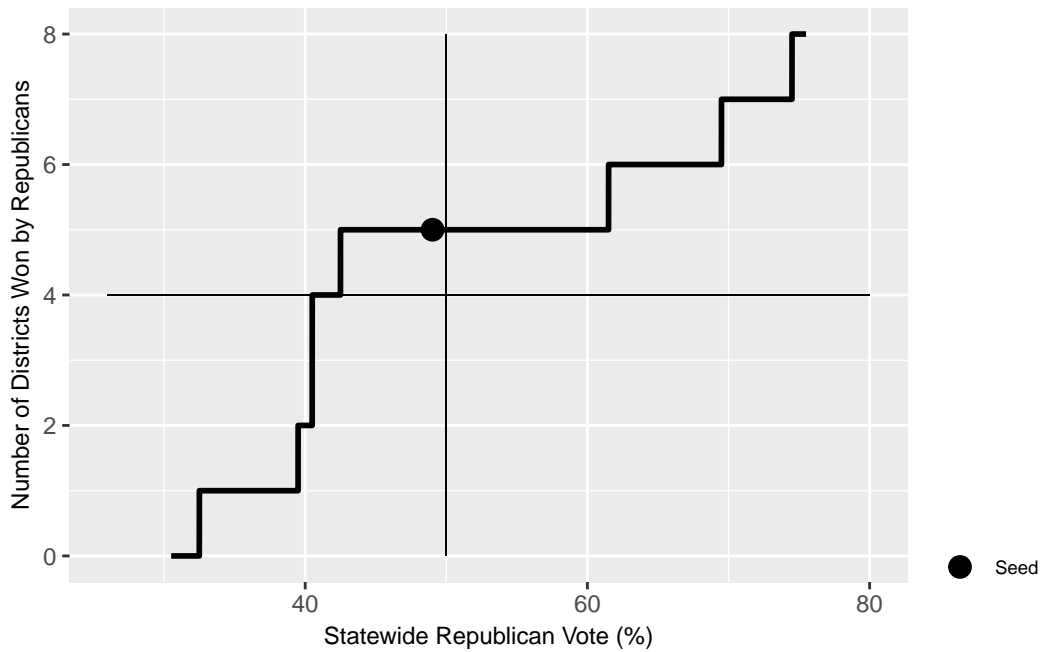


Figure 4.18: Bernstein-Style Simulations of Elections in Wisconsin. Using years 2012-2016 as the seed and an increment of 1.

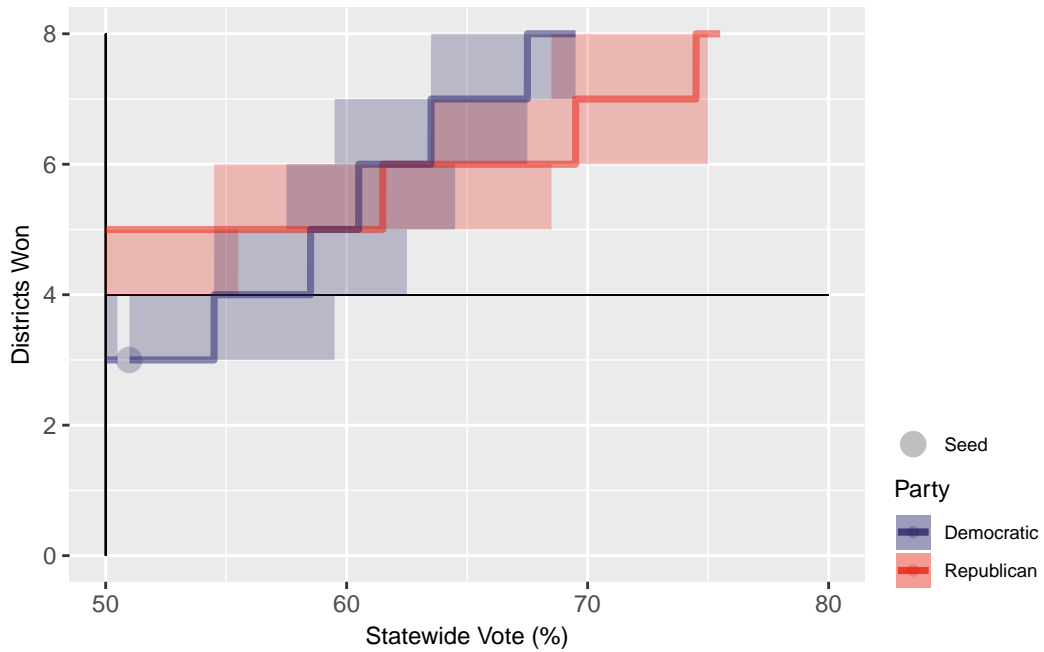


Figure 4.19: Step Chart of Statewide Vote and Seats Won by Party in Wisconsin. Simulations of elections in Wisconsin using years 2012-2016 as the seed. Shading represents values between 2.5th and 97.5th percentiles.  $n = 2000$  per step.

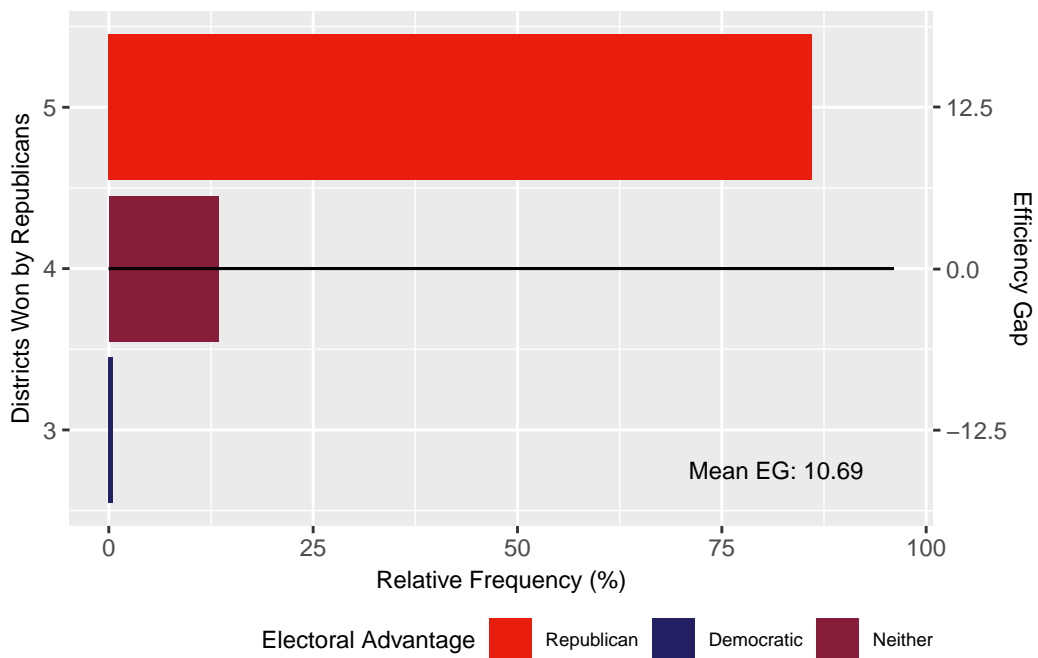


Figure 4.20: Bar Chart of Districts Won and Efficiency Gap in Wisconsin. Simulations of elections with 50 percent vote in Wisconsin using years 2012-2016 as the seed. Line segment indicates half of total districts.  $n = 2000$ .

### 4.3.3. Maryland

In November of 2018, a district court ruling in the case *Benisek v. Lamone* called for the mandatory redrawing of Maryland’s Congressional map prior to the 2020 elections. Unlike the previous states, Maryland’s Congressional map appears to favor Democrats rather than Republicans. In the 2016 Congressional elections, Republicans won 35.5% of the statewide vote but only one of the state’s eight districts for 12.5% of the seats.

It appears that Democrats have an advantage in Maryland’s current Congressional map. The simulation results are shown in figures 4.21 through 4.24. Of the simulated elections with a statewide vote of 50% (Figure 4.24), Democrats won more than half of the state’s eight districts over 80% of the time while Republicans won more than half less than 2% of the time. The resulting mean efficiency gap was  $-14.65\%$ , favoring Democrats.

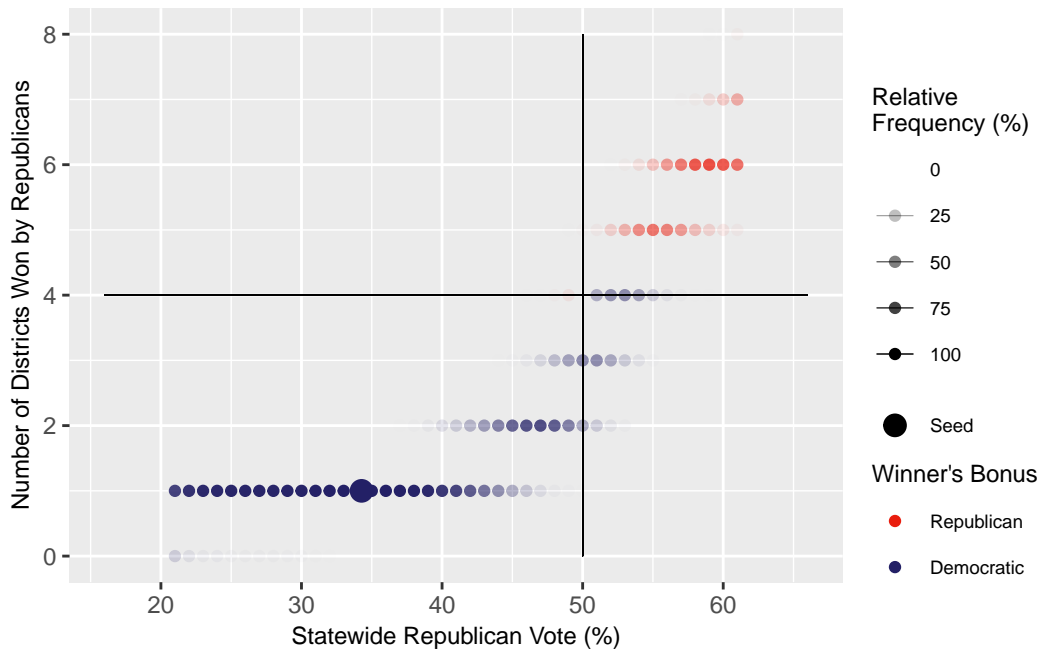


Figure 4.21: Scatter Plot of Statewide Vote and Seats Won in Maryland. Simulations of elections in Maryland using years 2012-2016 as the seed.  $n = 2000$  per step.

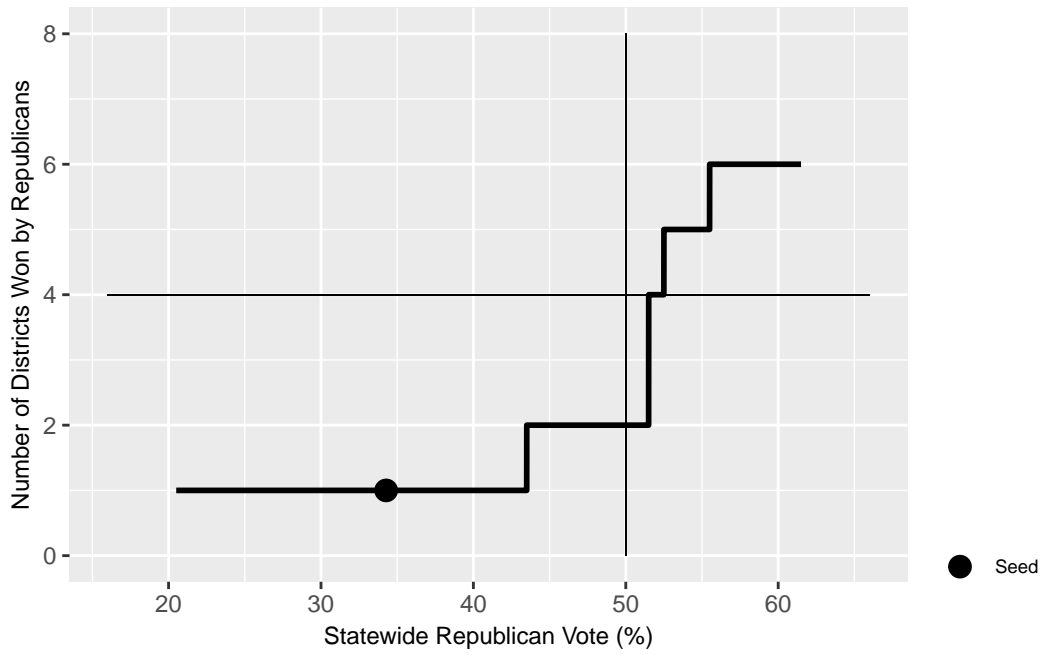


Figure 4.22: Bernstein-Style Simulations of Elections in Maryland. Using years 2012-2016 as the seed and an increment of 1.

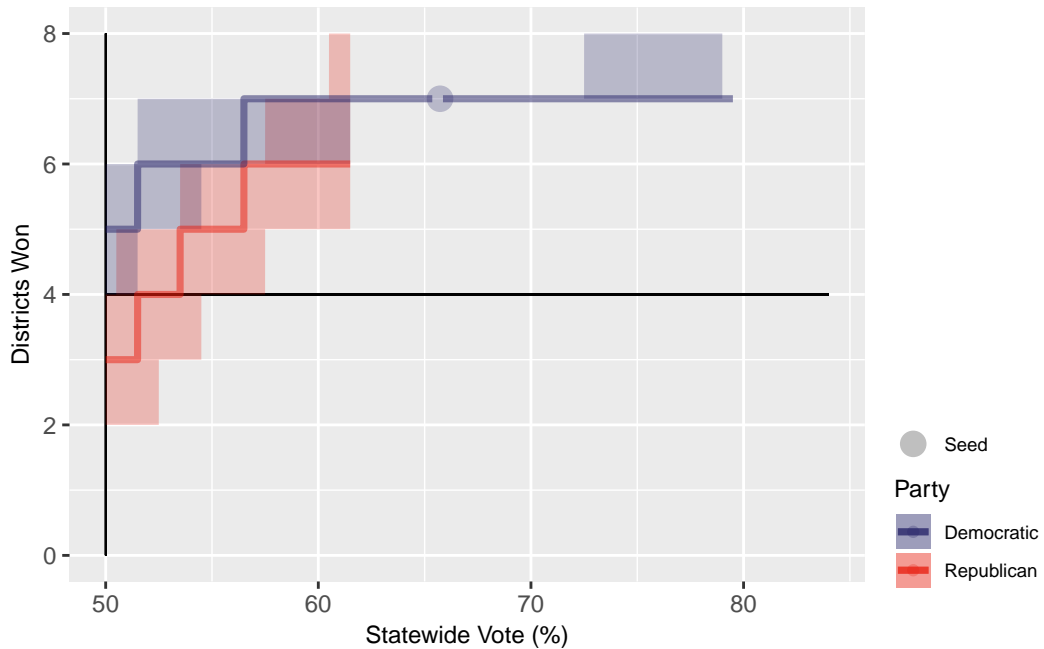


Figure 4.23: Step Chart of Statewide Vote and Seats Won by Party in Maryland. Simulations of elections in Maryland using years 2012-2016 as the seed. Shading represents values between 2.5th and 97.5th percentiles.  $n = 2000$  per step.

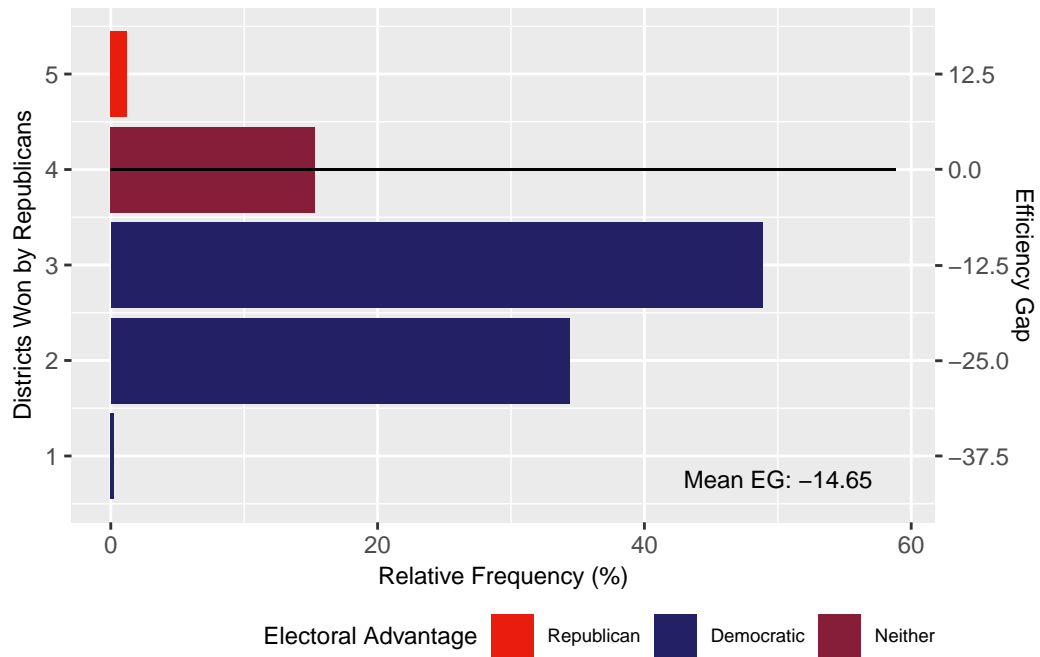


Figure 4.24: Bar Chart of Districts Won and Efficiency Gap in Maryland. Simulations of elections with 50 percent vote in Maryland using years 2012-2016 as the seed. Line segment indicates half of total districts.  $n = 2000$ .



