

George Mason University

Redesign Project

2016 US Election Map

Mitchell Breeden

STAT-515-009

Dr. Niloofar Ramezani

November 20, 2022

Original

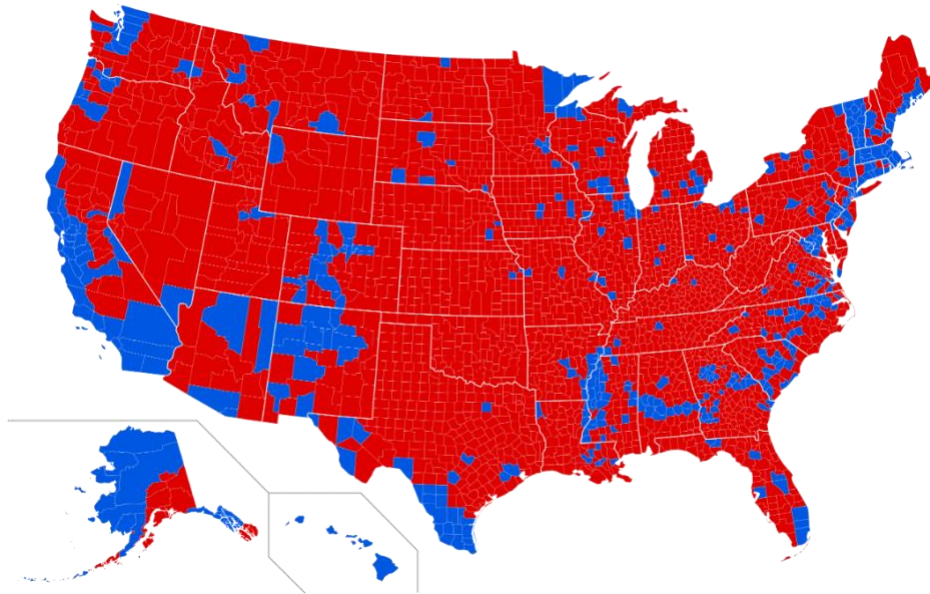


Figure 1: Original Election Map

For my redesign project, I chose to analyze a map of the United States which shows the 2016 election results by county (Figure 1). This map was created by Ali Zifan and shared on Wikimedia Commons. Since then, it has made the rounds on social media and respected institutions, such as Yale University, have used it. This graph does its job well: It clearly shows which presidential candidate won each county in the 2016 election. However, when I first saw this map, I believed it to be misleading for one main reason: land doesn't vote, people do. At first glance, it appears that most of the country is red, implying that the Republican candidate for president had a large margin of victory. This is untrue though because the United States does not use the popular vote to elect its presidents; rather, it uses the electoral college. "The number of electors each state gets is equal to its total number of Senators and Representatives in Congress. A total of 538 electors form the Electoral College. Each elector casts one vote following the general election. The candidate who gets 270 votes or more wins." (USAGov). With this as

background, as well as the basic design principles that the map is missing (any form of context), I decided to redesign it into a format that more accurately captures the electoral college results.

Redesign

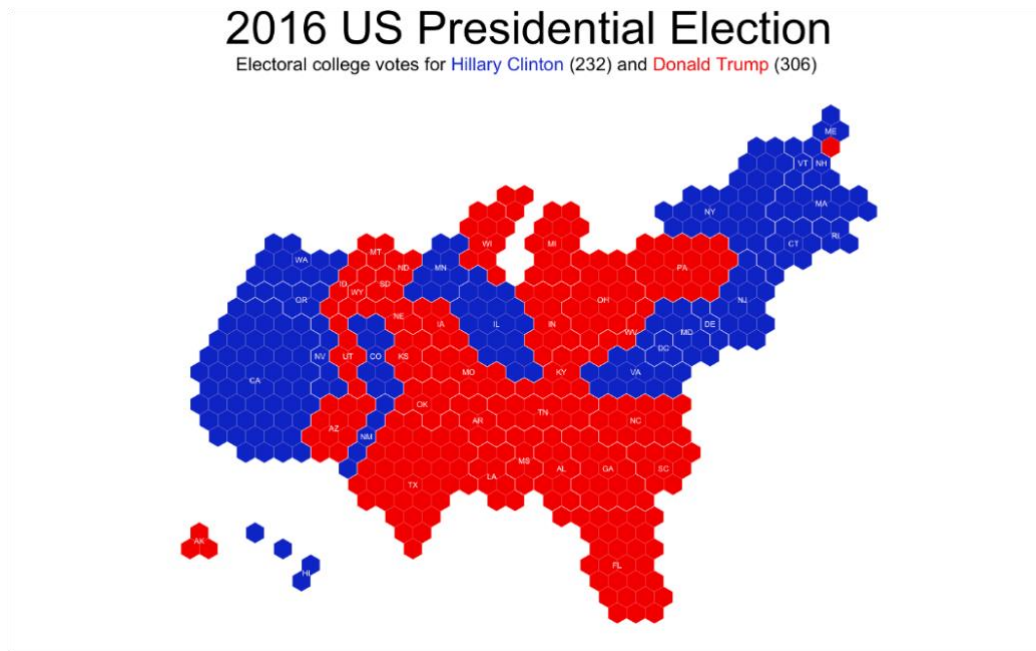


Figure 2: Redesigned Election Map

My redesign puts the focus on the electoral college. In the title section, I highlight how many electoral college votes each candidate received and their name is colored according to their party. In the map itself, the number of hexagons that comprises each state equals the number of each state's electoral votes. This allows a viewer to quickly see how much of an impact each state has towards a candidate reaching 270 electoral votes. Furthermore, I added state labels and colors to help the viewer make associations with the data. After analyzing my redesign, it is clear the Republican candidate for president won, but not by the margin portrayed in the original map. This is largely due to population centers like Chicago, Illinois, which caused the state to go blue.

