Project Deliverable III

Nicholas Occhipinti
https://github.com/nick-occ/dsba-6520-project/tree/main/deliverable_3

Graph Data Model Revisions

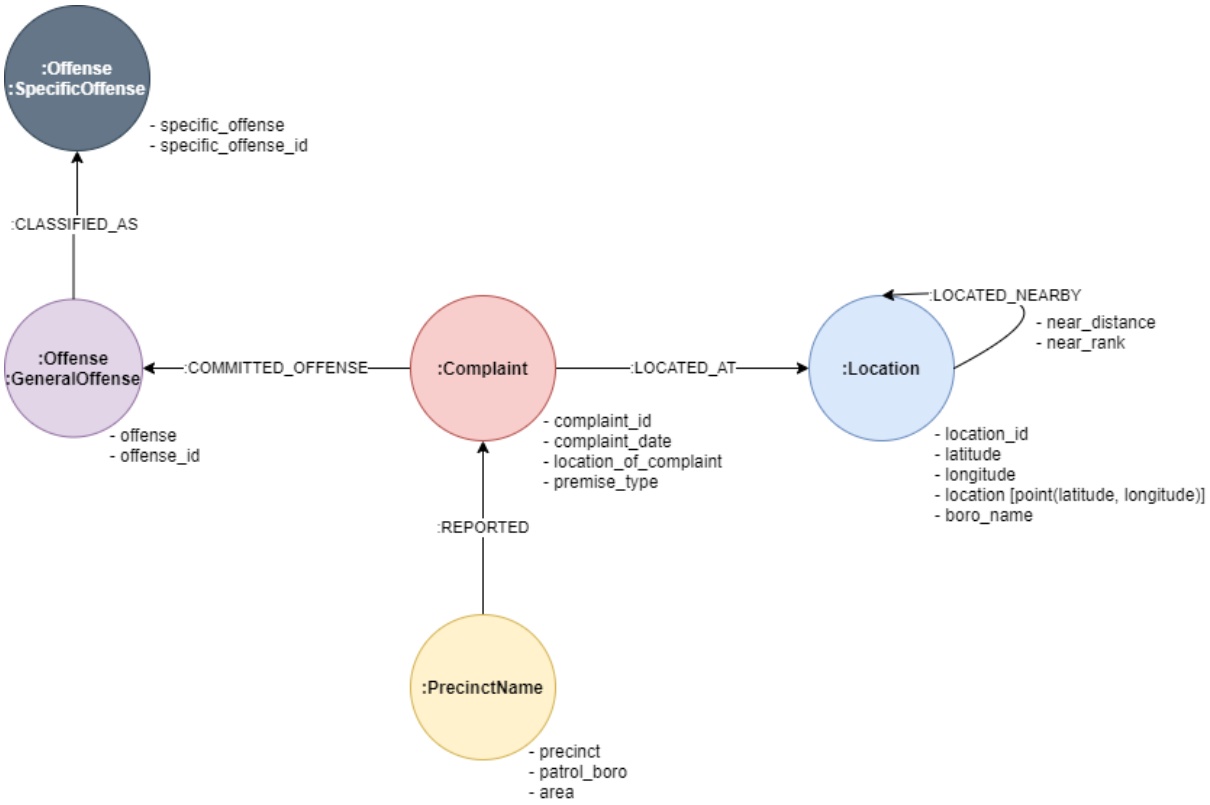
The data model contains 5 node types, **Complaint**, **PrecinctName**, **Location**, **GeneralOffense** and **SpecificOffense**. The **Complaint** node uses the “complaint_date” property which is when the complaint was first reported and uses the "complaint_id" property for the constraint to define uniqueness. The dates are a key to determining how complaints have changed over time. The **PrecinctName** node is the precinct where the complaint was reported and is represented by the “precinct” property. The relationship between **Complaint** and **PrecinctName** is a directional relationship where the precinct reports a complaint. The **Location** node is where the complaints occur and are spatially created as points through the latitude and longitude properties which represent the smallest level of granularity in this dataset. There is a directional relationship where complaints are located at locations.