#### 4.3.4. North Carolina

In 2017, the Supreme Court ruled that North Carolina's 1st and 12th districts were racially gerrymandered (*Cooper v. Harris*, 2017). Though this was not a partisan gerrymandering case, it still warrants examination due to the seemingly large electoral advantage Republicans have in the state. In the state's 2012 Congressional elections, Republicans won 9 of the state's 13 seats, or 69.2%, while winning only 48.7% of the statewide vote.

A simulation of North Carolina's Congressional map (Figures 4.25 through 4.28) seems to indicate a large advantage for Republicans. Democrats won less than half of the seats in every simulated election with a statewide vote of 50% (Figure 4.28) while Republicans won nine or more of the state's 13 seats over 95% of the time. North Carolina also has the highest 50% vote mean efficiency gap, 24.26%, of any of the states examined.

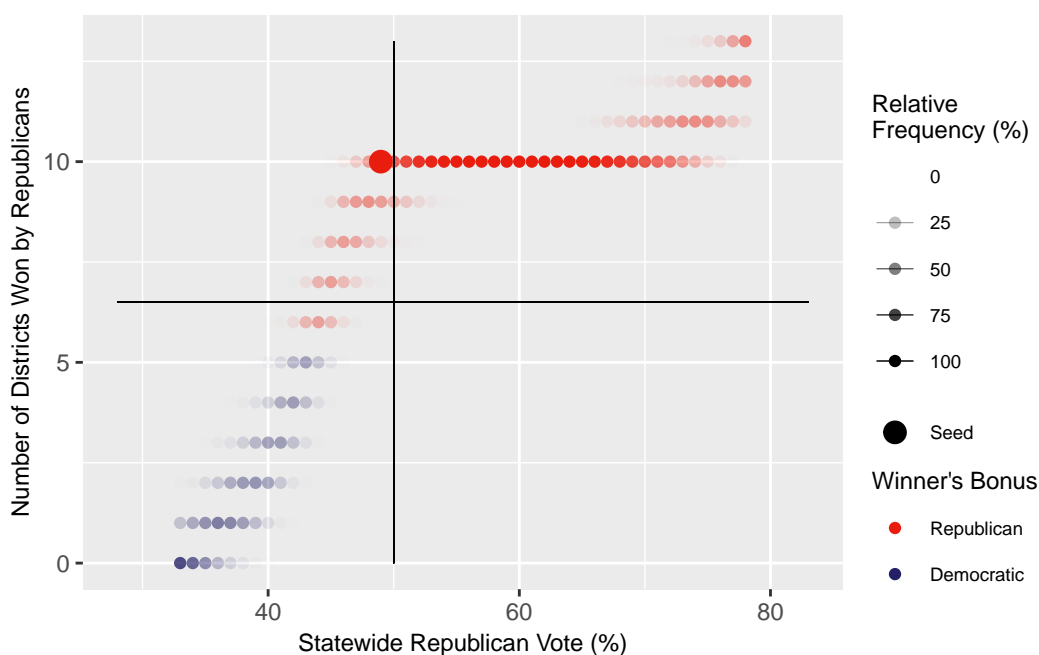


Figure 4.25: Scatter Plot of Statewide Vote and Seats Won in North Carolina. Simulations of elections in North Carolina using years 2012-2016 as the seed.  $n = 2000$  per step.

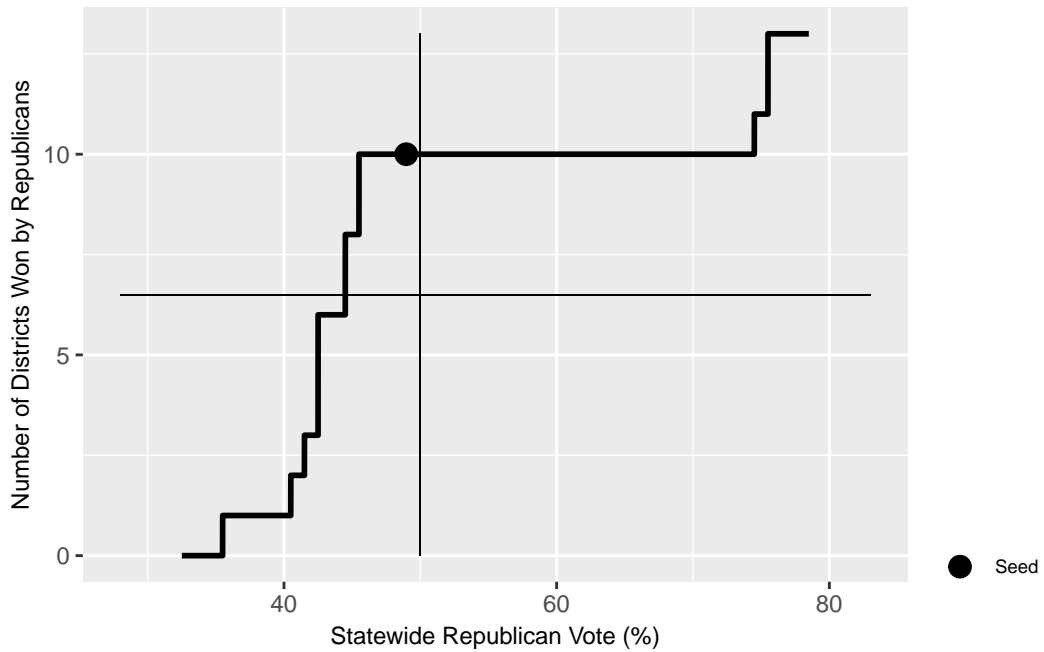


Figure 4.26: Bernstein-Style Simulations of Elections in North Carolina. Using years 2012-2016 as the seed and an increment of 1.

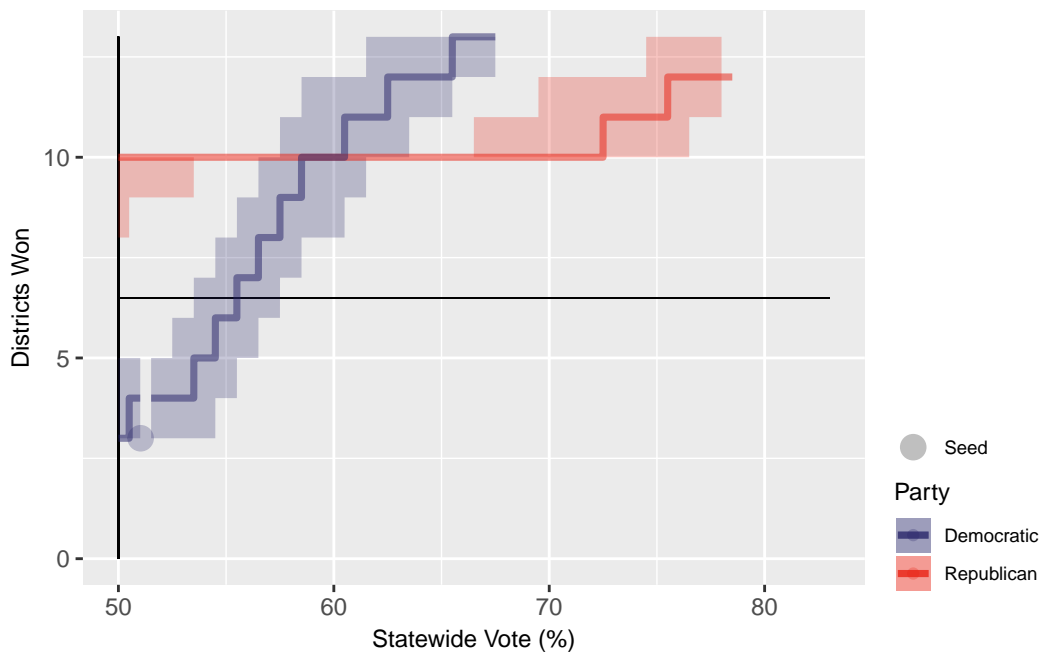


Figure 4.27: Step Chart of Statewide Vote and Seats Won by Party in North Carolina. Simulations of elections in North Carolina using years 2012-2016 as the seed. Shading represents values between 2.5th and 97.5th percentiles.  $n = 2000$  per step.

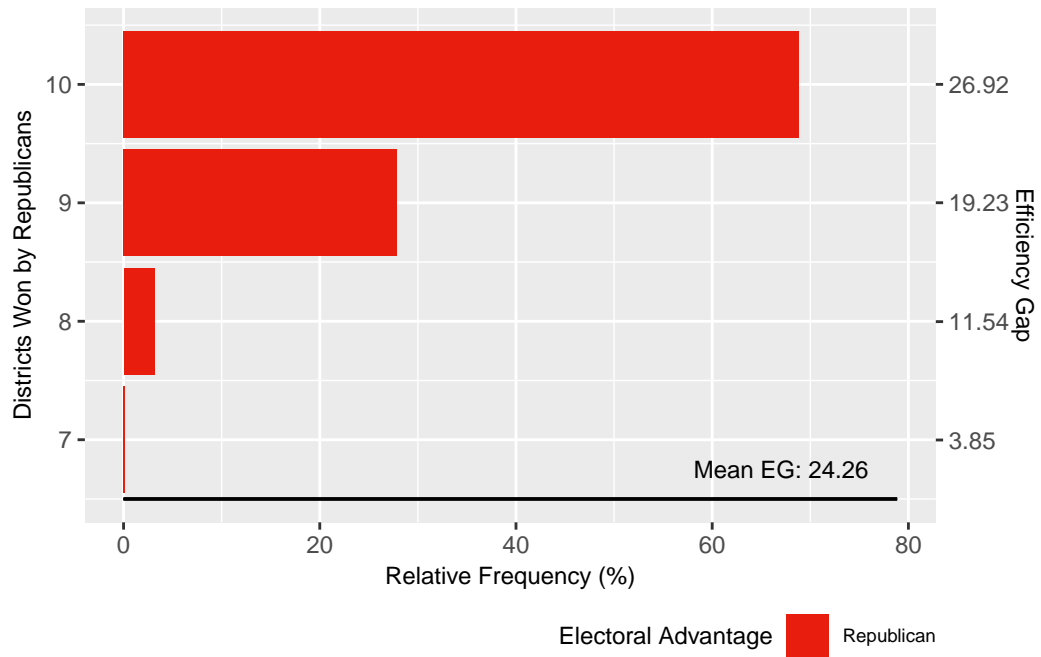


Figure 4.28: Bar Chart of Districts Won and Efficiency Gap in North Carolina. Simulations of elections with 50 percent vote in North Carolina using years 2012-2016 as the seed. Line segment indicates half of total districts.  $n = 2000$ .

#### 4.3.5. Arizona

In most states, Congressional maps are drawn by partisan legislative bodies. A few states use independent commissions instead. Arizona's Proposition 106 (2000) gave redistricting authority to a bipartisan independent commission, the Arizona Independent Redistricting Commission. The commission consists of two Republican members, two Democratic members, and one independent member. It is tasked with drawing new districts for both state legislative maps and Congressional maps. In 2016, Republicans won 52.4% of the statewide vote and five out of the state's nine Congressional seats, the most proportional result possible given the vote.

The results of simulating elections for Arizona's Congressional map are shown in figures 4.29 through 4.32. The map appears to be far more balanced than any of the other maps examined. Of the simulated elections with a statewide vote of 50% (Figure 4.32), Republicans won more than half of the nine districts 51.4% of the time while Democrats won more than half 48.6% of the time. This produced a mean efficiency gap of 0.33% in favor of Republicans, the lowest of all examined states. Additionally, as the share of the vote increases, it appears that each party converts the vote to seats at a similar rate (Figure 4.31).

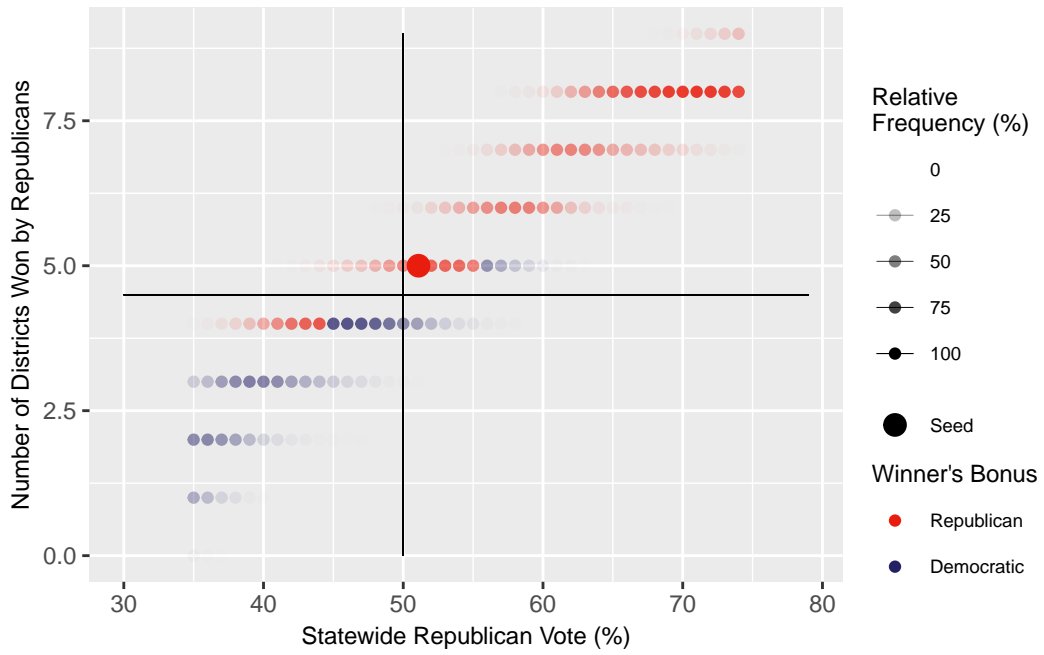


Figure 4.29: Scatter Plot of Statewide Vote and Seats Won in Arizona. Simulations of elections in Arizona using years 2012-2016 as the seed.  $n = 2000$  per step.

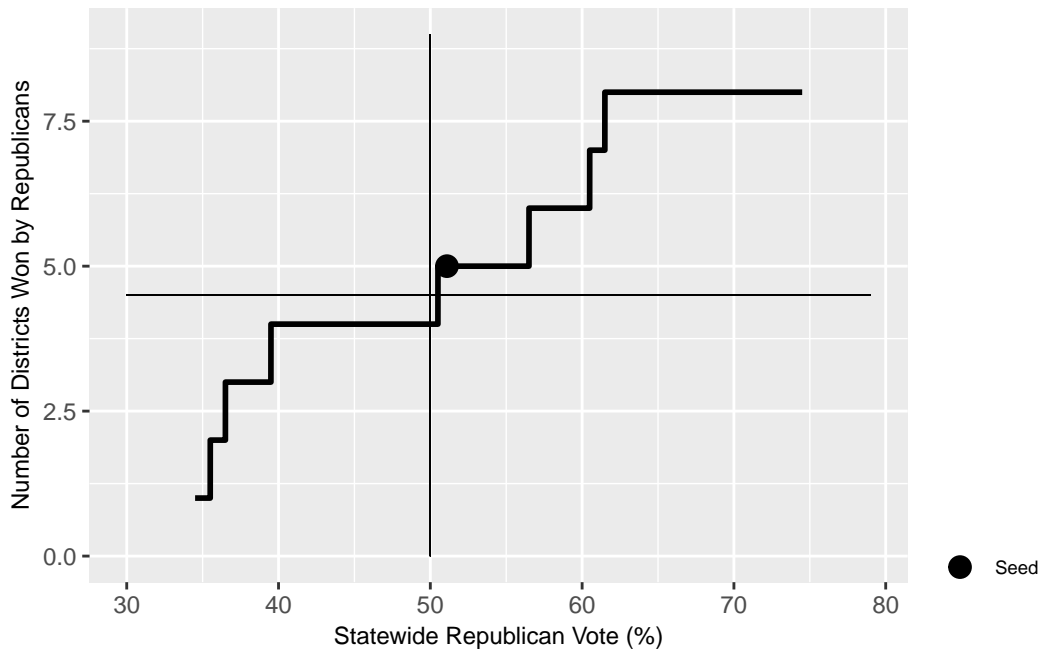


Figure 4.30: Bernstein-Style Simulations of Elections in Arizona. Using years 2012-2016 as the seed and an increment of 1.

