**Using Logistic Regression to Predict the 2020 Maricopa County Election Results by Geometric Coordinates**

Jessica Recker

B.S. Database Management and Data Analytics, Western Governor’s University

D195: DMDA Capstone

January 7th, 2024

## A. Create a proposal.

***Research Question:*** With this project we are attempting to understand to what extent election results in Maricopa County, AZ can be effectively analyzed and predicted using geospatial analysis and Logistic Regression, with the aim of providing insights into electoral trends and potential areas for strategic political outreach.

***Context:*** Maricopa County is divided along many different districts, including Congressional Districts, Legislative Districts, and Voting Precincts, which are the smallest political divisions. Political outreach is often structured around these geographic divisions, with political organizations and parties distributing resources across all divisions to cover as much of the county as possible.

The Arizona Secretary of State’s website publishes detailed election results by voting precinct for every election, going back many years. The results are published via XML and csv files. While these files are detailed, each voting precinct is re-mapped every 10 years by the Arizona Independent Redistricting Commission, in response to Federal Census results. This makes comparing historical election results extremely difficult, as the district and precinct boundaries change.

The Maricopa County Recorder’s office publishes shapefiles of each voting precinct, going back several years. These shapefiles specify longitudinal and latitudinal coordinates for each voting precinct’s outer boundaries. These geocoordinates can be used to map out corresponding interior coordinates within every precinct. By generating these interior coordinates, we would effectively create a static mesh of geometric points for the entire county, which can then be compared to historical precinct boundaries. This will tell us what precinct each point was a part of for each election year, allowing us to reference the historical election results for each of these coordinates.

After we have our historical trends defined, we can then train a Logistic Regression model to predict the 2020 results for each point in our mesh, using the prior elections as the training data. We can compare this to the actual 2020 election results to see how accurately our model performs.

If this project is successful, this would be significant for any political outreach efforts within the county, as the predicted results could be mapped and used to determine areas more likely to “flip” (change from having more Republican votes to more Democratic votes or vice versa.) It is these specific areas, not the areas in the county that are politically stable, that could benefit the most from both further analysis and heightened political outreach.

The 2020 Presidential Election was a highly controversial election. This project does not attempt to make any political or partisan declarations based on its results.

***Published References:***

Samuel Baltz (Baltz et al., 2022) studied and collected precinct-level election results across all 50 states. During this analysis, the researchers documented how important such granular views of election results can be. *“Granular vote count data are required for many important questions. Local election results are widely used in quantitative political science and one classic application is the study of legislative districts and gerrymandering reform. These data also have applications across many empirical sciences. Prominent applications of precinct- or county-level election results include modeling public health outcomes, particularly related to COVID-19; local-level analyses of municipal spending, policing and crime reporting, the effectiveness of public communication, and the usage or regulation of land, water, and energy; estimating neighbourhood-level demographics; modeling small-scale labour markets or the effects of macro-economic events; and even demonstrating how a new method in statistics or data science can be applied to important questions.”* This report effectively demonstrates the need for election result analysis in geographic detail more defined than county or state level reporting, which our project attempts to accomplish for Maricopa County.

Our project will utilize geospatial analysis to organize and analysis our data in detail. This geospatial analysis is crucial, especially considering the shifting boundaries of voting districts and precincts, as it will enable us to achieve a more defined geographic view of our election results. As the University of Southern California reports in their Spatial Sciences Institute’s blog (University of Southern California, 2021): *“Every 10 years, the 435 seats in the U.S. House of Representatives are apportioned across the 50 states based on population data collected by the Census Bureau. In turn, the states set district boundaries that determine what races the people who live in each voting precinct can participate in. Voting authorities must comply with an array of regulations and account for demographic shifts when making these adjustments. GIS simplifies the redistricting process by offering a clear view of all possible variations in the electoral map.”* This “clear view” is exactly what we are looking for in our analysis.

