Project Data Report

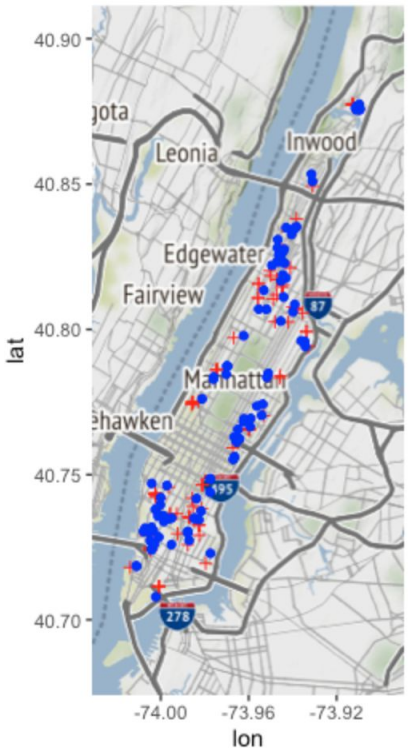
I started off with six datasets, all from the NYC Department of Finance and NYC Open Data:

NYC Property Rolling Sales [Manhattan]

- Description: This dataset provides information on all property sales from April 2019 to March 2020 in the borough of Manhattan. Variables include neighborhood, building type, square footage, and more. Since I'm focusing on home sales, I filter the building class category to keep only rows with "family dwellings" in their values.
- Variables to be included: After cleaning and wrangling the data, I kept all the variables. I then appended new columns, which came from geocoding the address variable, buffering the homes, and using the other datasets: **home_ID* (increasing unique indices for each row), **geometry* (polygons of the buffers), **lat*, **lon*, **num_schools*, **num_food*, **num_parks*, **num_subway*, **num_neighborhood_ord*. After feature selection, wherein I keep only the independent variables that are deemed important for my models, I am left with these variables: sale.price (dependent variable), residential.units, land.square.feet, gross.square.feet, *num_schools, *num_parks, *num_subway, *neighborhood_ord.
- Temporal resolution: Date sold (MM/DD/YY)
- Spatial resolution: Address, zip code, latitude, longitude
- File format: CSV
- Source:
    - *Rolling Sales Data*. May 17, 2020. Distributed by NYC Department of Finance.https://www1.nyc.gov/site/finance/taxes/property-rolling-sales-data.page.
- Visualization: I created a map of all the home sales in Manhattan with color as the sale price (yellow = lowest, red = highest)

NYC School Locations

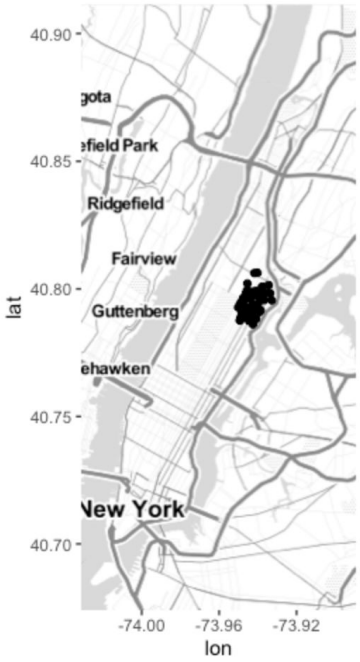
- Description: This dataset has information on all schools in NYC from 2019-2010.
- Variables to be included: When calculating the intersection between the schools and homes, I used only the 2 new columns that I created: *school_ID (increasing unique indices for each row) and *geometry(a re-projection of the longitude, latitude columns into a point). (Although I could have simply used the x and y coordinate variables, I wanted to practice re-projection). Once I calculated how many schools were present within each buffered home, I created a column of these values called **num_schools* and appended it to my home sales data frame.
- Temporal resolution: None
- Spatial resolution: Address, x coordinate, y coordinate, latitude, longitude
- File format: CSV
- Source:
    - *2019 - 2020 School Locations*. September 11, 2019. Distributed by NYC Open Data. https://data.cityofnewyork.us/Education/2019-2020-School-Locations/wg9x-4ke6.
- Visualization: Below is a map of only the schools and homes that intersect; homes are represented by the blue points and schools by the red crosses.



