**ETL Project – Member Household Income, Level and Zip code**

**Executive Summary**

Migrated a sample size of household member records from a local organization in a csv file and merged with a data file from the census API in order to determine average income level of each membership level based on zip code. Sensitive information is removed or blurred for security reasons. Initial csv file is also removed to protect privacy.

**Methodology\***

1. Collected two sources of data for this ETL project. The first data set was downloaded as a csv from a database (Altru) of a local cultural attraction and consisted of a sample of membership records. Fields included member level, status, expiration date, address, city, state, and zip code. The second data set was downloaded from the Census API. Fields included population, median age, household income, per capita income, poverty county, and zip code.
2. Imported the following dependencies and prepared setup for the data clean up and merge. Dependencies included pandas, numpy, matplotlib, requests, gmaps, os, and json. Not all dependencies were used in this exercise; however, there is a long-term goal to use these dependencies for future projects. From SQLAlchemy imported create\_engine.
3. Loaded member file as “active\_lapsed.csv”. Read and stored unclean member file in Pandas as member\_data.
4. Defined member dataframe as member\_data\_df.
5. Blurred address column for record protection and privacy.
6. Displayed dataframe with blurred column.
7. Calculated the total number of unique member addresses, which excluded duplicate records. The number of unique member addresses was 1,340.
8. Renamed Zip Code column to Zipcode for a simpler way to merge data. Blurred data for privacy.
9. Calculated the total number of rows/addresses, including duplicates. Total quantity was 1,890.
10. Revised member file by removing the last four digits of zip code and renamed dataframe as revised\_member\_df.
11. Confirmed that zip code was a string.
12. Retrieved second file via API. Pulled data and created a census\_pd. Fields included zipcode, population, median age, household income, per capita income, poverty count, and poverty rate.
13. Converted census\_pd to dataframe and renamed columns. Added in poverty rate count.
14. Confirmed that zip code type of census\_pd was also a string.
15. Merged census\_pd and revised\_member\_df on Zipcode column. Renamed file to merge\_df.
16. Removed null rows using the dropna.
17. Isolated data based to only include the following columns: zip code, household income, member level, status, expiration date, city, state. Dropped the other columns.
18. Created database connection to PostgreSQL, a relational database, using appropriate passwords. This database was chosen, because it is trusted for data integrity. It provides extensive data capacity and had a wide-range of user-defined data types.
19. Checked that one table included the merged member data with the following columns: index, Zipcode, population, Median Age, Household Income, Poverty Rate, Member Level, Expiration Date, City, and State.
20. Confirmed that data was added to by querying member merge.

**Note\*:** Please steps outlined in ETL\_project\_final.ipynb or ETL\_projetct\_final.py as #s for specific steps and code information. Original member cvs file that was placed in Resources removed for privacy.

**ETL Visual**

Census Data

Member Data

PostgreSQL

Python/Pandas