We want our results to be accurate and also easy enough for end users to interpret. As LinkedIn explains in their article “What are the advantages of using logistic regression for classification?” *(LinkedIn, 2023): “Logistic regression has several advantages over other classification algorithms, such as its ease of implementation and interpretation. You only need to specify the input and output variables, and the algorithm will do the rest. The parameters are also easy to understand, and how they affect the probability of each class is clear.”* As this article describes, Logistic Regression is a good choice for our use case, given the ease of use and interpretation.

***Data Analytics Solution:*** Our solution will be to utilize the power of the Python programming language to implement a programmatic analysis and prediction of election results, based on geospatial data. Our solution involves these key steps:

* Extract and format XML and CSV of election results
* Extract and format GeoJson files of precinct boundaries
* Determine (or generate) static geometric coordinates to reference to historical voting data
* Use a machine learning model to predict the 2020 election results
* Perform a statistical analysis to test how accurate the predictions were
* Visualize the results

***Benefit:***

* Successful prediction of election results would aid political outreach efforts within Maricopa County. Predicted results can be mapped to identify areas more likely to vote a certain way (“Republican” or “Democratic”), which enables political organizations and campaigns to target key areas more effectively.
* These predicted election results will also enable the identification of outliers or areas where the model does not accurately predict the results. These areas in particular warrant greater analysis and, potentially, political outreach to connect with “swing” voters.
* This project does all of the above in a programmatic and reliable way, removing the need for manual data processing and removing the obstacle of changing precinct boundaries.

## B. Describe your data analytics project plan.

***Goals, objectives, and deliverables***: The goal is to predict the 2020 election results by geometric point. We will measure success for this goal by comparing the proportion of accurately predicted geometric points to random chance. With an alpha of 0.05, this will tell us if the model performs better than chance in a statistically significant manner.

Our objectives will be:

* Extract and wrangle (clean/standardize) all necessary datasets, using the Python programming language.
* Create static geometric points within Maricopa County
* Merge these geometric points with their historical voting results (as defined by their historical precincts)
* Train and test a Logistic Regression model to predict the 2020 results.
* Create random “guesses” (randomly generated elected results), which we will use to compare the accuracy of the predicted results to random chance
* Complete a statistical analysis on the predictions to determine the model’s efficacy.
* Investigate the results and any trends we can identify from them.

Our deliverables will be:

* A csv file will be created and made available on GitHub for public use. This file will contain 1 row for each generated geometric shape with corresponding historical election results, the 2020 actual election results, the 2020 predicted results, and the 2020 randomly generated (“guessed”) results.
* A Jupyter Notebook file containing all the code to run the program
* A PowerPoint presentation will also be created which will explain the project, what it is attempting to solve for, how it attempts to solve it, and a simplified explanation of the results.

***Scope:*** The scope of our project will be to determine the efficacy of Logistic Regression in predicting election results, based solely on historical election results. This project will generate these static geometric points, map the points’ historical election results, create predicted election results, and compare the predictions to random chance.

Out of scope would be any attempt to analyze the “why” behind shifting election results. Also out of scope would be any attempt to make partisan or politically motivated declarations of the meaning behind the data.

***Planning Methodology:*** I used the Waterfall methodology to complete planning for this project. The sequence of this planning is below:

1. Requirements/Analysis

Here I define the project’s main goal and the objectives I need to accomplish to get to that goal. This involves identifying key components of the data, pros and cons of various data solutions, and understanding in depth what I am ultimately trying to solve for.

This is the stage where I researched the potential datasets available to me, including a preliminary investigation into their content to ensure they contained all data I would need for the project.

I also took some time reviewing the legal and regulatory limitations in working with the datasets I found to be available to me. I reached out to the AZ Secretary of State’s office via email, as I was unclear in the Arizona Revised Statutes if there were any legal limitations specific to this type of public data, and verified with their office that my project did not raise any concerns with them.