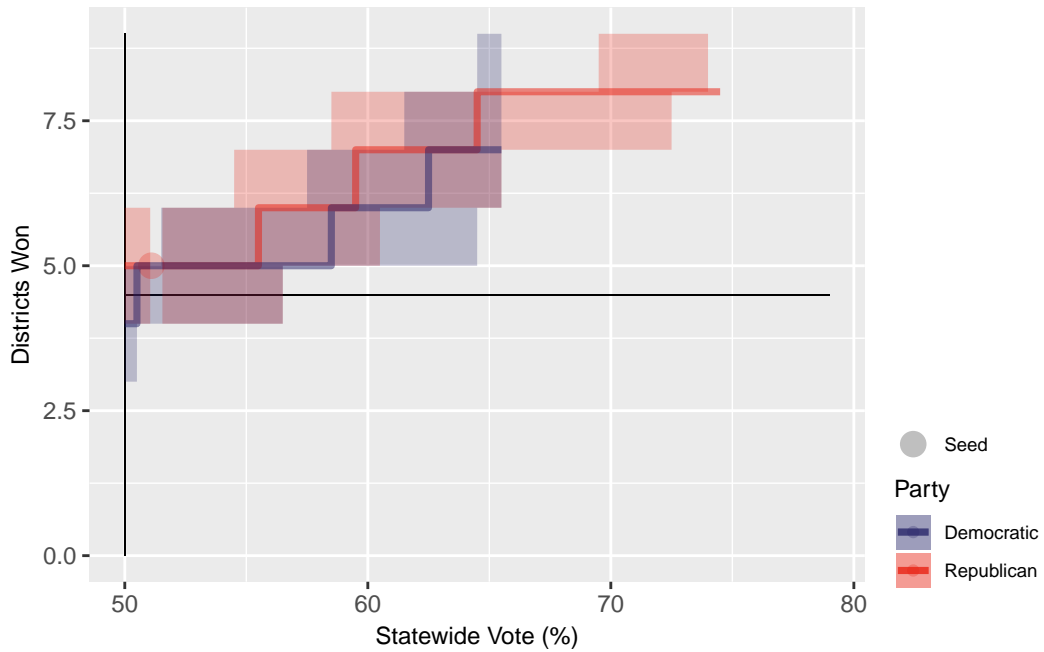


Figure 4.31: Step Chart of Statewide Vote and Seats Won by Party in Arizona. Simulations of elections in Arizona using years 2012-2016 as the seed. Shading represents values between 2.5th and 97.5th percentiles.  $n = 2000$  per step.

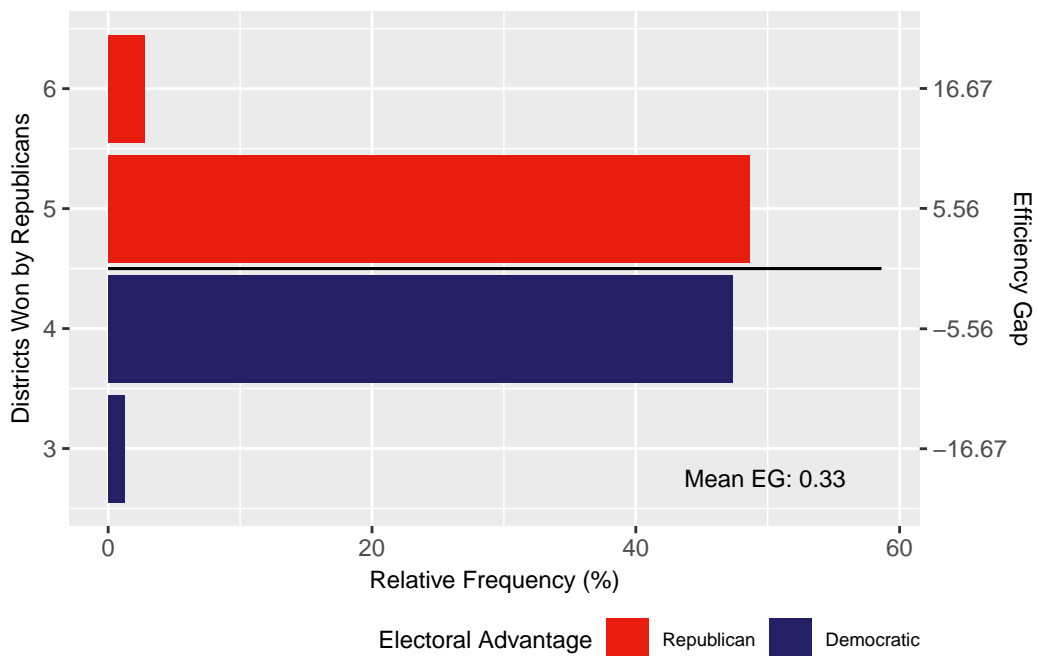


Figure 4.32: Bar Chart of Districts Won and Efficiency Gap in Arizona. Simulations of elections with 50 percent vote in Arizona using years 2012-2016 as the seed. Line segment indicates half of total districts.  $n = 2000$ .

### 4.3.6. Ohio

Ohio is typically a popular state to discuss for its political implications. Usually heralded as a battleground state every Presidential election, the state's Congressional map also warrants observation. Recent results suggest a Republican advantage as the party won 75% of districts with only 57.4% of the statewide vote in 2016 and 59% of the vote in 2014.

The results of a simulation of Ohio's Congressional map are shown in Figures 4.33 through 4.36. It appears the map is more advantageous for Republicans, who won more than half of the state's 16 seats in all of the simulated elections with a statewide vote of 50% (Figure 4.36). Additionally, the Republicans won 11 or more seats more than 95% of the time, producing a mean efficiency gap of 22.47%, the second largest behind only North Carolina.

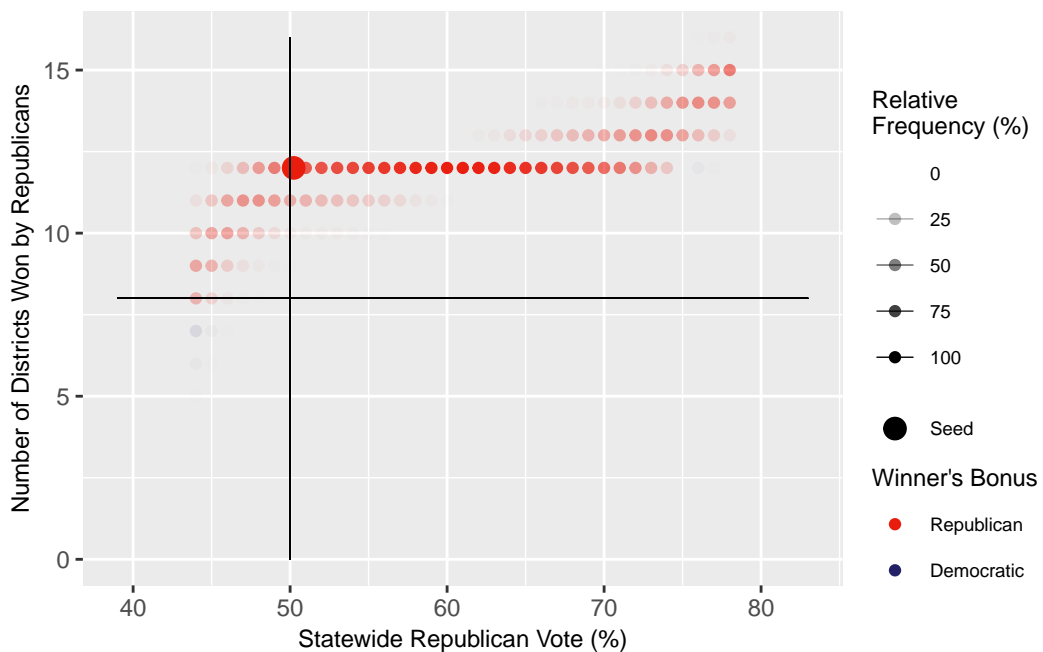


Figure 4.33: Scatter Plot of Statewide Vote and Seats Won in Ohio. Simulations of elections in Ohio using years 2012-2016 as the seed.  $n = 2000$  per step.

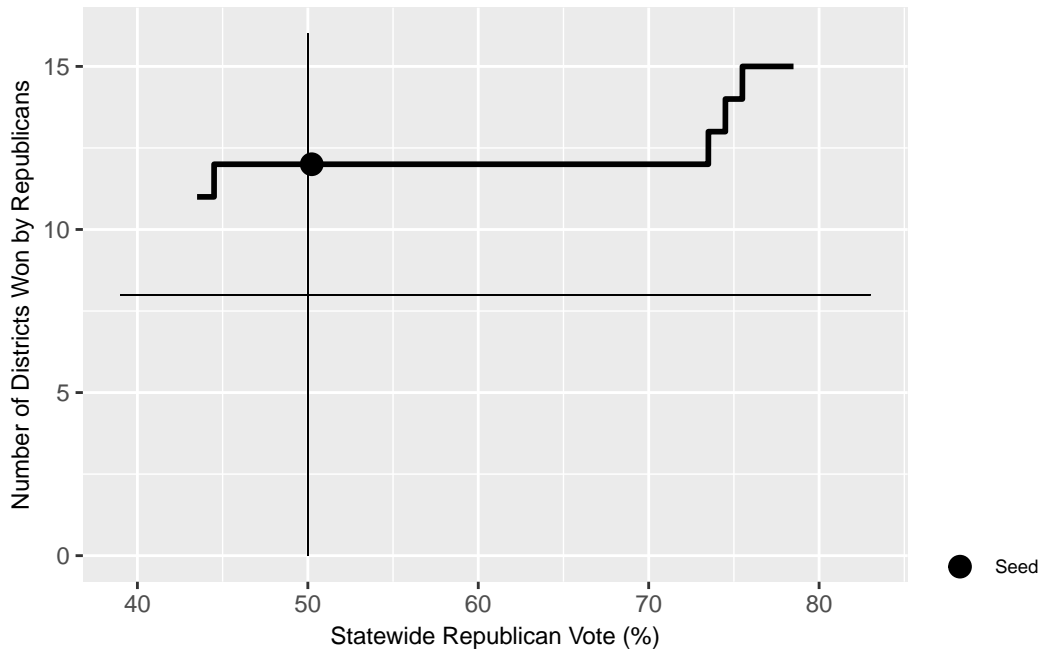


Figure 4.34: Bernstein-Style Simulations of Elections in Ohio. Using years 2012-2016 as the seed and an increment of 1.

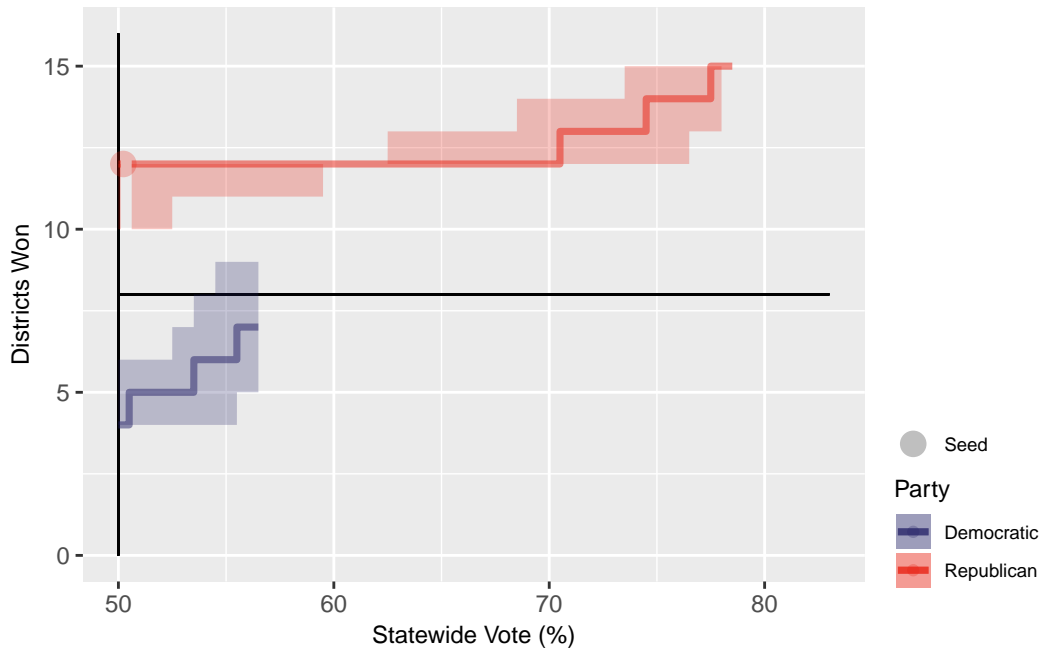


Figure 4.35: Step Chart of Statewide Vote and Seats Won by Party in Ohio. Simulations of elections in Ohio using years 2012-2016 as the seed. Shading represents values between 2.5th and 97.5th percentiles.  $n = 2000$  per step.



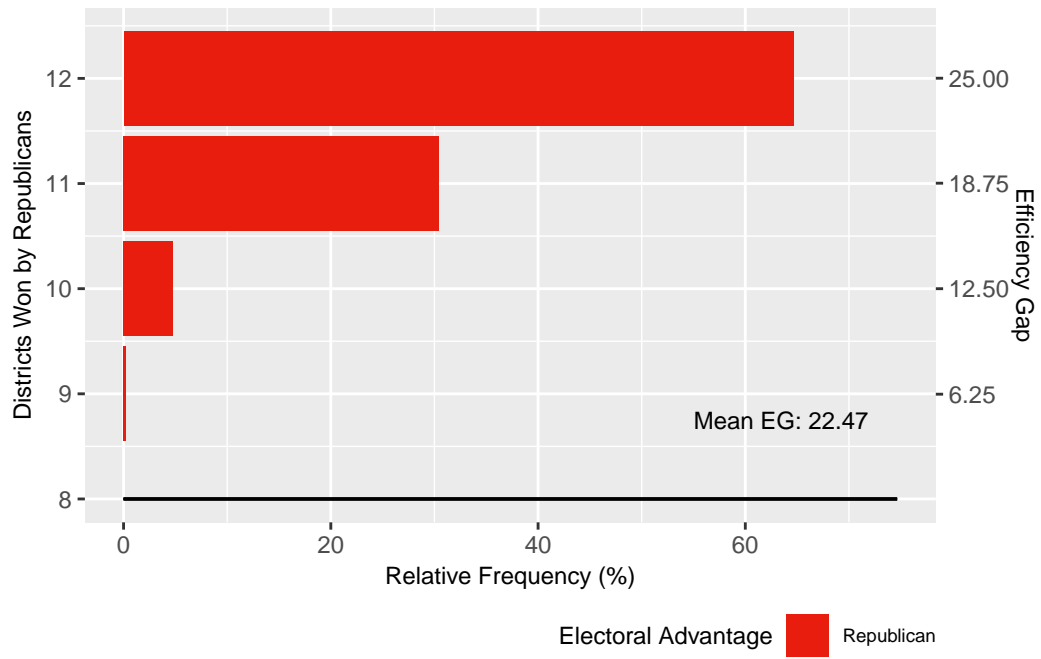


Figure 4.36: Bar Chart of Districts Won and Efficiency Gap in Ohio. Simulations of elections with 50 percent vote in Ohio using years 2012-2016 as the seed. Line segment indicates half of total districts.  $n = 2000$ .

## 5. Conclusions

Partisan gerrymandering will continue to be a topic of interest in the coming years as states will begin their redistricting process following the 2020 Census. Our simulations provide a new way of examining and visualizing the votes-to-seats relationship for a state Congressional map using past election data. We were able to build upon Mira Bernstein's method of uniformly simulating elections by injecting a data-driven component of variation into the simulations. Additionally, we were able to directly evaluate the accuracy of our simulations using a type of cross-validation.

When examining our states of interest, we see that the efficiency gap, Bernstein's uniform simulations, and our simulations are all capable of detecting a partisan advantage. Our simulation process, however, is more robust to random variation and provides a way of cutting through the noise and evaluating how likely a certain outcome is. By accounting for variation, we can more easily distinguish between an actual partisan advantage and a statistical anomaly in a relatively fair state.

