Haoxin Luo and Isabella Siu

NYC Stop and Frisk Visualization

Introduction:

For our research project, we want to analyze Stop and Frisk data for interesting statistics and correlations. We want to look at aspects like the gender and race of people stopped, how successful Stop and Frisk is at finding contraband, and how often force is applied by officers. We suspect that race will be a standout factor in who gets stopped and that the police may heavily target areas based on racial factors. As the effectiveness and the possible racial profiling involved in Stop and Frisk is a hot topic nowadays, our visualization is aimed at anyone from politicians to the general public who may care about this issue.

Our research questions were: “Are some races stopped disproportionately often? How often do stops translate into arrests? And how do the average income and predominant race of an area correlate to the people stopped there?” We hypothesized that some races would, in fact be stopped more often, particularly black people. We also suspected that the stops to arrests ratio would probably different amongst the different racial groups, but disagreed on how we thought things would swing. We believed that lower income neighborhoods and ones that are less white would be likely to have more stop and frisks.

Research:

As Stop and Frisks have been a hot topic for a while, it’s not surprising that there are several visualizations already existing that try to display its data.

One of the most well know visualizations for this is “All the Stops” (Rhiel), which represents each 1000 stops as a dot that, as the screen changes from representing race, to borough stopped in, to whether or not the suspect was armed, and so on. It’s a very powerful and effective visualization, that gives a good sense of the different quantities and gives a good sense of its subjects as people, as the dots rush to their next group between selections. While it’s an impressive visualization, it didn’t cover some of the big aspects we were interested in, like how the statistics of the different neighborhoods and locations corresponded and changed with the stop data.

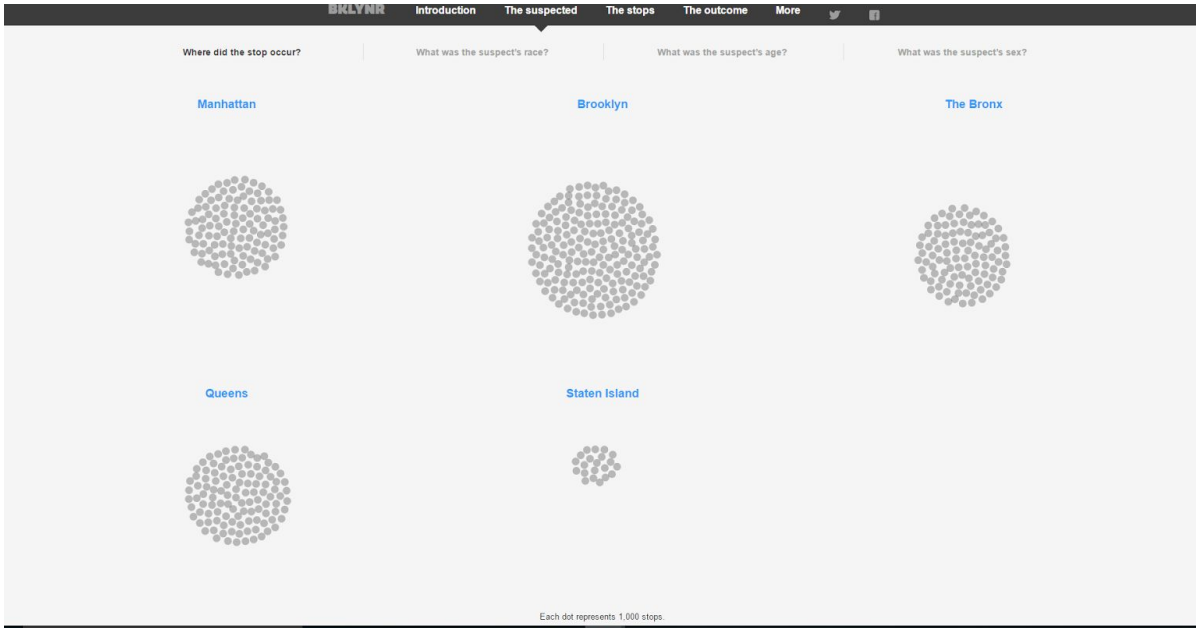

Figure 1. While a powerful visualization, “All the Stops” does not get nearly as exact with the Stop and Frisk locations as we wanted ours to get.

Another visualization for this is (Keefe) which shows the places where stops occurred as a choropleth with markers for where guns were found. The point of their visualization was that the locations where guns were found were not the same as the places with a high number of stops. While effective for its purpose, it displays a very different aspect of the data than what we’re interested in. However, it does contain an idea we would later employ, which is a choropleth as base layer to be compared with something shown on a different one.

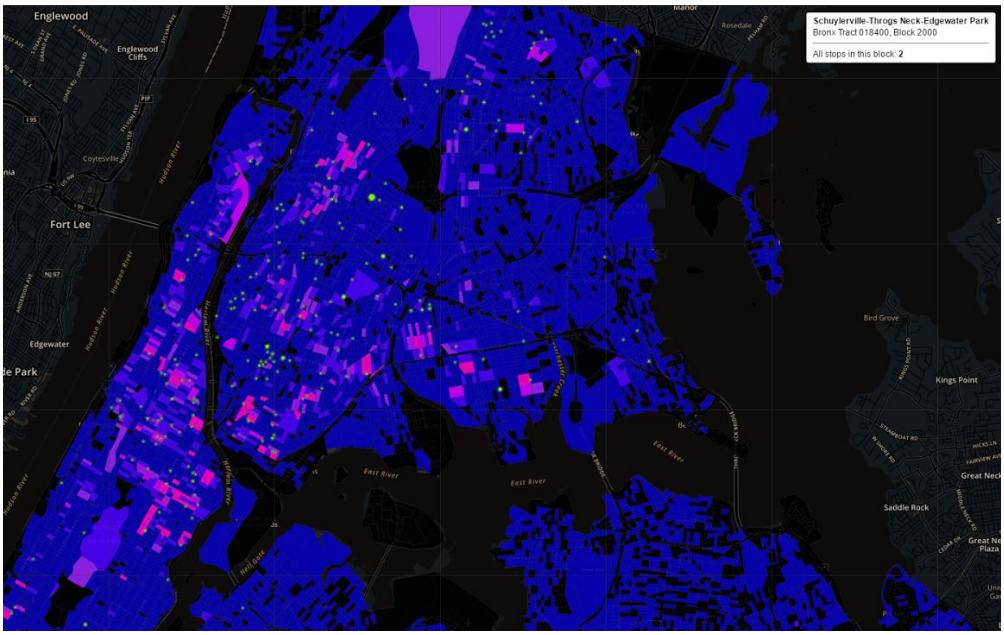

Figure 2. “Stop & Frisk | Guns” shows stops as a heat map and locations where guns were found on the suspects as green markers, which though very informative, does not get into the race versus neighborhood data we were looking into.