1. Design

After reviewing the requirements, I then worked on defining the logical design of my project. This stage is where I structured an outline of the data solution and broke down various logical steps to achieve my project's goal. These steps are defined as:

* + Extract and format XML, CSV, and GeoJson files
  + Determine (or generate) static geometric coordinates to reference to historical voting data
  + Use a machine learning model to predict the 2020 election results
  + Perform a statistical analysis to test how accurate the predictions were
  + Visualize the results

Once I understood the logical design of my data solution, I set out to define the physical design. Meaning, I started working on defining what tools and environments I would need to accomplish my logical design. This included many online searches as well as relying on prior experience with Python libraries to narrow down the set of resources I would need to accomplish my goal and objectives.

I already was familiar with xml and csv parsing, using Pandas for analysis, and the essentials of what Numpy could do for mathematical operations in Python. I also already knew I would be using a Jupyter Notebook as I was familiar with the environment and knew I could use it to provide a shareable program file, which is one of my deliverables. Not much analysis was needed, given my familiarity with what these tools and this environment could accomplish.

Where I took some time in the analysis phase was in researching the capabilities of geospatial analysis libraries in Python, as I was not quite as familiar with the tools available. After extensive research I settled on GeoPandas as my spatial analysis library. The factors that went into this decision include:

* + the ease of integration between Pandas and GeoPandas
  + the thorough documentation available on the GeoPandas library
  + the flexibility it offered in working with GeoJson files, which I determined in the Requirements phase was the type of files available to me for defining political boundaries.

I also utilized the Shapely library for my spatial analysis as I found online documentation discussing Shapely’s usefulness in extracting geometric coordinates from data. I determined this would be useful for the generation of my static coordinates, though at this point in my planning this was an assumption that I would have to test during Implementation and Testing.

I also took some time researching various machine learning models. I settled on Logistic Regression, as this model seemed easier to understand, implement, and interpret than others. I also wanted to minimize computational load in the program as much as possible, which my research suggested Logistic Regression to have a lower computational “cost” than other models.

I had experience already with hypothesis testing, but still worked to verify the specific tests that I logically planned to conduct were best for the project. After some time thinking about my options, I settled on a proportions z-test as my sample size would be much larger than 30 (t-tests are more appropriate for sample sizes smaller than 30.) Once I knew the tests I needed, it didn’t take much research to verify the Python libraries needed for these tests (Scipy and Statsmodel) as I had worked with them before. I did complete a quick search to see if there were any alternatives, but I found no alternatives that were superior in any relevant way.

1. Implementation

Once all of the prior steps were completed, this made the implementation phase more efficient. My steps were as followed:

* Data Gathering:
  + Manually downloaded data from the Arizona Secretary of State and Maricopa County Recorder’s websites.
  + Saved the files into folders in the Jupyter Notebook workspace.
* Data Cleaning and Formatting:
  + Used xml.ETree and csv libraries for cleaning and formatting XML and CSV files.
  + Structured the data into Pandas dataframes for ease of analysis.
* Geospatial Data Processing:
  + Employed GeoPandas, Shapely, and Contextily for processing GeoJSON files.
    - At this stage, Contextily was added as an afterthought to the geospatial analysis. This library imports background roadmaps for my geoplots, which makes the plots visually easier to interpret. This library was not at all necessary for my goal or objectives, so I did not revisit my Requirements phase, but this did mildly adjust the physical design of my project.
  + Generated interior geometric points using Numpy.
    - This step took a bit of incremental and iterative programming to get right, as my first couple of attempts were not successful for my goal or objectives. But I was able to accomplish this vital step in the program.
* Machine Learning:
  + Implemented Logistic Regression using the Sklearn library for training the model on historical election results and testing for the 2020 election.
* Random Result Generation:
  + Used Numpy for generating random election results for statistical testing.
* Statistical Analysis:
  + Applied Scipy and Statsmodel for proportions Z-test to evaluate the statistical significance of the Logistic Regression model.
* Plotting:
  + Utilized Matplotlib for creating scatter plots and an animated plot to visualize data.

