Exercise: Mapping 2016 Political Contributions in Michigan

Summary

This exercise maps contributions to presidential candidates by party in the 2016 election for zip codes in Michigan.

Input Data

There are three main input files. The first, **by_place_party.csv**, is on the class Google drive. It is built from the Federal Election Commission's data on 2016 presidential political contributions but aggregated by party rather than candidate. The remaining files, **cb_2018_us_zcta510_500k.zip**, **cb_2018_us_state_500k.zip**, are cartographic boundary shapefiles from the Census. They should be downloaded from the Census via the Cartographic Boundary Shapefiles link on the class web page under Census Shapefiles. Finally, one other file is optional: **cb_2018_26_place_500k.zip**. It provides boundaries and names for Census "places", which are generally cities and towns. In building the map you may want to include it if you're interested in what communities are associated with the zip codes.

Deliverables

There are three deliverables: a script called **mi_by_party.py**, a QGIS project file called **mi_by_party.qgz**, and a PNG file called **mi_by_party.png**.

Instructions

A. Script mi_by_party.py

1. Import `pandas`.
2. Set `usa` to the result of calling `pd.read_csv()` to read `'by_place_party.csv'`.
3. Set `is_mi` to `usa['STATE'] == 'MI'` to build a series indicating which records belong to Michigan.
4. Create `mi` to be the subset of `usa` where `is_mi` is `True` by setting `mi` to `usa[is_mi]`.
5. Apply the `set_index()` method to `mi` to set its index to be `'zip'` and `'party'`. The first argument to `set_index()` should be the list `['zip','party']` and the second argument should be `inplace=True`. Note that since the change is being done in place, there won't be an equals sign in this line: it will just be `'mi.set_index()'`.
6. Print `mi`. Notice that the index is the first two columns, `'zip'` and `'party'` (internally, the index is a tuple), and there are two columns of data, `'STATE'` and `'amt'`.
7. Set `wide` to be the result of calling the `unstack()` method on `mi`.
8. Print `wide`. Notice that the index is now one column, `'zip'`, and the column names are now tuples containing `'STATE'` or `'amt'` and a party name. There will be a lot of missing values (NaNs) because there weren't contributions to every party in every zip code. We'll take care of that shortly.
9. Set `amount` to be `wide['amt'].copy()`. This picks out the `amt` block of columns and makes a copy of it so it can be modified below.
10. Apply the `fillna()` method to `amount` with 0 (zero) as the first argument and `inplace=True` as the second argument. As before when `inplace` was used, there won't be an equals sign in this line. Setting the missing values to 0 is appropriate in this context since there weren't any corresponding contributions.
11. Create a new column in `amount` called `'total'` that is equal to the result of applying the `sum()` method to `amount` with the argument `axis='columns'`. As in earlier exercises, the `axis` argument tells Pandas that `sum()` should add the columns together.

12. Use the `to_csv()` method to write `amount` out to `'mi_by_party.csv'`. Have a look at it to make sure it's what you expect. Don't worry that a few of the zip codes are very different from the others: those are errors in the underlying data and will not be included in the map.

B. Map mi_by_party.png

1. Download the Census files from the cartographic boundary web site.
2. Start QGIS and load `cb_2018_us_state_500k.zip`. Filter it to select the polygons with `STATEFP` equal to Michigan's FIPS code, 26.
3. Save the Michigan boundary by right-clicking the layer and selecting 'Export' and then 'Save Features As...' or by selecting the layer and choosing "Save As..." from the Layer menu. On the menu that pops up, choose "GeoPackage" as the format and save the file in the GitHub directory for the assignment under the name `michigan`. The extension `.gpkg` will be added automatically and the layer name should be set to `michigan`. The new layer should be added to the map.
4. Remove the original `cb_2018_us_state_500k` layer by right-clicking it and selecting "Remove Layer..."
5. Add the zip code layer in `cb_2018_us_zcta510_500k.zip`.
6. Clip the zip code layer using `michigan` as the overlay layer. The new layer will be added to the map and called `Clipped`.
7. Save the clipped layer to a GeoPackage called `michigan-zips` following the steps used above. A new layer called `michigan-zips` should be added to the map.
8. Remove the `Clipped` and `cb_2018_us_zcta510_500k` layers.
9. Add `mi_by_party.csv` to the map.
10. Join `mi_by_party` to `michigan-zips` using `zip` as the join field from `mi_by_party` and `ZCTA5CE10` as the target field.
11. Build a heatmap of `mi_by_party_DEM/mi_by_party_total`. Use the RdBu color ramp. You'll need to click on the drop-down button at the right of the color ramp box and then choose "All Color Ramps" to find it. Use "Pretty Breaks" with 5 classes.
12. Add a pie chart to the layer with `mi_by_party_DEM` and `mi_by_party_REP` as blue and red segments. On the "Size" page of the diagrams settings choose "Scaled size" and choose `mi_by_party_total` for the attribute to use for the size.
13. Click 'Find' to fill in the maximum value of `mi_by_party_total`.
14. Set the Size field to 10 or so and click "Apply". Feel free to adjust the size field if the diagrams seem too large or small.
15. At this point, optionally add the places layer. Make it partially transparent so it won't obscure the layers below. You may want to make it hashed rather than a solid color as well.
16. Export the map as **mi_by_party.png**.
17. Save the project as **mi_by_party.qgz**.

Submitting

Once you're happy with everything and have committed all of the changes to your local repository, please push the changes to GitHub. At that point, you're done: you have submitted your answer.

Tips

- This exercise is just scratching the surface of what could be done to analyze the contributions data. For example, it would be very interesting to use the Census API to download populations by zip code in order to

calculate per capita contributions. Many other variables could be added as well, including median income, race, education, employment status, and so on.