One thing that was incredibly important for us to be able to create this project was a mapping API. While we initially considered using Google Maps, what we ended up using was leaflet.js, a lightweight map API (Agafonkin). It was incredibly important in getting our visualization off the ground, as we depended heavily upon its circle and polygon making capabilities initially, and even in our final project we use the layers functionality in order to display different aspects of the data. We also depended fairly heavily upon and its examples in order to create our visualization.

For our visualization, we used two data sources, the publicly released data for stop-and-frisk from the NYC.gov site, and the income data that we got from the US census reports. The first was obtained fairly easily, as it is released on the NYC police department site as a somewhat poorly formatted csv (nyc.gov). We had an issue where sometimes the data ended up offset either because of someone skipping a column (of 120 columns) or they separated a list entry by commas in what became a csv. We tried to write scripts for this, but gave up and just fixed the bad rows by hand. The second was surprisingly difficult to find as the census site did not have it compiled into an easily usable file. There is a search tool that allowed us to search for specific information from the census database, but each search brought up hundreds if not thousands of results, each different from the other in small but important ways. The columns of each file were also named rather cryptically, and led to us abandoning this path. After a more in depth search, we found an API that provided almost exactly the fields we need and returned a geoJson shape file along with the request (CitySDK). This allowed us to easily obtain both the data (income and race composition) and the geographical shape of each region.

Design Evolution:

The ideal final visualization should allow the user to quickly disprove or confirm both the (hypothesised) racial bias of law enforcement's search and frisk targeting practices, and the image that certain racial groups are perhaps more inclined toward criminal activity.

This concept was initially pitched as a streamgraph showing the data and its correlations. The feedback that we received from our classmates mentioned that handling the large data set could present its own issues, and that highlights were important. The idea of looking at the location data was also pitched, and it was recommended that we be careful of how we display the data. This is a very sensitive issue after all, and something simpler may be clearer. From their feedback, we concluded that something map based would probably be much more effective for our visualization than a streamgraph would be.

Now that we changed to a map based visualization, it became incredibly important for us to have a consistent variable for finding location. The data we had stored the information as one of three things: a proper address, an intersection of two streets, or a street and the two cross streets framing the section where the stop had occurred. Naturally, this was all stored in different columns. This lead to us creating a Python script in order to parse the data and while we looked very hard for an API to convert addresses to coordinates without a cap, we ended up just using the Google Maps Geocoding API (Google). We resorted to using eight keys to convert 2,500 addresses each from our collection of 22,000 stops over multiple days. This was helped by consolidating the stops by each location, so no location would be searched twice. This lead to their being fewer stops than we initially had.

We edited our concept somewhat for our proposal after the initial feedback. We then decided that ideally the very first view that greets the user should be the zoomed out map of New York City, with the default filter that would show the law enforcements' racial bias, or the lack thereof. The user would then be able to zoom and pan around the city to look at specific neighborhoods of interest. The data would update resolution as the user zooms in, either dynamically or algorithmically or just as a result of less crowding of data. There would also be a selection of characteristics other than race that the user can chose to be on the map. Preferably, multiple characteristics could be selected at the same time and accurately compared on the same map. In addition, the user would be able to select areas of interest (i.e. by zip code, street, radius, etc.) and choose to see a more traditional chart comparison of the selected data dimensions.

While actually producing the visualization, we quickly realized that representing the race and stop and frisk data alone on the map would be difficult. The first visualization we produced represented the stop and frisk data as circles on the map, with the area of the circle corresponding to the number of stops and the color corresponding to the race that was stopped there the most. This was not the best visualization, as it runs into the first-past-the-post problem where if something is second place for everything it will still look like it lost if something else was first half the time but last the other half. This visualization could lead to incorrect conclusions and be very misleading, so something had to change.

