Using Neighborhood and Sea Level Data to Determine Flood-Safe Zones in the Greater Boston Area

---

Introduction/Narrative

Climate change remains one of the defining issues of the 21st century. With sea levels rising around the world, coastal cities must be prepared to tackle new challenges.

Boston is no exception. In late 2016, Boston experienced what is known as a 'king tide' - the highest point that the natural tide gets to during the course of the year. The tide in that particular year was surprisingly high; water came over the wharf at almost two feet higher than the yearly average. As sea levels around the world continue to rise, it becomes increasingly important to understand where potential flooding can become a problem and act accordingly ahead of time. In this project, we collect neighborhood census data and pair it with neighborhood geospatial polygon data, then test where that intersects geospatial data of estimated regions of coastal flooding at different sea levels. From this information, a severity index can be applied.

Our project aims to help city planners prepare for this scenario by identifying new locations outside the affected flood zone to build/move essential utilities such as electric stations or water pumps. We attempted to find locations that were close to large population centers so that they can service the greatest amount of people. By proactively and preemptively building in spots less affected by the increase in water levels, we can lessen the effect that climate change has on the city of Boston.

Datasets and APIs Used

Greater Boston Area Sea Level Projections Source: http://bostonopendata-boston.opendata.arcgis.com/ Description: This data, generated by a geographical information system and formatted in GeoJSON, describes the sea level rise for the area surrounding Boston and is split into two subsets - one for 5 foot sea level rise and another for 7.5 foot sea level rise.
Boston Neighborhood Polygon Data Source: https://data.cityofboston.gov/ Description: Data taken from the Boston Open Budget web application, formatted in GeoJSON, which describes the shape and general geographic information of all the neighborhoods in Boston.
Cambridge Neighborhood Polygon Data Source: https://data.cambridgema.gov/ Description: Data from the Cambridge Open Data Portal, formatted in GeoJSON, which describes the shape and general geographic information of all the neighborhoods in Cambridge.
Cambridge Census Information Source: https://data.cambridgema.gov/ Description: Data from the Cambridge Open Data Portal, formatted in JSON, which outlines all relevant information from the 2010 census for the Cambridge-area neighborhoods.
Boston Neighborhood Census Information Source: http://datamechanics.io/

Description: Data from various online sources which we formatted into JSON such that it follows the structure of the Cambridge Census Information. Describes relevant census data for Boston neighborhoods around 2010.

Project Dependencies

This project requires no explicit API keys in order to collect information as most requests are simple JSON static content requests. Using a Socrata API token can help with throttling issues, but these are negligible.

There are two third-party Python library dependencies, the first of which is Shapely. This library is used to compare geometric information, which we use to check intersecting geospatial polygons in the context of this project. Shapely can be installed through pip, though it may have one additional dependency on 'libgeos' - both of these are easily accessible through most OS package managers. For further information, please view the documentation found here:

https://pypi.python.org/pypi/Shapely

The second dependency is NumPy, which is a package designed to assist in scientific computations. It offers efficient and convenient functions and data structures for manipulating large amounts of data. We use NumPy for running a weighted probability function needed in our K-Means++ function to select initial seeds for K-Means. For further information, please view the documentation found here:

http://www.numpy.org

Algorithms, Tools and Techniques

Algorithms and Heuristics

K-Means: K-Means is a clustering algorithm that calculates a set of “k” means that find the best localized means for a set of points.

K-Means++: K-Means++ is an algorithm that helps generate good seeds to be used in conjunction with K-Means.

Severity Index: A heuristic we developed to assign varying weights to different neighborhoods that represent how vulnerable they are to increased water levels. The severity index for a given neighborhood is as follows:

Severity Index = ((*estPeopleAffectedFive* * 2) + (*estPeopleAffectedSeven* - *estPeopleAffectedFive*)) / 1000

estPeopleAffectedFive and *estPeopleAffectedSeven* are calculated using the percentage of the neighborhood area that is flooded at five feet and seven feet, respectively, and scaled by the overall population of the area.

