Pradyoth Velagapudi

Prof. Sabiha Mahzabeen

DAT 301

23 November 2024

Interactive Geospatial Election Data Visualization in Python

One of the biggest stories of the 2024 election was data. Frightened of the failure of election models to accurately capture Donald Trump's vote share in 2016 and 2020, election pundits had been analyzing trends and revising in the months leading up to the election, even starting bitter feuds such as that between historian Allan Lichtman and renowned pollster Nate Silver. Now, in the aftermath of the election, we have concrete data to analyze and explain the direction that American society is heading. As my case study, I wanted to investigate Arizona's voting turnout and outcomes in the 2024 presidential election and compare it to 2020.

But election data is different from the data we dealt with in class. The raw numbers are intertwined with geospatial information that is crucial to understanding the data. From a technical standpoint, I wanted to investigate how I can represent data with a geospatial element in Python. Like many, I was poring over the endless election maps prior to and during the election, which offered countless fascinating ways to interact with voting data. In class we used charts and plots to visualize data, but I was curious about how I could visualize data on a map.

Some investigating led me to discover GeoPandas, an open-source Python library that allows seamless integration of geospatial data. Just like how the Pandas library can read CSV files to create dataframes, GeoPandas reads GIS shapefiles to create "geodataframe" objects. These geodataframes can be merged with Pandas dataframes to add geospatial data to a database.

The first thing I needed was a shapefile for a map of Arizona's counties. In GIS (geographic information systems), shapefiles are a common format used to store the location, shape, and attributes of geographic features as a collection of points, lines, and polygons. Shapefiles aren't a single file, but rather a collection of related files that depend on each other, including a .shp, .shx, .dbf, and .prj file. Industry standard GIS technologies such as ArcGIS use shapefiles as the primary way to store geographic data. Shapefiles are very difficult to produce on your own, but luckily there are publicly available shapefiles for many commonly used maps, for academic purposes.

I found a shapefile for Arizona's counties from the University of Arizona, and loaded it into my python code using GeoPandas' `read_file()` function. However, I encountered an early error when I realized the shapefile lacked a crucial .shx component. After some further research, I learned that the .shx component can be automatically generated from the other components of the shapefile by inserting the command `os.environ["SHAPE_RESTORE_SHX"]="YES"` at the beginning of my project.

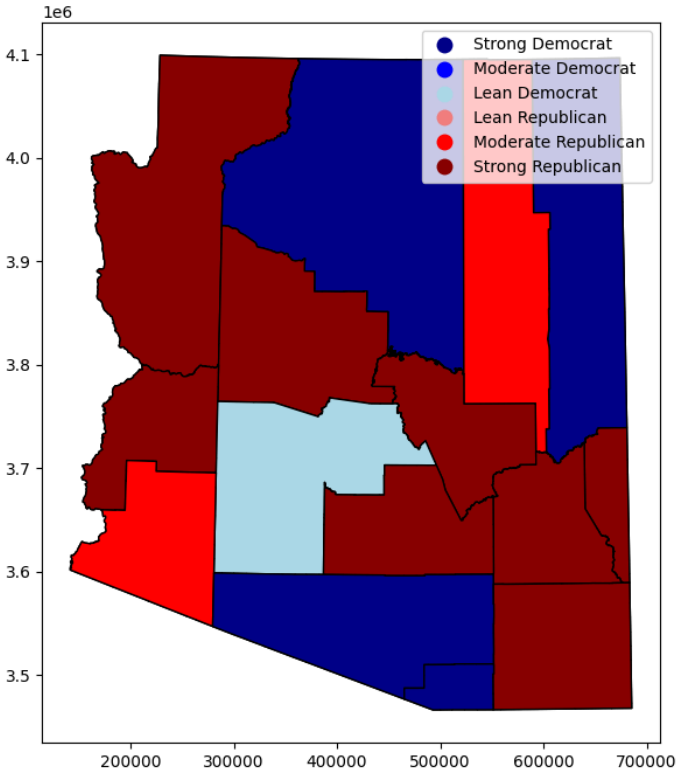
Next I found county-by-county voting data from 2020 and 2024. Though 100% of the results are not yet in for the 2024 election, unofficial numbers are available via the Arizona elections website. I cleaned this data and formatted it as a CSV, with rows `county,total_ballots,turnout_percent,votes_republican, votes_democrat`. I loaded this data into my project with the Pandas `read_csv()` function.