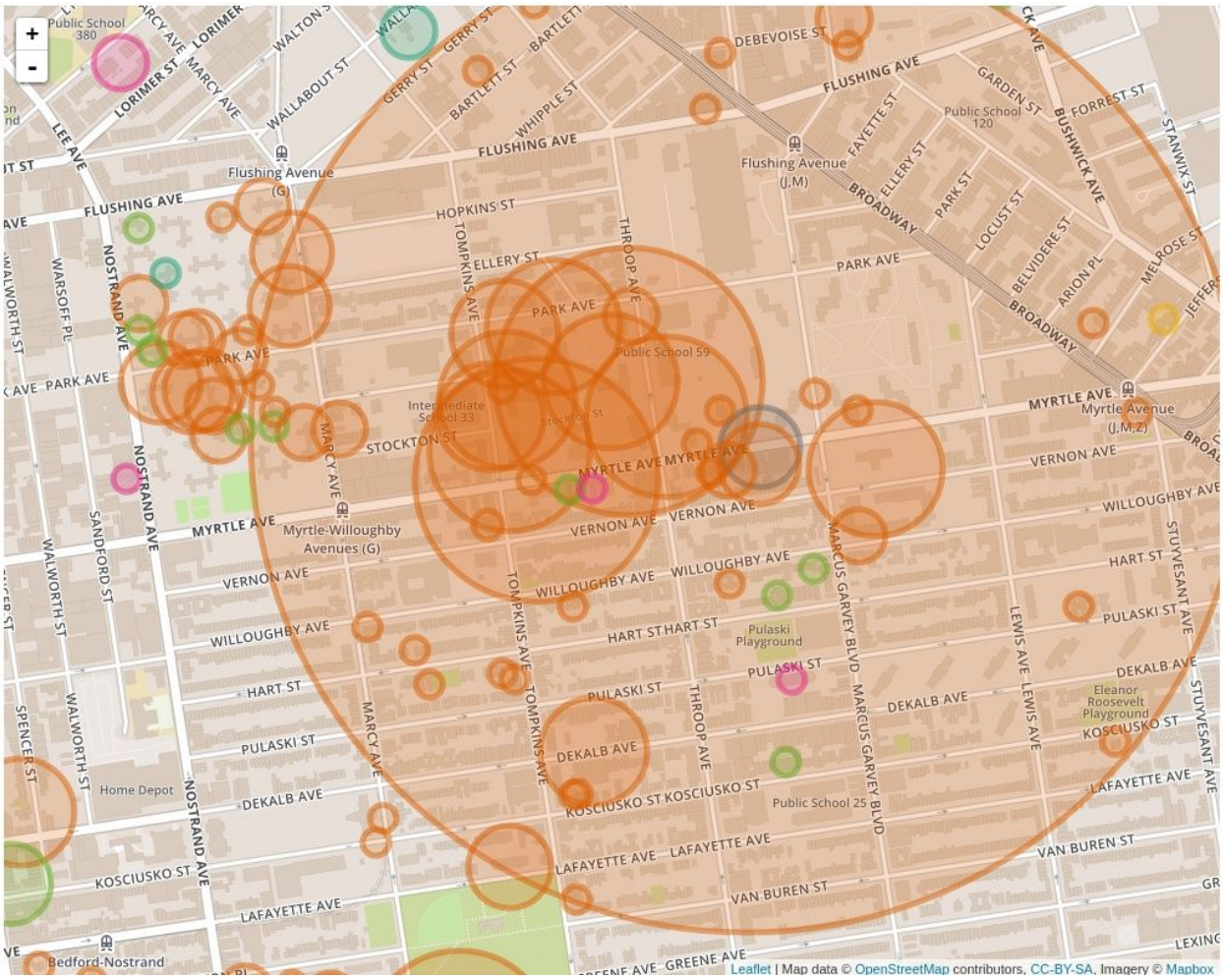

Figure 3. Our first visualization of the project. While it shows the data, it could be misleading as the circles are colored by whichever race was stopped the most, and thus could be hiding how much another race was stopped.

What we ended up deciding to do was for each of us to take one of two forking paths. Isabella worked on representing the different races at a point as concentric circles sized based on quantity, while Haoxin worked on creating a pie chart of who was stopped at each individual point using, initially, leaflet's Semicircle extension. We decided that we definitely wanted to be able to filter for different races no matter what approach we chose, so Isabella setup the filters that we would need, then Haoxin adjusted his pie charts to change based on which subgroups were included.

While we initially joked about using pie charts on top of a map, we found that our classmates did actually like them a lot, so we decided to instead base our visualization on that. Two important issues that we had to take care of was our lack of a legend, which Isabella applied, and the overlapping circles being confusing, so Haoxin did research on trying to merge the pie charts together. A large issue there was when pie charts were merged with the ones they touched they became larger and then touched more that they had to be merged with. Another being, the only merge library Haoxin found, MarkerCluster, and the pie chart library were incompatible.

Feedback:

When we presented the concentric circle (Isabella) and pie chart (Haoxin) versions of our visualization to the class, we got a lot of feedback, some of which I mentioned in our design evolution section.

The main issue with the concentric circle version was that it combines the lack of precision in area representations with even more overlapping circles than other methods. After giving a general explanation of how it worked, Isabella asked whether or not the demonstrated setup was intuitive and/or clear. Many people mentioned that they had initially thought the circles were showing area of influence, not stop number. They also felt that the parts where circles overlap were fairly confusing, though hard to fix. Some suggested perhaps using points with saturation or intensity where you could click on the points for specific racial breakdown at that stop instead. One person suggested leaving the sized circle filters as a separate option, with the other suggestion as the main. Also, they felt that the colors in the map did not help legibility.

The largest complaint about the pie chart visualization from the class class was not about the pie charts, as we expected; but about the overlapping pie charts (circles), which we did somewhat expect. Two of the people claimed they thought the pie charts represented the area that the pie physically covered rather than the center point of the pie. These two people also thought the small pies were showing a subset of the large overlapping pie. This is obviously a major misinterpretation. The fact that each race is represented by their slice of the pie with all the filters on had both positive and negative feedback. Some thought it enabled better understanding of whether the race in question is the sole contributor in the region or not, while others thought it was just weird and inconsistent depending on how many filters were selected at once.

