**Top 50 US Metropolitan Statistical Areas (MSA’s)**

**Spectral Clustering Binned by Population Density**

|  |  |  |
| --- | --- | --- |
| **Jupyter Notebooks** | *IHSDataProcessing.ipynb* | *IHSSpectralClustering.ipynb* |
| **Data Files Needed** | 1. *PumaCBSACrosswalk.csv* 2. *Top50MSAByPop.xlsx* 3. *AMI50MSA.xlsx* | 1. *50MSARental.csv* 2. *PopDensMissingPUMAs.xlsx* |

**Data Collection and Extraction:** *IHSDataProcessingl.ipynb*

\*Note: The census does not contain geography at the MSA level. It is called Metropolitan and Micropolitan Statistical Areas (CBSA). Therefore, CBSA & MSA terms are used interchangeably in this file.

1. A screenshot of a crosswalk

   Description automatically generated**Load ‘*PumaCBSACrosswalk.csv’* to retrieve corresponding PUMA’s and CBSA’s (MSA’s).**
   * Detect file encoding with *chardet.detect()*.
   * Add column names.
   * Convert numerical ‘ST’ (State) value into String type (necessary for housing/population dataset merge later).
2. **Load ‘*Top50MSAByPop.xlsx’* to get the top 50 MSA’s by population.**
   * A screenshot of a computer

     Description automatically generatedConvert dataset to a list.
   * Remove unnecessary punctuation (leading periods) and suffixes (‘Metro Area’).
3. **Run the *readACS()* function to scrape yearly housing and population data (for the top 50 MSA’s by population). Individual files can be seen at the US Census Bureau ACS 1-year PUMS data FTP site**[**[1]**](https://www.census.gov/programs-surveys/acs/microdata/access.html)**, and all results can be extracted utilizing the ACS 1-Year Data (2005-2021) API**[**[2]**](https://www.census.gov/data/developers/data-sets/acs-1year.html)**. Do this for 2012-2019.**
   * Extract the following housing and population attributes[[3]](https://www2.census.gov/programs-surveys/acs/tech_docs/pums/data_dict/PUMS_Data_Dictionary_2019.pdf) in JSON format and convert to pandas data frames.

|  |  |  |
| --- | --- | --- |
| Housing Dataset | | Population Dataset |
| * + - SERIALNO     - WGTP     - GRPIP     - RNTP     - HINCP     - TEN | * + - OCPIP     - VACS     - YBL     - BLD     - ST     - PUMA | * + - AGEP     - SERIALNO     - RAC1P     - SCHL     - HISP     - ESR     - SPORDER=1 |

* + Merge the housing and population datasets on the *SERIALNO* attribute. Use a left join to merge to the housing data. The housing data has more records than the population data because some households are vacant and will not have a corresponding record in the population file, but need to be included in records to calculate vacancies.
  + Add the corresponding MSA’s by merging ‘*PumaCBSACrosswalk.csv’* to the merged housing and population dataset on the *ST* and *PUMA* attributes. Also add an extra column for the appropriate year.
  + Return the merged data frame for the year and export as a CSV file (ex. *acs2019.csv*).

1. **Merge all ACS datasets**

**Feature Engineering:**

1. **SPSS Preprocessing**
   * *‘acsMerge\_50MSAs3.sav’* looks like the processed file using the ‘*Syntax 50MSAs.sps*’ syntax file to run the needed scripts. It already has the AMI values as well
   * Convert
2. **Add the Area Median Income (AMI) values to its corresponding MSA and year.**
   * Load ‘*AMI\_50\_MSAs.xlsx’* to obtain the Area Median Income (AMI) values.
   * Merge the previously created ACS datasets into one DataFrame.
   * Load merged ACS file ‘*acs50MSAs.csv’* and add AMI values to corresponding year.

Cluster50MSAs\_filtered\_2.csv = Clustering variables.xlsx + fullMSA + outMSA + partMSA + MostInMSA

**Spectral Clustering Preprocessing:** *IHSSpectralClustering.ipynb*

1. **Load ‘*50MSARental.csv’* to get the normalized features used in clustering.** 
   * *A screenshot of a screen

     Description automatically generated*All features have been normalized except *‘PopulationDensity2021.’*
   * 72 features
   * 1335 PUMA’s
2. **Remove PUMA’s partially in or outside of MSA territories.**
   * Remove PUMA’s flagged as ‘1’ in *outMSA* and *partMSA* columns. Keep only PUMA’s flagged in *fullMSA* and *MostInMSA*.
   * 1287 PUMA’s remaining (48 removed)
3. **A screenshot of a computer