There are limits to what our simulation process can do. The presence of a partisan advantage does not necessarily imply partisan gerrymandering has taken place. There are numerous factors such as human geography and compliance with the Voting Rights Act which can create a partisan advantage for one party even without gerrymandering taking place. The simulations only examine the partisan advantage present in a Congressional map and do nothing to evaluate the causes of it. Despite this limitation, we believe our simulation process, when paired with important context, provides a powerful tool for evaluating and visualizing the partisan fairness of a Congressional map.

## 6. Ideas for Future Research

### 6.1. Possible Test for Partisan Asymmetry

The simulation process we use for a single vote total is similar to the construction of a bootstrap confidence interval. We repeatedly sample with replacement from our residual distribution. We then apply the residuals to our seed value to determine the number of districts won for each sample. In our analysis, we observe the percentiles of the number of districts won similar to how one would select the 2.5th and 97.5th percentiles to construct a 95% bootstrap confidence interval. Despite the similarities, there are added layers, such as selecting or constructing a seed value and repeating the process for multiple vote totals, which separate this process from a bootstrap interval construction. The similarities do, however, suggest that a method of statistically testing for partisan asymmetry could be developed using bootstrap confidence intervals and this simulation process as a framework.

### 6.2. State Legislatures

Partisan gerrymandering is not just a concern at the national level, but also the state level as well. In addition to redrawing U.S. Congressional maps, states are also in charge of redrawing maps for their own state legislatures. This means partisan gerrymandering could be used to provide a partisan advantage in state houses as well. Our simulation process could be employed to examine the relationship between statewide vote and seats won for these state legislative bodies as well.

### 6.3. Analysis of Residuals

As mentioned previously, there did not appear to be any homoskedacity of the residuals across states. It would be of interest to examine what factors are driving the differing levels of variation in the state residuals. This could perhaps be done by collecting more historical Congressional election data, or maybe by examining the effects of different election laws by state. Differences could also be

due to external political or demographic factors. Whatever the reasons, a better understanding of what drives the variability in the residuals could allow for better informed and more accurate simulations.

#### 6.4. Effect of Fewer Data Restrictions

For the sake of simplicity, we chose to examine only situations in which there was low third-party vote and there were no candidates running unopposed. In reality, elections are not always that simple. It would be beneficial to observe how effective our simulation process could be at examining less simple situations using less simple data. Currently, we are incapable of conducting simulations of California's current Congressional map because there are a few districts in which Republicans have yet to have a candidate on the general election ballot. Perhaps performing a simulation on the maximal subset of usable districts could be an adequate substitute for a simulation of the entire state. Regardless of the methods used, a generalization of the simulation process to be able to reliably handle quirks such as the one in California would be a valuable improvement.

## References

- Arizona Prop. 106. (2000). *Citizens Independent Redistricting Commission Initiative*.
- Benisek v. Lamone, 585 U.S. \_\_\_ (2018)
- Bernstein, M. (2017, November). *Measures of partisan fairness*. Paper presented at Geometry of Redistricting conference, Durham, NC. Retrieved from <https://sites.duke.edu/gerrymandering/files/2017/11/MB-duke-slides.pdf>
- Chen, J., Rodden, J., et al. (2013). Unintentional gerrymandering: Political geography and electoral bias in legislatures. *Quarterly Journal of Political Science*, 8(3), 239–269.
- Cooper v. Harris, 581 U.S. \_\_\_ (2017)
- Davis v. Bandemer, 478 U.S. 109 (1986)
- Federal Election Commission. (n.d.). Election Results. Retrieved from <https://transition.fec.gov/pubrec/electionresults.shtml>
- Gill v. Whitford, 585 U.S. \_\_\_ (2018)
- Hyndman, R. J., & Fan, Y. (1996). Sample quantiles in statistical packages. *The American Statistician*, 50(4), 361–365.
- Stephanopoulos, N. O., & McGhee, E. M. (2015). Partisan gerrymandering and the efficiency gap. *U. chi. l. Rev.*, 82, 831.
- Tufte, E. R. (1973). The relationship between seats and votes in two-party systems. *American Political Science Review*, 67(2), 540–554.
- Vieth v. Jubelirer, 541 U.S. 267 (2004)
- Wang, S. S.-H. (2016). Three tests for practical evaluation of partisan gerrymandering. *Stan. L. Rev.*, 68, 1263.

*Who draws the maps? legislative and congressional redistricting.* (2018, Jun).  
Retrieved from <https://www.brennancenter.org/analysis/who-draws-maps-states-redrawing-congressional-and-state-district-lines>

## APPENDICES

## Appendix A: Additional Tables



Appendix A: Additional Tables

Table A.1: Full Cross-Validation Results. Districts won column contains actual results for each election. Inner 50% and Inner 95% column denote whether or not the actual result was between the 25th/75th and 2.5th/97.5th percentiles respectively.

State	Year	Districts Won	Percentage	Inner 50%	Inner 95%
Maryland	2008	1 (12.5%)	0%	No	No
Nebraska	2006	3 (100%)	0%	No	No
New Hampshire	2014	1 (50%)	0%	No	No
Kansas	2010	4 (100%)	1.05%	No	No
New Mexico	2010	1 (33.3%)	1.6%	No	No
North Carolina	2004	7 (53.8%)	2.95%	No	Yes
Kentucky	2010	4 (66.7%)	3.6%	No	Yes
New Hampshire	2012	0 (0%)	5.85%	No	Yes
Iowa	2010	2 (40%)	7%	No	Yes
Mississippi	2010	3 (75%)	10.4%	No	Yes
Colorado	2004	4 (57.1%)	10.5%	No	Yes
Indiana	2004	7 (77.8%)	12.3%	No	Yes
Virginia	2012	8 (72.7%)	13.4%	No	Yes
Ohio	2006	11 (61.1%)	15.95%	No	Yes
Connecticut	2004	3 (60%)	18.7%	No	Yes
Wisconsin	2010	5 (62.5%)	18.9%	No	Yes
North Carolina	2010	6 (46.2%)	19%	No	Yes
Mississippi	2008	1 (25%)	20.7%	No	Yes
New Mexico	2006	2 (66.7%)	21%	No	Yes
West Virginia	2010	2 (66.7%)	22.95%	No	Yes
Washington	2006	3 (33.3%)	26.25%	Yes	Yes
Iowa	2004	4 (80%)	30.25%	Yes	Yes
Kansas	2006	2 (50%)	38.65%	Yes	Yes
New Jersey	2012	6 (50%)	40%	Yes	Yes
Connecticut	2014	0 (0%)	41%	Yes	Yes
North Carolina	2012	9 (69.2%)	42.8%	Yes	Yes
Arkansas	2010	3 (75%)	43.75%	Yes	Yes
Washington	2004	3 (33.3%)	49.7%	Yes	Yes
Washington	2008	3 (33.3%)	50.8%	Yes	Yes
Colorado	2008	2 (28.6%)	54.35%	Yes	Yes
Connecticut	2006	1 (20%)	55%	Yes	Yes
Hawaii	2012	0 (0%)	65.15%	Yes	Yes
Michigan	2012	9 (64.3%)	65.5%	Yes	Yes
Nebraska	2008	3 (100%)	72.15%	Yes	Yes
New Hampshire	2006	0 (0%)	72.25%	Yes	Yes
Hawaii	2014	0 (0%)	73.65%	Yes	Yes
New Mexico	2004	2 (66.7%)	73.95%	Yes	Yes
Connecticut	2016	0 (0%)	75.45%	Yes	Yes
Wisconsin	2012	5 (62.5%)	77.1%	Yes	Yes

Table A.1 (continued)

State	Year	Districts Won	Percentage	Inner 50%	Inner 95%
Maryland	2014	1 (12.5%)	82.8%	Yes	Yes
North Carolina	2008	5 (38.5%)	83.7%	Yes	Yes
Iowa	2008	2 (40%)	83.9%	Yes	Yes
Wisconsin	2014	5 (62.5%)	85%	Yes	Yes
Indiana	2006	4 (44.4%)	89.25%	Yes	Yes
Indiana	2014	7 (77.8%)	89.9%	Yes	Yes
Kansas	2008	3 (75%)	90.15%	Yes	Yes
South Carolina	2006	4 (66.7%)	91.8%	Yes	Yes
New Jersey	2014	6 (50%)	94.35%	Yes	Yes
Kentucky	2012	5 (83.3%)	94.95%	Yes	Yes
Kentucky	2014	5 (83.3%)	95.75%	Yes	Yes
Nebraska	2004	3 (100%)	96.8%	Yes	Yes
Connecticut	2008	0 (0%)	96.9%	Yes	Yes
North Carolina	2016	10 (76.9%)	97.65%	Yes	Yes
Maryland	2004	2 (25%)	99.05%	Yes	Yes
South Carolina	2008	4 (66.7%)	99.6%	Yes	Yes
Connecticut	2012	0 (0%)	99.75%	Yes	Yes
Missouri	2004	5 (55.6%)	99.8%	Yes	Yes
South Carolina	2016	6 (85.7%)	99.85%	Yes	Yes
Maryland	2012	1 (12.5%)	99.9%	Yes	Yes
Missouri	2006	5 (55.6%)	99.95%	Yes	Yes
Hawaii	2004	0 (0%)	100%	Yes	Yes
Hawaii	2006	0 (0%)	100%	Yes	Yes
Hawaii	2008	0 (0%)	100%	Yes	Yes
Hawaii	2010	0 (0%)	100%	Yes	Yes
Maine	2016	1 (50%)	100%	Yes	Yes
New Hampshire	2004	2 (100%)	100%	Yes	Yes
New Hampshire	2008	0 (0%)	100%	Yes	Yes
New Mexico	2012	1 (33.3%)	100%	Yes	Yes
New Mexico	2014	1 (33.3%)	100%	Yes	Yes
New Mexico	2016	1 (33.3%)	100%	Yes	Yes
Oregon	2006	1 (20%)	100%	Yes	Yes
West Virginia	2004	1 (33.3%)	100%	Yes	Yes
West Virginia	2006	1 (33.3%)	100%	Yes	Yes

## Appendix B: R Scripts

## Appendix B: R Scripts

### Raw Data Script

```
#####  
#  
# program: raw_data.r  
# author: Zachary Morgan  
#  
# purpose: To take FEC election spreadsheets and compile a  
#           congressional election raw dataset  
#  
# inputs:  CSV files which were converted from the .xls files  
#           available on the FEC website.  
#           https://transition.fec.gov/pubrec/electionresults.shtml  
#  
# outputs: a raw dataset which can be later molded  
#           into an analysis dataset.  
#  
# run order: 1  
#  
#####  
  
library(dplyr)  
  
trimwsnb <- function(S) {  
  # Function for trimming leading a trailing whitespace including non-breaking  
  gsub("(^\\s+)|(\\s+$)", "", S)  
}  
  
states <- c('Alabama', 'Alaska', 'Arizona', 'Arkansas', 'California', 'Colorado',  
            'Connecticut', 'Delaware', 'Florida', 'Georgia', 'Hawaii', 'Idaho',  
            'Illinois', 'Indiana', 'Iowa', 'Kansas', 'Kentucky', 'Maine',  
            'Maryland', 'Massachusetts', 'Michigan', 'Minnesota', 'Mississippi',  
            'Missouri', 'Montana', 'Nebraska', 'Nevada', 'New_Hampshire',  
            'New_Jersey', 'New_Mexico', 'New_York', 'North_Carolina',  
            'North_Dakota', 'Ohio', 'Oklahoma', 'Oregon', 'Pennsylvania',  
            'Rhode_Island', 'South_Carolina', 'South_Dakota', 'Tennessee',  
            'Texas', 'Utah', 'Vermont', 'Virginia', 'Washington', 'West_Virginia',  
            'Wisconsin', 'Wyoming')  
  
# Creating empty raw dataset with desired structure  
raw.shell <- data.frame(  
  Year <- numeric(),  
  State <- character(),  
  District <- numeric(),  
  Republican <- numeric(),  
  Democratic <- numeric(),  
  Other <- numeric(),  
  Total <- numeric(),  
  Unopposed <- logical()  
)  
  
raw <- raw.shell  
  
# Importing CSV files and performing minor cleanups  
for (y in seq(from=2004, to=2016, by=2)) {  
  d <- read.csv(paste0(y, ".csv"), stringsAsFactors=FALSE)  
  d <- d[, colSums(is.na(d)) < nrow(d)]  
  d <- d[is.na(d[, c(1)]) == FALSE, ]  
  chars <- sapply(d, is.character)  
  d <- data.frame(cbind(sapply(d[, chars], trimwsnb), d[, !chars]))  
  
  d %>% mutate_if(is.factor, as.character) -> d  
  
  # Removing rows for non-Congressional elections, rows for Louisiana,  
  # and rows for non-states.  
  d <- d[d$STATE %in% states, ]  
  house.races <- grep("[0-9]{1,2}", d$D)  
  d <- d[house.races, ]  
}
```

```

# Removing some special elections
specs <- grep("\\*#",d$D)
if (length(specs) > 0) d <- d[- specs, ]

# Renaming variables for consistency
if (y < 2012) {
  names(d)[names(d) == 'GENERAL'] <- 'GENERAL.VOTES'
  names(d)[names(d) == 'DISTRICT'] <- 'D'
}
ind <- grep("COMBINED",names(d))[1]
colnames(d)[ind] <- 'COMBINED'

# Recoding unopposed elections to -1
unopp <- grep("Unopposed",d$GENERAL.VOTES)
d$GENERAL.VOTES[unopp] <- rep("-1", length(unopp))

# Only retaining rows with general election data
d <- d[! d$GENERAL.VOTES %in% c("", "n/a"),]

# Converting GENERAL.VOTES and COMBINED to numeric (if after 2004)
d$GENERAL.VOTES <- as.numeric(gsub(",","",d$GENERAL.VOTES))
if (y > 2004) d$COMBINED<- gsub(",","",d$COMBINED)

# Removing * from PARTY
d$PARTY <- gsub("\\*","",d$PARTY)

# Recoding PARTY values when necessary
d$PARTY <- gsub("REP|GOP","R",d$PARTY,ignore.case=TRUE)
d$PARTY <- gsub("DEM","D",d$PARTY,ignore.case=TRUE)
# Handling two states with different Democratic Party names
d$PARTY[d$STATE == "Minnesota"] <- gsub("DFL","D",
                                         d$PARTY[d$STATE == "Minnesota"])
d$PARTY[d$STATE == "North_Dakota"] <- gsub("DNL","D",
                                         d$PARTY[d$STATE == "North_Dakota"])

# Handling scenario where candidate is listed for multiple parties
d$PARTY <- gsub("R/.*/R","R",d$PARTY)
d$PARTY <- gsub("D/.*/D","D",d$PARTY)

# Removing rows where GENERAL.VOTES contained invalid numeric data
d <- d[is.na(d$GENERAL.VOTES) == FALSE, ]

# Using "combined" column where relevant (if after 2004)
if (y > 2004) {
  d$GENERAL.VOTES[! d$COMBINED == ""] <-
    as.numeric(d$COMBINED[! d$COMBINED == ""])
}

# Cleaning up district column
to.remove <- grep("UNEXPIRED",d$D)
if (length(to.remove) > 0) d <- d[- to.remove, ]
d$D <- trimws(d$D)
d$D <- as.numeric(substr(d$D,1,2))
at.large <- which(d$D == 0)
d$D[at.large] <- rep(1,length(at.large))

# Looping through each state and district to extract the desired values
raw.temp <- raw.shell
for (s in states){
  num_dis <- max(unique(d$D[d$STATE == s]))
  for (i in 1:num_dis){
    d.sub <- d[d$STATE == s & d$D == i, ]

    # Calculating Republican total
    # If multiple Republicans ran, their vote totals were added together.
    rep <- d.sub$GENERAL.VOTES[d.sub$PARTY == "R" & d.sub$TOTAL.VOTES==""]
    if (length(rep) > 1) rep <- sum(rep)
    if (length(rep)==0) rep <- c(0)

    # Similarly calculating Democratic total
    dem <- d.sub$GENERAL.VOTES[d.sub$PARTY == "D" & d.sub$TOTAL.VOTES==""]
    if (length(dem) > 1) dem <- sum(dem)
    if (length(dem)==0) dem <- c(0)
  }
}

```

```

# Extracting total votes
tot <- d.sub$GENERAL.VOTES[grepl("District_Votes:",d.sub$TOTAL.VOTES)]
if (length(tot)==0) tot <- c(0)

# Determining if a candidate was running unopposed
if (rep==-1 | dem==-1) {
  un <- TRUE
  rep <- NA
  dem <- NA
}
else un <- FALSE

# Putting the data in desired format
row <- data.frame(
  Year = c(y),
  State = c(s),
  District = c(i),
  Republican = c(rep),
  Democratic = c(dem),
  Total = c(tot),
  Unopposed = c(un)
)
row$Other <- row$Total - row$Republican - row$Democratic
raw.temp <- rbind(raw.temp, row)
}
}
raw <- rbind(raw,raw.temp)
}

# Outputting raw dataset
write.csv(raw, file="raw.csv",row.names = FALSE)