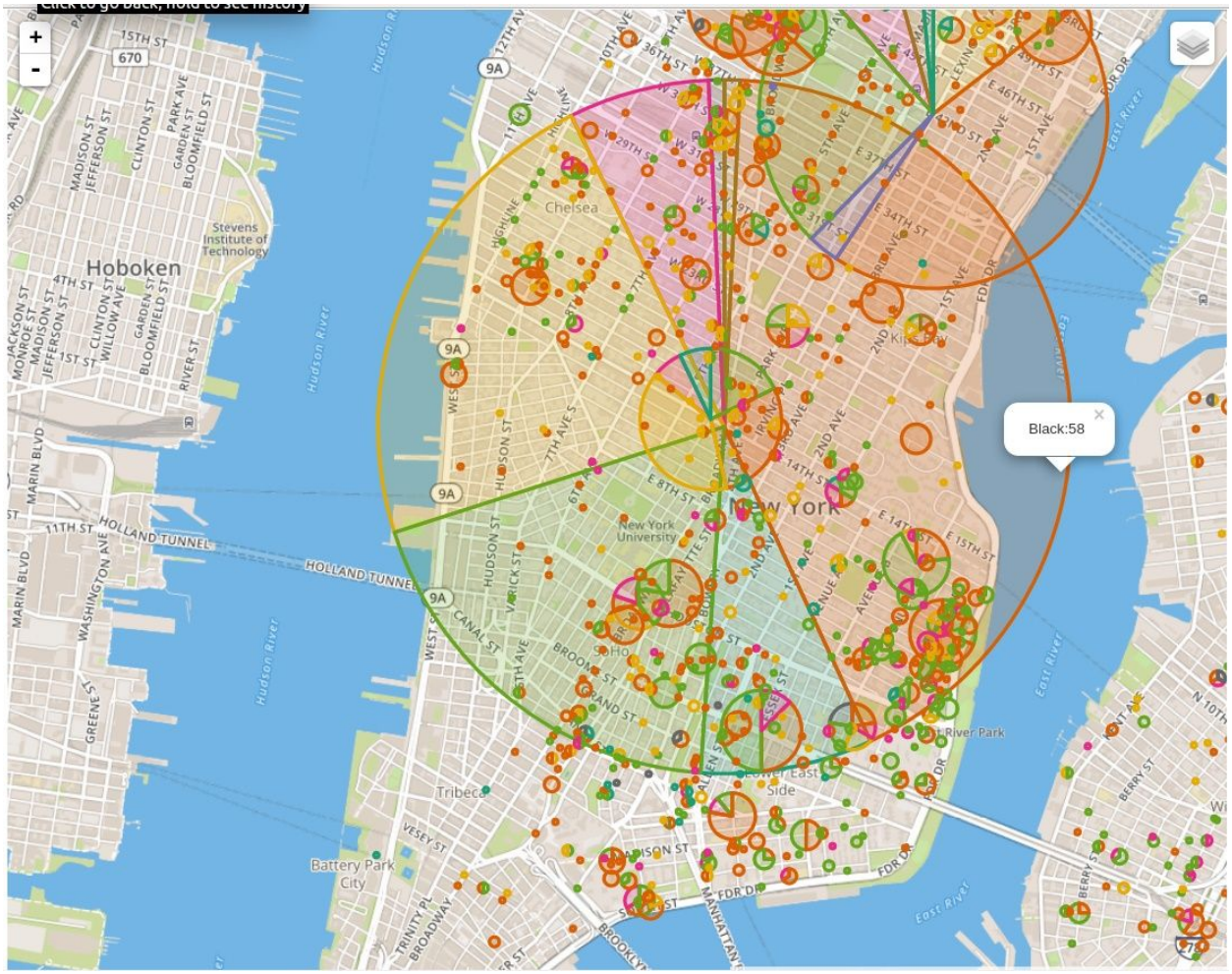

Figure 4. A look at Downtown Manhattan with all 22 thousand incidents displayed via pie charts. The radius of the pie represents how many total incidents occurred at the location in the center of the pie. Each slice of the pie is a different color that represents a different race that was stopped at the location.

In the end, the big takeaway from our class feedback was that the overlap was really terrible. Also, while pie charts aren't very good in general, they are more effective than one would think they would be for this visualization.

Design Evolution (continued):

In the end, we decided that using size to show quantity was going to be more of a hassle than it was worth, and removed that from the visualization. That didn't change the fact that our big priority became finding some way to stop them from overlapping so much. After Haoxin's many difficulties in trying to combine the MarkerCluster library's dynamic merging with the Semicircle library's support for pie charts, he eventually found sample code for generating merging pie charts on a leaflet map with d3js (Romstad). While it showed the number represented by each pie inside the circle, there was no other obvious visual cue. The MarkerCluster library also makes it so that when you hover over a pie chart a leaflet polygon appears that shows the area covered by that particular pie chart. It does do something odd

visually where the polygons covering the sections sometimes seem to overlap, but we haven't found a solution yet