There are two types of offenses, those that are more general and another category of more specific offenses which are broken up into two nodes, **GeneralOffense** and **SpecificOffense** respectively. There is a directional relationship between complaints where the related complaints contain details of an offense that was committed, and the general offense can get the classification of the specific offense category.

Latest Revisions
In the latest revision of the data model the from and to date and time properties that weren't being used were removed from the **Complaint** node. The majority of the changes occurred in the **Location** node. Using Geographic Information Systems (GIS) software different analyses were performed to get information assigned to the locations dataset before it was loaded into Neo4J. A "boro_name" property was created by taking a polygon spatial file of the NYC Boroughs from NYC Open Data and calculating the borough for all the location points within the 5 polygon features.

The second GIS related analysis performed was getting the nearby distances of location nodes. A unique id was created to make it easier to link the data between the Jupyter notebook and the GIS software, so as a result a "location_id" property was added to the **Location** node. The location data in my Jupyter notebook was imported into the GIS and XY points were generated to display on the map. The Generate Near Table tool calculated the distance for each location node to every other location node within 1000 feet. The output generated “to node” and “from node” fields of location ids and the distance and ranking between the nodes. The output was exported to a CSV and loaded into Neo4J as seen in the **LOCATED_NEARBY** relationship of the updated data model below along with the "near_distance" and "near_rank" properties of the relationship.

Updated Graph Data Model



Search Phrases

Complaints by Precinct In the Past Number of Days

Search Phrase: Complaints in $precinct in the last $number days

Purpose: This allows precincts to filter data only relevant to their precinct and get a view of the number of recent complaints that are happening and the offenses they are related to. It is designed to filter complaints by a user specified number of days back from the latest date on record in the database. So if the user just wants to look a week back they can just enter 7, a month back 30. This is more focused on what is happening currently and not for analyzing historical data months or years back. The output returns the complaint, precinct and offense nodes and allows you to expand the relationship and identify which offenses had a high number of complaints within a short time period in order to find patterns of offenses.

Cypher Query:

```
MATCH (c:Complaint)
with date(max(c.complaint_date)) - duration('P' + $number + 'D') as past_x
with min(past_x) as min_date
MATCH
(o:GeneralOffense)<-[:COMMITTED_OFFENSE]-(c:Complaint)<-[:REPORTED]-(p:PrecinctNam
e)
where date(c.complaint_date) >= min_date and p.precinct = $precinct
return c, p, o
```