   Description automatically generatedLoad *‘PopDensMissingPUMAs.xlsx’* to fill 30 missing *PopulationDensity2021* column values with corresponding 2016 values.**
   * Rename *PopDens2016* column to *PopulationDensity2021.*
   * Merge (using *StateID* and *PUMA* columns as key) to fill missing population density values.
   * Confirm values were correctly filled. 0 null values should exist and PUMA 6513 = 5067.2.
4. **Remove rural areas (PUMA’s where *PopulationDensity2021* is less than 500).**
   * 1195 PUMA’s remaining (92 removed).
   * Confirm there are no values below 500. New lowest *PopulationDensity2021* should be 501.6.
5. **Create new feature *PopDens\_binned* to group *PopulationDensity2021* values into 5 equal-sized bins.**
   * Use the *qcut()*[*[4]*](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.qcut.html)function to sort and separate the *PopulationDensity2021* column into 5 bins.
   * Between 238 and 240 PUMA’s in each bin.
6. **Create a new dataset by removing non-numerical and categorical columns, then normalize using *MinMaxScaler()***[[5]](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html)**for clustering.**
   * Remove 13 columns:

|  |  |  |
| --- | --- | --- |
| * *StateID* | * *Share\_1\_Unit\_perm17\_19* | * *MostInMSA* |
| * *PUMA* | * *Share\_2\_Unit\_perm\_17\_19* | * *partMSA* |
| * *MSA* | * *Share\_3\_4\_Unit\_perm17\_19* | * *fullMSA* |
| * *PopulationDensity2021* | * *Share\_5more\_Unit\_perm17\_19* | * *outMSA* |
|  | * *Subsidized\_share* |  |

* + Use SciKit-Learn’s *MinMaxScaler()* to normalize all values between (0, 1).
  + Final dataset for clustering should have 60 features, 1195 PUMA’s

**Spectral Clustering and Interpretation:** *IHSSpectralClustering.ipynb*

1. **Perform spectral clustering algorithm on the final numerical dataset (1195 PUMA’s) using the *spectralClust()* function.**
   * UseScikit-Learn’s *SpectralClustering*()[[6]](https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html) function to construct an affinity matrix using the default Gaussian (‘rbf’) kernel with Euclidean distance and setting *n\_clusters* to 6.
   * For consistency (due to different OS and configurations) find the *random\_state* that gives the Silhouette Score[[7]](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html) ~0.078448226. The cluster enumeration may vary, but the results will be consistent.
   * Add the new cluster labels to each PUMA as the *spec\_clus* column. Label numbers may vary.
   * Use the *switchClusters()* function convert clusters numbers to match the CZI Noah maps created with CARTO. Cluster distribution is shown below, with the correct numerical cluster label.
   * Final dataset includes 74 features, 1195 PUMA’s

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Cluster** | **1** | **2** | **3** | **4** | **5** | **6** |
| **# of PUMAs** | 267 | 75 | 85 | 206 | 276 | 286 |

1. **Group PUMA’s by cluster enumeration, create boxplots (for cluster interpretation) for each feature of the dataset, and plot the overall cluster median line.**
   * Set the color palette to use for each cluster.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Cluster** | **0** | **1** | **2** | **3** | **4** | **5** |
| **Color** | Red #FF0000 | Blue #0000FF | Orange #FFA500 | Yellow #FFFF00 | Purple #A020F0 | Green #1B8D2E |

* + Create boxplots for each numerical feature using Seaborn’s *boxplot()*[*[8]*](https://seaborn.pydata.org/generated/seaborn.boxplot.html)function.
  + **A comparison of a diagram

    Description automatically generated**Using Matplotlib’s *axhline()*[*[9]*](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.axhline.html)function, add the overall cluster median line (and legend) for each feature. All plots are shown in pairs, ex. below.

1. **Run ANOVA**[**[10]**](https://www.statsmodels.org/dev/generated/statsmodels.stats.anova.anova_lm.html) **to check for significance in the variance of each feature.**
   * A screenshot of a computer

     Description automatically generatedCreate a copy of the full dataset (74 features) and remove the *StateID, PUMA, and MSA* columns.
   * Create a list of the 71 column names.
   * Fit an ordinary least squares model (OLS)[[11]](https://www.statsmodels.org/stable/generated/statsmodels.formula.api.ols.html)for each feature.
   * Perform ANOVA and evaluate: p-value (PR(>F)) < 0.05 means significant variance
   * Retrieve ANOVA table as R like output.
   * Signal which features are significant according to ANOVA results.

