Chicago Eats

Rebecca Connie (RConnie)
Bill Lennon (lennonwk)
Andrew Warfield (awarfield)

Project Overview:
Our project creates an interactive Chicago map allowing the user to select population demographic and food source location overlays for 2013, 2016 and 2019. We also have supplemental graphs and maps (in jupyter notebooks) that analyze the data by Chicago regions (North Side, West Side, Downtown, South Side, East Side).

We use Chicago Department of Public Health food inspection results from the Chicago Data Portal to compile all food source types and locations for the different year periods. Population demographic data is drawn from 2013, 2016, and 2019 American Community Surveys.

Group Member Responsibilities
Andrew: Cleaned and organized the ACS data (cenpy_fetcher.py). Created the program's command line interface (app.py). Created a layered visualization dashboard out of Folium and Panel with unified map classifications across years from our ACS and food sources data (viz.ipynb). Conducted some Point Pattern Analysis on data (PPA.ipynb). Refactored food data cleaning for code efficiency and readability (clean_data.py).

Bill: Responsible for the food_source_data_api.py file which calls the Chicago Data Portal API to download specific attributes of the Food Inspection dataset and writes to a csv. Responsible for install.sh and requirements.txt to create a virtual environment for running the app.

Rebecca: Responsible for cleaning the csv file returned from food_source_data_api.py. Wrote the script for creating a clean csv file (clean_data.py). Responsible for creating a jupyter notebook to create graphs representing the data by Chicago regions(zip_data_cleaning.py, zipcodeviz.ipynb).

User Interaction and Application Output
1. Navigate into the `chicagoeats` folder from the command line
2. Create the virtual environment: `source install.sh`
3. Enter the virtual environment: `source env/bin/activate`
4. Interact with the command line:
   a. To re-run the process of fetching/cleaning the data(found in the 'data' directory) for the project enter the following command: `python app.py --data`
   b. To launch interactive visualizations enter the following command: `python app.py --viz`. This should print out a localhost address. Navigate to the address in your browser.
5. User can choose between different population demographic attributes, years, and food types via the panel with dropdown menus on the right side of the map to display desired information.

To access supplemental visuals:
1. Run `python app.py --notebooks`. This will open Jupyter Notebooks in your browser.
2. Navigate from `chicagoeats` → `analytics` module.
3. Open `PPA.ipynb` and `zipcodeviz.ipynb` to explore.

Project Accomplishments
Created a layered visualization dashboard with the 3 drop down options: Population demographics, Year, and Food Source. Population demographics include the categories: median income, median home value, racial percentages and poverty percent.

The notebook 'PPA.ipynb' does a Point Pattern Analysis that is concerned with the visualization and statistical characterization of point patterns. In this analysis we visualize and characterize the patterns of food sources in Chicago.

The jupyter notebook 'zipcodeviz.ipynb' highlights the data based on Chicago regions. The bar graph displays total food sources for each Chicago region in the years 2013, 2016, and 2019. A note on analysis, the bar graph does not take into account the geographical or population differences of each region, meaning it is only showing the total food sources (i.e. comparing Chicago's North Side to East Side is skewed since the East Side is relatively smaller in population and geographical size).

The pie graphs in 'zipcodeviz.ipynb' focus on each region in 2019. A slight year change to the code will produce graphs for 2013 or 2016. These pie graphs display the percent of each type of food source category ('Restaurant', 'Convenience Store/Gas Station', 'Grocery Store', 'Liquor Store') in each region. Analysis note, due to changing food inspection guidelines around gas stations/convenience stores it appears data for that food source category was inconsistent or non-existent in almost all regions.

Software Structure:

- **proj-chicago-eats**
  - **README.md** Contains basic directions to run the project
  - **chicagoeats**
    - **__init__.py**
    - **__main__.py**
    - **install.sh** Creates a virtual environment for the user to interact with the software.
    - **requirements.txt** Contains all third party packages/libraries necessary to run the software. All are installed when the user creates the virtual environment.
    - **app.py** Contains the code to run the software from the command line.
    - **analytics**
      - **__init.py**
      - **PPA.ipynb**
      - **zipcodeviz.ipynb** Uses Matplotlib and Numpy packages to create bar graph and pie charts for data visualization based on Chicago regions.
    - **data**
      - **acs.parquet** Downloaded and processed American Community Survey data.
      - **'Boundaries – ZIP Codes.geojson'** Dataset containing Chicago ZIP code boundaries downloaded from the Chicago Data Portal.
      - **food_source_final.csv** Downloaded and processed food inspection data from the Chicago Data Portal API. The output of **clean_data.py**
      - **food_source_data_set.csv** Contains filtered columns of food inspection data from the Chicago Data Portal API.
    - **notes**
      - **proj-d1.md**

- **proj_d2.pdf**
- **proj-d2-feedback.md**
- **proj-paper.pdf**

data_processing

- **cenpy_fetch.py** Utilizes cenpy package to download and process desired American Community Survey data.
- **clean_data.py** Utilizes pandas and geopandas to process and clean original food inspection file from the Chicago Data Portal.
- **food_source_data_api.py** Utilizes Socrata package to access the Chicago Data Portal API and download filtered columns of Chicago Food Inspection data.
- **__init__.py**

visualizations

- **__init__.py**
- **viz.ipynb** Uses Panel package to create interface for data visualization.