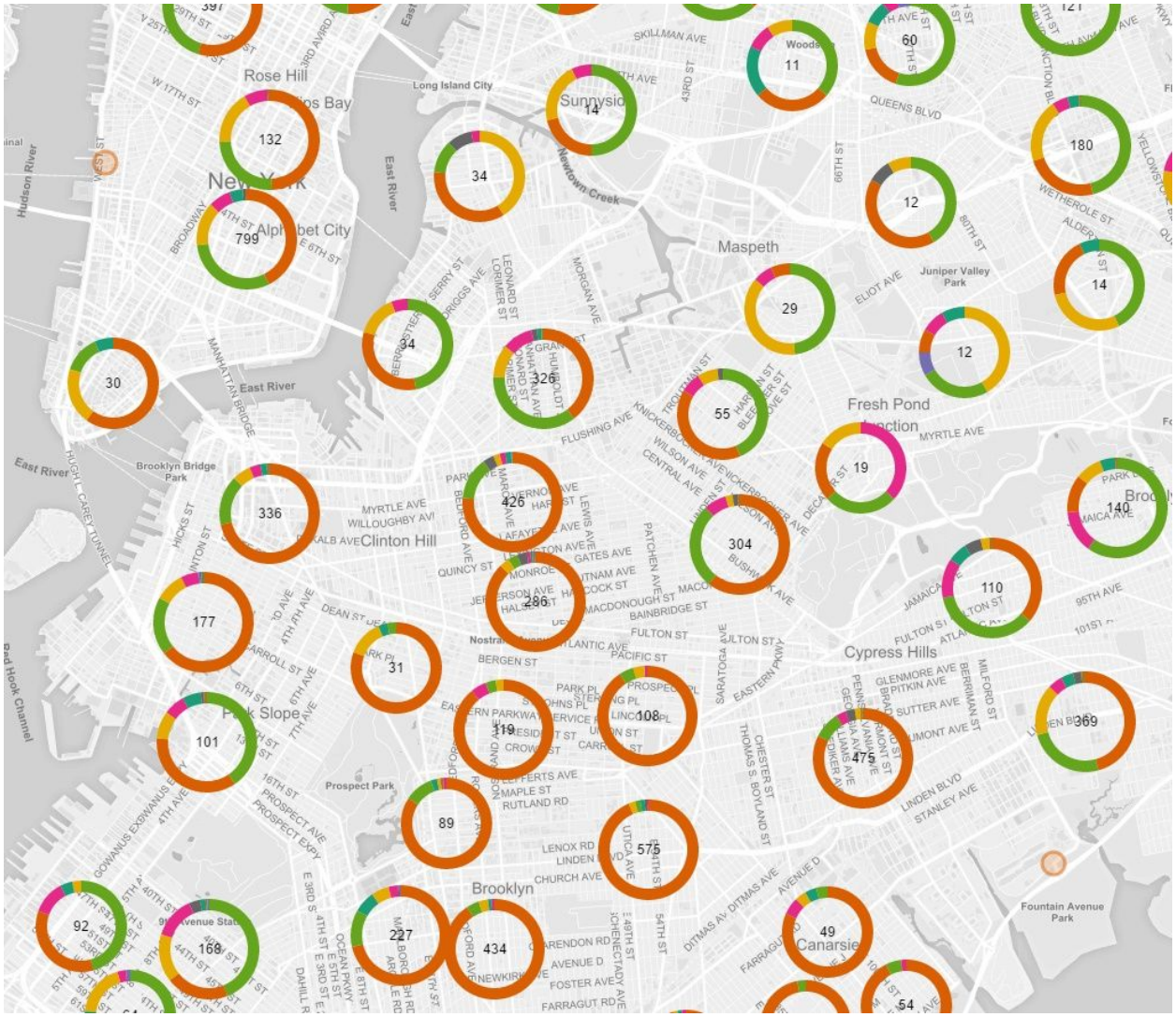

Figure 5. The visualization using the mergeable circles, which vastly improved legibility of the pies, but was unclear about which pies represented larger numbers of people.

However, we felt that there was an issue with the fact that there was no obvious visual difference between a chart representing 700 people and one representing 100 people. Thus, we added a transparency element to the image, where the more opaque a circle was the more people it represented. Haoxin made it so that the transparency would rescale depending on what was in frame. This ended up solving our problems with both the overlap and the circles seeming like area data, as it eliminated overlap for the most part and the hollow circles were more obviously not representing area. It was also adjusted so that the pie chart changes depending on which races are selected.

Then, we wanted to add one of our earlier goals- getting other data, like for the majority race in the area or the average income in the area on. This had the unexpected hurdle of getting the data, as mentioned before. After Haoxin's many struggles, we obtained the very

large data, and proceeded to map the races onto the background of the map. Here we had further issues, in that there were a lot of color scales going on that had to be dealt with. Initial attempts were not great.

Figure 6. The colored income choropleth interfered with the opacity coded colors for race. Our initial attempt to include race for census tracts, which made the map really hard to look at.

Then, after getting the data on the map at all, Isabella worked on finding smarter color choices for the graph. We ended up using a grayscale for the income data, as its sequential data, and pastel versions of the colors for races for the majority race data. This vastly improved the legibility of the visualization.

Final Result:

In the latest version of our project, the visualization is comprised of a grayscale themed leaflet map of the New York City area, three leaflet control groups, and a toolbar of “controls” and legends for the map. The control groups are what controls the actual overlays of the map, while the toolbar is what we want the users to be able to control the map with; work is under way to shift over to using the toolbar, but as of now we only have the design for the most part. The map itself is dynamically sized to fill the entire view, while the toolbar takes up about 18% of the right hand side view. In the toolbar and the control groups are selections for all eight race categories, five boroughs, and income or predominant race of each county or census tract.