We weigh *estPeopleAffectedFive* higher because these people represent the people that will be affected by the flooding the earliest. They are also more vulnerable to flash floods and natural disasters, so we wanted the severity index to reflect that. We also did not want to double count the people described by *estPeopleAffectedSeven*, so we subtract the people already accounted for in *estPeopleAffectedFive*. Lastly, we divide by a thousand to make the resulting severity index easier to work with (on a more reasonable scale).

This scoring mechanism was especially vital in weighting the K-means clustering algorithm in regards to considering the different neighborhoods. This will be later explained in the technique section.

Transformations

Transformation #1 (Neighborhood Synchronization)

The first transformation consists of a simple unionization of the two neighborhood census data sets into one set. This requires a few projections in the last step, as the Boston census data does not explicitly match the Cambridge census data. We then use this set in applying the second transformation.

Transformation #2 (Neighborhood Polygon and Centerpoint Data)

In the second transformation, we compare the master list of neighborhoods from the first transformation against the polygon data in both the Boston and Cambridge geospatial information; this allows us to correlate a set of coordinates to a neighborhood name and population. In addition, we add the calculated center point of the neighborhood into the dataset, as such a value could come in handy later if applying a k-means sort of algorithm proves useful.

Transformation #3 (Calculating Flood Ratios for 5 and 7.5 Feet Rise)

Finally, we use the Shapely python library to check for intersections between the neighborhood polygons and the sea level information. By comparing the intersection area to the size of the neighborhood, we can determine a ratio of flooded space and estimate how much of the population is affected. From there a severity index can be calculated for that neighborhood, which could later on be used in calculating a weighted k-means clustering algorithm which is biased towards area with large impact on humans. Note that this transformation could take a while (since it applies an expensive operation), but should take no longer than 5 minutes on most processors.

Transformation #4 (K-Means)

In the fourth transformation, we run K-Means on the neighborhoods, using the centerpoints of each neighborhood. This ensures that all the neighborhoods have a location that is relatively close. We use a seeding algorithm for K-Means known as K-Means++, which uses a weighted random walk over the original neighborhood center points to find ideal seeds for K-Means.

Transformation #5 (Calculating Safe Point)

In the fifth transformation, we take the locations from the fourth transformation and attempt to find safe points outside of the flood zone. We iterate over the means from the fourth transformation, the proposed safe points. Because we converted the flood zone into a multi-polygon, we check the proposed safe points by comparing each of the points to each of the polygons that make up the flood zone. If the mean is already safe, i.e not in the flood zone, we do nothing. If it is in

the flood zone, we perform a calculation to find the closest point from that mean to the safe zone.

Techniques

Weighting K-Means: When generating the set of points we would run K-Means against, we made use of the severity index of each neighborhood. We converted the severity indexes into an integer, and a corresponding number of points to represent that neighborhood in the set. Neighborhoods that had a high severity index would have many points in the set. Neighborhoods that are unaffected by the flooding, and thus have a severity index of zero, added no points to the set.This was to ensure that the means were clustered around only the most heavily affected areas.