NYC Healthy Stores

- Description: This dataset is on bodegas and grocery stores that have participated in the Shop Healthy NYC program's Retail Challenge and were recognized by the Borough President's Office.
- Variables to be included: I combined this dataset with the NYC Farmers' Market dataset because a) there weren't enough data points for healthy stores alone, and b) I wanted to create one variable to describe healthy food vendors. From the healthy stores' dataset, I kept the name, address, latitude, and longitude columns when I merged it with the farmers' market data frame. In the farmers' market dataset description below I will go into which variables I used from the combined dataset.
- Spatial resolution: Address, zip code, latitude, longitude, council district, census tract
- File format: CSV
- Source:
    - *Recognized Shop Healthy Stores*. February 7, 2020. Distributed by NYC Open Data. https://data.cityofnewyork.us/Health/Recognized-Shop-Healthy-Stores/ud4g-9x9z.

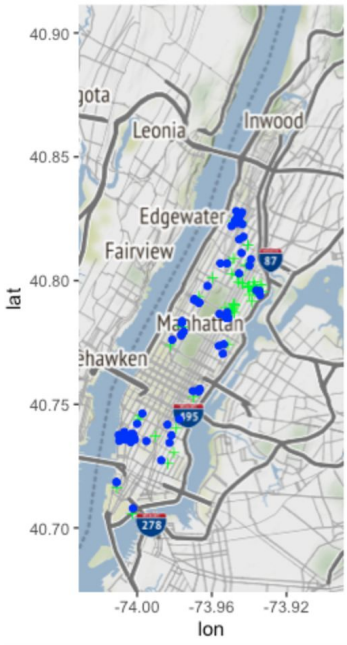
- Visualization: I created a map of all the healthy stores in Manhattan.



NYC Farmers' Markets

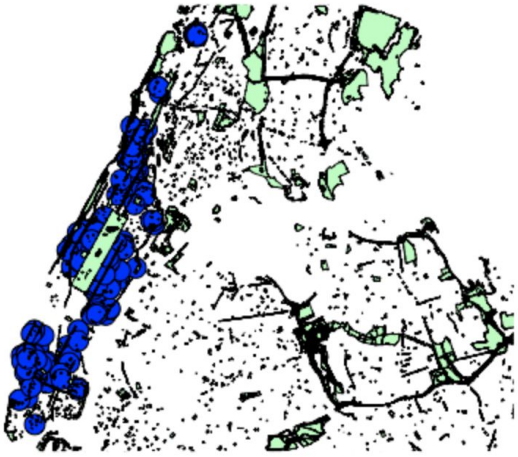
- Description: Each row in this dataset represents a farmers market in any of the five boroughs. It includes data on the vendors, products, EBT acceptance, and more.
- Variables to be included: As with the healthy stores' data frame, I kept the market name, address, latitude, and longitude columns when I merged it. When I calculated the intersections between the home buffers and these healthy food vendors, I used two new columns I created: *food_ID (increasing, unique indices for each row) and *geometry (a re-projection of the longitude, latitude columns into a point). Once I calculated how many healthy food vendors were within each buffer, I used those values for a new column called **num_food* that I appended to the home sales data frame.
- Temporal resolution: Hours of operation, season dates
- Spatial resolution: Address, latitude, longitude, location point (latitude and longitude as a pair)
- File format: CSV
- Source:
    - *DOHMH Farmers Markets*. February 7, 2020. Distributed by NYC Open Data. https://data.cityofnewyork.us/dataset/DOHMH-Farmers-Markets/8vwk-6iz2.

- Visualization: Below is a map of only the healthy food vendors (green crosses) and homes (blue points) that intersect with each other.