1. Testing

I spent quite a bit of time reviewing the results of each step of the implementation phase, ensuring that I maintained data quality, data cleanliness, and analysis best practices. My quality checks involved several programmatic tests and investigations into the dataframes I created in Pandas, including:

* Checking for nulls
* Verifying that merges and pivots completed as intended (did not result in lost data.)
* Verifying that all precincts are represented in both the results data and the geometric data and that there are no significant mismatches when joining the two sources.
* Checking that any dropped columns or rows were not statistically significant for the project.
* Confirming that the final dataset produced was accurate and clean, meaning all data is consistent to the original datasets and that any columns included provided value to the end user.

1. Maintenance

I plan to upload all my files (including the Jupyter notebook itself) onto GitHub for use by others. Should I or others determine that there are bugs in the code or inaccuracies in the report, I plan to triage where appropriate and re-upload an updated version of the project.

Also, part of the maintenance phase is completing documentation of the project. For this I will use the outline of the WGU forms, as defined in Tasks 1, 2, and 3.

***Milestones:***

The completion of this project, up to the testing phase, has already been completed as this is a project I completed last calendar year. The end goal for the documentation of the project, which includes the documentation required by WGU, is 3/31/2024.

The phases completed thus far are reflected in the timeline below:

|  |  |  |
| --- | --- | --- |
| **Phase** | **Start Date** | **End Date** |
| **Requirements** | **11/1/2023** | **11/15/2023** |
| Define main goal | 11/1/2023 | 11/2/2023 |
| Define key objectives | 11/2/2023 | 11/5/2023 |
| Determine datasets available | 11/5/2023 | 11/10/2023 |
| Research legal and regulatory limitations | 11/10/2023 | 11/15/2023 |
| **Design** | **11/15/2023** | **11/29/2023** |
| Outline the logical design of the project | 11/15/2023 | 11/20/2023 |
| Research geospatial tools | 11/20/2023 | 11/26/2023 |
| Research machine learning models | 11/26/2023 | 11/28/2023 |
| Research statistical tests | 11/28/2023 | 11/29/2023 |
| Finalize the physical design of the project | 11/29/2023 | 11/29/2023 |
| **Implementation** | **11/29/2023** | **12/20/2023** |
| Data gathering | 11/29/2023 | 11/30/2023 |
| Data cleaning and formatting | 12/1/2023 | 12/4/2023 |
| Geospatial data processing | 12/4/2023 | 12/13/2023 |
| Train and test the Logistic Regression model | 12/13/2023 | 12/13/2023 |
| Generate random results for comparison | 12/13/2023 | 12/13/2023 |
| Complete statistical testing | 12/13/2023 | 12/18/2023 |
| Plot results | 12/18/2023 | 12/20/2023 |
| **Testing** | **12/20/2023** | **12/27/2023** |
| Complete data quality checks | 12/20/2023 | 12/23/2023 |
| Complete data cleanliness checks | 12/26/2023 | 12/27/2023 |
| **Maintenance** | **1/1/2024** | **3/31/2024** |
| Upload all files to GitHub | Planned 1/31/2024 | Planned 1/31/2024 |
| Document project | 1/1/2024 | Planned 3/31/2024 |

***Resources and Cost:*** I used my personal laptop to install and run the Anaconda platform. This was free and the laptop was a resource already available to me. Once Anaconda was installed, I used Jupyter Notebook as my sole environment for running the Python scripts.

The datasets were obtained from publicly available websites and were free to use.

My freelance rate would normally apply, except this project was not for any client. As someone with a fascination for politics and political data, this was a project I had on the back of my mind for quite some time. So, all work performed for the project was done as a volunteer.

In sum, there were no costs associated with this project.