Issues

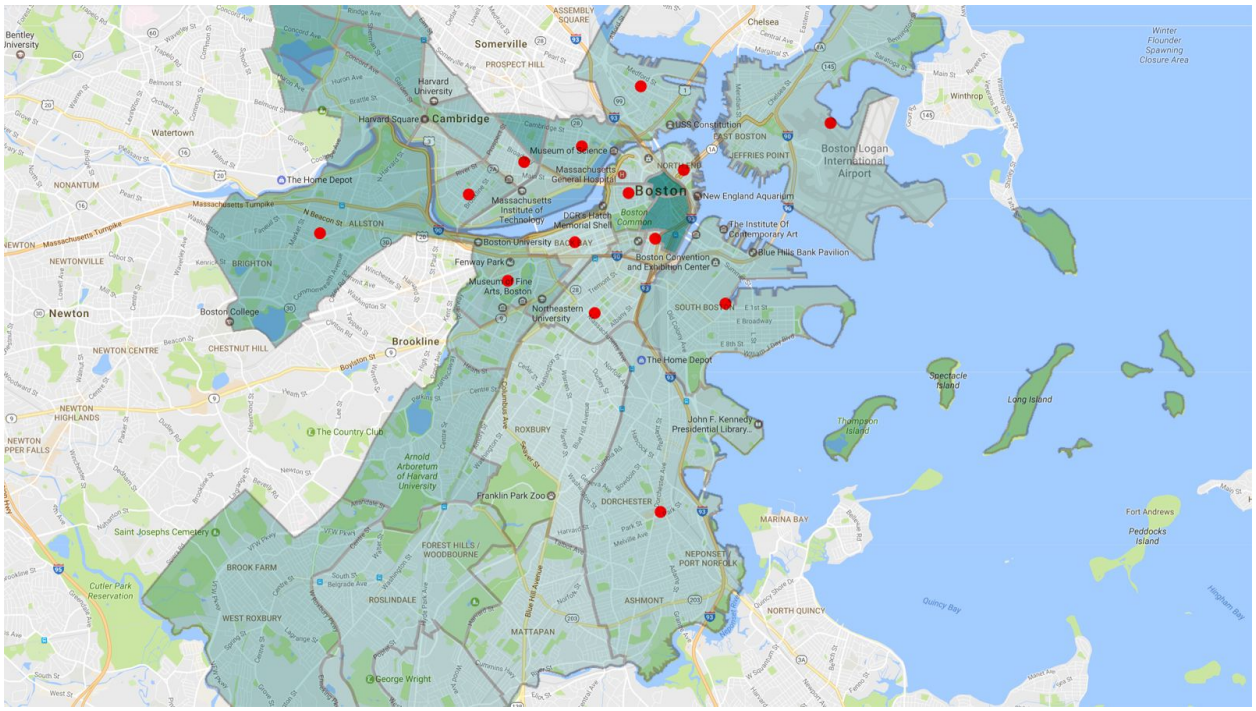
We had difficulty using this algorithm at first, as it would always return one mean, regardless of the number of means and the values of means we started with. To counter this, we used K-Means++ as a method of seeding the initial mean values.

Another interesting issue we encountered was when we were asked to implement a “trial” mode. We reduced the number of neighborhoods, which in turn reduced the number of data points being fed into K-Means. This lead to K-Means failing, as it attempted to find more means than there were data points (i.e find 5 means, when there were only 3 neighborhoods). This is working as intended, but limits the functionality of trial mode for our program.

Trial mode continued to be a feature that did not really suit the nature of our project. One of the operations we run - wherein we calculate the intersection of the sea level polygon information with the neighborhood polygon information - is computationally expensive and takes several minutes to run. Again, it is running as intended, but the time constraints of the trial mode are difficult for our project to meet without losing functionality.