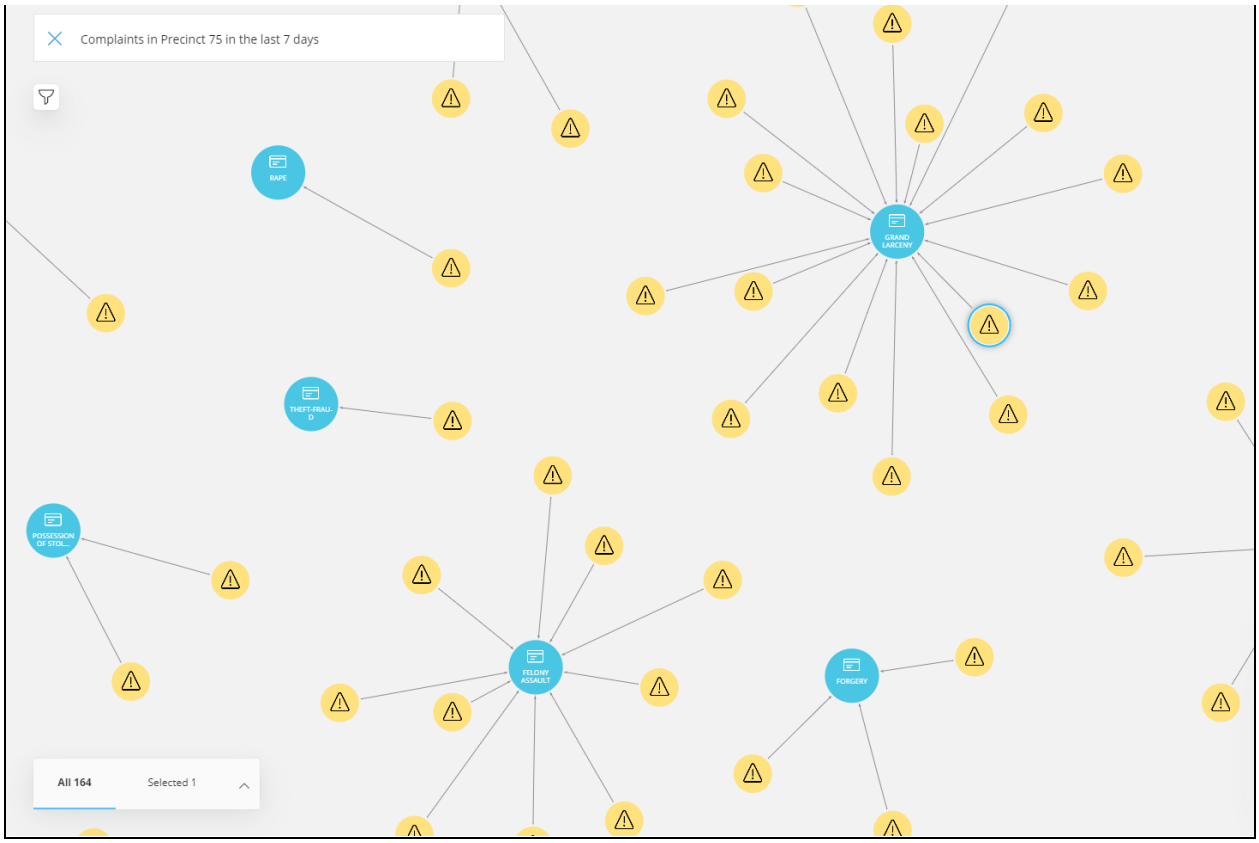
Parameters:

$precinct - PrecinctName({precinct})
$number - String

Example:

Complaints in Precinct 75 in the last 7 days

By looking at the related visualization we can see which offenses (GRAND LARCENY and FELONY ASSAULT) stand out as having a higher degree by looking at their relationship between complaints. This aims to answer the question about what is happening in each precinct.



Locations of Complaints by Precinct Within a Certain Date Range

Search Phrase: Locations of Complaints Reported by $precinct from $complaint_from_date to $complaint_to_date

Purpose: This allows precincts to filter data only relevant to their precinct and explore the relationship between complaints and locations. The user can provide a from and to date range as part of the search phrase to filter complaints only in the relevant time period they are interested in. This allows the user to find patterns where a high degree of complaints are happening at the same location. The user can also expand upon the locations they find and bring in the offenses to see if the high volume of complaints involve the same offense multiple times or if there are a variety of offenses being committed at the same location.

Cypher Query:

