API Junkies - Project Report

Github repository: https://github.com/bkarduck/bkarduck_clairesyFinal

IMPLEMENTED CHANGES FROM PRESENTATION:

- Added an additional visualization, changed what was being read into the csv file, added in a function called richestAvgIncome to calculate the average income of the counties in the states with multiple counties that have a poverty rate below 5%, changed what was being put into the db to avoid duplicate values!

Original Project Goals

Poverty Data: https://www.census.gov/data/developers/data-sets/Poverty-Statistics.html
Covid Data: https://apidocs.covidactnow.org/#historic-data-for-all-states-counties-or-metros
We initially planned to use APIs from poverty and covid content. We were going to compare between covid and poverty, finding correlations between the two. By finding the correlation between the two, we were going to collect and store into the database from the poverty data different countries from the State of Michigan, finding poverty rates of top 5 highest poverty rate and top 5 lowest poverty rates, and then from the covid content collect data consisting of covid cases, deaths, and number of vaccines, we would then calculate the difference between the top 5 most poverty and top 5 least poverty in terms of covid metrics.

Updated Goal & Goals Achieved

APIs Used: Poverty Data, Counties Data Obtained From
https://www.census.gov/data/developers/data-sets/Poverty-Statistics.html

Counties Data:
https://api.census.gov/data/2019/acs/acs1/subject?get=NAME,S0101_C01_001E&for=county:*

Poverty Data:
https://api.census.gov/data/timeseries/poverty/saipe?get=SAEPOVRTALL_PT,SAEMHI_PT,GEOID&for=county:*&in=state:*&time=2020

After working on part of our project, we changed our project, changing it to focus the data on finding the states with the most poor and the most rich counties. With our new goal, we were able to achieve finding the top 100 highest poverty rates and top 100 lowest poverty rates of counties in the US. With these two, we were able to make graphs referring to the poverty rates for the counties with the most poverty counties and most low poverty counties. Among the states, we found that Georgia, Mississippi, Louisiana, Texas, and Arizona were states with multiple large counties that have a poverty rate above 22 as those that have a poverty rate below 5 were Minnesota, Virginia, Wisconsin, New Jersey, and Colorado. We noticed that the states with high poverty rates tended to be in the south and the states with low poverty rates tended to be in the north

Problems Faced

Problem 1: Difficulty in using API to find a set of counties where the population was defined

- **Solution:** we scoured the census website until we found something that would work for us. It took way longer than it should have.

Problem 2: Difficulty when trying to figure out how to make a database with just the counties we defined in the low and high 100 county lists instead of reading in all 800+ counties

- **Solution:** we ended up finding a way to loop through the counties and make an if statement determining if the codes were the same. We wanted to do this for program efficiency and we're glad it ended up working out!

Problem 3: When trying to figure out how to make the graphs, we first tried to make the function in the main file but found that to be too complicated and too much effort for what we were looking for, since we had already read the data into a table

- **Solution:** We reviewed the SQL lectures and discovered a better way to sort the data using SQL. We put this graph-making code in a new python file!

Calculation File: countyGraphs.py

```
    for county in richestCounties:
        nameList = county[1].split(",")
        stateName = nameList[1][1:]
        if stateName in incomeDict:
            incomeDict[stateName] += 1
            incomeCalcDict[stateName] += county[3]
            incomeAvg[stateName] = incomeCalcDict[stateName] / incomeDict[stateName]
        else:
            incomeDict[stateName] = 1
            incomeCalcDict[stateName] = county[3]
    return incomeAvg
```

In the function - richestAvgIncome - we calculated the the average income for the states that have more than one county with a poverty rate below 5