```

## Analysis Data Script

```
#####  
#  
# program: analysis_data.r  
# author: Zachary Morgan  
#  
# purpose: To take previously constructed raw dataset to create an  
#          analysis dataset for use in statistical analysis and  
#          simulations.  
#  
# inputs:  raw dataset created in raw_data.r program  
#  
# outputs: an analysis dataset which can be easily used to perform  
#          desired analysis and simulations.  
#  
# run order: 2  
#  
#####  
  
library(dplyr)  
  
raw <- read.csv("raw.csv")  
elections <- raw  
  
# Creating Percentage Columns  
elections$Rep_perc <- (elections$Republican /elections$Total)*100  
elections$Dem_perc <- (elections$Democratic /elections$Total)*100  
elections$Oth_perc <- (elections$Other / elections$Total)*100  
  
# Since new congressional maps were used beginning in 2002,  
# data from 2000 doesn't add anything to our analysis  
included <- elections[elections$Year != 2000 ,]  
included <- included[rowSums(is.na(included)) != ncol(included),]  
  
# Creating an adjusted unopposed column which also includes districts  
# where only one major party candidate ran which we deem to be an  
# "essentially unopposed" election.  
ind = which(included$Rep_perc > included$Dem_perc)  
included$min_party_perc <- included$Rep_perc  
included$min_party_perc[ind] <- included$Dem_perc[ind]  
  
included$Unopp.adj <- included$Unopposed  
included$Unopp.adj[which(included$min_party_perc == 0)] <- rep.int(  
  TRUE, table(included$min_party_perc)[1])  
  
# The following section creates a column, map, which serves to  
# keep different congressional maps separate.  
state_level <- data.frame(State=character(), Year=numeric(),  
  map=numeric())  
  
for (s in unique(included$State)) {  
  for (y in unique(included$Year)) {  
    if (! s %in% c("Maine", "Georgia", "Texas", "Florida")) {  
      if (y < 2012) m <- 1  
      else m <- 2  
    }  
    else if (s=="Maine") {  
      if (y == 2002) m <- 1  
      else if (y < 2012) m <- 2  
      else m <- 3  
    }  
    else if (s=="Georgia") {  
      if (y < 2006) m <- 1  
      else if (y == 2006) m <- 2  
      else if (y < 2012) m <- 3  
      else m <- 4  
    }  
    else if (s=="Texas") {  
      if (y == 2002) m <- 1  
      else if (y == 2004) m <- 2  
      else if (y < 2012) m <- 3  
    }  
  }  
}
```

```

    else m <- 4
  }
  else if (s=="Florida") {
    if (y<2012) m <- 1
    else if (y<2016) m<-2
    else m<-3
  }
  row <- data.frame(State=c(s), Year=c(y), map=c(m))
  state_level <- rbind(state_level, row)
}
}

included <- merge(included, state_level, by=c("State", "Year"))

# Adjusting Republican and Democratic Percentages for Third Party Vote
main.party.total <- included$Rep_perc + included$Dem_perc
included$Rep_perc.adj <- (included$Rep_perc / main.party.total) *100
included$Dem_perc.adj <- (included$Dem_perc / main.party.total) *100

# Computing lag columns which have the previous election's
# district percentages for the same congressional map, if applicable.
# Also adding a lag column for adjusted unopposed
included <- included[order(included$State, included$District, included$Year), ]

included <- included %>% group_by(State, District, map) %>%
  mutate(lag.Rep_perc = dplyr::lag(Rep_perc, n = 1, default=NA))

included <- included %>% group_by(State, District, map) %>%
  mutate(lag.Dem_perc = dplyr::lag(Dem_perc, n = 1, default=NA))

included <- included %>% group_by(State, District, map) %>%
  mutate(lag.Rep_perc.adj = dplyr::lag(Rep_perc.adj, n = 1, default=NA))

included <- included %>% group_by(State, District, map) %>%
  mutate(lag.Dem_perc.adj = dplyr::lag(Dem_perc.adj, n = 1, default=NA))

included <- included %>% group_by(State, District, map) %>%
  mutate(lag.Oth_perc = dplyr::lag(Oth_perc, n = 1, default=NA))

included <- included %>% group_by(State, District, map) %>%
  mutate(lag.Unopp.adj = dplyr::lag(Unopp.adj, n = 1, default=NA))

# Computing the shift in party support from the previous election, if applicable
included <- included[order(included$State, included$Year, included$District),]
included$Rep_shift.adj <- included$Rep_perc.adj - included$lag.Rep_perc.adj
included$Dem_shift.adj <- included$Dem_perc.adj - included$lag.Dem_perc.adj

# Writing the shifts as NA where exclusion criteria apply
# If third party support is over 5% for current or previous year
# If the district was essentially unopposed for the current or previous year

included$Rep_shift.adj[is.na(included$Rep_shift.adj) == FALSE &
  (included$Unopp.adj == TRUE | included$lag.Unopp.adj == TRUE |
  included$Oth_perc > 5 | included$lag.Oth_perc > 5)] <-
  rep(NA, length(included$Rep_shift.adj[is.na(included$Rep_shift.adj) == FALSE &
  (included$Unopp.adj == TRUE | included$lag.Unopp.adj == TRUE |
  included$Oth_perc > 5 | included$lag.Oth_perc > 5)]))

included$Dem_shift.adj[is.na(included$Dem_shift.adj) == FALSE &
  (included$Unopp.adj == TRUE | included$lag.Unopp.adj == TRUE |
  included$Oth_perc > 5 | included$lag.Oth_perc > 5)] <-
  rep(NA, length(included$Dem_shift.adj[is.na(included$Dem_shift.adj) == FALSE &
  (included$Unopp.adj == TRUE | included$lag.Unopp.adj == TRUE |
  included$Oth_perc > 5 | included$lag.Oth_perc > 5)]))

# function for writing certain observations as NA where appropriate
drop <- function (id) {
  out <- included
  out$Rep_shift.adj[out$State == id[1] & out$Year == id[2] &
    out$District == id[3]] <- NA

  out$Dem_shift.adj[out$State == id[1] & out$Year == id[2] &
    out$District == id[3]] <- NA
}

```



```

    return(out)
  }

# excluding appropriate observations
included <- drop(c("Ohio",2006,18))

# Writing the shifts as NA if only one shift is available for a given state & year
n.adj <- aggregate.data.frame(
  included$Rep_shift.adj[is.na(included$Rep_shift.adj) == FALSE],
  by=c(list(included$State[is.na(included$Rep_shift.adj) == FALSE]),
  list(included$Year[is.na(included$Rep_shift.adj) == FALSE])), FUN=length)

names(n.adj) <- c("State","Year","n.adj")
included <- merge(included, n.adj, by=c("State", "Year"), all.x=TRUE)

ind <- which(is.na(included$n.adj) == FALSE & included$n.adj == 1)
included$Rep_shift.adj[ind] <- rep(NA, length(ind))
included$Dem_shift.adj[ind] <- rep(NA, length(ind))

# Computing state-wide mean party shifts and creating a column for the
# number of districts in a state.

means <- aggregate.data.frame(included[, c("Rep_shift.adj", "Dem_shift.adj")],
  by=c(list(included$State), list(included$Year)), FUN=mean, na.rm=TRUE)

counts <- aggregate.data.frame(included$District, by=c(list(included$State),
  list(included$Year)), FUN=length)

names(means) <-c("State","Year","Rep_shift_sw","Dem_shift_sw")
names(counts) <- c("State","Year","num_dis")

# Labeling states as small/medium/large based on the number of districts
size <- function (nums) {
  s <- character()
  for (n in nums) {
    if (n <= 9) s<-c(s, "Small_(9_or_Fewer)") else if (n <= 19) s<-
      c(s,"Medium_(Between_10_and_19)") else s<-c(s,"Large_(20_or_More)")
  }
  return (s)
}

counts$size <- sapply(counts$num_dis, FUN=size)
summary <- merge(means, counts, by=c("State", "Year"))
analysis <- merge(included, summary, by=c("State", "Year"))

# Computing residual shifts as the shifts in excess of the state wide shifts.
analysis$Rep_shift.resid <- analysis$Rep_shift.adj - analysis$Rep_shift_sw
analysis$Dem_shift.resid <- analysis$Dem_shift.adj - analysis$Dem_shift_sw

# Removing the state of Louisiana
analysis <- analysis[analysis$State != "Louisiana", ]

# Creating an inclusion variable
analysis$incl <- rep(TRUE,nrow(analysis))
where.false <- which(analysis$Oth_perc >= 5 | analysis$Unopp.adj == TRUE)
analysis$incl[where.false] <- rep(FALSE,length(where.false))

# Outputting dataset
write.csv(analysis, file="analysis.csv",row.names = FALSE)

```

## Plots Script

```
#####  
#  
# program: plots.r  
# author: Zachary Morgan  
#  
# purpose: To produce plots and summaries which help to understand  
#           and visualize the analysis dataset  
#  
# inputs:  analysis dataset created in analysis_data.r program  
#  
# outputs: various plots and summaries.  
#  
# run order: 3  
#  
#####  
# --- preplot  
library(dplyr)  
library(ggplot2)  
  
# Importing dataset for analysis  
imported <- read.csv("analysis.csv")  
analysis <- imported[is.na(imported$Rep_shift.adj) == FALSE &  
                     imported$num_dis > 1, ]  
  
# Changing applicable variables to factor type  
analysis$Year <- factor(analysis$Year)  
analysis$Unopposed <- factor(analysis$Unopposed)  
analysis$Unopp.adj <- factor(analysis$Unopp.adj)  
analysis$map <- factor(analysis$map)  
analysis$size <- factor(analysis$size)  
  
#####  
# Theme to be used for plots  
  
theme <- theme(plot.title=element_text(size=12,  
                                       face="bold",  
                                       #family="American Typewriter",  
                                       color="black",  
                                       hjust=0.5,  
                                       lineheight=1.2),  
              plot.subtitle=element_text(size=9,  
                                       #face="bold",  
                                       #family="American Typewriter",  
                                       color="black",  
                                       hjust=0.5,  
                                       lineheight=1.2),  
              axis.title=element_text(size=9),  
              plot.caption=element_text(size=7),  
              legend.title = element_text(size=9, color = "black"),  
              legend.text=element_text(size=7),  
              legend.justification=c(1,0),  
              legend.position='right',  
              legend.background = element_blank(),  
              legend.key = element_blank(),  
              legend.margin=margin(-3,0,-3,0))  
  
colors <- c("#E91D0E", "#232066")  
  
# ---  
#####  
# Plot of data points vs variance in shifts  
# grouped by state and year  
  
grouped <- group_by(analysis, State, Year)  
spread <- summarize(grouped, iqr=IQR(Rep_shift.resid),  
                    var=var(Rep_shift.resid),  
                    range=range(Rep_shift.resid)[2]-range(Rep_shift.resid)[1],  
                    data.points=length(Rep_shift.resid))
```

```

ggplot(spread, aes(x=State,y=var,color=data.points,shape=Year)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

ggplot(spread, aes(x=data.points, y=var)) + geom_point()

#####
# Histogram and Normal Prob Plot of Repub District Shift Residuals
# ---- residhist

resids <- unlist(analysis$Rep_shift.resid)

# Creating fitted histogram
bw <- 0.5
hist <- qplot(resids, geom = "histogram",
  breaks = seq(min(resids)-bw/2,max(resids)+bw/2, bw),
  colour = I("black"), fill = I("white"),
  xlab = "Shift_in_District_Level_Support(%)",
  ylab = "Count",
  main=NULL) +
  theme +
  stat_function(
  fun = function(x, mean, sd, n, bw){
    dnorm(x = x, mean = mean, sd = sd) * n * bw
  },
  args = c(mean = mean(resids), sd =sd(resids),
    n = length(which(is.na(resids) == FALSE)), bw = bw))

# Normal Probability Plot
npp <- ggplot(data.frame(resids=(resids-mean(resids))/sd(resids)),
  aes(sample = resids)) +
  stat_qq() +
  geom_segment(aes(x=-4,y=-4,xend=4,yend=4)) +
  labs(title=NULL,x="Theoretical",y="Sample") +
  theme

grid.arrange(hist,npp,ncol=2)

#####
# Boxplots of residual values by state, by year, and by size.

# ---- boxstate
ggplot(analysis, aes(y=Rep_shift.resid, x=State)) +
  geom_boxplot(aes(group=State)) +
  labs(title=NULL,
  y="Residual_Shift(%)", x="State") + theme +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

# ---- boxyear
ggplot(analysis, aes(y=Rep_shift.resid, x=Year)) +
  geom_boxplot(aes(group=Year)) +
  labs(title=NULL,
  y="Residual_Shift(%)", x="Year") + theme

# ---- boxsize
ggplot(analysis, aes(y=Rep_shift.resid, x=size)) +
  geom_boxplot(aes(group=size)) +
  labs(title=NULL,
  y="Residual_Shift(%)", x="Year") + theme +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust=1))

```

## Simulation Script

```
#####  
#  
# program: simulation.r  
# author: Zachary Morgan  
#  
# purpose: To perform simulations using analysis dataset  
#  
# inputs: analysis dataset created in analysis_data.r program  
#  
# outputs: simulation results and plots  
#  
# run order: 4  
#  
#####  
# --- presim  
library(tidyr)  
library(dplyr)  
library(car)  
library(ggplot2)  
library(grid)  
library(gridExtra)  
  
# Importing dataset for analysis  
imported <- read.csv("analysis.csv")  
analysis <- imported[is.na(imported$Rep_shift.adj) == FALSE &  
  imported$num_dis > 1, ]  
  
# Changing applicable variables to factor type  
analysis$Year <- factor(analysis$Year)  
analysis$Unopposed <- factor(analysis$Unopposed)  
analysis$Unopp.adj <- factor(analysis$Unopp.adj)  
analysis$map <- factor(analysis$map)  
analysis$size <- factor(analysis$size)  
  
#####  
# Theme to be used for plots  
  
theme <- theme(plot.title=element_text(size=12,  
  face="bold",  
  #family="American Typewriter",  
  color="black",  
  hjust=0.5,  
  lineheight=1.2),  
  plot.subtitle=element_text(size=9,  
  #face="bold",  
  #family="American Typewriter",  
  color="black",  
  hjust=0.5,  
  lineheight=1.2),  
  axis.title=element_text(size=9),  
  plot.caption=element_text(size=7),  
  legend.title = element_text(size=9, color = "black"),  
  legend.text=element_text(size=7),  
  legend.justification=c(1,0),  
  legend.position='right',  
  legend.background = element_blank(),  
  legend.key = element_blank(),  
  legend.margin=margin(-3,0,-3,0))  
  
colors <- c("#E91D0E", "#232066")  
  
#####  
# Efficiency Gap (simplified version) function  
# --- effgap  
EG <- function(vote, seats) {  
  if (! length(vote) == length(seats)) {  
    print("Vector lengths do not match")  
    return(NULL)  
  }  
}
```

```

    (100*seats- 50) - 2*(100*vote - 50)
  }

#####
# Uniform simulation
# ---- uniformsim

step.func <- function(X,votes,seats,step) {
  Y <- numeric()
  for (x in X) {
    d <- abs(x-votes)
    ind <- which(d == min(d))
    if (length(ind) == 1 & d[ind[1]] < step/2) {
      Y = c(Y,seats[ind])
    } else {
      Y = c(Y,NA)
    }
  }
  return(Y)
}

uniform.sim <- function(state,year,step){
  to.use <- imported[imported$State == state & imported$Year == year, ]
  seed <- to.use$Rep_perc.adj
  num_dis <- length(seed)

  min_shift <- -30
  max_shift <- 20

  points <- data.frame(vote=numeric(),seats=numeric(),seed=character())
  shifts <- seq(from=min_shift, to=max_shift, by=step)
  for (shift in shifts) {
    elec <- seed + shift
    v <- mean(elec)
    s <- length(which(elec > 50))
    if (shift==0){
      is.seed <- "Seed"
    }
    else is.seed <- ""
    point <- data.frame(vote=c(v),seats=c(s),seed=c(is.seed))
    points <- rbind(points, point)
  }

  # Scatter Plot of Simulation
  scat <- ggplot(points, aes(x=vote, y=seats)) +
    geom_point(alpha=1,aes(size=seed)) +
    geom_segment(aes(x=50,y=0,xend=50,yend=num_dis),
                 color=I('black'),size=.25) +
    geom_segment(aes(x=0,y=num_dis/2,xend=100,yend=num_dis/2),
                 color=I('black'),size=.25) +
    stat_function(fun=step.func,args=list(
      votes=points$vote[points$seed==""],
      seats=points$seats[points$seed==""],
      step=1),
      n=10000, size=1,na.rm=TRUE) +
    labs(title=NULL, subtitle=NULL,
         x="Republican_Vote_("%", y="Number_of_Seats_Won",size="") +
    scale_size_manual(breaks = c("Seed"), values=c(0,3.5)) + theme

  return(list(data=points,scat=scat))
}

#####
# ---- sim
# Simulation of Partisan Symmetry

# Seed generation function
gen.seed <- function(to.use, y) {
  # Standardizing for year to year shifts by expressing each
  # percentage in year.start terms
  shift.from.year1 <- 0