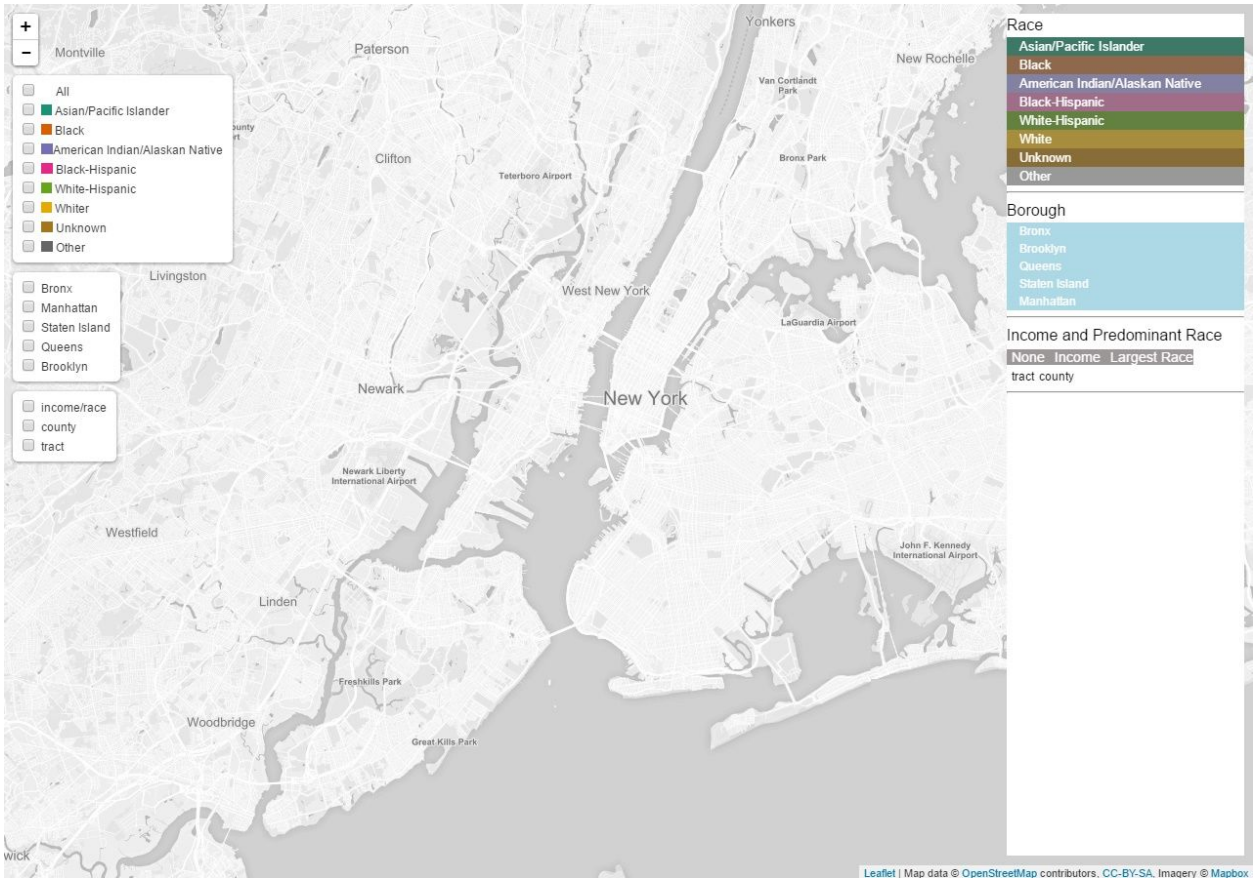

Figure 7. shows the map and sidebar with no data selected.

The selection panels allow the user to mix and match any combination of boroughs and races, and will dynamically update the map to visualize the data using pie charts. Income and predominant race selections are mutually exclusive because the data is represented using a choropleth style instead. All options are in one of two states: selected and unselected. The selected state is usually shown as the actual color with black text, while the unselected are represented by low saturation versions in the race section, darker colors in the borough and income and predominant race section, with white text to show they are unselected.

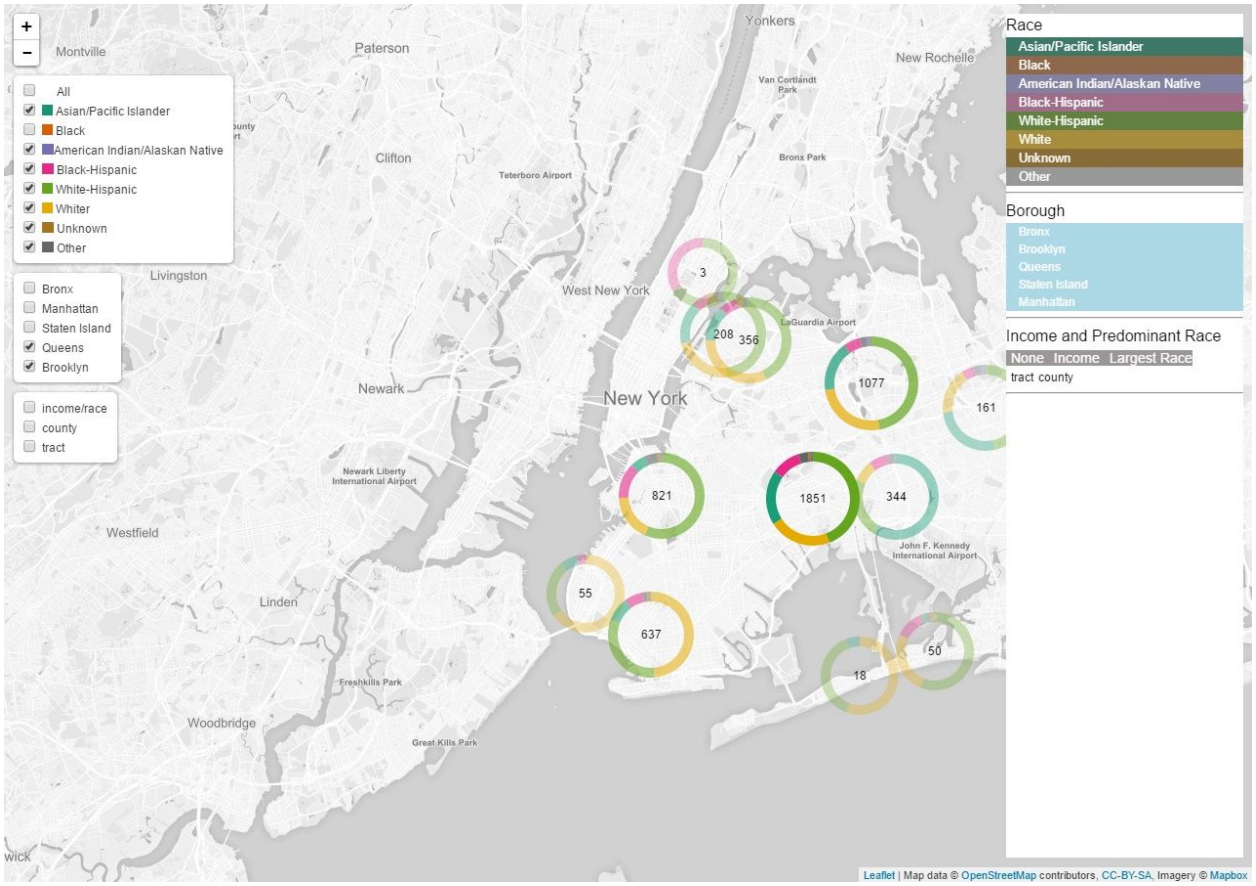

Figure 8. shows the map with all races but Black in Queens and Brooklyn.

The borough options limit the geographical region where any data is visualized, and by default all boroughs are selected. On the other hand, all race options start out unselected, which means no data is shown. With each race that the user selects, the map adds a circle to each location where an individual of the selected race(s) was stopped. The color of the circle represents the race of the individual stopped, and the center of the circle is the location where he or she was stopped. When circles are within a certain pixel radius of each other, they merge into a pie chart. Each pie chart has a number in the center indicating the number of circles (individual incidents) that the pie represents. Each slice of the pie chart is sized and colored to represent the amount of times a race was stopped out of all incidents the merged into the pie chart. Since the merge condition is based on pixel distance between the two points rather than physical distance between the two latitude-longitude, the circles merge to different degrees on each zoom level. As the user zooms out, the circles merge into larger and larger pie charts, and vice versa; however, at the max zoom there might still be pie charts, usually this means there are many circles on the same point.