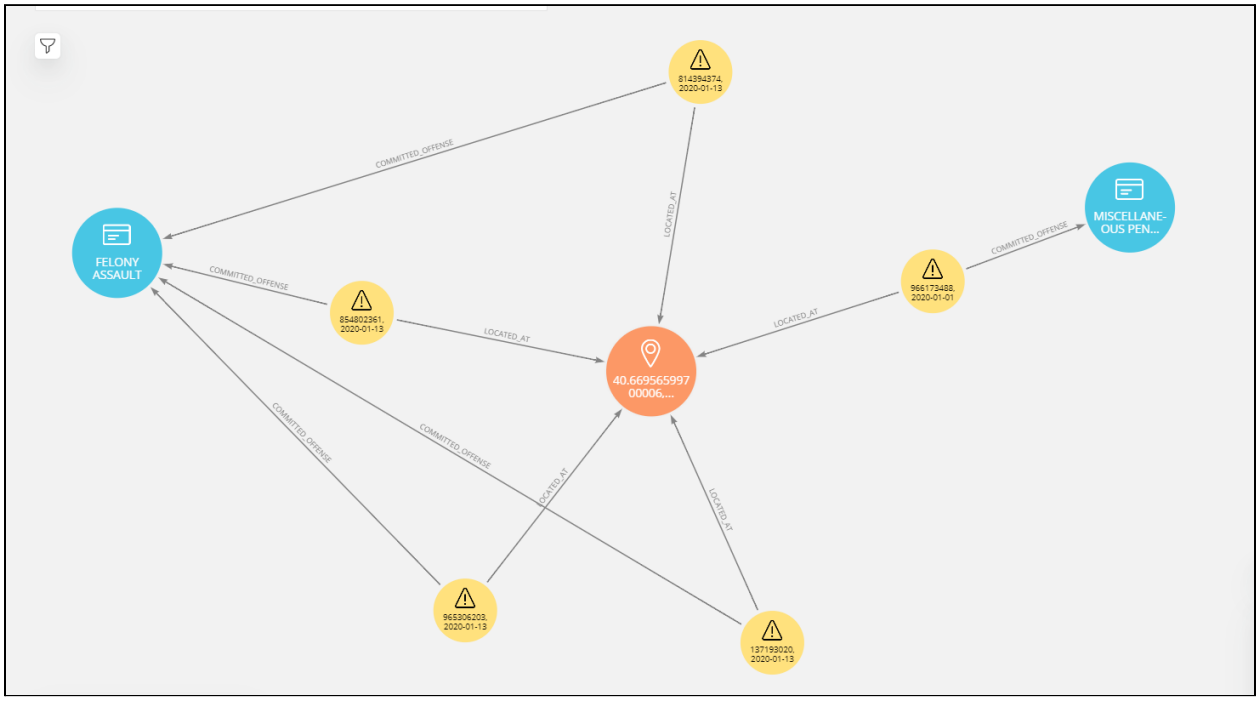
```
Match (p:PrecinctName)-[:REPORTED]->(c:Complaint)-[:LOCATED_AT]->(l1:Location)
where p.precinct = $precinct and c.complaint_date >= $complaint_from_date and
c.complaint_date <= $complaint_to_date
return  l1,c
```

Parameters:

$precinct - PrecinctName({precinct})
$complaint_from_date - Complaint({complaint_date])
$complaint_to_date - Complaint({complaint_date])

Example:
Locations of Complaints Reported by Precinct 75 from 2020-01-01 to 2020-01-14

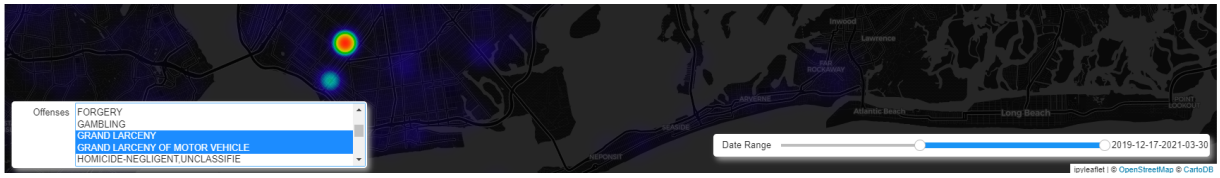
In this example we are exploring Precinct 75 during a two week time period in 2020. The relationships between Complaint and Location are shown and the user can expand the relationships between the two nodes and see which locations had numerous complaints within the short two week window. The example below shows one location had 5 complaints reported. Offenses were expanded to find that 4 out of the 5 complaints involved a felony assault and those 4 complaints were reported on the same day, 1/13/2020. This aims to answer the question about where complaints are happening in each precinct.



Mapping User Interfaces

In addition to the work done in Bloom, mapping interfaces were built and integrated with the Jupyter notebook through the “ipyleaflet” and “ipywidgets” modules. These maps have user interfaces to

allow the user the flexibility to filter by dates and being able to select one or multiple offenses. This will be elaborated on in the **Graph Visualizations** section.



Graph Visualizations

For the graph visualizations, the following visualizations were made outside of Bloom and use the Python map module “ipyleaflet” that provides the flexibility to develop an interactive map in Jupyter notebook. The research is heavily focused on where things are happening whether at the precinct level or the latitude and longitude coordinates, and maps help tell that story of what and where complaints are happening. Patterns can be shown geographically and user interface widgets allow for the flexibility of users to choose different times periods and offenses.