```

```

if (length(y) > 1) {
  for (i in 2:length(y)) {
    shift.from.year1 <- shift.from.year1+
      to.use$Rep_shift_sw[to.use$Year == y[i]][1]

    to.use$Rep_perc.adj[to.use$Year == y[i]] <-
      to.use$Rep_perc.adj[to.use$Year == y[i]] - shift.from.year1

    to.use$Dem_perc.adj[to.use$Year == y[i]] <-
      100 - to.use$Rep_perc.adj[to.use$Year == y[i]]
  }
}

to.use <- to.use[! is.na(to.use$Rep_perc.adj), ]

# Deriving seed values if possible
grouped <- group_by(to.use[to.use$incl == TRUE, ], District)
rep_seed <- unlist(summarize(grouped, val=mean(Rep_perc.adj))$val)
num_dis = to.use$num_dis[1]
if (length(rep_seed) < num_dis) {
  print("Unable to complete seed.")
  return(NULL)
}

return(rep_seed)
}

# Core simulation function
sim<- function(s,year.start, year.end, means=NA, n=2000, fill.incom=TRUE,
  nw.distrib=FALSE,cross.validate=FALSE,invert.seed=FALSE){
  # s <- State
  # year.start <- First year to use in seed
  # year.end <- last year to use in seed
  # means <- statewide mean to be used for simulation
  # n <- Number of Simulated Datapoints per step
  # fill.incom <- Determine whether or not to fill in seed gaps
  # nw.distrib <- Logical variable stating whether or not to use
  #   nationwide distribution of residuals
  # cross.validate <- Specify whether or not to include selected
  #   years in residual distribution to sample from
  # invert.seed <- Specify whether or not to select the years NOT
  #   specified which share the same congressional map

  # Selecting distribution of residuals to sample from
  if (nw.distrib) {
    resids <- analysis[,c("Year","State","Rep_shift.resid")]
  } else if (! nw.distrib) {
    resids <- analysis[analysis$State == s,
      c("Year","State","Rep_shift.resid")]
  }

  if (cross.validate == TRUE) {
    resids <- resids[! (resids$State == s & resids$Year %in%
      seq(from=year.start, to=year.end+2,by=2)),]
  }

  resids <- unlist(resids$Rep_shift.resid)
  if (length(resids) == 0) {
    print("No residuals available for desired year(s)")
    return(NULL)
  }

  # Creating sequence of years to be used
  y <- seq(from=year.start, to=year.end, by=2)

  # Ensuring all selected years come from the same congressional map
  if (length(unique(imported$map[imported$State==s &
    imported$Year %in% y])) > 1) {
    print("Years come from different congressional maps.")
    return(NULL)
  }

  map <- unlist(imported$map[imported$State==s &

```

```

imported$Year %in% y)][1]

# Taking relevant subset of data
to.use <- imported[imported$State==s & imported$Year %in% y, ]
if (is.null(nrow(to.use))) {
  print("No included records for desired year")
  return(NULL)
}

# Detecting incomplete seeds if necessary
if (fill.incom==FALSE & length(which(to.use$incl==FALSE)) > 0) {
  print("Seed incomplete.")
  return(NULL)
}

num_dis = to.use$num_dis[1]

if (num_dis == 1) {
  print("Single district state")
  return(NULL)
}
rep_seed <- gen.seed(to.use, y)
if (is.null(rep_seed)) return (NULL)
seed_mean <- mean(rep_seed)

# If means is NA, assigning it to be mean of supplied years
if (is.na(means[1]) == TRUE) means <- mean(rep_seed)

# Calculating wins
rep_wins <- length(which(rep_seed > 50))

# Initializing data frame
cum_freq <- data.frame(dis_won=c(rep_wins),mean=c(seed_mean),
                      freq=c(NA),perc=c(NA),seed=c(TRUE))

# Inverting seed if necessary
if (invert.seed) {
  to.use <- imported[imported$State==s &
                    (! imported$Year %in% y) & imported$map == map, ]
  if (is.null(nrow(to.use))) {
    print("No included records for desired year")
    return(NULL)
  }
  rep_seed <- gen.seed(to.use,
                      unique(imported$Year[(! imported$Year %in% y) &
                                           imported$map == map & imported$State == s]))

  if (is.null(rep_seed)) return (NULL)
  seed_mean <- mean(rep_seed)
}

# Performing a simulation
for (m in means) {
  #m <- 50
  i <- m-mean(rep_seed) # Calculating specified statewide shift
  simulation <- rep(NULL,n)
  for (j in 1:num_dis) {
    # Ensureing no impossible values are generated by truncating the
    # sample distribution.
    to.sample <- resid[rep_seed[j] + resid + i >= 0 & rep_seed[j]
                      + resid + i <= 100]
    sim <- sample(to.sample,size=n,replace=TRUE)
    simulation <- cbind(simulation, sim)
  }
  simulation <- as.data.frame(apply(simulation, MARGIN = 2,
                                  FUN="-", rowMeans(simulation)))

  simulation <- as.data.frame(t(apply(simulation, MARGIN = 1,
                                    FUN="+", rep_seed))) + i

  simulation$dis_won <- rowSums(simulation[,1:num_dis] > 50)
  simulation$mean <- rowMeans(simulation[,1:num_dis])
}

```

```

simulation$seed <- rep(FALSE, n)

freqs <- as.data.frame(table(simulation$dis_won))
names(freqs) <- c("dis_won", "freq")
freqs$dis_won <- as.numeric(as.character(freqs$dis_won))
freqs$perc <- 100 * freqs$freq / n
freqs$mean <- rep(m, nrow(freqs))
freqs$seed <- rep(FALSE, nrow(freqs))
cum_freq <- rbind(cum_freq, freqs)
}
return(cum_freq)
}

# Simulation function with plots
sim.range <- function(s, year.start, year.end, n=2000, step=1,
                     fill.incom=TRUE, nw.distrib=FALSE) {
  # s <- State
  # year.start <- First year to use in seed
  # year.end <- last year to use in seed
  # n <- Number of Simulated Datapoints per step
  # step <- Difference between simulated shifts
  # nw.distrib <- Logical variable stating whether or not to use
  # nationwide distribution of residuals

  # Creating sequence of years to be used
  y <- seq(from=year.start, to=year.end, by=2)

  # Taking relevant subset of data
  to.use <- imported[imported$State==s & imported$Year %in% y, ]
  if (is.null(nrow(to.use))) {
    print("No included records for desired year")
    return(NULL)
  }

  # Ensuring all selected years come from the same congressional map
  if (length(unique(to.use$map)) > 1) {
    print("Years come from different congressional maps.")
    return(NULL)
  }

  # Detective incomplete seeds if necessary
  if (fill.incom==FALSE & length(which(to.use$incl==FALSE)) > 0) {
    print("Seed incomplete.")
    return(NULL)
  }

  # Standardizing for year to year shifts by expressing each
  # percentage in year.start terms
  shift.from.year1 <- 0
  if (length(y) > 1) {
    for (i in 2:length(y)) {
      shift.from.year1 <- shift.from.year1 +
        to.use$Rep_shift_sw[to.use$Year == y[i]][1]
      to.use$Rep_perc.adj[to.use$Year == y[i]] <-
        to.use$Rep_perc.adj[to.use$Year == y[i]] - shift.from.year1
      to.use$Dem_perc.adj[to.use$Year == y[i]] <-
        100 - to.use$Rep_perc.adj[to.use$Year == y[i]]
    }
  }

  # Deriving seed values if possible
  grouped <- group_by(to.use[to.use$incl == TRUE, ], District)
  rep_seed <- unlist(summarize(grouped, val=mean(Rep_perc.adj))$val)
  seed_mean <- mean(rep_seed)
  num_dis = to.use$num_dis[1]
  if (length(rep_seed) < num_dis) {
    print("Unable to complete seed.")
    return(NULL)
  }

  # Determining minimum and maximum shifts.
  # Ensuring no more than 5% of resid dist would lead to impossible percentages

```



```

q <- quantile(analysis$Rep_shift.resid,probs=c(.05,.95))

min_shift <- - q[1] - min(rep_seed)
max_shift <- 100 - q[2] - max(rep_seed)

min_mean <- ceiling(min_shift + seed_mean)
max_mean <- floor(max_shift + seed_mean)

# Creating sequence of shifts
#state_shifts <- seq(min_shift, max_shift, by=step)
state_means <- seq(min_mean, max_mean, by=step)
steps <- length(state_means)

# Simulating shifts
cum_freq <- sim(s=s,year.start = year.start, year.end=year.end, means=state_means,
              fill.incom=TRUE, nw.distrib = nw.distrib, n=n)

cum_freq$eg <- EG(cum_freq$mean/100, cum_freq$dis_won/num_dis)

cum_sim <- cbind(cum_freq[cum_freq$seed == TRUE, c("dis_won","mean","seed")],est=c(NA))
for (m in unique(unlist(cum_freq$mean))) {
  data <- rep(cum_freq$dis_won[cum_freq$mean == m & cum_freq$seed==FALSE],
            cum_freq$freq[cum_freq$mean == m & cum_freq$seed==FALSE])
  q <- quantile(data, probs=c(0.025,0.25,0.75,0.975))
  sim_means <- data.frame(dis_won=c(round(mean(data))),mean=c(m),
                        seed=c(FALSE),est=c("mean"))
  sim_025 <- data.frame(dis_won=round(q[1]),mean=c(m),seed=c(FALSE),est=c(2.5))
  sim_25 <- data.frame(dis_won=round(q[2]),mean=c(m),seed=c(FALSE),est=c(25))
  sim_75 <- data.frame(dis_won=round(q[3]),mean=c(m),seed=c(FALSE),est=c(75))
  sim_975 <- data.frame(dis_won=round(q[4]),mean=c(m),seed=c(FALSE),est=c(97.5))
  cum_sim <- rbind(cum_sim, sim_means)
  cum_sim <- rbind(cum_sim, sim_025)
  cum_sim <- rbind(cum_sim, sim_25)
  cum_sim <- rbind(cum_sim, sim_75)
  cum_sim <- rbind(cum_sim, sim_975)
}
cum_sim$Bonus <- (100*cum_sim$dis_won / num_dis) - cum_sim$mean
cum_freq$Bonus <- (100*cum_freq$dis_won / num_dis) - cum_freq$mean

cum_sim$seed.plot <- rep("",nrow(cum_sim))
cum_sim$seed.plot[cum_sim$seed] <- "Seed"

cum_freq$seed.plot <- rep("",nrow(cum_freq))
cum_freq$seed.plot[cum_freq$seed] <- "Seed"

cum_sim$Bonus.party <- rep(NA, nrow(cum_sim))
adv <- which(cum_sim$Bonus > 0)
cum_sim$Bonus.party[adv] <- rep("Republican",length(adv))
disadv <- which(cum_sim$Bonus < 0)
cum_sim$Bonus.party[disadv] <- rep("Democratic",length(disadv))

cum_freq$Bonus.party <- rep(NA, nrow(cum_freq))
adv <- which(cum_freq$Bonus > 0)
cum_freq$Bonus.party[adv] <- rep("Republican",length(adv))
disadv <- which(cum_freq$Bonus < 0)
cum_freq$Bonus.party[disadv] <- rep("Democratic",length(disadv))

cum_sim$effgap <- (100*cum_sim$dis_won / num_dis - 50) - 2*(cum_sim$mean-50)
absmean.effgap <- mean(abs(cum_sim$effgap))
#print(absmean.effgap)

# Labels for plots
t <- "Statewide_Vote_and_Seats_Won"
if (length(y) == 1) {
  st <- paste("Simulations_of_Elections_in",s,"_Using_",y,"_as_the_Seed",sep="")
} else if (length(y) > 1) {
  st <- paste("Simulations_of_Elections_in",s,"_Using_Years_",
            paste0(year.start,"-",year.end),"_as_Seed",sep="")
}
xlab <- "Statewide_Republican_Vote_%"
ylab <- "Number_of_Districts_Won_by_Republicans"
capt <- paste("n=",n,"_/step",sep="")

```

```

# Scatter Plot of Simulation
scat <- ggplot(cum_freq, aes(x=mean, y=dis_won, color=Bonus.party, alpha=perc,
                             size=seed.plot),na.rm=TRUE) +
  geom_point(na.rm=TRUE) +
  geom_segment(aes(x=50,y=0,xend=50,yend=num_dis),color='black',size=.125) +
  geom_segment(aes(x=max(min(cum_freq$mean)-5,0),y=num_dis/2,
                   xend=min(max(cum_freq$mean)+5,100),yend=num_dis/2),color='black',size=.125) +
  labs(title=NULL, subtitle=NULL, x=xlab, y=ylob,color="Winner 's Bonus",caption=NULL) +
  scale_color_manual(breaks=c("Republican","Democratic"),values=rev(colors)) +
  scale_alpha("Relative\nFrequency (%)",range = c(0, 1),limits=c(0,100)) +
  scale_size_manual(NULL,breaks = c("Seed"), values=c(1.5,3.5)) +
  theme

# Scatter Plot By Party
cum_freq.party <- cum_freq
repub <- which(cum_freq.party$mean > 50)
dem <- which(cum_freq.party$mean < 50)
repdem <- which(cum_freq.party$mean == 50)

cum_freq.party <- rbind(cum_freq.party,cum_freq.party[repdem, ])
repdem <- which(cum_freq.party$mean == 50)

cum_freq.party$party <- rep(NA,nrow(cum_freq.party))
cum_freq.party$party[repub] <- "Republican"
cum_freq.party$party[dem] <- "Democratic"
cum_freq.party$party[repdem] <- c(rep("Republican",length(repdem)/2),
                                  rep("Democratic",length(repdem)/2))

cum_freq.party$mean[cum_freq.party$party=="Democratic"] <-
  100 - cum_freq.party$mean[cum_freq.party$party=="Democratic"]

cum_freq.party$dis_won[cum_freq.party$party=="Democratic"] <-
  num_dis - cum_freq.party$dis_won[cum_freq.party$party=="Democratic"]

scat.party <- ggplot(cum_freq.party, aes(x=mean, y=dis_won, color=party,
                                         size=seed.plot, alpha=perc),na.rm=TRUE) +
  geom_point(na.rm=TRUE) +
  geom_segment(aes(x=50,y=0,xend=50,yend=num_dis),color='black',size=.25) +
  geom_segment(aes(x=max(min(cum_freq.party$mean)-5,50),y=num_dis/2,
                   xend=min(max(cum_freq.party$mean)+5,100),yend=num_dis/2),
              color='black',size=.25) +
  labs(title=NULL, subtitle=NULL, x="Statewide Vote (%)", y="Districts Won",
       color="Party",size="") +
  scale_color_manual(breaks=c("Republican","Democratic"),values=rev(colors)) +
  scale_alpha("Relative\nFrequency (%)",range = c(0, 0.5),limits=c(0,100)) +
  scale_size_manual("",breaks = c("Seed"), values=c(2,4)) +
  theme # + theme(plot.title=element_blank(), plot.subtitle=element_blank())

# Plot of step functions

cum_sim2 <- cum_sim[,c("dis_won","mean","est","seed","seed.plot")] %>%
  gather(variable, value, dis_won) %>%
  unite("var", est, variable, sep = "|") %>%
  spread(var, value, sep="|")

names(cum_sim2)[4:9] <- c("dis_won2.5","dis_won25","dis_won75","dis_won97.5",
                        "dis_won_mean","dis_won_seed")

step.func <- function(X,votes=cum_sim2$mean[is.na(cum_sim2$seed)==FALSE],seats) {
  Y <- numeric()
  for (x in X) {
    d <- abs(x-votes)
    ind <- which(d == min(d))
    if (length(ind) == 1 & d[ind[1]] < step/2) {
      Y = c(Y,seats[ind])
    } else {
      Y = c(Y,NA)
    }
  }
  return(Y)
}

xs <- seq(min(cum_sim2$mean,na.rm=TRUE),max(cum_sim2$mean,na.rm=TRUE),