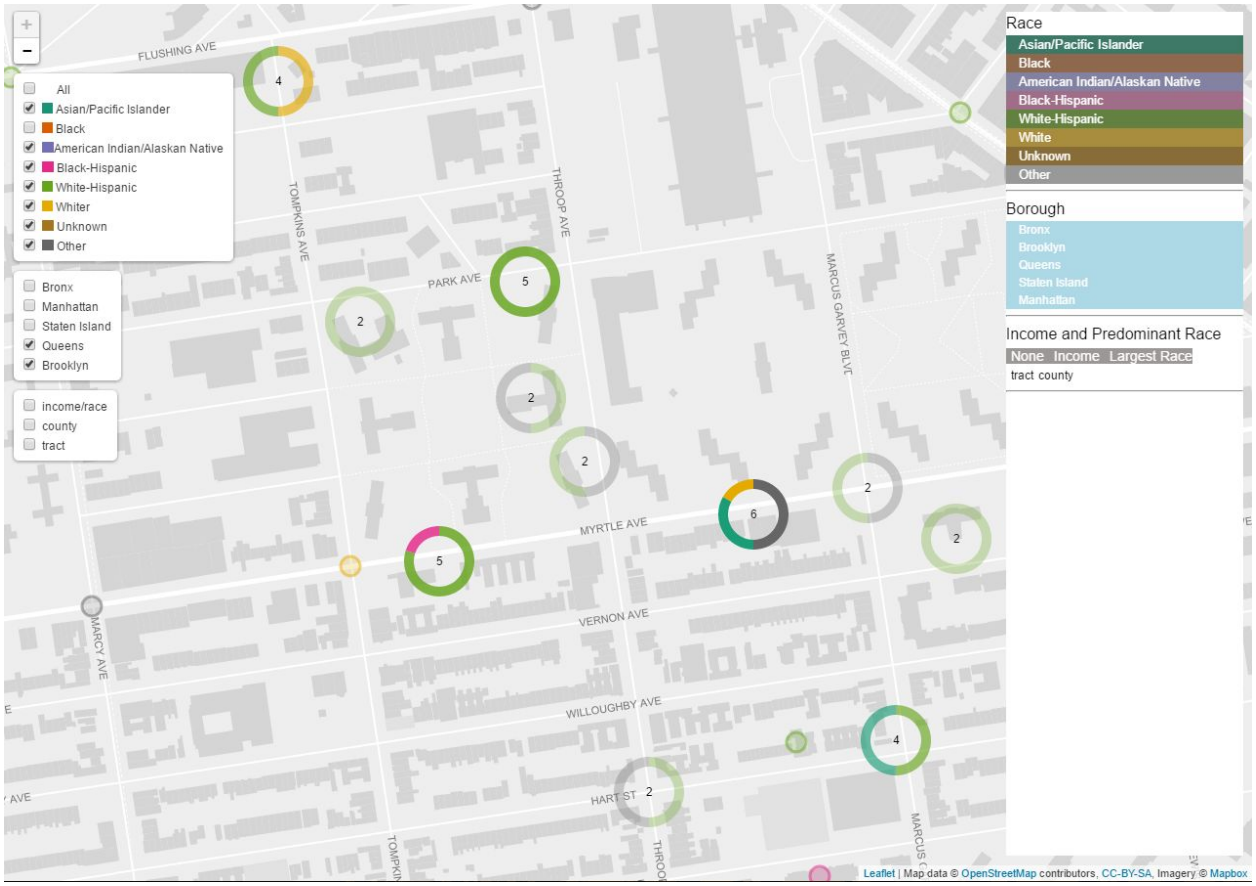

Figure 9. the same as Figure 8., but zoomed in to show pie chart at max zoom.

The income and predominant race selections toggle the choropleth map that either shows the 2014 median household income for each region or the race that comprises the largest portion of each region's population. By toggling either of these datasets on, the toolbar also updates to show the legend for each dataset. The regions are divided either by county or by census tract to provide both a general and detailed view of income and racial distribution. If income is selected, a grayscale layer appears where darker colors correspond to wealthier neighborhoods. If majority race is selected, that's shown with pastel versions of the race colors overlaid onto the background. Both have their own legends.

The current state of the visualization enables users to extract interesting pieces of information; such as how much portion of the total stop and frisk incidents each race makes up, and whether these proportions are consistent within different geographical regions. The income choropleth overlay allows users to spot correlations between regional income and the number of stop and frisk incidents. Furthermore by comparing the racial makeup in different income bracket regions, users can infer whether income plays a role in who gets stopped. On the other hand, the predominant race overlay allows users to see if a disproportionate number of a certain race gets stopped in a region relative to the region's racial distribution.

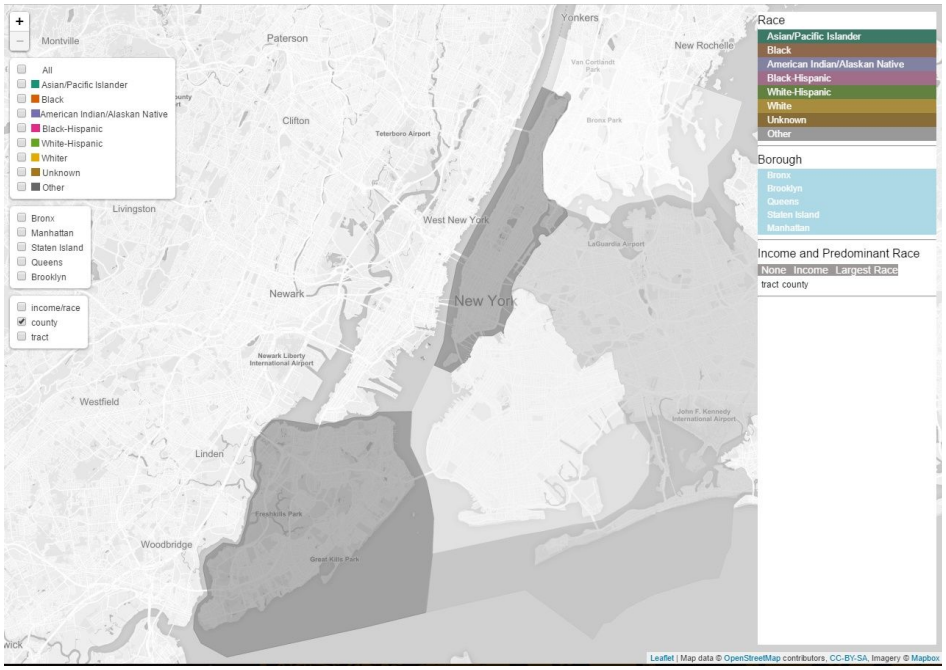

Figure 10. Income choropleth at the county level.

