NYPD Crime 2006-2016 Fact Book

By:
Amitesh Sah (aks629)
Prasad Bhagwat(pgb252)
Tejaswi Vinod (tvg226)

Github link: https://github.com/amiteshsah/Big-Data-Project/

ABSTRACT:

Our aim is to provide a reference to get an idea about safer neighbourhood by running certain analysis against NYPD Crime dataset. Also, we are trying to incorporate the dataset of Census i.e. Population and financial status to see if these factors affects the crime rate or not. Also, to decide the better neighbourhood, we can use the complaints data to support the results generated by analysis on crime dataset.

INTRODUCTION:

In a city as big as New York City, it is very difficult to find out the safe neighbourhood and people are often worried if they do not have good statistical information.

So through our analysis on Crime data, we would like to find out which neighbourhood is safe by below factors:

1. What are types of crime committed in neighbourhood? Analyse them year to year.
2. Which time of the day is it committed i.e when is the most of the crimes are happening?
3. What is the day is it happening - weekday or weekend ?
4. What is the population of that neighbourhood?
5. What is the unemployment rate of that neighbourhood?
6. What is the financial status - median income and poverty rate of the neighbourhood?
7. Are the population, unemployment rate and financial status affecting the rate of the crime?
8. What would be a better choice of neighbourhood based on crimes and complaints?

DATASET :

1. NYPD Crime Dataset
    a. https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i
2. Complaints
    a. (2010-2016) https://data.cityofnewyork.us/Social-Services/311/wpe2-h2i5
    b. (2009) https://data.cityofnewyork.us/Social-Services/new-311/9s88-aed8.
3. Population, Unemployment rate and Poverty rate neighborhood wise 2006-2014
    a. http://furmancenter.org/research/sonychan

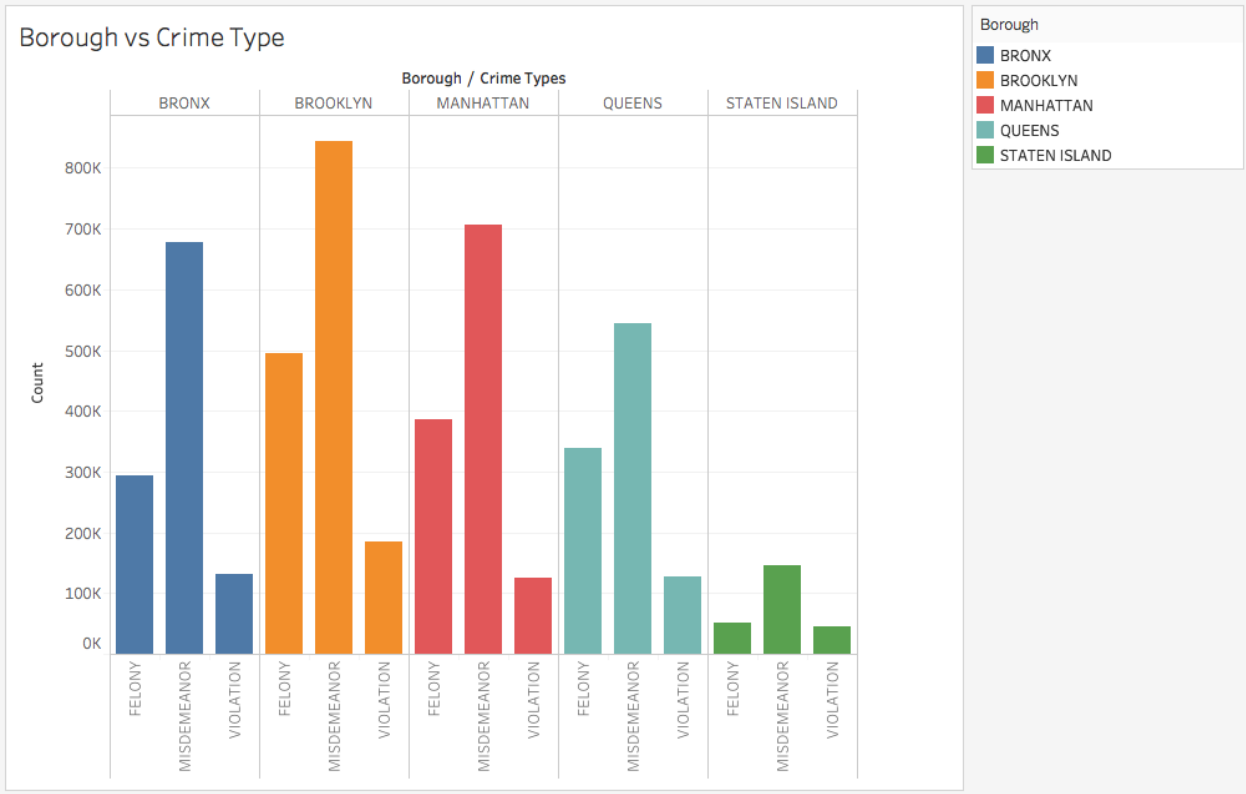
PART 1

AIM : Our aim in project part 1 was to clean the NYPD Crime dataset and find out valid, Invalid entries.