***Measuring Success:*** I will first measure success by writing a Python program that will complete all of the objectives leading up to the statistical analysis. This includes extracting and cleaning the original data, generating the static geocoordinates, linking the election results data to these coordinates, training the logistic regression model, testing the model, and generating random “guesses” with Numpy to compare to the predicted results. If this program is able to successfully perform all of these tasks, then my efforts will pass my first measure of success.

Then I will perform a statistical analysis of the predicted results. I will measure the proportion of accurate predictions against the proportion of accurate randomly generated “guesses.” The null hypothesis will be that the proportion of accurately predicted results will be lower or equal to the proportion of accurate randomly generated results. The alternative hypothesis will be that the proportion of accurately predicted results is higher than the proportion of accurate random guesses. We will set the alpha level to 0.05 for our analysis.

**H0**: accurate\_results(predicted) <= accurate\_results(random)

**H1**: accurate\_results(predicted) > accurate\_results(random)

## C. Design of the data analytics solution.

***Hypothesis:*** My hypothesis is that Logistic Regression can be used to effectively predict election results based solely on historical election results and geocoordinates.

***Analytical Method:*** The analytical method employed in this project is predictive analytics, specifically utilizing Logistic Regression. This method involves training a model on historical election results to predict the 2020 election results for each geometric point.

Justification: The goal of the project is to predict the 2020 Maricopa County election results. Predictive analytics, by definition, is designed to forecast future outcomes by identifying patterns and relationships in historical data. In the realm of politics, understanding and anticipating future electoral outcomes is crucial for effective decision-making.

***Tools and Environments:*** I will complete my project in a Jupyter Notebook, utilizing the Python programming language. Jupyter Notebook is a server-client application that allows editing and running notebook documents via a web browser.

I use several Python libraries, which are important to completing my project. They are detailed below.

I use xml.ETree to extract and format the data I needed from the XML file because it is a library familiar to me that makes XML parsing faster and easier to read within the code. I decided to use the CSV library for the CSV files for the same reason.

I decided to use Pandas for the storing and formatting of the data as it allows for easy storage and viewing of the data in a format that can be best described as a table. Pandas also works extremely well with GeoPandas “geodataframes” which I use to analyze and format the geographic data.

I use Numpy, which is a Python mathematical library, to generate random election results for each geometric point. This is necessary to test the statistical veracity of the model. Numpy’s “random.choice” function makes generating these random results easy and straightforward.

I decided to use GeoPandas to process and format the GeoJSON files as this library is very easy to work with, the documentation is clear, and it functions almost seamlessly with Pandas dataframes. GeoPandas uses GeoDataFrames to store data into table like structures, much like Pandas, but with a special “GeoSeries” column. This unique column is necessary as it defines the geometric shape and spatial coordinates of said shape. These spatial coordinates can be stored in this column as “Points” (single latitude and longitude coordinates), Polygons (a closed geometric shape with straight lines), MultiPolygons(a collection of Polygons), and much more.

I use the Shapely library to work with extraction of the geometric points and polygons contained in the GeoPandas GeoDataFrames. This library allows me to extract necessary geometric data, which I use to generate the static geocoordinates I will use as my reference points throughout the project.

I use contextily in my plotting of the spatial data because this library creates backdrops of maps that sync with the GeoDataFrames to create more visually appealing map plots.

I chose Logistic Regression because it works well with categorical, binary data. The model is relatively quick to train, making it computationally efficient. It’s also a relatively simple Machine Learning model, making its results easier to analyze and interpret.

I use Sklearn as my machine learning library because it is a well-known and reliable library for training and testing machine learning models.

I use Scipy and Statsmodels libraries to complete my statistical analysis. Scipy is used briefly to run correlation coefficients of the election results. Statsmodels is the backbone of my statistical analysis, as it contains the proportions z-test function I use to test the efficacy of the model.