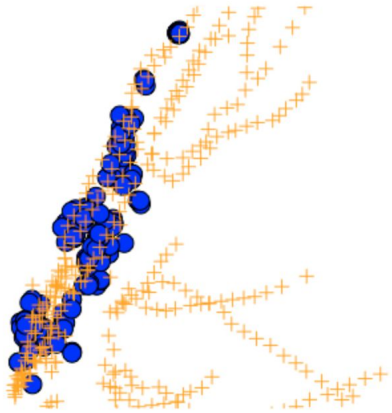
NYC Parks

- Description: This is a planimetric base map polygon layer of parks in NYC.
- Variables to be included: When calculating the intersections between the parks and home buffers, I used the geometry column (multipolygon) and a new column I created called *park_ID (increasing unique indices for each row). Once I calculated the number of parks within each buffer, I created a new column called **num_parks* with these values and appended it to the home sales data frame.
- Temporal resolution: None
- Spatial resolution: Shape length, shape area, geometry
- File format: GeoJSON
- Source:
    - *Open Space (Parks)*. September 10, 2018. Distributed by NYC Open Data. https://data.cityofnewyork.us/Recreation/Open-Space-Parks-/g84h-jbjm.
- Visualization: I created a map of all the home buffers (blue points) and parks (green geometries).



NYC Subway Stations

- Description: Each row of the dataset has information on the subway station name, line, and its point geometry.
- Variables to be included: When calculating the intersections between the subway stations and home buffers, I used the geometry column (point) and a new column I created called *subway_ID (increasing unique indices for each row). Once I calculated how many stations were within each buffer, I created a new column called **num_subway* with these values and appended it to the home sales data frame.
- Temporal resolution: None
- Spatial resolution: Geometry
- File format: GeoJSON
- Source:
    - *Subway Stations*. August 22, 2019. Distributed by NYC Open Data. https://data.cityofnewyork.us/Transportation/Subway-Stations/arq3-7z49.
- Visualization: Here is a plot of all the home buffers (blue points) and subway stations (orange crosses).