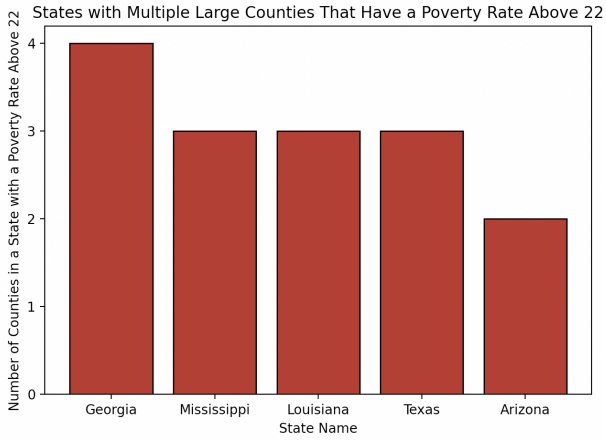
richestCountiesAvgIncome

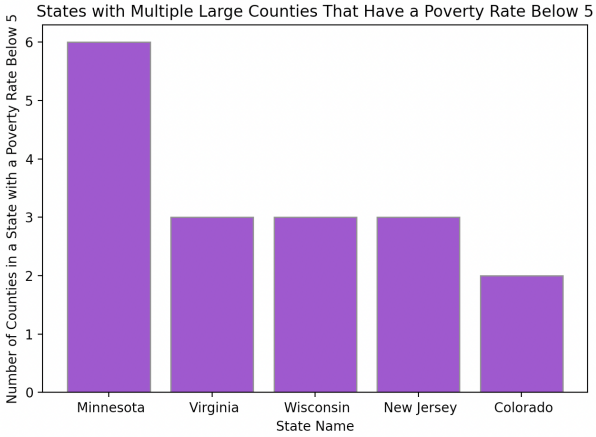
State Name	Average Income
Virginia	119396.66666666700
Minnesota	96392.5
New Jersey	114429.0
Wisconsin	86143.33333333330
Colorado	114276.5

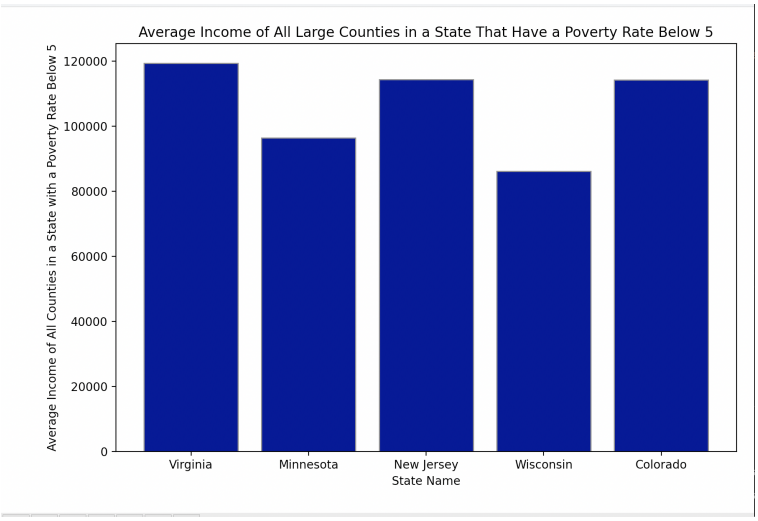
CSV Output:

Calculations are explained more in detail in the section in Code Documentation for countyGraphs.py

Visualizations







Instructions for running code

SIFinal.py must be run first, 4 times, and then after that countyGraphs.py can be run.

SIFinal.py

This file collects and obtains from the APIs of counties with a population greater than 65,000 in the US and poverty data from the US census. With the obtained data, it creates a list of the counties, highest 100 poverty and the lowest 100 poverty. With these lists, it then creates the database while then creating the tables for CountyFIP, Low100Poverty, and High100Poverty in the database county.db.

countyGraphs.py

With this file, it creates visualizations portraying the relationship between median income for the counties with the most poverty counties and the relationship between median income for counties with the most low poverty countries as well as average income of all large counties in states with poverty rate below 5. It creates the bar graphs based on what is returned from SIFinal.py

Code Documentation

countyGraphs.py

- def low100BarGraph(cur, conn)
    - Selects with cur the countyID and povertyRate from Low100Poverty table and the name from countyFIP table where the poverty rate is under 5%
    - Split the state and county names in the returned names
    - Returns a bar graph depicting the states that have more than one large (greater than 65,000 people) county with a poverty rate under 5%
- def high100BarGraph(cur, conn)
    - Selects with cur the countyID and povertyRate from High100Poverty table and the name from countyFIP table where the poverty rate is over 22%
    - Split the state and county names in the returned names
    - Returns a bar graph depicting the states that have more than one large (greater than 65,000 people) county with a poverty rate over 22%
- def richestCountiesIncomeGraph(cur, conn)
    - Obtained the state name, average median income from both the richestAvgIncome function and appended the states and income values into two different lists (state and income)
    - Returns a bar graph depicting the average income for states that have more than one large (greater than 65,000 people) county with a poverty rate under 5%
- def richestAvgIncome(filename, cur, conn)

    - Selects with cur the countyID and povertyRate from Low100Poverty table and the name from countyFIP table where the poverty rate is under 5%
    - Calculated the average income of all counties in the state where the poverty rate is under 5%
    - Returned a dictionary of the calculated average incomes
- def writePovertyDataFile(filename, cur, conn)
    - Input
        - Took in a filename to write a csv file to
        - Wrote the header
        - Obtained the state name, average median income from both the richestAvgIncome function and wrote the calculated average income for the states that had multiple counties with a poverty rate below 5%
    - Output
        - The csv file containing the data
- def main()
    - Sets up the connection to the database (county.db), calls the low100BarGraph and high100BarGraph functions, closes the connection to the database

