**Mapping Distribution Centers**

Cole Anderson, GIS 5578

**Introduction/Motivation:**

The project was motivated by discussions about spatial division of labor in a university geography classroom. This division of labor, or where each element of a corporation’s commodity production takes place, is often highly dependent on class and minority status. Factories might be entirely staffed by low income females in rural areas, or by immigrants with few other possibilities in urban cores, for example. Many electronic retail houses have placed factory-like distribution centers across the world to keep their enormous logistics chains running smoothly. However, the effects of these centers on local economies have been mixed. While cities fight to bring the jobs associated with such enormous corporations, the labor abuses and working conditions at Amazon fulfillment centers have sparked protests and strikes. Local wages have crashed in the communities around fulfillment centers in a race-to-the-bottom for firm survival. This led to the motivating questions: do distribution centers target certain neighborhoods with specific demographic or economic traits? Are immigrants, minorities, and poor areas being targeted for these working conditions because they have few options? This project explored possible correlations between the electronic retail distribution center density and three demographic traits: income, racial breakdown, and immigration status. {add deliverables}

**Data Types and Sources:**

Data on the selected demographic traits for three years were downloaded from IPUMS NHGIS, tabulated by ZCTA (ZIP Code Tabulation Area). Unfortunately, this site did not offer the needed data at the correct granularity for individual non-census years. 5-year ranges (2008-12, 2010-14, 2013-17) from the American Community Survey, centered on the year of interest, were selected as proxies. There was some concern about the use of year-range proxies with single year data files, but the next option would have been to significantly reduce the data granularity to county level. The margin of error was anticipated to be smaller by using range files than county data. Each data sheet came as a .csv with a .txt metadata file. The metadata provided a codebook of what measures corresponded with what data codes. Polygon vectors for ZCTAs were downloaded from IPUMS NHGIS site. These files were fortunately offered in individual years (2010, 2012, 2015) in ESRI shapefile format (.shp).

The industry density files came from directly from the US Census’s County Business Patterns page. Each yearly .txt file contained counts of industrial facilities by respective NAICS codes for each ZIP code. This data was only offered by ZIP and not ZCTA, causing some methodology concerns. ZCTA and ZIP essentially measure the same geographic space by polygons and addresses, respectively, and can be joined with minimal error.

Summarily, each year analyzed used data from two year-range proxy ZCTA demography files, a yearly ZCTA shapefile, and a ZIP industry count file.

**Methods:**

**Model\*:**

Rename CSV header on ZCTA demography files

Join ZCTA demography files

Edit ZIP industry file columns and group rows

Join ZIP industry files

Merge ZIP and ZCTA files by common column

Join data to ZCTA shapefile

Calculate location coefficient (LQ) for each ZCTA

Calculate percentage black, immigrant for each ZCTA

Calculate R-value between each demography marker and LQ

Create LQ choropleth

Repeat for 2010, 2012, 2015

\*See Figure 1.1 for visual representation of code model

**Methods Description:**

The first component will be creating scripts to automate sorting and processing the large number of data files. All of the functions are defined in the header, and occur for each data year in the order specified here.

*ReadDataDictionary:* this function opens the codebook associated with demography files and breaks it to individual key: value pairs at each ‘:’ in the book. Each pair is added to an uncleaned variable dictionary, which is returned

*ParseMultipleDictionaryOccurances:* this function counts the number of occurrences of each value in the variable dictionary from *ReadDataDictionary*, and eliminates duplicate entries. It returns the cleaned variable dictionary

*ReadCSV:* this function opens each demography CSV and aquires the column headers and values to an ordered dictionary for processing via the variable dictionary

*WriteFile:* this function writes the values in the variable dictionary over their respective code key headers in the CSV file, so they are finally easier to interpret.

*JoinDemoToIndustry:* this is the major joining function in this project. All of the CSVs should be cleaned via the four above scripts prior to use. First, both ZCTA .csv files and the industry ZIP .txt file are read into data frames via pandas. The ZIP file is modified to keep only the ZIP, NAINCS, and count estimates, and the ZIP column has letters added to match GISJOIN format. The ZIP file is split into two frames for modification. ZIP1 is modified to only rows containing the NAICS of interest in each ZIP. All rows of the same ZIP are grouped to create total counts, the data-frame index re-added, and the industry estimate column renamed ‘45411x Industry Count.’ ZIP2 undergoes the same processes, except all industry codes are kept before grouping, and the name of the estimate column is replaced with ‘All Industry Count.’ ZIP1 and 2 are then remerged by ZIP. All ZIPS with values are preserved. The ZCTA csv’s are joined by GISJOIN on all rows with all demography markers. ZCTA and ZIP with all demography markers are then joined by ZIP and GISJOIN (same values) and exported to file.

