Exercise: Mapping Median Earnings for New York Counties

Summary

This exercise combines use of the Census API with mapping via QGIS. It examines median earnings over the last 12 months for the population 16 years and older who had any earnings. Just FYI, 'earnings' refers labor income and excludes income from capital (physical or financial assets) and transfer payments from the government.

Input Data

The only input file is **tl_2016_us_county.zip** from the earlier exercise. The remaining data will be obtained from the Census via its API.

Deliverables

There are four deliverables: (1) a script called **earn_by_county.py** that will request data from the Census API; (2) a CSV file called **earn_by_county.csv** that will contain the data obtained from the Census; (3) a QGIS project file called **earn_by_county.gqz**; and (4) a PNG file called **earn_by_county.png** that will be a heat map of median earnings.

Instructions

A. Script earn_by_county.py

1. Using an approach similar to that of the earlier Census API assignment, download variable `'B20002_001E'` (median earnings in the last 12 months) and create a dataframe from the results. Please call the dataframe `results`. As before, include `'NAME'` in the API query so the county names will be included. Also note that this task only involves a single Census variable so the steps in the earlier assigment related to `var_info` are not needed here.
2. Create a new column in `results` called `'geoid'` and set it to the result of concatenating the `'state'` and `'county'` columns of `results`. The result should be a 5-digit string that begins with 36. If you're not sure how to do the concatenation, see `demo.py` for an example.
3. Use the `set_index()` method of `results` to set the index to `'geoid'`. Remember to use `inplace=True`.
4. Create a new dictionary called `newnames` with one entry, `'B20002_001E':'median'`. This will be a convenient way to rename the Census variable to something easier to understand.
5. Call the `rename()` method of `results` with the following arguments: `newnames`, `axis='columns'`, and `inplace=True`. The dictionary `newnames` shows which names are to be changed as a collection of old_name:new_name pairs. The argument `axis='columns'` indicates that the renaming is to apply to columns. Pandas will look up each existing column name in `newnames` and, if the name appears there, it will rename the column to the corresponding value. Any columns that aren't found in the keys of `newnames` are left unchanged.
6. Use the `to_csv()` method of `results` to write out a file called `'earn_by_county.csv'`.

B. Map earn_by_county.png

1. Using an approach similar to that of the previous QGIS exercise, build a heat map of the median earnings in `'earn_by_county.csv'` using `tl_2016_us_county.zip`
2. Please save the QGIS project as `earn_by_county` in the GitHub directory for the assignment.
3. Export the image to the GitHub directory as well and call it `earn_by_county.png`.

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

- Using the same base name, `earn_by_county`, for all of the files is handy because it keeps related files together in directories with large numbers of files. It also helps reduce confusion since it's very clear that the four files are all part of a single conceptual task.
- Although there are other ways to rename columns or rows, using a dictionary is especially handy when there are lots of variables to rename because it is very clear and compact.