2016 US Presidential Election

Maine - 4 Electoral Votes

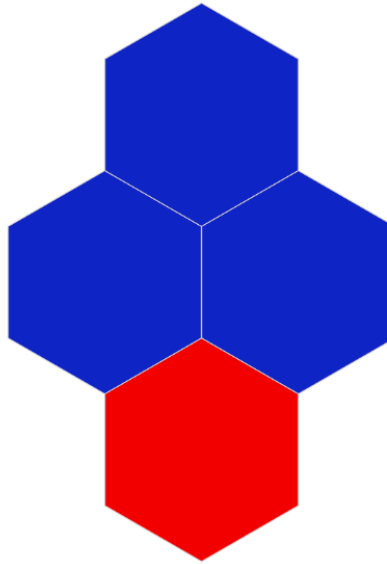


Figure 3: Enlarged State

In addition to the full-sized map, I also created a function that allows the user to enter a state's abbreviation and see an enlarged version of it. Figure 3 demonstrates this feature: Maine has faithless electors who can have their congressional districts retain an electoral vote that differs from the rest of the state. The function allows the user to see the breakout by depicting the red hexagon along with the majority blue hexagons. The user can also see the state's full name along with the number of electoral votes that have been assigned to that state.

Software

This project was created using R in RStudio along with a multitude of libraries. Namely, `tilegramsR` and `ggplot2`. `TilegramsR` is a library created by Bhaskar Karambelkar which provides the spatial data necessary to create the map. With this library, I was able to plot the hexagons using `ggplot2` and further manipulate them to gain the desired appearance. Since the original map

used county level data, I had to create my own mapping file with a list of states and then which party those states voted for as an input for the script. Learning how to deal with spatial data was a bit of a struggle for me. For example, approaching the labeling of the states required identifying the state object rather than the individual hexagons that illustrate each electoral vote. I also struggled with creating borders that were clear and decisive; however, I was able to overcome this challenge by using hex codes to identify the correct color.

Concluding Remarks

My redesign focuses on what makes or breaks presidential elections, the electoral college. The original map put the focus on the micro level of the outcomes from county elections. The original map is not necessarily wrong in displaying this detail; however, given there is no context related to this information, the user is left to make their own assumptions which could lead to an inaccurate understanding of the results and lead to misinformation.

If I were to continue working on this project, there are a few things I would like to add. My job requires me to work with Tableau. This program has a feature where you can hover over a data point (e.g., state) and display a tooltip with more detailed information about this selection. I would like to place my enlarged state view from the function I created into this tooltip to allow a user to quickly get more information about each state. Additionally, I would like to have the color of each state be proportional to the percent of vote that the winning candidate received. Currently, each state is depicted in a solid red or blue color regardless of the candidate's margin of victory; with Tableau, this could be improved by displaying varying shades of red and blue to depict the final vote. I enjoyed this project and learned a lot about plotting information in R as well as manipulating data within RStudio. With these skills and the ones gained in the remainder of the course, I believe I will be able to better spot bad designs and work to correct the flaws.

Sources

Cummings, M. (2021, April 21). Swing vote “trumped” turnout in 2016 election. YaleNews.
<https://news.yale.edu/2021/04/21/swing-vote-trumped-turnout-2016-election>

USA Gov. (2016). Presidential Election Process | USAGov. Usa.gov.
<https://www.usa.gov/election>

RStudio Sources

Arnold J (2021). `_ggthemes`: Extra Themes, Scales and Geoms for 'ggplot2'_. R package version 4.2.4, <<https://CRAN.R-project.org/package=ggthemes>>.

H. Wickham. `ggplot2`: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

Karambelkar B (2017). `_tilegramsR`: R Spatial Data for Tilegrams_. R package version 0.2.0, <<https://CRAN.R-project.org/package=tilegramsR>>.

Wickham H (2022). `_stringr`: Simple, Consistent Wrappers for Common String Operations_. R package version 1.4.1, <<https://CRAN.R-project.org/package=stringr>>.

Wickham H, François R, Henry L, Müller K (2022). `_dplyr`: A Grammar of Data Manipulation_. R package version 1.0.10, <<https://CRAN.R-project.org/package=dplyr>>.

Wilke C, Wiernik B (2022). `_ggtext`: Improved Text Rendering Support for 'ggplot2'_. R package version 0.1.2, <<https://CRAN.R-project.org/package=ggtext>>.

R version 4.2.1 (2022-06-23)