Hotspot Cluster Analysis Using Louvain Community Detection

This first analysis will calculate hotspots of complaint clusters based on the user selecting one or more offenses and a date range. The user can click on the Show Clusters button and it will run the following Louvain data science algorithm to determine communities of locations based on the nearby locations, complaints and offenses.

```
'CALL gds.louvain.stream({  \n\
   nodeQuery: "MATCH
(l:Location)<-[:LOCATED_AT]-(c:Complaint)-[:COMMITTED_OFFENSE]->(o:Offense) ' \
   'where o.offense in '+ str(offenses) + ' and c.complaint_date >= ' + from_date + ' and
c.complaint_date <= ' + to_date + '  return distinct id(l) as id",\n\
   relationshipQuery: "MATCH
(c:Complaint)-[:LOCATED_AT]->(l1:Location)-[:LOCATED_NEARBY]->(l2:Location) return id(l1)
as source, id(l2) as target",\n\
   validateRelationships: false,\n\
   maxIterations: 50})\n\
   yield nodeId, communityId\n\
   RETURN gds.util.asNode(nodeId).location_id AS location_id,
gds.util.asNode(nodeId).latitude AS latitude, gds.util.asNode(nodeId).longitude AS longitude,
communityId as community'
```

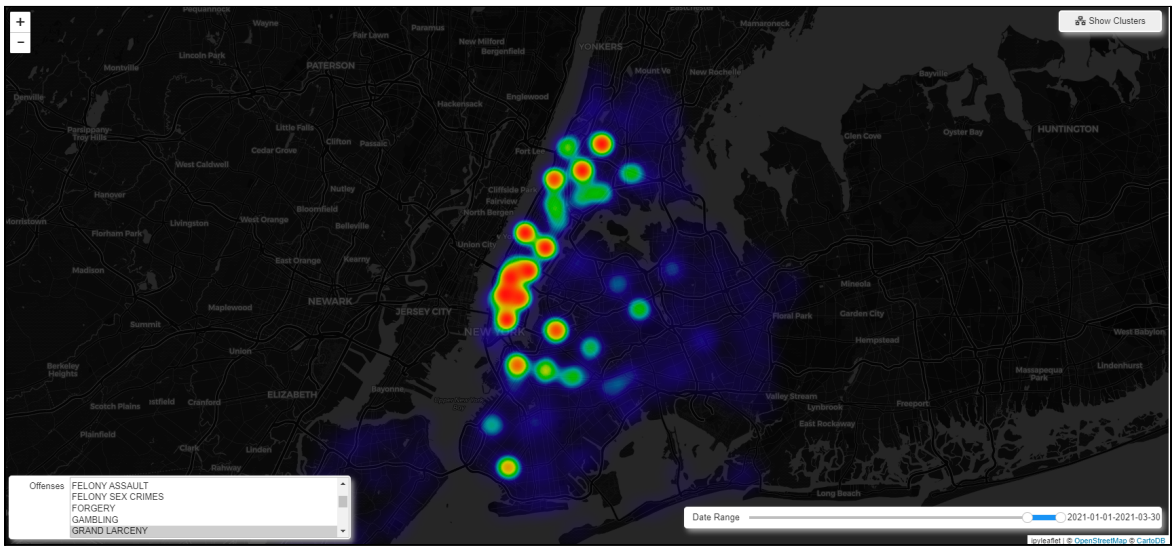
The community assignment, location_id, latitude, longitude and number of complaints are stored in a data frame and the average latitude and longitude and the sum of the number of complaints is calculated per community to get the cluster locations. The size of the cluster is based on the total number of complaints related to that community.

```
match (c:Complaint)-[:COMMITTED_OFFENSE]→(g:GeneralOffense)
return g.offense as offense, count(c) as complaints
order by complaints desc
limit 5
```

	offense	complaints
1	"GRAND LARCENY"	78517
2	"FELONY ASSAULT"	38555
3	"BURGLARY"	27879
4	"ROBBERY"	25727
5	"MISCELLANEOUS PENAL LAW"	23935

Grand Larceny is the most frequently reported offense in the dataset by a large margin. The results show a lot of hot spots in Manhattan. As we zoom in closer we see they occur in many high populated areas where not only New Yorkers live, but many people work and tourists visit. One advantage of this tool is that you can look at different time periods to spot any noticeable patterns.