Finally, I use matplotlib as my plotting library. Matplotlib is a well-known, robust, and flexible plotting library commonly used to plot data in Python. I use it often for several scatter plots, map plots, and even an animated plot of historical election results.

***Statistical Methods and Metrics:*** I have chosen to run a proportions Z-test to measure the statistical significance of my solution. A Z-test for proportions is used when comparing two independent proportions. This test is best for binary outcomes, which is well suited for our use case as we are measuring “accurate” vs “inaccurate” results in both the predicted results and the randomly generated results. I use a Z-test instead of a T-test, as I will have much more than 30 samples to analyze, and the Z-test is considered to be more reliable for larger datasets.

I implement this test in the code using the Statsmodels’ library's “proportions\_ztest” function, which makes the analysis easier to interpret when reading the code and also consumes less “page real estate” than manually completing the test in Python code.

***Practical Significance:*** The Accuracy Score of the predictions(proportion of accurate predictions) will be one assessment of the practical significance of the results. This score will quickly tell us if the majority of the results were accurately predicted or not.

After I have completed my statistical analysis, I will also produce several plotted maps with both the predicted results and the historical election results. Comparing these plots would allow us to see if the areas isolated as outliers (not easily predicted) roughly correspond to areas more clearly in flux (experiencing shifting political trends.) If so, then the model will not only practically identify the areas steady and consistently voting along party lines, but also the areas where greater political outreach would be justified.

***Visual Communication:*** I use matplotlib for plotting my data because it is a widely used Python library for plotting static and animated visualizations. I use it in my program for several scatter plots, map plots, and also an animated plot showing the shifting voting trends for each geometric point.

I also added in contextily to my map plots, as it provides a nice background road map behind the plots. This is not necessary for the project but makes for map plots that are a little easier to interpret.

My specific visualizations are as follows:

* A map plot of the precincts within Maricopa County (2020 boundaries.) This visualization shows us the distribution of the precincts across the county, providing useful context for future plots.
* 1 map + scatter plot displaying the original boundaries of a sample precinct (Carriage Lane.) This plot allows us to see and explain what is contained in the original GeoJson files.
* 1 map + scatter plot displaying the generated coordinates –within- Carriage Lane precinct. This plot allows us to see how we are taking the original boundary points to create interior coordinates for each precinct. This plot is important to explain the concept of our coordinate generation.
* 1 map + scatter plot displaying the distribution of the incorrectly predicted coordinates.
* 4 map + scatter plots displaying the distribution of Republican and Democrat results (1 map for each election year in our data)
* 4 duplicates of the above map + scatter plots, narrowed into the urban center of Maricopa County.
* 1 animated heatmap plot that overlaps the display of the 4 maps of historical voting results, allowing for easier interpretation of how results shifted from election to election.

## D. Descriptions of the Dataset(s).

***Sources:*** Arizona Secretary of State’s Historical Election Results: <https://azsos.gov/elections/results-data/voter-registration-statistics/historical-election-results-information>

Maricopa County Recorder’s Mapping Files: <https://data-maricopa.opendata.arcgis.com/search?tags=Election>

These sources are appropriate because they are the official election result files and GIS mapping files as provided by the government authorities responsible for the collection and publication of such data.

***Collection Methods:*** I chose to use manual downloads of the files needed for this project as building a web scraper, while entirely possible, would not add any value to the project. There are only a few files, which are easily found and downloaded from the Secretary of State’s and County Recorder’s website. In addition, considering the frequency in which these websites tend to change in structure (almost every time there is a new Secretary of State or Recorder), any web scraper built would have to be re-built for use in future elections.

To download, I visited the websites as detailed in the Sources section above and downloaded the files to my personal laptop. I then uploaded these files directly into folders in my Jupyter Notebook workspace for use by the Python program.