I had my election data and my geospatial data, but now I needed to combine them. GeoPandas allows for joining geodataframes with regular Pandas dataframes based on a common column—in this case, county name. After cleaning up the data to match between the shapefile and election data, I was able to create 2 merged dataframes—one for each election—using the

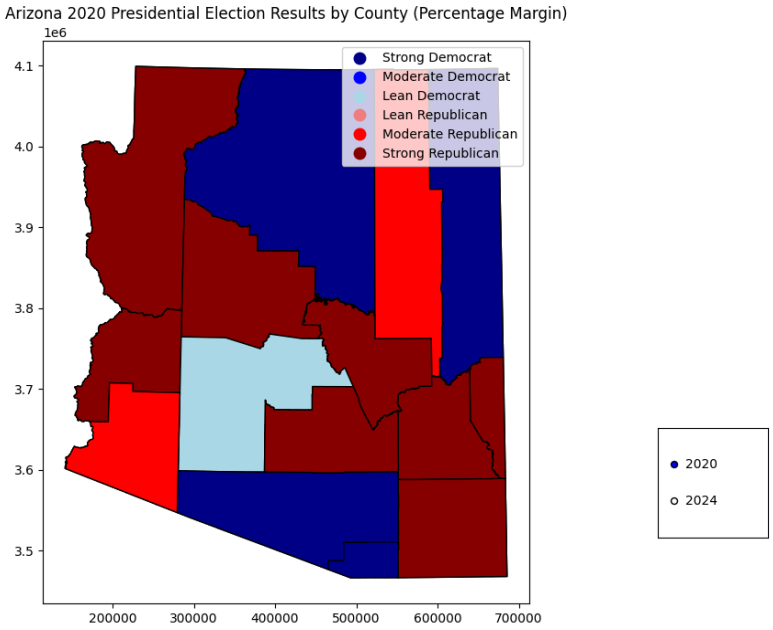
`merge()` function with tag `on="county"`. To display the map itself, it was as easy as using matplotlib's `plot()` function on the dataframe, choosing a column to represent with a graded colormap.

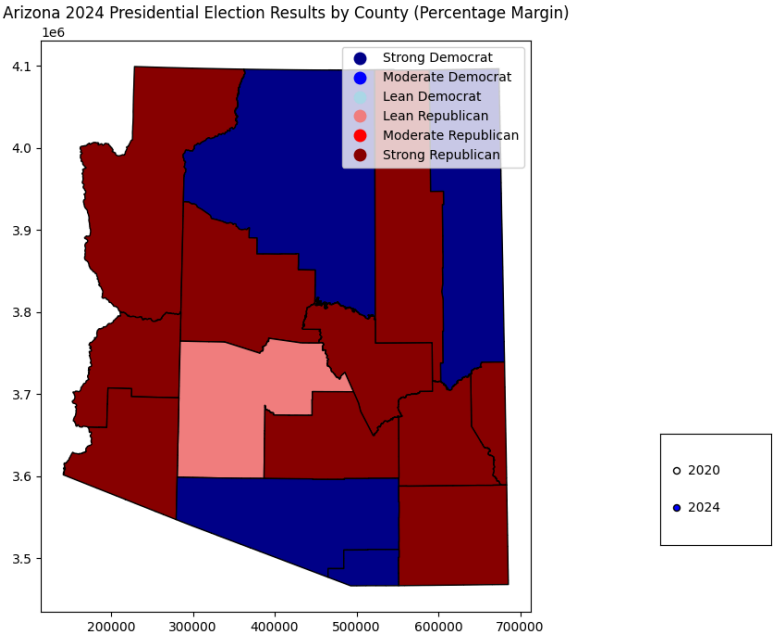
However, this only allowed me to display the results of one election, and the gradient colormap made understanding the data at a glance difficult.

First I defined discrete bins for the data. I calculated a column to represent the percentage margin of victory—that is, how many more percentage points the winning party got in a county. I defined and labelled bins for Strong Democrat, Moderate Democrat, Lean Democrat, Lean Republican, Moderate Republican, and Strong Republican. Then I used the pandas `cut()` function to chop the column into these discrete bins. After defining a custom colormap with colors corresponding to these bins, I was able to modify the plot to color counties based on the bin they fell into. This made it easier to understand the data at a glance.



But this still only displayed the results for one election at a time. So I added a radiobutton widget from matplotlib, and defined an `update_plot()` function that would change the displayed results based on what the user selected.





Finally, I was able to visualize the change from 2020 to 2024. But this still lacked the interactivity or detail of popular online election map tools. I wanted to add a hover element where users could hover their cursor over a county to see the actual margin numbers between 2020 and 2024, along with the county name and turnout. This proved to be a little tougher than I expected.