Heat map of Grand Larceny from 1/1/21 to 3/30/21

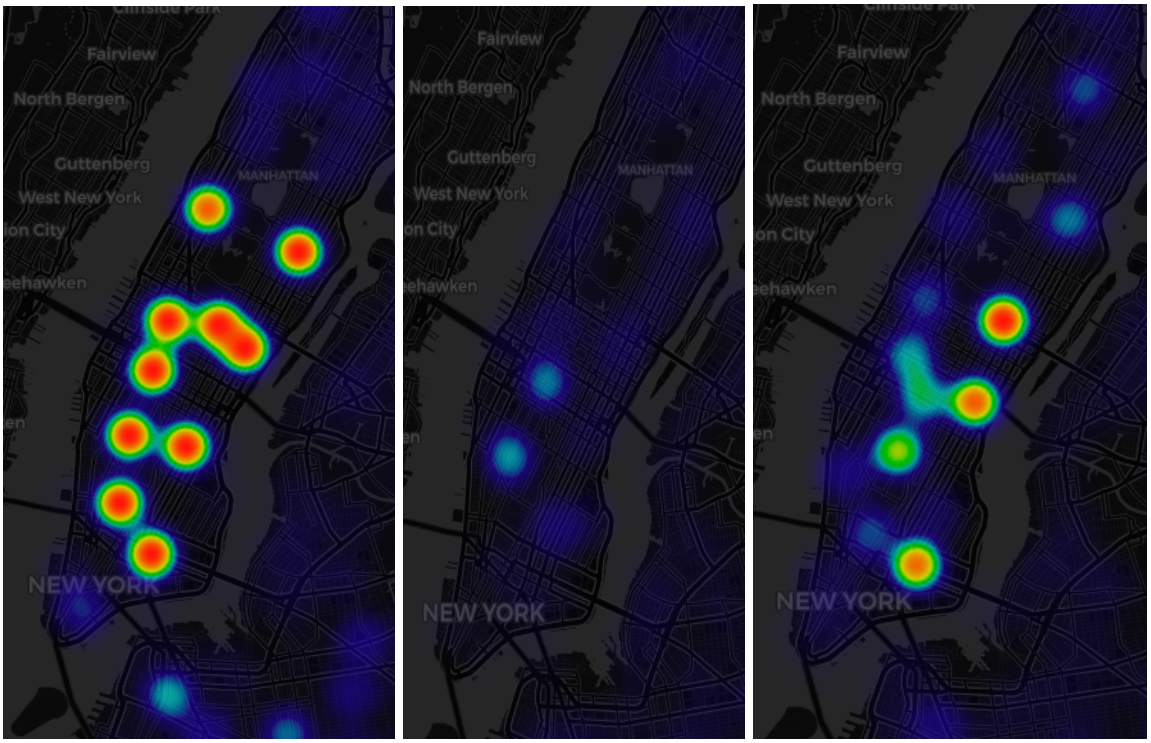


Zoomed in version of Grand Larceny Heat Map in Manhattan

To demonstrate this we will use another use case; seeing if it can be determined how offenses have changed as a result of the pandemic. Sticking with the Grand Larceny offense, 3 time periods were chosen: 4/1/2019-4/30/2019, 4/1/2019-4/30/2020 and 3/1/2021-3/30/2021. These time periods represent a normal month, pandemic month, and a transitioning to normal month in the pandemic calendar.