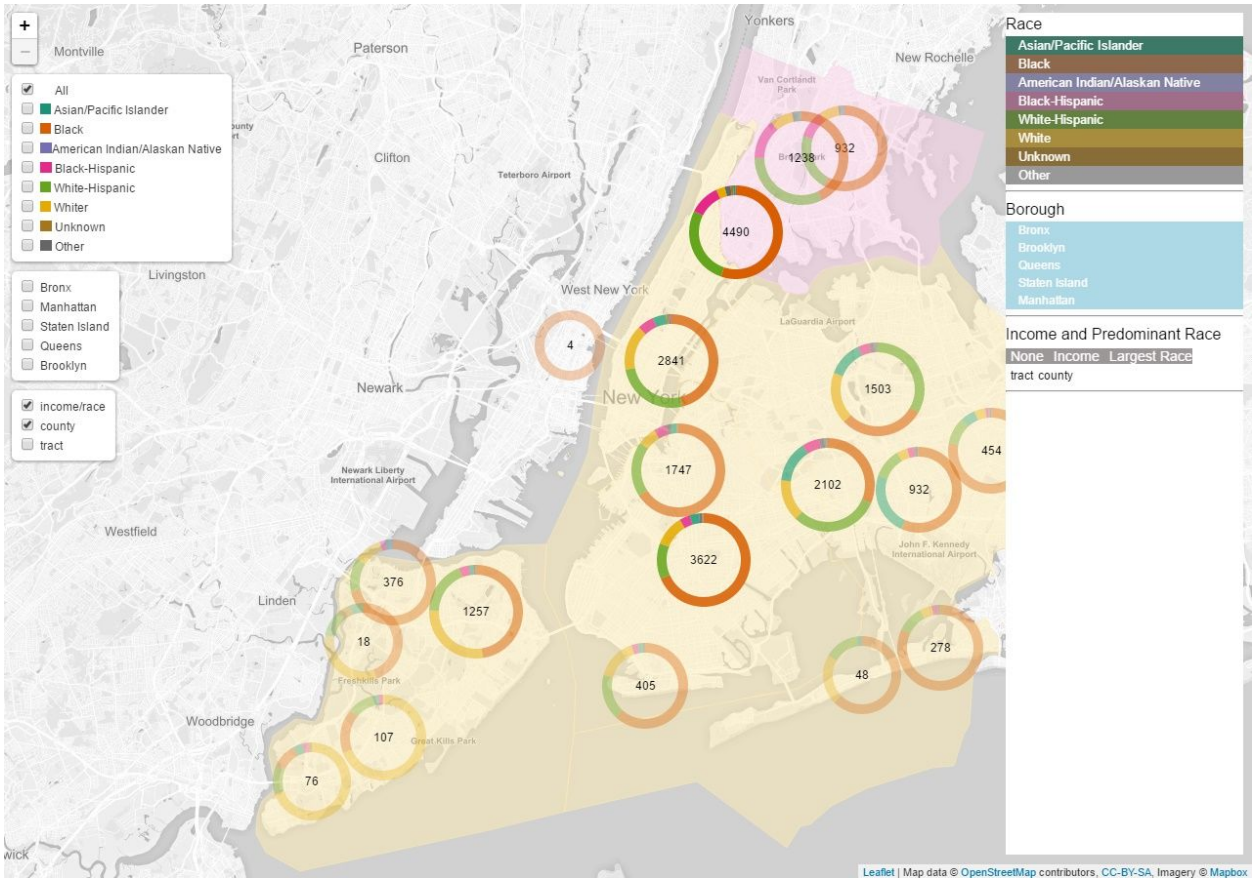

Figure 11. Broadly speaking, all but Bronx has predominantly White population, yet the people stopped were predominantly Black except in Southern Staten Island (the lower left).

Future Work:

If we had more time to work on our visualization, other things we would like to work on would be filters to show other aspects of the data, for example how many stops in a location involved arrests or violence. This would be a bit difficult, but worth exploring, as it is an important aspect of the data. However, we would probably have to deal a bit more with lag, one of our existing problems due to the sheer amount of data involved in our visualization. If we had the time, playing with the data organization and algorithms to try to improve that would be a priority. We also had started working on but hadn't completed a sidebar addition that would show a nice, more exact bar chart breakdown of a selected region or pie, which would be helpful for better analysis, as pie charts are not generally considered the best method. Another really cool element would be time, as we do have exact dates for all the stops and data is available for other years, so a slider and/ or a moving graphic for that would be useful for looking at trends over time.

Contributions:

Haoxin worked on data manipulation, making the pie chart view, laying down the census data, and setting up the functionality of the sidebar.

Isabella set up the map and initial circle representation, did the layers and filter selection, made the concentric circle view, and figured out colors and sidebar phrasing and layout.

Bibliography:

"2015 Stop, Question and Frisk Data." *Nyc.gov*. N.p., n.d. Web. 3 Apr. 2016.
"Google Maps Geocoding API." *Google Maps.* Google. 03 March 2016. Code. 12 April 2016.
"Guides." *CitySDK*. US Census Bureau, n.d. Web. 06 May 2016.
Agafonkin, Vladimir. "Leaflet.js." 2015. Code. 12 April 3016.
Keefe, John. "Stop & Frisk | Guns." *WNYC News*. WNYC, 16 July 2012. Web. 09 April 2016.
Rheil, Thomas. "All the Stops." *BKLYNR.* N.p. 2013. Web. 09 April 2016.
Romstad, Bård. "Markercluster pie charts." bl.ocks.org, 08 April 2016. Code. 28 April 2016.