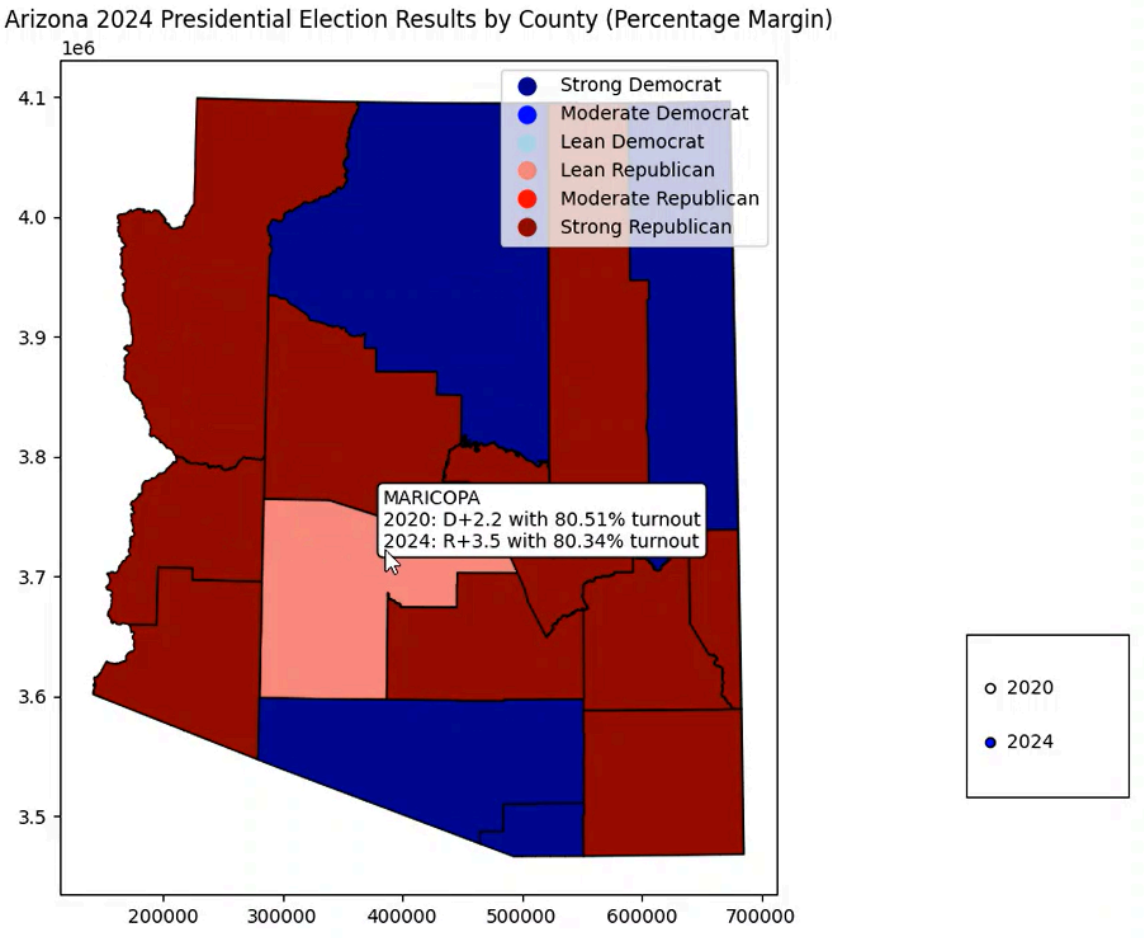
First, I attempted to use Python's mplcursors library, which is the most common way to introduce interactive mouse elements to matplotlib plots. However, after I got it working, I realized that it only recognized when I interacted with county borders, which were polygons plotted in matplotlib. I needed a solution that recognized when the cursor was within the boundaries of a particular county.

Exploring the GeoPandas library further, I realized that it had a function called `points_from_xy()`, which takes x/y plot coordinates as inputs and returns an array of shapely Point geometries. Using matplotlib's event handling functionality, I could grab the coordinates of the cursor and input these to the `points_from_xy()` function. I could then check which county's geometry contained those coordinates.

```
73
74      #mouse hover event handler
75      def on_hover(event):
76          if event.inaxes == ax:  #ensure the click is within the map
77              #convert cursor coordinates to GeoDataFrame CRS
78              coords = gpd.points_from_xy([event.xdata], [event.ydata], crs=counties.crs)
79              point = coords[0]  # Only one point
80
81              #determine which county was clicked
82              clicked_2020 = counties_2020[counties_2020.geometry.contains(point)]
83              clicked_2024 = counties_2024[counties_2024.geometry.contains(point)]
```

```
109     fig.canvas.mpl_connect("motion_notify_event", on_hover)
```

Using this information, it was possible to display the name and information of a given county upon hover. Adding a bit of code to format victory margins in the familiar “D+X/R+X” format added to comprehensibility.



With this, I was successfully able to create an accessible and interactive visualization of geospatial data in Python. This widget makes it easy to understand the changes in voter turnout and election outcomes between the 2020 and 2024 elections. I learned a lot about how geospatial data elements can be handled in Python in ways that integrate seamlessly with industry standard GIS software and with widely used Python data handling and visualization libraries.

A next step for this project could be to investigate how to display different layers on the same map. Adding data such as average level of education, ethnic demographics, or the number

of independent voters in each county could help reveal more insights about the election, but it would require a more complex user interface to choose which data is displayed. Potentially displaying multiple information maps overlaid on top of one another could show patterns that can provoke further investigation.

The full source code for my project, along with the necessary CSV and shapefile dependencies, is available here: https://github.com/PradyothVelagapudi/Election-Data-Visualization.

Resources

2020 election results: 2020 | The American Presidency Project

2024 election results: Arizona Election Results

Arizona counties shapefile: Arizona Counties Shapefile

GeoPandas user guide and documentation: GeoPandas 1.0.1 — GeoPandas documentation