Grand Larceny Comparison

4/1/2019 - 4/30/2019	4/1/2020 - 4/30/2020	3/1/2021 - 3/30/2021

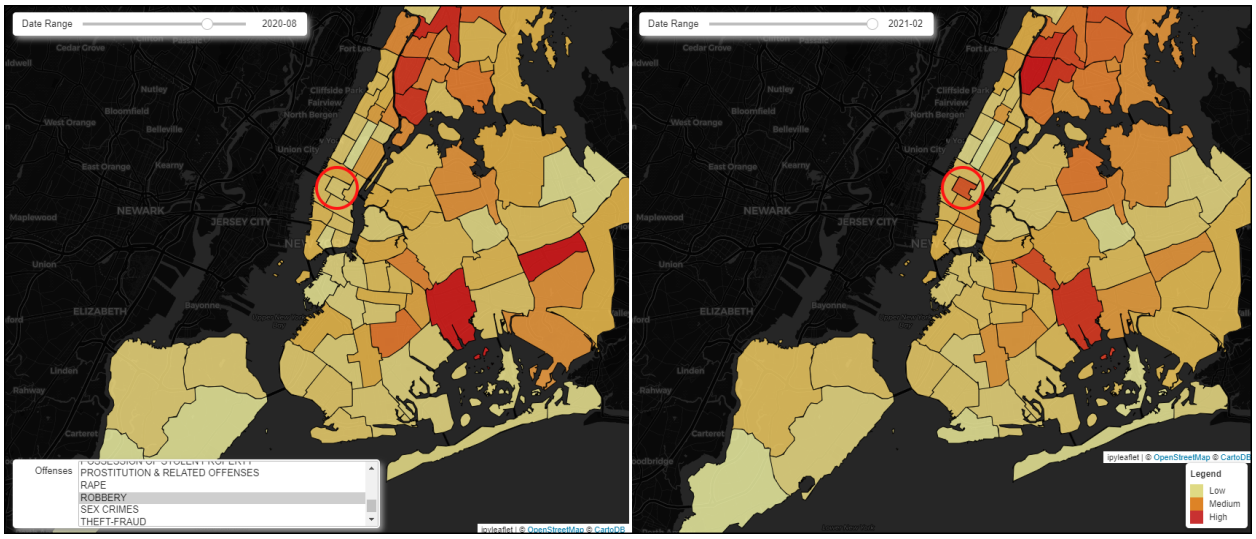


The results show a big drop in grand larceny complaints between 2019 and 2020 and we can see an emergence of hotspots appearing in 2021 which could be an indicator of returning to normal which unfortunately also means rising grand larceny cases. An article from June 2021 in the New York Post (https://nypost.com/2021/06/08/grand-larcenies-rebound-as-nyc-opens-back-up/) also mentions the trend of rising grand larceny complaints as the city reopens and note that 2020 had the lowest number of grand larceny complaints in 20 years.

Comparison of Precinct Complaints by Dates

This visualization provides a high level view of how precincts are performing in terms of the number of complaints by comparing different time periods. To study complaint patterns people need to be able to look historically and compare previous data with the current time period. The following visualization is two choropleth maps that allow you to adjust the dates and see what the level of complaints were for each selected date. The data is resampled at the month level. Similar to the previous map the user can select one or multiple offenses to analyze. The precinct layers were brought into the map as a GeoJSON layer and are colored based on the amount of complaints.

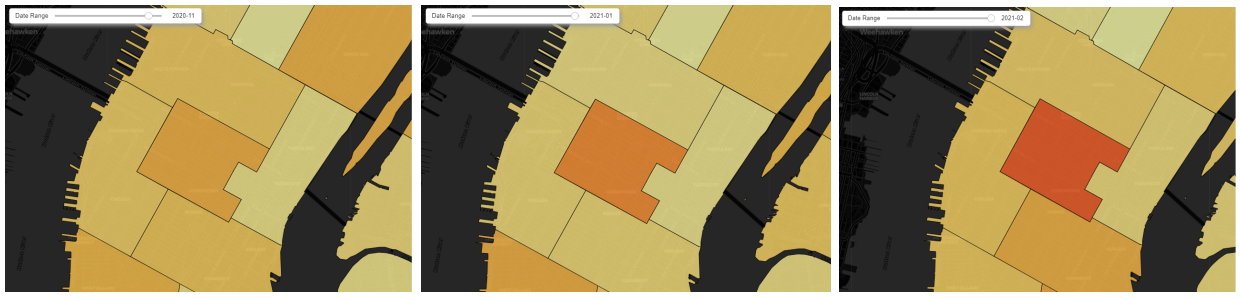
Investigating the Robberies in August 2020 and February 2021, a lot of the complaint levels seem fairly similar for most precincts, One noticeable difference is the precinct highlighted in the red circle below where it was low in August 2020 and six months later it is high.