*JoinStatstoSHP:* This function takes in the master joined file from *JoinDemoToIndustry*, and the corresponding ZCTA shapefile. The shapefile is read into a geo-data frame by geopandas, the joined csv into a data frame by pandas. The files are joined on the GISJOIN column. The percent black, white, and immigrant are calculated and added to new columns in the dataframe. Per capita income is already present. The LQ for each ZCTA is calculated from the industry count columns, and the file is exported as a new shapefile.

The next major script component is calculating correlations and mapping or tabulating the results. This was done in a separate script for cleanliness and ease of use. The completion of *JoinStatstoSHP* is required, for input.

*CorrMeasures:* The inputs for this function are joined shapefiles, an area name, and a geographic extent, default value 0. The shapefile is read to a geo-data frame via geopandas and is cut down by the specified extents. The correlations between LQ and each demographic marker are calculated for the modified extend, and displayed. Next, a map display of LQs is set up. The displayed column, color ramp, and classification scheme are defined by the user. The figure is named according to the area, and exported at 500 dpi to file. The process of map creation is covered in depth in the Visualization section of this report.

**Modules:**

CSV was used to read and write initial ZCTA csv header to re-assign names from codebook. Ordered Dict was used to store key:value pairs to re-assign names ZCTA csvs. Pandas and Geopandas were by far the most utilized modules in this project due to their ability to modify large data-frames in a relatively user-friendly way. Pandas was used for the following: reading data CSVs into data-frames, rename column headers, joining ZCTA and ZIP data into single sheet, and dropping unneeded data rows and columns. Geopandas was used for the following: joining data sheets to ZCTA shapefiles, calculating percent by demography, calculating location coefficient measures, and interfacing with matplotlib for coefficient and map making. Matplotlib was used to interface with geopandas and create choropleth maps from shapefiles. Matplotlib maps are extremely customizable, almost to the point of being difficult to use.

**Data analysis, description, outcomes/Deliverables and Results:**

The expected final results were a conclusion about correlation direction and strength between distribution centers and low income, immigrant, or minority neighborhoods. A table of correlation measures was created from the *CorrMeasures* outputs for each year, see Figure 1.2. The conclusions from the correlations, for Minnesota/Wisconsin, are:

* Small positive correlation between 45411x LQ and percent black
* Small positive correlation between 45411x LQ and percent foreign born
* Small positive correlation between 45411x LQ and per capita income.
* More minority weakly correlates with more distribution centers, while poverty seems to work in reverse.

Yearly maps of distribution center LQs were also produced, seen in Figure 1.3

It is important to know how LQ is calculated, since it has been extensively referenced in the analysis section of this report:

*LQ = (ZCTA Industry of Choice ∕ ZCTA All Industry)/*

*(Nation Industry of Choice ∕ Nation All Industry)*

**Visualization**

The visualization tactic of choice for this project was choropleth maps. For each year, a map was produced to show which ZCTA had what density of distribution centers. Matplotlib was the chosen tool for this, although it can be used much more in depth in the future. The customization options offered for choropleth maps are rather extensive. These were the selected options for the ZCTA maps:

*column*: the chosen data frame column to map

*alpha*, *linewidth*, *edgecolor*: the color intensity of shading, width, and color of boundaries *cmap*: the color ramp for the choropleth. This option either seemed far too light or intense.

*scheme = 'user\_defined':* the classification scheme was defined by user

*classification\_kwds = {'bins':[1,5,10,25,50,100,200]}):* The defined classification scheme.

The map shows a handful of spatial patterns. The first is that the highest density ZCTAs do not always seem to occur around the largest urban area, like the Twin Cities. Most of the geographies in the Metro are mid-level density, although the total number of 45411x industry locations per ZCTA may tell a different story. Another conclusion is that, except in the Metro, areas with high density are not clustered into groups. There are ‘heavy zones’ interspersed in a region of overall light density. The third conclusion is that there are in fact heavy areas in the northern and agricultural regions of the state.

**Learned and Challenges**

**Challenges:**

There were compatibility problems getting geopandas to function inside ArcPro’s Spyder 3, Either the versions of another module in ArcPro were incorrect and not found, or there are modules that are difficult to utilize in each particular Spyder version. Anaconda’s Spyder 3 worked fine, however, and did not have similar problems loading other modules except arcpy. Arcpy was unneeded for this particular project. Matplotlib also proved to be a difficult module to use. The maps it creates can be modified in almost every possible way, sometimes making it difficult to keep all the options straight. It is obvious after seeing other data presentations that it is indeed possible to make the produced maps look nice.