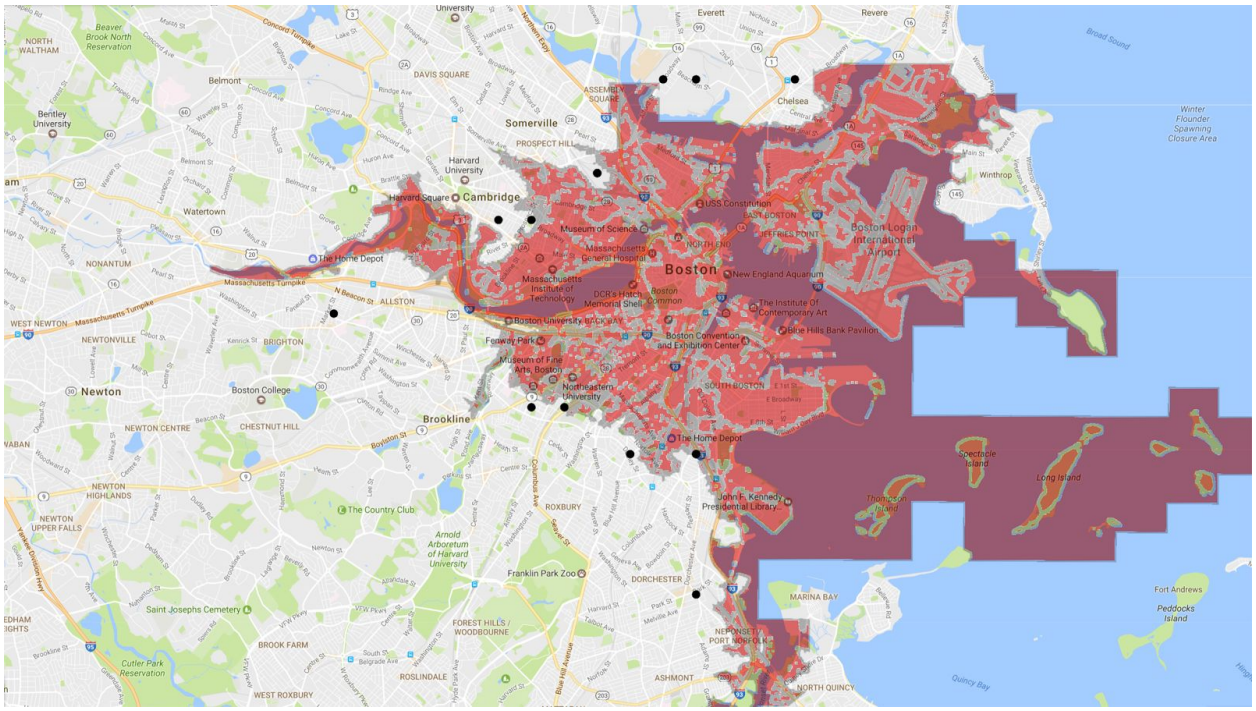
Results and Conclusion

Weighted K-Means of Flooded Population



The K-Means that we calculated, which represented the geographical averages of the flooded neighborhoods weighted by the severity index, mostly feel in the center of their respective neighborhood. This makes sense, considering the technique we used to seed the set of points. Financial Center and Government Center were the only neighborhoods in the floodzone that did not have their own means (instead, their closest means were in the adjacent neighborhoods). This makes sense, as these districts have a low population, and thus a low severity index.

Public Service Points Outside the Floodzone



The service points that were generated were excellent. They all fell outside of the 7.5 Feet Sea Level rise zone, and were an areas suitable for construction. We noticed that the area north of Boston was particularly safe, with 3 of our 14 points clustered there.

Future Development

If we had more time to fine-tune our project, we would have focused our efforts on doing K-Means with more precision. Our K-means that we calculated tended to fall in the center of the respective neighborhoods that they fell in. This is because of the way we added points while running K-Means. In addition, we could have increased the accuracy by using the specific polygons of the neighborhoods and their intersection with the Sea Level rise information. In this way, we could have generated points that better represented the people affected by the rise in sea level.

For our project to remain relevant and useful, we would need more methods of fine-tuning the generation of the K-Means. To do this, we would use more precise

data of where people in a neighborhood live. This would allow us to give more weight to parts of the neighborhood that are more densely populated.

With the data we created, city officials could proactively build critical infrastructure and services in locations that would be less affected by the rising sea levels, as well as plan out evacuation points in an event of an emergency. There's only so much a city can do to combat climate change on its own, but it can definitely brace for its consequences.

Greater Boston Area Sea Level Projections Source: http://bostonopendata-boston.opendata.arcgis.com/
Boston Neighborhood Polygon Data Source: https://data.cityofboston.gov/
Cambridge Neighborhood Polygon Data Source: https://data.cambridgema.gov/
Cambridge Census Information Source: https://data.cambridgema.gov/
Boston Neighborhood Census Information Source: http://datamechanics.io/