Optimizing Police Dispatch Locations

Thomas Hervieu, Jason Kolbush, Chris Kwan

Introduction:

The city of Baltimore has one of the highest crime rates in the country. It is well above the national average with an estimated 344 homicides in 2015. About 0.4% of the entire population is directly involved or impacted by a violent crime every year. Many crimes seem to occur in 'drug clusters' located around the city, which tells us that crime is unevenly distributed throughout Baltimore's area. We wonder if there is a way to shorten police response time to an urgent call, where even seconds matter. Thus, the goal of our project is to determine the best dispatch locations for police officers to be in order to arrive to the crime scene as quickly as possible. We will also find unique locations for daytime (6am-6pm) and nighttime (6pm-6am) Ideally, we are increasing the number of patrol cars in high-crime areas while keeping other areas secure.

Data Set: https://catalog.data.gov/dataset/calls-for-service

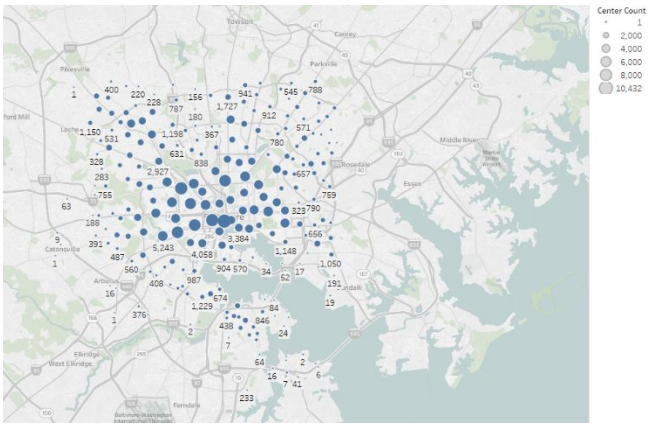
Through data.gov, we could find a "911 Calls for Service" from data.baltimorecity.gov, which contains many different data collections for the city of Baltimore. This dataset contains 2016 data for every 911 call made prior to its location (i.e. latitude and longitude coordinates). The data set is contained in a CSV file containing the date/time, priority, description, and location in three different forms: district, street #, and longitude/latitude coordinates. We filtered out the "Non-Emergency" and "Low" Priority data points and separated our data into 2 files, 'Day.csv' and 'Night.csv', each containing over 300,000 records.

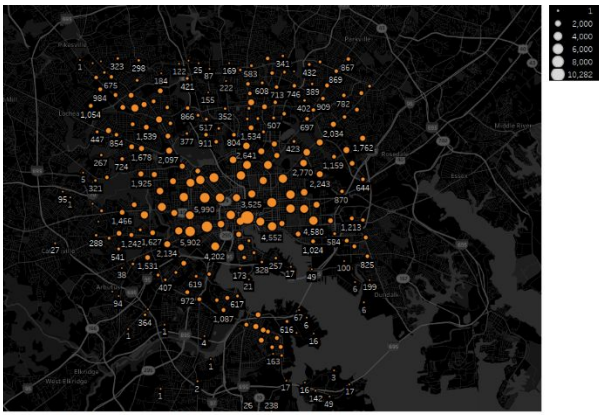
Machine Learning Problem:

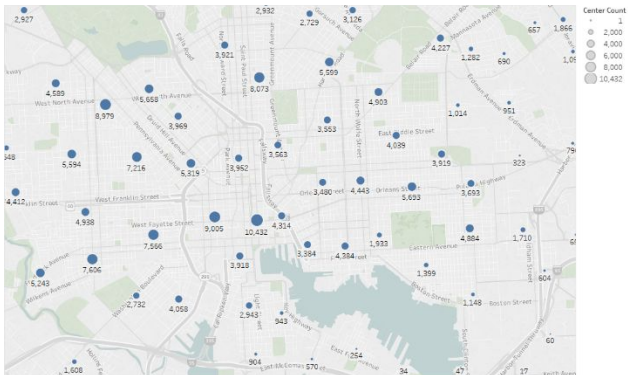
This is a K-Clustering problem. We displayed the ideal dispatch locations per our analysis of the 2016 data on a map of Baltimore with nodes. The optimal dispatch locations are represented by the cluster centers that we created. We will generate two maps for both nighttime/daytime hours and will adjust the size of the dots per the number of data points, or emergency calls, assigned to each cluster center.

