*Hoa Tran*

*Project 1: Polarized politics*

*03/07/2022*

**I. Introduction**

Polarization in politics has always been a topic of much analytical research, especially in America where the central government, with its two-party system. This is because the two parties must work together cooperatively to implement a policy or pass a law. Therefore, the goal of this project is to use data from the U.S House of Representatives to analyze the voting behavior and polarization of the U.S House chamber over time. With this knowledge in mind, the results of this analysis will be of immense importance regarding helping the general public gain a more thorough understanding of the subject matter. Moreover, other skate holders such as politicians, political scientists and so on could also be interested in the resulting analysis. Because of that, one cannot overlook the sheer importance of this project.

**II. Method explanation:**

**A/ Explaining the algorithm:**

In order to complete the first part of the project, we divided our algorithm into three small components:

1. Determine whether a House roll call vote number from a given year is a party line vote:

* To do this, we must create an input so that people can access a roll call vote number for a specific year of their choice by manipulating the url for the website. Then, we search for specific strings containing the information regarding the voting information of Democrats and Republicans. To be more specific, we calculate the number of Democrats and Republicans who voted yes and no excluding votes that are recorded in other ways.

1. Calculate the fraction of party line votes:

* In order to accomplish this task, we use the result of the step number but incorporate it in an iterations of loops in the range of the number of roll call votes for of the given year to count the number of party line votes. To make things simple, we assume the maximum number of roll calls for each year is 450. Then, we divide the number of party line votes with the 450 to get the fraction of votes.

1. Plotting the fraction of party line votes over time.

* We use the results of step two to plot the results over the span of 20 years by iterating each year of the 20 years in a loop. For each iteration, we use the results of the previous step until the loop ends. The final result should be a complete plot of the fraction of party line votes over time with the fraction on the y-axis and the years on the x-axis.

**B/ Algorithm implementations:**

Our program has 4 functions: main(), partyLine(), countPartyLine(), and plotPartyLine().

The **main () function** is where the program begins. We use it to call the partyLine function and the countPartyLine function with the input for a specific year and roll call number.

The **partyLine()** functions take year and number as its parameters, both of which are integer values. This function is to find the party line votes in the House chamber and return whether or not a roll call number has party line votes (returns True) and if otherwise returns False. To do this, we must first initialize the URL for the website and manipulate it by combining it with the year and number variable so that we could access the desirable information. We also initialize and set the value of the necessary variables as 0 to calculate the number of Democrats and Republicans who voted and count the number of lines in the XML files. Then, we use a for loop to encode the XML files so that Python could understand it and count the number of lines. Next, we use another for loop to go through each line to find the information of voters in the Democratic and Republican Party who voted yes (Yea or Aye) and no (Nay and No) by finding the specific strings containing these information. When the loop ends, we determine whether or not each party has more than half its members voting in different ways. If so, we return True ( is party line vote) otherwise, we return False (not party line vote).

The **countPartyLine()** function is used to calculate the fraction of party line votes. To do this, we initialize the number of party line votes as zero. Then, we use a for loop in the range of the given year’s roll call vote numbers and call the result of the function partyLine. In each iteration, if the results of the previous function is true, we increment the number of party line votes by 1. When the loop ends, we divide the number of votes by the maximum number of votes for that given year.

The **plotPartyLine** function takes no parameters and is only used to plot the fraction of party line votes. In order to do this, we use the results of the countPartyLine function in a for loop in the range of 20 years ( we decide to choose from 1999 to 2019). For each iteration, we call the results of the previous function and append to the list as well as append the value of the year to a list. When the loop ends, we plot both of these values on the same graph.

**III. Results and analysis:**

**1. Investigate the Affordable Care Act and Drywall Safety Act.**

The vote on the Affordable Care act was a party line vote as in 2010, the year it was passed. We searched on the website provided in the project and tested all of the roll call numbers that were associated with this act and all of them proved that this was a party line vote. (111th Congress roll call number: 165,163,162,160,159)

Another vote that I care about is the Drywall Safety Act. It was a party line vote based on the result of our algorithm and also based on the result of the website. (112th congress roll call number 657)

**2. Fraction of party line vote over time (1999-2020).**



According to this graph, from 1999 to 2019, all of the years have more than 50% of their votes as party line votes. This means that the fraction of votes from all of the years all went along the party line vote.

The plot shows that all of the years surveyed all followed the party line vote but with varying degree of differences. That is to say that the curve showing the fraction of party line votes fluctuates throughout the years, with the lowest fraction being around 60% in 2011 to a peak of around 90% in 2019.

At the beginning, we can see that the fraction of party line vote was around 0.7 and then it plummeted to 0.65 6 years later. After some fluctuations, the figure for this dropped to around 0.6 in 2011. However, from 2012 and onward the overall trend was that the fraction of party line votes was increasing at a steady rate with a few small declines in around 2016 and 2018.

🡺 All of this information proves that American politics have always been polarized with many of its members possessing different political ideology.

**3. Further investigation of voting behavior:**

Link to article: <https://www.elonnewsnetwork.com/article/2022/03/cheat-sheet-political-polarization-elon-university-journalism-professor>

This article discussed how polarization in recent years has increased with the author saying: “The overall share of Americans who express consistently conservative or liberal opinions has doubled over the past two decades.” And based on the resulting of the graph, we can see that from around 2013 to 2019, the general trend for fraction of party line votes increased by a large amount. This proves that the results of the news report are in accordance with the results of our analysis. The motivation of this news article is to discuss how and why polarization has been increasing. With that in mind, the similar results between the article and the plot shown in our analysis indicate that the general trend in American politics is that polarization has become more prevalent.

. **IV. Conclusion:**

This project has helped us gain a deeper understanding of American politics and how polarized it is. With this knowledge in mind, we now realize the importance of investigating data and incorporating computation problem solving in data analysis. Such a large amount of data needs to be polished so that we only need to focus on the specific to formulate the correct conclusions and inferences. That is what this project has guided us to do. Moreover, we also learn that we must be thorough when coming up with the algorithm and implementing it into Python as the code for certain parts of the project took quite some time to finish. This is our biggest challenge so far in terms of solving the problems at hand and creating an efficient algorithm to the best of our understanding.