**Mapping PUMA’s:** *IHSSpectralClustering.ipynb*

1. **Extract GEO data and map PUMA’s in top MSA’s.**
   * Use the *read\_file()* function from the *geopandas*[[12]](https://geopandas.org/en/stable/docs/user_guide/io.html) package to load zip files containing geospatial data from the US Census Bureau. (11 features)
   * 12 MSA’s and the USA coordinates are extracted to use as examples.
   * A screenshot of a computer

     Description automatically generatedAn example of the first 3 rows of Chicago saved as *va\_tract1* is below.
   * A yellow and orange rectangle

     Description automatically generatedCreate a color palette to match the boxplot colors.
   * Run the *map\_plot()* function to merge the *GEOID10* data extracted in the *va\_tract#* datasets to the dataset including the cluster labels. Then the maps of the 12 MSA’s and USA with colored coded PUMA’s using the map palette are plotted.

A map of different colored squares

Description automatically generated

**A map of the united states

Description automatically generated**

1. **Create cross-tabulation tables for the MSAs**
   * Show the number of PUMAs in each cluster for every MSA.
   * Show the proportion of each cluster inclusive for every MSA.
2. **Plot the MSA clusters using GeoID (Chicago, NY, Seattle)**

**References**

1. [US Census ACS 1-year PUMS data](https://www.census.gov/programs-surveys/acs/microdata/access.html). FTP site available in CSV and SAS formats. Retrieved June 14, 2023.
2. [ACS 1-Year Data API (2005-2021)](https://www.census.gov/data/developers/data-sets/acs-1year.html). Access and use US Census Bureau data. Retrieved June 14, 2023.
3. [2019 ACS PUMS Data Dictionary](https://www2.census.gov/programs-surveys/acs/tech_docs/pums/data_dict/PUMS_Data_Dictionary_2019.pdf). Includes variables available for each PUMS release and how each variable is coded.
4. [Pandas qcut().](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.qcut.html) Discretize variable into equal-sized buckets based on rank or sample quantiles.
5. [Scikit-learn MinMaxScaler()](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html). Transform features by scaling each feature to a given range.
6. [Scikit-Learn SpectralClustering()](https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html). Apply clustering to a projection of the normalized Laplacian.
7. [Scikit-Learn silhouette\_score()](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html). Compute the mean Silhouette Coefficient of all samples. The best value is 1 and the worst value is -1. Values near 0 indicate overlapping clusters.
8. [Seaborn boxplot().](https://seaborn.pydata.org/generated/seaborn.boxplot.html) Draw a box plot to show distributions with respect to categories.
9. [Matplotlib Pyplot axhline().](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.axhline.html) Add a horizontal line across the Axes.
10. [Statsmodels anova\_lm().](https://www.statsmodels.org/dev/generated/statsmodels.stats.anova.anova_lm.html) Anova table for one or more fitted linear models.
11. [Statsmodels ols().](https://www.statsmodels.org/stable/generated/statsmodels.formula.api.ols.html) Fit statistical model using R-style formulas.
12. [Geopandas read\_file().](https://geopandas.org/en/stable/docs/user_guide/io.html) Can read almost any vector-based spatial data format and returns a GeoDataFrame object

**Appendix**

1. **Access W: Drive**
   * <https://webdrive.depaul.edu/FileAccess/>
   * W 🡪 com 🡪 rec 🡪 IHS\_User 🡪 sbabu4 🡪 Top 50 MSAs Data

**Glossary**

**ACS (American Community Survey)** – The American Community Survey is an ongoing survey that provides data every year – giving communities the current information they need to plan investments and services. The ACS covers a broad range of topics about social, economic, demographic, and housing characteristics of the US population. [US Census Bureau - ACS](https://www.census.gov/data/developers/data-sets/acs-1year.html)

**MSA (Metropolitan Statistical Area)** – geographical region with a relatively high population density at its core and close economic ties throughout the area. [Wikipedia - MSA](https://en.wikipedia.org/wiki/Metropolitan_statistical_area)

**PUMA (Public Use Microdata Area)** – non-overlapping, statistical geographic areas that partition each state or equivalent entity into geographic areas containing no fewer than 100,000 people each. They cover the entirety of the United States, Puerto Rico, and Guam. [US Census Bureau](https://www.census.gov/programs-surveys/geography/guidance/geo-areas/pumas.html) - PUMA

**PUMS (Public Use Microdata Sample)** – PUMS files enable data users to create custom estimates and tables, free of charge, that are not available through ACS pretabulated data products. The ACS PUMS files are a set of records from individual people or housing units, with disclosure protection enables so that individuals or housing units cannot be identified. [US Census Bureau - PUMS](https://www.census.gov/programs-surveys/acs/microdata.html)