Taking the “L” on Public Transit

Roberto Barroso, Charmaine Runes, Hana Passen
barrosoluquer-crunes-passen

Project Summary:

What does life expectancy or blood pressure look like along Chicago’s elevated rail lines? Are diabetes and race correlated among Chicago’s metro system? We set out to answer these kinds of questions by creating a visualization of the relative health of the neighborhoods along Chicago’s L-stops. We wrote a MetroLine class that scrapes the web to produce an ordered list of MetroStop objects, links each MetroStop with geographic data from the Chicago Open Data Portal, and uses the MetroStop geographic data to use each stop to build shapefiles for the MetroLine that contain public health data from the City Health Dashboard and the American Communities Survey (ACS). We also calculated a health adversity score for each census tract along a MetroLine. We then wrote functions to visualize as many metros lines as a user wants, with a heat map of a single public health indicator in the census tracts under that line, compared to a demographic indicator. A text-based user interface allows the user to generate and save a plot with the health and demographic indicators they care about.

Overall Structure of the Software:

Python Version:

3.7.4

External libraries: Please run the following commands, in order:

```
-   python3.7 -m venv env
-   source env/bin/activate
-   sudo pip install --upgrade pip
-   sudo apt-get install libgeos++-dev
-   sudo apt-get install libproj-dev
-   sudo apt install python3-rtree
-   sudo pip3.7 install -r requirements.txt
```

To Run the Software: in terminal, run:

```
- python3.7 text_ui.py
```

NOTE: if it fails to run in the virtual environment, run in outside the virtual environment in the VM

NOTE: You may see this error message:

```
/usr/local/lib/python3.7/site-packages/geopandas/plotting.py:335:
UserWarning: The GeoSeries you are attempting to plot is empty. Nothing has
been displayed.
```

If so, the plot has still been created - this the automated warning for any NaN values, which we have coded gray in our plots.

Python Libraries: see requirements.txt

Files:

- **requirements.txt:** a txt file containing the requirements for the virtual environment
- **cta_classes.py:** a module containing the methods for constructing MetroLine and MetroStop classes, generating shapefiles for each MetroLine, and plotting the lines
- **cta_data_wrangle.py:** a module containing the functions to request CTA stop data through an API, and process that data
- **line_scraper.py:** a module for webscraping CTA line data for the ordered list of stops
- **health_wrangle.py:** a module for collecting City of Chicago public health indicator data from the City Health Portal API, and cleaning that data
- **census_data_wrangle.py:** a module for loading and cleaning a CSV with the demographic data for each census tract downloaded from the census website
- **acs_ph_combine.py:** a module that merges the public health and demographic dataframes, and uses indicators from each to calculate an aggregated health score, and an adversity index for each census tract
- **make_figures.py**: a module that plots the figures, based on the dataframes and shapefiles from the CTA classes
- **context.py:** a module that containing methods to construct and represent MapContext objects, which stores the necessary information to create a plot
- **text_ui.py:** a module that runs a text-based user interface, allowing the user to choose what MetroLines, public health indicator, and socioeconomic/demographic indicator to construct and plot a MapContext object.

Directory:

- **data:** a directory containing the ACS data downloaded from Census Bureau

Accomplishments and Further Opportunities

We accomplished our goal of visualizing public health indicators along CTA L-lines and comparing those indicators to various demographic metrics for each metro stop. We succeeded in building an interface through which a user can interact with the data and plot indicators they care about. Due to our plotting libraries, we were not able to plot the health adversity score as a categorical index, rather than continuous variable. We also have questions we were not able to answer visually, such as the line with the lowest health adversity index overall. Finally, while we were able to plot the healthiest metro stops on a given line under a certain budget, we were not able to program that functionality into the UI. Over spring break, we plan to conduct further analysis and build a more interactive UI.