We can compare what has happened between those months and see if it is just a one off spike or an emerging pattern. In the visualization below we see that in November 2020 the number of complaints rose to a medium level, in January 2021 it rose again to a medium-high level before reaching a high level in February 2021. Based on the results it appears to be a pattern emerging over time. It is also important to note that this precinct covers the area between 34th Street and 42nd Street which is a major tourist hotspot. This could indicate once again that as things get back to normal after the pandemic and as tourism rises there is the risk that complaints will rise as well.

Robberies

November 2020	January 2021	February 2021



Top 10 Locations of Offenses using Page Rank

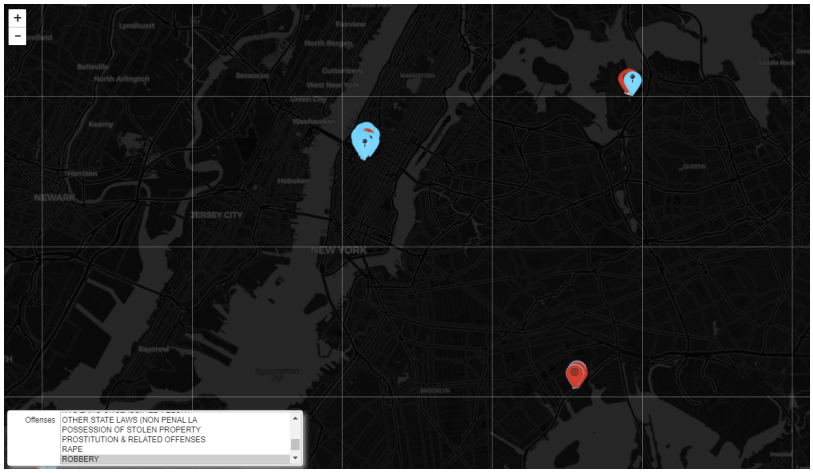
This visualization uses the Page Rank centrality algorithm to find the top central locations to where offenses are occurring. The algorithm also takes into account related locations nearby and the amount of complaints to calculate the top locations.

```
'CALL gds.pageRank.stream({ \n\
   nodeQuery: "match (l:Location) return distinct id(l) as id",\n \
   relationshipQuery: "match
(o:Offense)<-[:COMMITTED_OFFENSE]-(c:Complaint)-[:LOCATED_AT]->(l:Location)-[:LOCAT
ED_NEARBY]->(l2:Location) where o.offense in ' + str(offenses) + ' and c.complaint_date >= ' +
from_date + ' return id(l) as source, id(l2) as target",\n\
   validateRelationships: false})\n\
   YIELD nodeId, score\n\
   with gds.util.asNode(nodeId).location_id AS location_id, gds.util.asNode(nodeId).latitude AS
latitude, gds.util.asNode(nodeId).longitude AS longitude, score\n\
   where location_id is not null\n\
   return location_id, latitude, longitude, score\n\
   order by score desc\n\
   limit 10'
```

The user has the ability to select one or multiple offenses and can search as far back as 90 days since the last date a complaint was reported to determine the central locations. By knowing the central locations you can better target your policing strategies. An example could be a location with a large volume of complaints but on the outskirts of where most of the police patrol is currently occurring. Leadership can take this data and realign their patrols to match where the central locations of complaints are happening.

In the example below the user selected Robbery going 90 days back. The red markers are the top 10 central locations selected by the Page Rank algorithm and the blue markers are locations nearby that had the same offense occur within the same time frame.

Robbery 90 Days Back



One of the areas the Page Rank algorithm chose was in midtown Manhattan where 3 of the 10 Page Rank assigned locations were close to one another. This example shows why these locations were chosen given how central they are to the large number of other nearby locations where the same offenses occurred.

Robbery 90 Days Back - Zoomed In