SIFinal.py

- def getCounties()
    - Input
        - Calls a request to the URL provided in the function (which is an API of counties with population greater than 65,000 in the US), returns a json object of the read in file, loops through that abject to make a list that obtains the name, stateFIP, countyFIP, and fullFIP codes for each county read in
    - Output
        - Returns the created list of counties
- def get2020CountyPoverty():
    - Input
        - Calls a request to the URL provided in the function (which is the poverty data from US census about poverty rate, median income for all counties in the states)
        - Makes a new list of all of the FIP codes of the counties in getCounties() function
        - Returns a json object of the read in file, loops through the json object and if the full FIP code of the poverty API matched a county in the overall

counties list, appended the full FIP code, poverty rate, and median income of that county to a new county poverty list
    - Output
        - Returned the county poverty list
- def getHighest100Poverty()
    - Input
        - Created a list of the poverty counties, sorted by poverty rate with reverse = True, looped through the sorted list and appended the counties to a new list until 100 counties were added to the highest 100 poverty rate list
    - Output
        - Returned the highest 100 poverty rate list
- def getLowest100Poverty()
    - Input
        - Created a list of the poverty counties, sorted by poverty rate, looped through the sorted list and appended the counties to a new list until 100 counties were added to the lowest 100 poverty rate list
    - Output
        - Returned the lowest 100 poverty rate list
- def setUpDatabase(db_name)
    - Input
        - Database name
        - Used the database name to create a connection and determine cur
    - Output
        - Returned cur and conn
- def setUpCountyTable(cur, conn)
    - Input
        - Cur and conn
        - Created three tables: CountyFIP (countyID, name), Low100Poverty(CountyID, povertyRate, medianIncome), and High100Poverty(CountyID, povertyRate, medianIncome)
        - CountyID is the same between all tables; it is the stateFIP code plus the countyFIP code. It's a 5 number code unique to a certain county in the US
    - Output
        - Used as basically a void function for table set up purposes
- def makeCountyPlusLowPovTable(cur, conn)
    - Input
        - Cur and conn
        - Called getCounties to make a general list, called getLowest100Poverty() to make a list of the lowest 100 counties in terms of poverty rate.

        - Then selected countyID from countyFIP, defined a start number based on the length of the CountyFIP table
        - Looped first for a single county in county list, then did a second loop looking at for a county in the low poverty list and if the county code matched in both lists, defined variables to add into the tables based on the two lists
        - Inserted the data into the Low100Poverty and CountyFIP tables
        - Selected countyID from the countyFIP table again to check the length, if the new length of the table - the start length of the table was greater than or equal to 25, break the loop
    - Output
        - The output was committing this to the table so the tables are updated accordingly
- def makeCountyPlusHighPovTable(cur, conn)
    - Input
        - Cur and conn
        - Called getCounties to make a general list, called getHighest100Poverty() to make a list of the highest 100 counties in terms of poverty rate.
        - Then selected countyID from countyFIP, defined a start number based on the length of the CountyFIP table
        - Looped first for a single county in county list, then did a second loop looking at for a county in the low poverty list and if the county code matched in both lists, defined variables to add into the tables based on the two lists
        - Inserted the data into the High100Poverty and CountyFIP tables
        - Selected countyID from the countyFIP table again to check the length, if the new length of the table - the start length of the table was greater than or equal to 25, break the loop
    - Output
        - The output was committing this to the table so the tables are updated accordingly
- def main()
    - Input
        - Called getCounties(), defined cur, conn by setting up the database with county.db
        - Set up the table using cur and conn
        - Ran both of the makeCounty table functions
        - Closed the connection
    - Output
        - The output is the table

Resources

Date Issue	Description	Location of Resource	Result
4/5	Needed help to use API to pull search information while figuring out how to retrieve data with counties and population	https://www.census.gov/data/developers/data-sets/Poverty-Statistics.html	API reference index
4/6	Needed help to use API to pull search information while figuring out how to retrieve data with counties and poverty rates	https://www.census.gov/data/developers/data-sets/Poverty-Statistics.html	Another API reference index
4/17	We needed help on making the visualizations while changing details especially with the values showing up as floats as we rather wanted them to show up as integers	https://www.geeksforgeeks.org/matplotlib-pyplot-yticks-in-python/	We learned how to display the values as integers within the y-axis label of the graphs