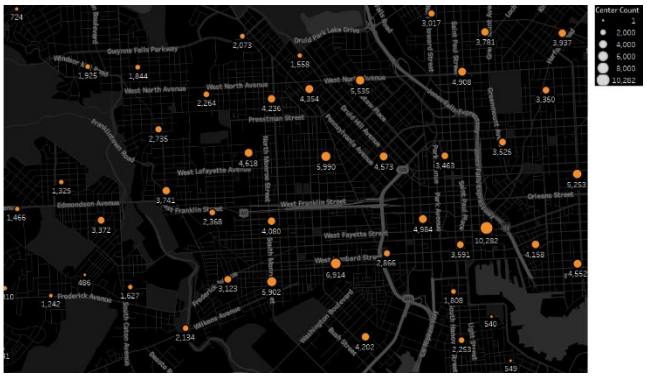
Algorithm/Method/Result:

We began by writing a python function (split.py) that took in our original dataset, eliminated low-priority calls, and split the dataset into day and night records. Then, we applied the K-Means Algorithm (project2.py) to both datasets using our own implementation of K-Means with 300 cluster centers. Since dispatch units are driving cars on the streets, we can view the path of travel as a grid. Therefore, we used Manhattan distance when allocating each data point to its designated cluster. We ran the algorithm until convergence for both datasets. We also confirmed that our python code printed the latitude/longitude coordinates and size of each cluster to 'DayCenters.csv' and 'NightCenters.csv'. We eliminated all the empty cluster in excel and ended up with 228 clusters in 'DayCenters.csv' and 239 clusters in 'NightCenters.csv'. Then, we imported both files into Tableau to visualize our result (DayLocations.twb, NightLocations.twb):







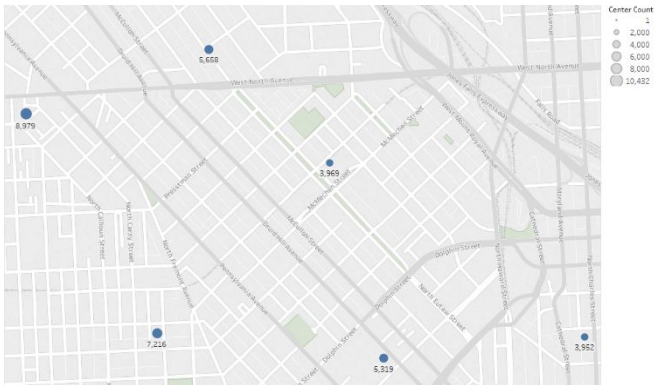


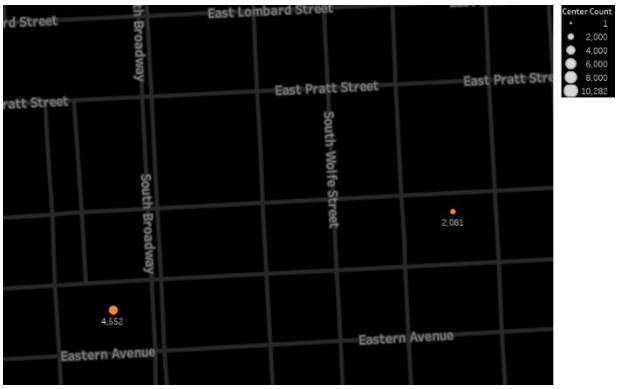
Analysis/Discussion/Challenges

The visualizations we have obtained show us where 220+ dispatch cars should be located in order to maximize the average response time to high-priority calls. Here, we see as expected that cluster centers in the populated downtown area are larger than those on the perimeter. However, we are still able to identify certain areas with lower crime rates inside the perimeter. Judging from the density and size of the dots, we are able to decipher between the 'good' and 'bad' neighborhoods. This information could be very useful for individuals looking to move to Baltimore.

Throughout our project, we encountered a few challenges. Cluster centers formed over bodies of water. To solve this, we eliminated all of the empty clusters in our output file. The reason for the high number of empty clusters (~70) was due to the boundary we defined for our latitude/longitude coordinates. Since it would be difficult and tedious to 'draw' boundaries for the city of Baltimore, we decided to bound all the coordinates within a rectangle, surrounding the city. Therefore, our cluster centers do not align directly with the jurisdiction of the Baltimore Police Department. However, we used the maximum and minimum coordinates to fully cover the city of Baltimore. Moreover, manhattan distance, while practical, was not the theoretically best

distance to use. Taking the two examples below, we see that the ‘nighttime’ map on the right displays a grid-like street layout. Manhattan distance is optimal for calculating this layout. However, the ‘daytime’ map on the right contains streets running diagonally, where a combination of Euclidean and Manhattan distance should be used to calculate the shortest distance.





In addition to distance, despite police cars having the right of way, traffic can still affect response times to emergency calls. While it would an endeavour beyond the scope of this course, we could make this project much more elaborate my incorporating elements accessible via Google API including real-time traffic reports and accounting for each angle and connectedness of the streets of Baltimore. Testing our data would require us to compare average distance travelled by police responding to a call in 2016, and comparing it to the average distance we obtained (i.e. average all of the manhattan distances of all the points to their respective cluster centers).

Ultimately, our project was successful in localizing areas of crimes. While we could create more cluster centers, we can use the size of each cluster to indicate the number of police cars that should be roaming the area. For example, clusters of size 5,000+ imply more than 13 high-priority calls occurring in a certain location, daily. Therefore, we would expect more than one dispatch unit to be present in the area. Unintentionally, the maps above also serves to aid incoming residents to decipher between ‘good’ and ‘bad’ neighborhoods by avoiding areas with many or large centers.