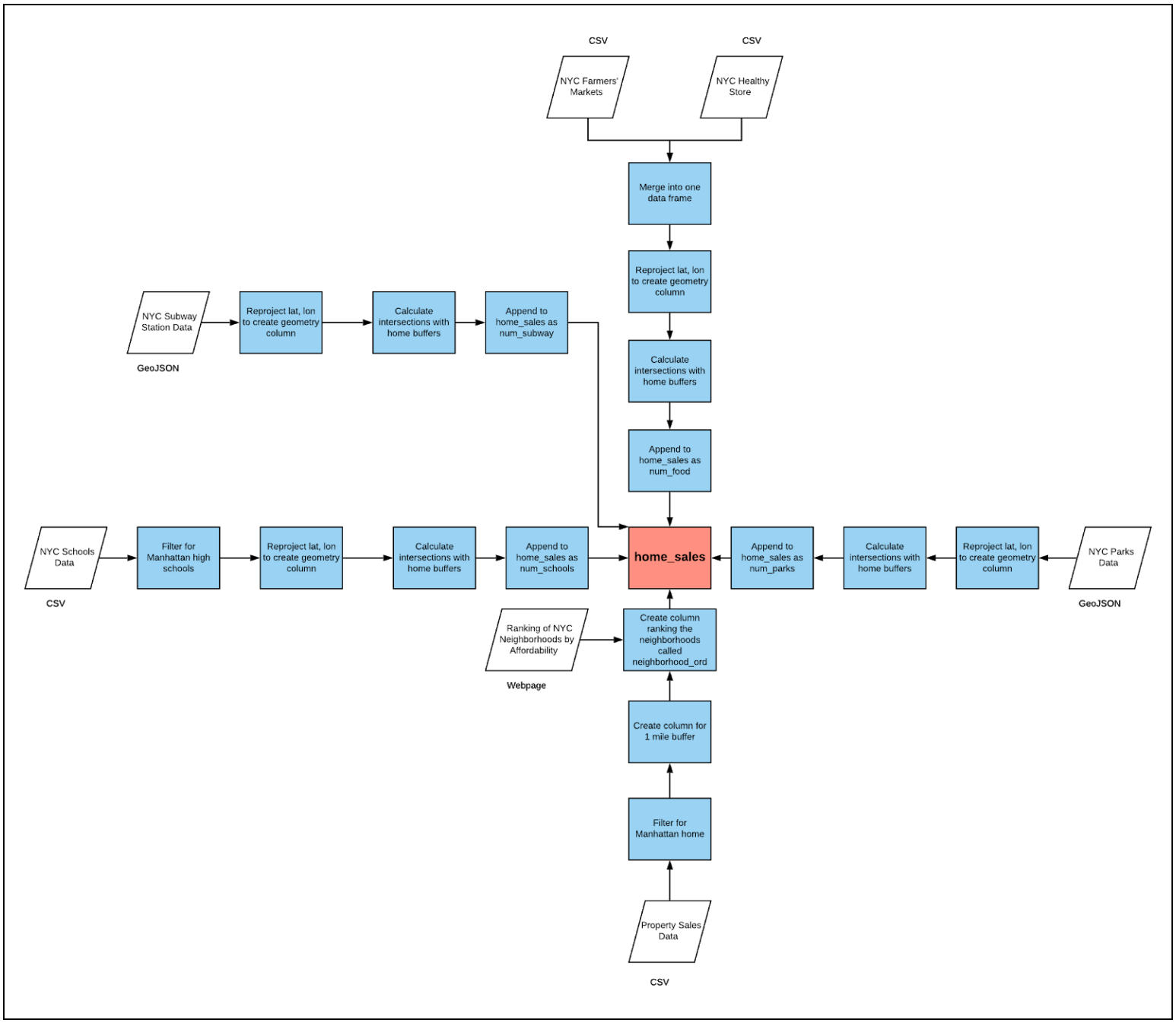


I created the **neighborhood_ord* column that I mentioned at the beginning by ranking each neighborhood according to an article on Curbed New York about New York's most and least affordable neighborhoods. The full citation is as follows:

Rosenberg, Zoe. "New York's Most and Least Affordable Neighborhoods." Curbed NY. Curbed NY, August 4, 2017. New York's most and least affordable neighborhoods.

As mentioned in the description of the homes sales dataset, I created two new columns--*lat, *lon--after geocoding the address variable. I also exported my output as a CSV called geocoded_addresses because I didn't want to run the geocoding chunk multiple times, as it took almost 5 minutes to load each time. It has three columns: query (the address), lat (latitude), lon (longitude). It can be viewed here: https://github.com/lcao21/GIS3_Final_Project/blob/master/Data/geocoded_addresses.csv

Lastly, I created a rough data and process diagram to visualize the relationships between all the data sources. I didn't include some small processes like omitting rows with NA values and creating ID columns but instead indicated the bigger processes like calculating the intersections between the homes and each new independent variable:


CSV
CSV
NYC Farmers' Markets
NYC Healthy Store
Merge into one data frame
Reproject lat, lon to create geometry column
Calculate intersections with home buffers
Append to home_sales as num_food
NYC Subway Station Data
Reproject lat, lon to create geometry column
Calculate intersections with home buffers
Append to home_sales as num_subway
GeoJSON
NYC Schools Data
Filter for Manhattan high schools
Reproject lat, lon to create geometry column
Calculate intersections with home buffers
Append to home_sales as num_schools
CSV
home_sales
Append to home_sales as num_parks
Calculate intersections with home buffers
Reproject lat, lon to create geometry column
NYC Parks Data
GeoJSON
Ranking of NYC Neighborhoods by Affordability
Webpage
Create column ranking the neighborhoods called neighborhood_ord
Create column for 1 mile buffer
Filter for Manhattan home
Property Sales Data
CSV