***Quality and Completeness of Data:*** I found all files from both sources to be of very high quality and completeness. There were some seemingly erroneous precinct names between the two sources, as it looked to me that in very isolated cases the Secretary of State was using pre-redistricted precinct names and numbers whereas the Recorder office was using post-redistricting names and numbers. Where the mismatch could be accurately identified, I fixed this issue in the data wrangling process. Where I could not correct this issue, those precincts were dropped. This represented a very small number of geometric points (less than 1%) and were not statistically significant.

The data was complete for both sources, as all precincts were contained in both sets of files.

Overall, though, both files were cleanly organized and complete with very little data cleaning required.

***Data governance, privacy, and compliance:*** All data we use are publicly available on government websites.

* Data governance: Even though the data is publicly available, it was crucial to ensure we used the most recent versions. Multiple files are produced on election night and afterwards, showing incremental changes as more vote tallies are reported. Care was taken to ensure that only the final, official results were used in the project. Continued precautions would include ensuring future analysis files performed with the results of this project are conducted with only final, officially published election results.
* Privacy: There is a small chance in precincts with very low amounts of voters to interpret the results in a way that you can isolate what voter voted for which candidate. However, nothing in our project enhances or limits this ability. The election results files themselves enable this interpretation in their original, published form. To gain this sort of insight, one would need to reference “Precinct Register” files which contain voter specific data, including whether they voted in the most recent election. We did not use this voter specific type of data anywhere in our project. Should this type of data be integrated with this project in future iterations, precautions should be taken to ensure that the insights produced are summarized effectively enough to not violate privacy rights.
* Security: Given the public nature of the original data, little security is required for the files. Indeed, the intention is for the files to be made publicly available for use. However, precautionary data security steps to ensure the preservation of the files will be completed by creating redundancy in the file storage. Final files will be placed on GitHub for public access and a copy of these final files will be kept in my personal electronic records.
* Ethical: Care has been taken in the publication of our results to make sure they do not justify misinterpretation for political purposes. The 2020 election was highly controversial, with many accusations (especially in Maricopa County) of voter fraud and election tampering. In our Jupyter notebook file and also in summary reports on the analysis, I have made a point to mention that I am not answering the “why” behind the coordinates that are not easily predicted by the model. Precaution should be taken in communicating the results to ensure that there is no suggestion that the data is inherently indicative of foul play.
* Legal: This project and the publication of the results are well within the legal boundaries of what we are allowed to do with the public data. There are specific Arizona Revised Statutes pertaining to the use of “Precinct Registers” (ARS 16-168), which are not used in this analysis. The registers would include voter specific information, including voting history (what elections they voted in, not who they voted for.) None of this information is used in our analysis. Precautions should be taken to ensure that any future iterations of this project which may incorporate these Precinct Registers are conducted in a way which adheres to legal limitations.
* Regulatory Compliance: I was unable to find specific Arizona Revised Statutes pertaining to the use of summary election data or precinct GIS data. It seems from my research that such statutes do not exist. I did reach out to the Arizona Secretary of State’s office, describing what I was doing with the data, to which they respond via email that “they had no concerns with this activity.” Otherwise, I am not aware of any regulatory concern with our use of this data. Precautions should still be taken to understand changing regulations in regard to this type of data to ensure continued compliance.

**References:**

**Baltz, S., Agadjanuan, A; Chin, D., Curiel, J., DeLuca, K., Dunham, J., Miranda, J., Phillips, C., Uhlman, A., Wimpy, C., Zarate, M, Stewart III, C. (2022).** American election results at the precinct level. *Nature.com* <https://www.nature.com/articles/s41597-022-01745-0>

**LinkedIn (2023).** What are the advantages of using logistic regression for classification? *Linkedin.com* <https://www.linkedin.com/advice/1/what-advantages-using-logistic-regression-classification-p6eaf>

**University of Southern California (2021).** *Mapping America’s Future: 4 Ways GIS Data Impacts Elections.* <https://gis.usc.edu/blog/4-ways-gis-is-impacting-elections/>