There were a large number of files and folders to keep straight, which proved challenging when writing processing scripts. A handful of globals were used to lay out the folder structure. Files of the same type were named similarly across years, allowing the use of string formatting for many repeated function calls. The OS Module was suggested to manage files in a clean manner.

**Learned:**

Data projects inevitably end up being far more tedious than originally thought. The original intent of this project was to analyze data from specific metropolitan areas for well over a dozen years. There are always tighter time, frustration, and data limitations that anticipated, and this project ran into all three. Data is often messy to work with, and unknown problems arise often. As a result, there were significant cutbacks. It is also extremely important to add function descriptions and comments right as the code itself is written. It is easy to become confused with exactly what each line accomplishes if errors show up at the end, and there is little documentation. Github proved invaluable for managing this project and will be used in the future. The ability to manage files across the net while simultaneously being able to modify them on a personal computer is superb, rivaling Google’s services. However, it can be frustrating to resolve version conflict if the repository is not updated correctly.

**Future goals for this project**:

clusters,

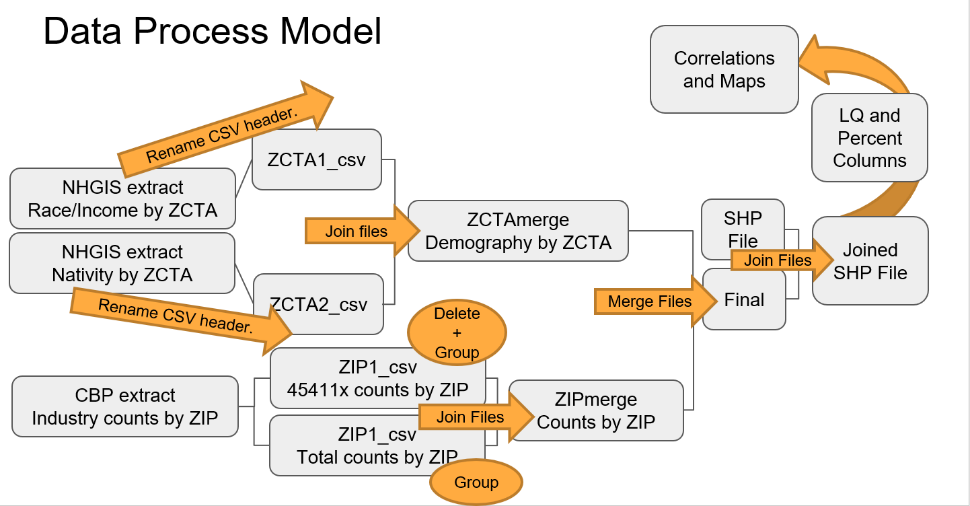
how to better utilize visualization modules,

spatial significance,

better error catching

**Appendix:**

**Fig. 1.1**

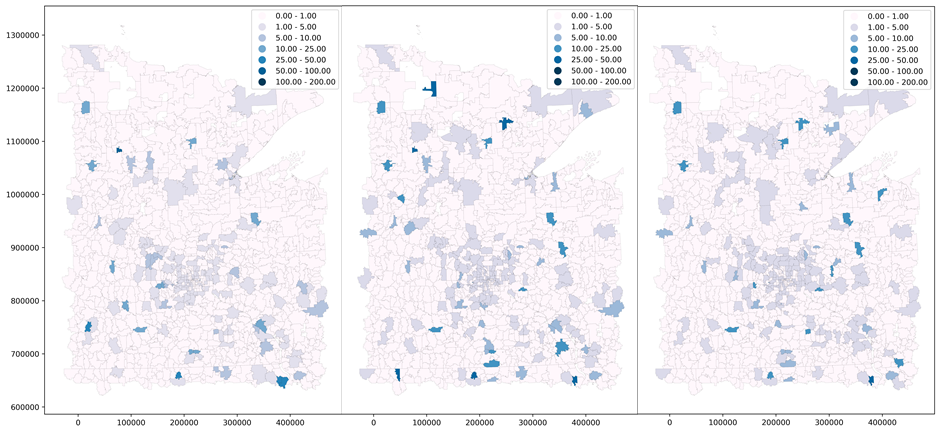


**Fig. 1.2**

|  |  |  |  |
| --- | --- | --- | --- |
| Correlation R-values  MN 45411x LQ | Percent Foreign Born | Percent Black | Per Capita Income |
| 2010 | .1935 | .1267 | .1052 |
| 2012 | .0956 | .0588 | .1180 |
| 2015 | .0011 | .0026 | .1060 |

**Fig 1.3**

2010



2012

2015