```

```

length.out = 10000)

ymins <- step.func(xs,votes=cum_sim2$mean[cum_sim2$seed==FALSE],
                  seats=cum_sim2$dis_won25[cum_sim2$seed==FALSE])

ymaxs <- step.func(xs,votes=cum_sim2$mean[cum_sim2$seed==FALSE],
                  seats=cum_sim2$dis_won75[cum_sim2$seed==FALSE])

shading <- data.frame(xs,ymins,ymaxs,dis_won_seed=rep(0,10000))

stepchart <- ggplot(cum_sim2, aes(x=mean,y=dis_won_seed)) +
  geom_segment(aes(x=50,y=0,xend=50,yend=num_dis),color=I('black'),size=.25) +
  geom_segment(aes(x=max(min(xs)-5,0),y=num_dis/2,xend=min(max(xs)+5,100),
                  yend=num_dis/2),color=I('black'),size=.25) +

  stat_function(fun=step.func, args=list(votes=cum_sim2$mean[cum_sim2$seed==FALSE],
                                       seats=cum_sim2$dis_won_mean[cum_sim2$seed==FALSE]),
               n=10000, size=1, aes(color='Mean',linetype='Mean'),na.rm=TRUE) +
  stat_function(fun=step.func, args=list(votes=cum_sim2$mean[cum_sim2$seed==FALSE],
                                       seats=cum_sim2$dis_won2.5[cum_sim2$seed==FALSE]),
               n=10000, size=0.75, aes(color='2.5 and 97.5 Percentiles',
                                       linetype='2.5 and 97.5 Percentiles'),na.rm=TRUE) +
  stat_function(fun=step.func, args=list(votes=cum_sim2$mean[cum_sim2$seed==FALSE],
                                       seats=cum_sim2$dis_won97.5[cum_sim2$seed==FALSE]),
               n=10000, size=0.75, aes(color='2.5 and 97.5 Percentiles',
                                       linetype='2.5 and 97.5 Percentiles'),na.rm=TRUE) +

  geom_ribbon(data=shading, aes(x=xs, ymin=ymins, ymax=ymaxs, fill='IQR'), alpha=0.5) +
  geom_point(aes(x=mean, y=dis_won_seed,size=seed.plot),na.rm=TRUE) +

  scale_size_manual(element_blank(), breaks=c("Seed"), values=c(0,3)) +
  scale_color_manual(element_blank(), values=c("darkgoldenrod1","Black")) +
  scale_linetype_manual(element_blank(), values = c("dashed","solid")) +
  scale_fill_manual(element_blank(), values=c("darkgoldenrod1")) +

  labs(title=NULL, subtitle=NULL, x=xlab, y=ylob,color="Winner's Bonus",size="") +
  theme

# Stepchart by party
cum_sim2.party <- cum_sim2
repub <- which(cum_sim2.party$mean > 50)
dem <- which(cum_sim2.party$mean < 50)
repdem <- which(cum_sim2.party$mean == 50)

cum_sim2.party <- rbind(cum_sim2.party,cum_sim2.party[repdem, ])
repdem <- which(cum_sim2.party$mean == 50)

cum_sim2.party$party <- rep(NA,nrow(cum_sim2.party))
cum_sim2.party$party[repub] <- "Republican"
cum_sim2.party$party[dem] <- "Democratic"
cum_sim2.party$party[repdem] <- c(rep("Republican",length(repdem)/2),
                                rep("Democratic",length(repdem)/2))

cum_sim2.party$mean[cum_sim2.party$party == "Democratic"] <-
  100 - cum_sim2.party$mean[cum_sim2.party$party == "Democratic"]
cum_sim2.party[cum_sim2.party$party == "Democratic",4:9] <-
  num_dis - cum_sim2.party[cum_sim2.party$party == "Democratic",4:9]

cum_sim2.party <- cum_sim2.party[order(cum_sim2.party$mean), ]

xs.party <- seq(min(cum_sim2.party$mean,na.rm=TRUE),
               max(cum_sim2.party$mean,na.rm=TRUE),length.out = 10000)

ymins.r <- step.func(xs.party,
                    votes=cum_sim2.party$mean[cum_sim2.party$seed==FALSE &
                                                cum_sim2.party$party == "Republican"],
                    seats=cum_sim2.party$dis_won2.5[cum_sim2.party$seed==FALSE &
                                                      cum_sim2.party$party == "Republican"])

ymaxs.r <- step.func(xs.party,
                    votes=cum_sim2.party$mean[cum_sim2.party$seed==FALSE &
                                                cum_sim2.party$party == "Republican"],
                    seats=cum_sim2.party$dis_won97.5[cum_sim2.party$seed==FALSE &
                                                      cum_sim2.party$party == "Republican"])

```

```

      cum_sim2.party$party == "Republican"])
shading.r <- data.frame(xs.party, ymins.r, ymaxs.r, dis_won_seed=rep(0,10000))

ymins.d <- step.func(xs.party,
  votes=cum_sim2.party$mean[cum_sim2.party$seed==FALSE &
    cum_sim2.party$party == "Democratic"],
  seats=cum_sim2.party$dis_won2.5[cum_sim2.party$seed==FALSE &
    cum_sim2.party$party == "Democratic"])
ymaxs.d <- step.func(xs.party,
  votes=cum_sim2.party$mean[cum_sim2.party$seed==FALSE &
    cum_sim2.party$party == "Democratic"],
  seats=cum_sim2.party$dis_won97.5[cum_sim2.party$seed==FALSE &
    cum_sim2.party$party == "Democratic"])
shading.d <- data.frame(xs.party, ymins.d, ymaxs.d, dis_won_seed=rep(0,10000))

stepchart.party <- ggplot(cum_sim2.party, aes(x=mean, y=dis_won_seed)) +
  geom_segment(aes(x=50, y=0, xend=50, yend=num_dis), color=I('black'), size=.25) +
  geom_segment(aes(x=max(min(xs.party)-5, 50), y=num_dis/2,
    xend=min(max(xs.party)+5, 100),
    yend=num_dis/2), color=I('black'), size=.25) +

  geom_ribbon(data=shading.r, aes(x=xs.party, ymin=ymins.r, ymax=ymaxs.r,
    fill='Republican'), alpha=0.25) +
  geom_ribbon(data=shading.d, aes(x=xs.party, ymin=ymins.d, ymax=ymaxs.d,
    fill='Democratic'), alpha=0.25) +
  geom_point(aes(x=mean, y=dis_won_seed, color=party, size=seed.plot),
    alpha=0.25, na.rm=TRUE) +

  stat_function(fun=step.func, args=list(
    votes=cum_sim2.party$mean[cum_sim2.party$seed==FALSE &
      cum_sim2.party$party == "Republican"],
    seats=cum_sim2.party$dis_won_mean[cum_sim2.party$seed==FALSE &
      cum_sim2.party$party == "Republican"]),
    n=10000, alpha=0.5, size=1.25, aes(color='Republican'), na.rm=TRUE) +

  stat_function(fun=step.func, args=list(
    votes=cum_sim2.party$mean[cum_sim2.party$seed==FALSE &
      cum_sim2.party$party == "Democratic"],
    seats=cum_sim2.party$dis_won_mean[cum_sim2.party$seed==FALSE &
      cum_sim2.party$party == "Democratic"]),
    n=10000, alpha=0.5, size=1.25, aes(color='Democratic'), na.rm=TRUE) +

  scale_size_manual(element_blank(), breaks=c("Seed"), values=c(0,4)) +
  scale_color_manual("Party", values=rev(colors)) +
  scale_fill_manual("Party", values=rev(colors)) +

  labs(title=NULL, subtitle=NULL, x="Statewide Vote (%)",
    y="Districts Won", caption=NULL) +
  theme +
  theme(legend.key.height=unit(1, "line")) +
  theme(legend.key.width=unit(1, "line"))

# alternate step plot

bonus.as.number <- function(dis_won, mean) {
  out <- numeric()
  for (i in 1:length(dis_won)) {
    if (is.na(dis_won[i])==FALSE) {
      if (100*dis_won[i]/num_dis - mean[i] > 0) out <- c(out, 1)
      else if (100*dis_won[i]/num_dis - mean[i] < 0) out <- c(out, -1)
    }
    else out <- c(out, NA)
  }
  return (out)
}

stepchart.alt <- ggplot(cum_sim2, aes(x=mean, y=dis_won_mean,
  color=bonus.as.number(dis_won_seed, mean))) +
  geom_segment(aes(x=50, y=0, xend=50, yend=num_dis), color=I('black'), size=.25) +
  geom_segment(aes(x=20, y=num_dis/2, xend=80, yend=num_dis/2), color=I('black'), size=.25) +
  geom_line(aes(y=dis_won_mean, color=bonus.as.number(dis_won_mean, mean)),
    na.rm=TRUE, size=1.25) +
  geom_line(aes(y=dis_won2.5, color=bonus.as.number(dis_won2.5, mean)), na.rm=TRUE) +

```

```

geom_line(aes(y=dis_won97.5, color=bonus.as.number(dis_won97.5,mean)),na.rm=TRUE) +
geom_ribbon(data=cum_sim2, aes(ymin=pmin(dis_won25,mean/100*num_dis),
                                ymax=pmin(dis_won75,mean/100*num_dis),
                                color=NA, fill=colors[2], alpha="0.35") +
geom_ribbon(data=cum_sim2, aes(ymin=pmax(dis_won25,mean/100*num_dis),
                                ymax=pmax(dis_won75,mean/100*num_dis),
                                color=NA, fill=colors[1], alpha="0.35") +
labs(title=NULL, subtitle=NULL, x=xlab, y=ylob,color="Winner 's Bonus",size="") +
geom_point(aes(x=mean, y=dis_won_seed, size=seed)) +
scale_size_manual(breaks = c("Seed"), values=c(0,5)) +
scale_color_gradient(low=colors[2],high=colors[1]) +
theme

# Box Plots of Simulation

box <- ggplot(cum_sim, aes(y=mean, x=dis_won)) +
  geom_segment(aes(y=50,x=0,yend=50,xend=num_dis),color=I('red'),size=.25) +
  geom_segment(aes(y=0,x=num_dis/2,yend=100,xend=num_dis/2),color=I('red'),size=.25) +
  geom_boxplot(aes(group=dis_won)) + coord_flip() +
  labs(title=NULL, subtitle=NULL,
        y=xlab, x=ylob) + theme

# Histogram of Efficiency Gap at 50% vote
cum_freq$Bonus.party[cum_freq$Bonus == 0] <-
  rep("Neither",length(which(cum_freq$Bonus == 0)))
cum_freq$Bonus.party <- factor(cum_freq$Bonus.party,
                              levels=c("Republican","Democratic","Neither"),ordered=TRUE)
eg.mean <- round(sum(cum_freq$eg[cum_freq$mean==50]*
                    cum_freq$perc[cum_freq$mean==50]/100),2)
cum_freq$eg <- round(cum_freq$eg,2)

freq50 <- ggplot(cum_freq[cum_freq$mean == 50,],
                aes(x=dis_won,y=perc,fill=Bonus.party)) +
  geom_bar(stat='identity') +
  geom_segment(aes(x=num_dis/2,xend=num_dis/2,y=0,
                  yend=min(max(cum_freq[cum_freq$mean == 50,c("perc")])+10,100))) +
  coord_flip() +
  scale_x_continuous(sec.axis=sec_axis(~./num_dis*100-50,
                                       name="Efficiency Gap",
                                       breaks=unique(cum_freq[cum_freq$mean == 50,c("eg")])) +
  scale_fill_manual(breaks=c("Republican","Democratic","Neither"),
                    values=c(colors,"#861e3a")) +
  labs(title=NULL,
        subtitle=NULL,
        x="Districts Won by Republicans",
        y="Relative Frequency (%)",
        fill="Electoral Advantage",
        caption = NULL) +
  theme +
  theme(legend.position='bottom') +
  annotate("text", x=min(cum_freq$dis_won[cum_freq$mean==50])-0.25,
          y=0.85*min(max(cum_freq[cum_freq$mean == 50,c("perc")])+10,100),
          label= paste0("Mean EG: ",as.character(eg.mean)),
          size=3)

# Bernstein Style Shift
points <- data.frame(vote=c(seed_mean),seats=c(length(which(rep_seed >= 50))),
                    seed=c("Seed"))
for (mean in min_mean:max_mean) {
  elec <- rep_seed + (mean-seed_mean)
  v <- mean(elec)
  s <- length(which(elec > 50))
  is.seed <- ""
  point <- data.frame(vote=c(v),seats=c(s),seed=c(is.seed))
  points <- rbind(points, point)
}

bern <- ggplot(points, aes(x=vote, y=seats)) +
  geom_point(alpha=1,aes(size=seed)) +
  geom_segment(aes(x=50,y=0,xend=50,yend=num_dis),color=I('black'),size=.125) +
  geom_segment(aes(x=max(min(cum_freq$mean)-5,0),y=num_dis/2,
                  xend=min(max(cum_freq$mean)+5,100),

```

```

        yend=num_dis/2), color=I('black'), size=.125) +
stat_function(fun=step.func, args=list(
  votes=points$vote[points$seed==""],
  seats=points$seats[points$seed==""]),
  n=10000, size=1, na.rm=TRUE) +
labs(title=NULL, subtitle=NULL,
  x="Statewide Republican Vote (%)",
  y="Number of Districts Won by Republicans", size="") +
scale_size_manual(breaks = c("Seed"), values=c(3.5,0)) + theme

return(list(data=cum_freq, scat=scat, scat.party = scat.party,
  step=stepchart, step.party=stepchart.party, freq50=freq50, bern=bern))
}

#####
# Looking at simulations of states
# --- sim_states
set.seed(34734)
pa <- sim.range(s="Pennsylvania", year.start = 2004, year.end=2010)
wi <- sim.range(s="Wisconsin", year.start = 2012, year.end=2016)
md <- sim.range(s="Maryland", year.start = 2012, year.end=2016)
nc <- sim.range(s="North Carolina", year.start = 2012, year.end=2016)
az <- sim.range(s="Arizona", year.start = 2012, year.end=2016)
oh <- sim.range(s="Ohio", year.start = 2012, year.end=2016)

save(pa, wi, md, nc, az, oh, file="sims.rds")

```

## Evaluation Script

```
#####  
#  
# program: evaluation.r  
# author: Zachary Morgan  
#  
# purpose: To examine the accuracy of the simulation process  
#  
# inputs: analysis dataset created in analysis_data.r program  
#  
# outputs: plots and measures of simulation accuracy  
#  
# run order: 5  
#  
#####  
  
# --- init  
library(ggplot2)  
  
source("simulation.R")  
  
# Importing dataset for analysis  
imported <- read.csv("analysis.csv")  
analysis <- imported[is.na(imported$Rep_shift.adj) == FALSE &  
                      imported$num_dis > 1, ]  
  
# Changing applicable variables to factor type  
analysis$Year <- factor(analysis$Year)  
analysis$Unopposed <- factor(analysis$Unopposed)  
analysis$Unopp.adj <- factor(analysis$Unopp.adj)  
analysis$map <- factor(analysis$map)  
analysis$size <- factor(analysis$size)  
  
#####  
# Theme to be used for plots  
  
theme <- theme(plot.title=element_text(size=12,  
                                         face="bold",  
                                         #family="American Typewriter",  
                                         color="black",  
                                         hjust=0.5,  
                                         lineheight=1.2),  
               plot.subtitle=element_text(size=9,  
                                           #face="bold",  
                                           #family="American Typewriter",  
                                           color="black",  
                                           hjust=0.5,  
                                           lineheight=1.2),  
               axis.title=element_text(size=9),  
               plot.caption=element_text(size=7),  
               legend.title = element_text(size=9, color = "black"),  
               legend.text=element_text(size=7),  
               legend.justification=c(1,0),  
               legend.position='right',  
               legend.background = element_blank(),  
               legend.key = element_blank(),  
               legend.margin=margin(-3,0,-3,0))  
  
colors <- c("#E91D0E", "#232066")  
  
states <- c('Alabama', 'Alaska', 'Arizona', 'Arkansas', 'California', 'Colorado',  
            'Connecticut', 'Delaware', 'Florida', 'Georgia', 'Hawaii', 'Idaho',  
            'Illinois', 'Indiana', 'Iowa', 'Kansas', 'Kentucky', 'Maine',  
            'Maryland', 'Massachusetts', 'Michigan', 'Minnesota', 'Mississippi',  
            'Missouri', 'Montana', 'Nebraska', 'Nevada', 'New Hampshire',  
            'New Jersey', 'New Mexico', 'New York', 'North Carolina',  
            'North Dakota', 'Ohio', 'Oklahoma', 'Oregon', 'Pennsylvania',  
            'Rhode Island', 'South Carolina', 'South Dakota', 'Tennessee',  
            'Texas', 'Utah', 'Vermont', 'Virginia', 'Washington', 'West Virginia',  
            'Wisconsin', 'Wyoming')
```

```

#####
# Determining number of complete seed years
# ---- completeseed

com <- 0
incom <- 0
for (s in states) {
  for (y in seq(from=2004,to=2016,by=2)) {
    if (length(which(imported$incl[imported$State == s &
                          imported$Year == y]==FALSE)) == 0) {
      com <- com + 1
    } else {
      incom <- incom + 1
    }
  }
}

#####
# Cross Validation
# ---- crossval
set.seed(514568)
eval <- data.frame(State = character(), Year=numeric(),
                   in95 = logical(), in50 = logical())

for (s in states) {
  for (y in seq(from=2004,to=2016,by=2)) {
    print(paste(s,y))
    res <- sim(s=s,year.start=y,year.end=y,cross.validate = TRUE,
              invert.seed=TRUE,nw.distrib = FALSE,fill.incom = TRUE)
    print(res)
    if (! is.null(res)) {
      data <- rep(res$dis_won[res$seed == FALSE],
                 res$freq[res$seed == FALSE])
      pct <- mean(data < res$dis_won[res$seed])
      pct.high <- mean(data <= res$dis_won[res$seed])
      pct.mid <- (pct + pct.high) / 2
      dis_won <- res$dis_won[res$seed==TRUE]
      num_dis <- imported$num_dis[imported$State==s &
                                  imported$Year == y][1]
      prob <- res$perc[res$seed==FALSE & res$dis_won==dis_won]
      if (length(prob)==0) {prob=0}
      q <- quantile(data, probs=c(0.025,0.25,0.75,0.975))
      q <- round(q)
      e <- data.frame(State=s,
                     Year=y,
                     dis_won = dis_won,
                     dis_won.pct = dis_won/num_dis,
                     num_dis = num_dis,
                     pct=pct,
                     pct.mid=pct.mid,
                     pct.high=pct.high,
                     prob=prob,
                     in95 = c(q[1] <= res$dis_won[res$seed] &
                               res$dis_won[res$seed] <= q[4]),
                     in50 = c(q[2] <= res$dis_won[res$seed] &
                               res$dis_won[res$seed] <= q[3]))
      eval <- rbind(eval,e)
    }
  }
}

(in95 <- length(eval$in95[eval$in95]) / nrow(eval))
(in50 <- length(eval$in50[eval$in50]) / nrow(eval))

#####
# Comparison of statewide versus nationwide residual usage
# ---- sunw

sw.vs.nw <- function(s,year.start,year.end,n=2000,step=1,fill.incom=TRUE) {
  # s <- State
  # year.start <- First year to use in seed
  # year.end <- last year to use in seed

```



```

# n <- Number of Simulated Datapoints per step
# step <- Difference between simulated shifts

# Creating sequence of years to be used
y <- seq(from=year.start,to=year.end,by=2)

# Taking relevant subset of data
to.use <- imported[imported$State==s & imported$Year %in% y, ]
if (is.null(nrow(to.use))) {
  print("No_included_records_for_desired_year")
  return(NULL)
}

# Ensuring all selected years come from the same congressional map
if (length(unique(to.use$map)) > 1) {
  print("Years_come_from_different_congressional_maps.")
  return(NULL)
}

# Detective incomplete seeds if necessary
if (fill.incom==FALSE & length(which(to.use$incl==FALSE)) > 0) {
  print("Seed_incomplete.")
  return(NULL)
}

# Standardizing for year to year shifts by expressing each
# percentage in year.start terms
shift.from.year1 <- 0
if (length(y) > 1) {
  for (i in 2:length(y)) {
    shift.from.year1 <- shift.from.year1+
      to.use$Rep_shift_sw[to.use$Year == y[i]][1]
    to.use$Rep_perc.adj[to.use$Year == y[i]] <-
      to.use$Rep_perc.adj[to.use$Year == y[i]] - shift.from.year1
    to.use$Dem_perc.adj[to.use$Year == y[i]] <-
      100 - to.use$Rep_perc.adj[to.use$Year == y[i]]
  }
}

# Deriving seed values if possible
grouped <- group_by(to.use[to.use$incl == TRUE, ], District)
rep_seed <- unlist(summarize(grouped,
  val=mean(Rep_perc.adj,na.rm=TRUE))$val)
seed_mean <- mean(rep_seed)
num_dis = to.use$num_dis[1]
if (length(rep_seed) < num_dis) {
  print("Unable_to_complete_seed.")
  return(NULL)
}

# Determining minimum and maximum shifts.
# Ensuring no more than 5% of resid dist would lead to impossible percentages
q <- quantile(analysis$Rep_shift.resid,probs=c(.05,.95))

min_shift <- - q[1] - min(rep_seed)
max_shift <- 100 - q[2] - max(rep_seed)

# Creating sequence of shifts
state_shifts <- seq(ceiling(min_shift), floor(max_shift), by=step)
state_means <- state_shifts + seed_mean
steps <- length(state_shifts)

# Simulating shifts using statewide distribution
cum_freq.sw <- sim(s=s,year.start = year.start, year.end=year.end,
  means=state_means, fill.incom=fill.incom, nw.distrib = FALSE, n=n)
if (is.null(cum_freq.sw)) return(NULL)
cum_sim.sw <- cbind(cum_freq.sw[cum_freq.sw$seed == TRUE,
  c("dis_won","mean","seed")],est=c(NA))
for (m in unique(unlist(cum_freq.sw$mean))) {
  data <- rep(cum_freq.sw$dis_won[cum_freq.sw$mean == m &
    cum_freq.sw$seed==FALSE],
    cum_freq.sw$freq[cum_freq.sw$mean == m & cum_freq.sw$seed==FALSE])
  q <- quantile(data, probs=c(0.025,0.25,0.75,0.975))
}

```

```

sim_means <- data.frame(dis_won=c(round(mean(data))),mean=c(m),
                        seed=c(FALSE),est=c("mean"))
sim_025 <- data.frame(dis_won=round(q[1]),mean=c(m),seed=c(FALSE),est=c(2.5))
sim_25 <- data.frame(dis_won=round(q[2]),mean=c(m),seed=c(FALSE),est=c(25))
sim_75 <- data.frame(dis_won=round(q[3]),mean=c(m),seed=c(FALSE),est=c(75))
sim_975 <- data.frame(dis_won=round(q[4]),mean=c(m),seed=c(FALSE),est=c(97.5))
cum_sim.sw <- rbind(cum_sim.sw, sim_means)
cum_sim.sw <- rbind(cum_sim.sw, sim_025)
cum_sim.sw <- rbind(cum_sim.sw, sim_25)
cum_sim.sw <- rbind(cum_sim.sw, sim_75)
cum_sim.sw <- rbind(cum_sim.sw, sim_975)
}
cum_sim.sw$Bonus <- (100*cum_sim.sw$dis_won / num_dis) - cum_sim.sw$mean
cum_sim.sw$seed.plot <- rep("",nrow(cum_sim.sw))
cum_sim.sw$seed.plot[cum_sim.sw$seed] <- "Seed"

cum_sim.sw$Bonus.party <- rep(NA, nrow(cum_sim.sw))
adv <- which(cum_sim.sw$Bonus > 0)
cum_sim.sw$Bonus.party[adv] <- rep("Republican",length(adv))
disadv <- which(cum_sim.sw$Bonus < 0)
cum_sim.sw$Bonus.party[disadv] <- rep("Democratic",length(disadv))

cum_sim.sw$distrib <- rep("State",nrow(cum_sim.sw))

# Simulating shifts using nationwide distribution
cum_freq.nw <- sim(s=s,year.start = year.start, year.end=year.end,
                 means=state_means, fill.incom=fill.incom, nw.distrib = TRUE, n=n)
cum_sim.nw <- cbind(cum_freq.nw[cum_freq.nw$seed == TRUE,
                              c("dis_won","mean","seed")],est=c(NA))
for (m in unique(unlist(cum_freq.nw$mean))) {
  data <- rep(cum_freq.nw$dis_won[cum_freq.nw$mean == m &
                                cum_freq.nw$seed==FALSE], cum_freq.nw$freq[cum_freq.nw$mean == m &
                                cum_freq.nw$seed==FALSE])

  q <- quantile(data, probs=c(0.025,0.25,0.75,0.975))
  sim_means <- data.frame(dis_won=c(round(mean(data))),mean=c(m),
                          seed=c(FALSE),est=c("mean"))
  sim_025 <- data.frame(dis_won=round(q[1]),mean=c(m),seed=c(FALSE),est=c(2.5))
  sim_25 <- data.frame(dis_won=round(q[2]),mean=c(m),seed=c(FALSE),est=c(25))
  sim_75 <- data.frame(dis_won=round(q[3]),mean=c(m),seed=c(FALSE),est=c(75))
  sim_975 <- data.frame(dis_won=round(q[4]),mean=c(m),seed=c(FALSE),est=c(97.5))
  cum_sim.nw <- rbind(cum_sim.nw, sim_means)
  cum_sim.nw <- rbind(cum_sim.nw, sim_025)
  cum_sim.nw <- rbind(cum_sim.nw, sim_25)
  cum_sim.nw <- rbind(cum_sim.nw, sim_75)
  cum_sim.nw <- rbind(cum_sim.nw, sim_975)
}
cum_sim.nw$Bonus <- (100*cum_sim.nw$dis_won / num_dis) - cum_sim.nw$mean
cum_sim.nw$seed.plot <- rep("",nrow(cum_sim.nw))
cum_sim.nw$seed.plot[cum_sim.nw$seed] <- "Seed"

cum_sim.nw$Bonus.party <- rep(NA, nrow(cum_sim.nw))
adv <- which(cum_sim.nw$Bonus > 0)
cum_sim.nw$Bonus.party[adv] <- rep("Republican",length(adv))
disadv <- which(cum_sim.nw$Bonus < 0)
cum_sim.nw$Bonus.party[disadv] <- rep("Democratic",length(disadv))

cum_sim.nw$distrib <- rep("Nation",nrow(cum_sim.nw))

cum_sim <- rbind(cum_sim.nw,cum_sim.sw)

# Labels for plots
t <- "Statewide_Vote_and_Seats_Won"
if (length(y) == 1) {
  st <- paste("Simulations_of_Elections_in",s,"Using",y,
             "as_the_Seed.(n=",n,"/step)",sep="")
} else if (length(y) > 1) {
  st <- paste("Simulations_of_Elections_in",s,"Using_Years",
             paste0(year.start,"-",year.end),"as_Seed.(n=",n,"per_step)",sep="")
}
xlab <- "Statewide_Republican_Vote_%"
ylab <- "Number_of_Districts_Won_by_Republicans"

```

```

# Stepchart by distrib
cum_sim2 <- cum_sim[,c("dis_won","mean","est","distrib","seed.plot","seed")] %>%
  gather(variable, value, dis_won) %>%
  unite("var", est, variable, sep = "|") %>%
  spread(var, value, sep="|")

names(cum_sim2)[5:10] <- c("dis_won2.5","dis_won25","dis_won75","dis_won97.5",
  "dis_won_mean","dis_won_seed")

step <- function(X,votes=cum_sim2$mean[is.na(cum_sim2$seed)==FALSE],seats) {
  Y <- numeric()
  for (x in X) {
    for (i in 1:(length(votes)-1)) {
      if (x >= votes[i] & x < votes[i+1]) {
        Y <- c(Y,seats[i])
        break
      }
      else if (i==length(votes)-1) {
        Y <- c(Y,NA)
      }
    }
  }
  return(Y)
}

xs <- seq(min(cum_sim2$mean,na.rm=TRUE),
  max(cum_sim2$mean,na.rm=TRUE),length.out = 10000)

ymins.sw <- step(xs,
  votes=cum_sim2$mean[cum_sim2$seed==FALSE & cum_sim2$distrib == "State"],
  seats=cum_sim2$dis_won2.5[cum_sim2$seed==FALSE & cum_sim2$distrib == "State"])
ymaxs.sw <- step(xs,
  votes=cum_sim2$mean[cum_sim2$seed==FALSE & cum_sim2$distrib == "State"],
  seats=cum_sim2$dis_won97.5[cum_sim2$seed==FALSE & cum_sim2$distrib == "State"])
shading.sw <- data.frame(xs,ymins.sw,ymaxs.sw,dis_won_seed=rep(0,10000))

ymins.nw <- step(xs,
  votes=cum_sim2$mean[cum_sim2$seed==FALSE & cum_sim2$distrib == "Nation"],
  seats=cum_sim2$dis_won2.5[cum_sim2$seed==FALSE & cum_sim2$distrib == "Nation"])
ymaxs.nw <- step(xs,
  votes=cum_sim2$mean[cum_sim2$seed==FALSE & cum_sim2$distrib == "Nation"],
  seats=cum_sim2$dis_won97.5[cum_sim2$seed==FALSE & cum_sim2$distrib == "Nation"])
shading.nw <- data.frame(xs,ymins.nw,ymaxs.nw,dis_won_seed=rep(0,10000))

stepchart <- ggplot(cum_sim2, aes(x=mean,y=dis_won_seed)) +
  geom_segment(aes(x=50,y=0,xend=50,yend=num_dis),color=I('black'),size=.25) +
  geom_segment(aes(x=max(min(xs)-5,0),y=num_dis/2,xend=min(max(xs)+5,100),yend=num_dis/2),
  color=I('black'),size=.25) +

  geom_ribbon(data=shading.sw, aes(x=xs, ymin=ymins.sw, ymax=ymaxs.sw, fill='State'),
  alpha=0.25, na.rm=TRUE) +
  geom_ribbon(data=shading.nw, aes(x=xs, ymin=ymins.nw, ymax=ymaxs.nw, fill='Nation'),
  alpha=0.25, na.rm=TRUE) +
  geom_point(aes(x=mean, y=dis_won_seed, color=distrib, size=seed.plot),
  alpha=0.5, na.rm=TRUE) +

  stat_function(fun=step,args=list(
  votes=cum_sim2$mean[cum_sim2$seed==FALSE & cum_sim2$distrib == "State"],
  seats=cum_sim2$dis_won_mean[cum_sim2$seed==FALSE & cum_sim2$distrib == "State"]),
  n=10000, alpha=0.5, size=1, aes(color='State'), na.rm=TRUE) +

  stat_function(fun=step,args=list(
  votes=cum_sim2$mean[cum_sim2$seed==FALSE & cum_sim2$distrib == "Nation"],
  seats=cum_sim2$dis_won_mean[cum_sim2$seed==FALSE & cum_sim2$distrib == "Nation"]),
  n=10000, alpha=0.5, size=1, aes(color='Nation'), na.rm=TRUE) +

  scale_size_manual(element_blank(), breaks=c("Seed"), values=c(0,3)) +
  scale_color_manual("Residual_Distribution", values=c("Darkgoldenrod1","Cyan")) +
  scale_fill_manual("Residual_Distribution", values=c("Darkgoldenrod1","Cyan")) +

  labs(title=NULL, subtitle=NULL, x=xlab, y=ylob,caption=NULL) +
  theme + theme(legend.position = c(1, 0)) +

```

```

    theme(legend.key.height=unit(1,"line")) +
    theme(legend.key.width=unit(1,"line")) +
    theme(legend.margin=margin(5,5,5,5))

    mean.diff <- length(which(! cum_sim2$dis_won_mean[cum_sim2$distrib=="Nation"]==
                                cum_sim2$dis_won_mean[cum_sim2$distrib=="State"]))
    mean.diff.perc <- 100 * mean.diff / length(state_means)
    return(list(plot=stepchart,diff=mean.diff,diff.perc=mean.diff.perc))
}

sw.vs.nw.df <- function() {
  num.resids <- data.frame(state=character(),n=numeric())
  for (s in states) {
    n.resids <- nrow(analysis[is.na(analysis$Rep_shift.resid)==FALSE &
                            analysis$State == s, ])
    num.resids <- rbind(num.resids, data.frame(state=c(s),n=c(n.resids)))
  }

  sw.vs.nw.res <- data.frame(state=character(),year=numeric(),
                             diff=numeric(),diff.perc=numeric())
  for (s in states) {
    for (y in seq(2004,2016,by=2)) {
      out <- sw.vs.nw(s=s,y,y)
      if (! is.null(out)) {
        o <- data.frame(state=c(s),year=c(y),diff=c(out$diff),
                        diff.perc=c(out$diff.perc))
        sw.vs.nw.res <- rbind(sw.vs.nw.res,o)
      }
    }
  }
  sw.vs.nw.res <- merge(sw.vs.nw.res,num.resids,by=c("state"))

  names(spread)[1:2] <- c("state","year")
  sw.vs.nw.res <- merge(sw.vs.nw.res,spread,by=c("state","year"))

  write.csv(sw.vs.nw.res,"sw.nw.csv")
}

# ---- sunwscat
sw.vs.nw.res <- read.csv("sw.nw.csv")
ggplot(sw.vs.nw.res, aes(x=var,y=diff.perc)) +
  geom_point() +
  labs(
    title=NULL,
    x="Variance",
    y="Mean Disagreement (%)"
  ) +
  theme

# ---- sunwez
out <- sw.vs.nw(s="Nebraska",2006,2006)
out$plot

```