Also, we are trying to find out some interesting patterns from the dataset and give insight into dataset with some facts.

Below are facts and findings from the dataset.

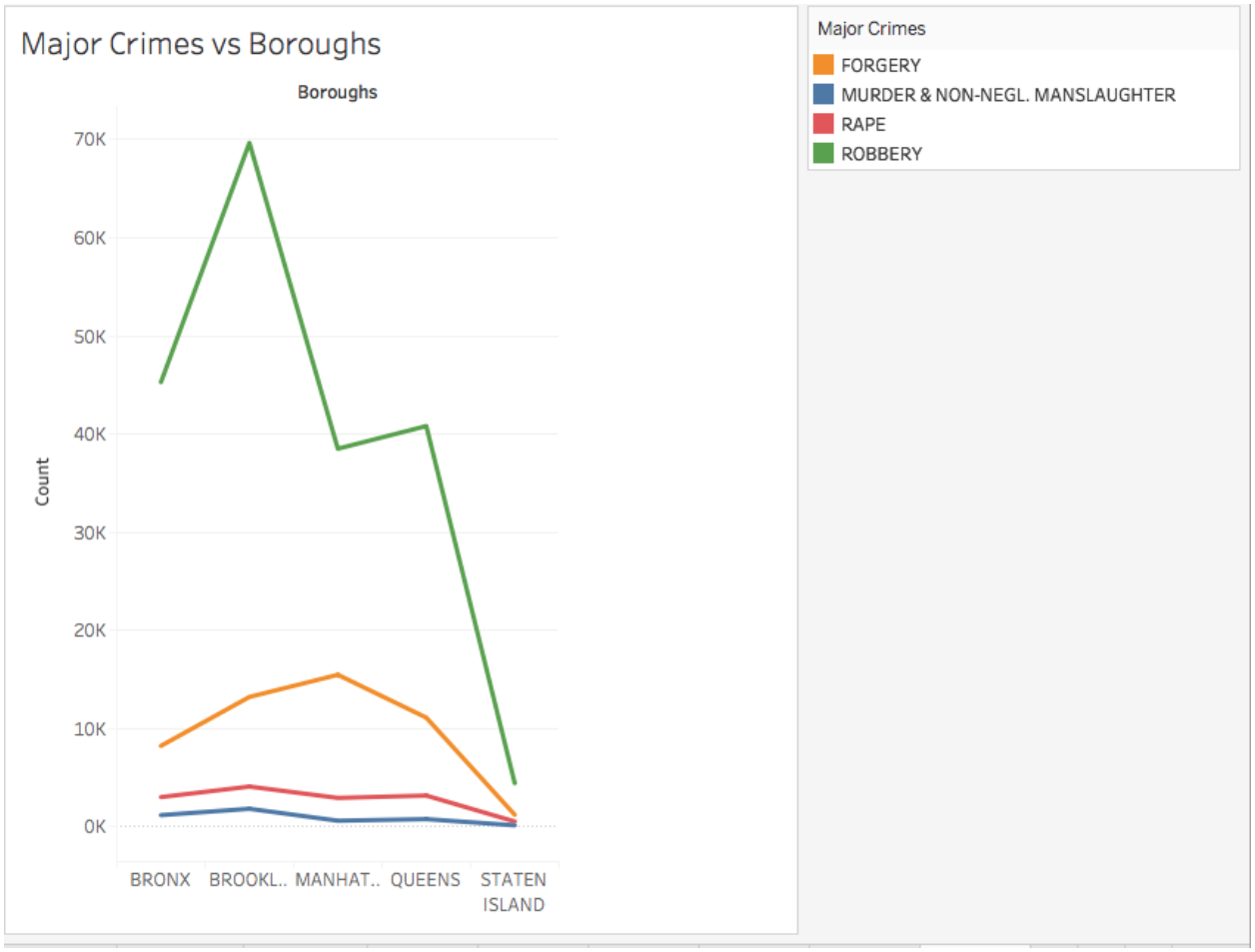
1. **Borough vs Crime Type:**
   A graph to give an idea about crime rate occurred and with respect to different boroughs and different crime categories. In here, Brooklyn has most number of crime rates in all categories.



Script Used:

CrimeTypes_vs_Boroughs_map.py
CrimeTypes_vs_Boroughs_reduce.py

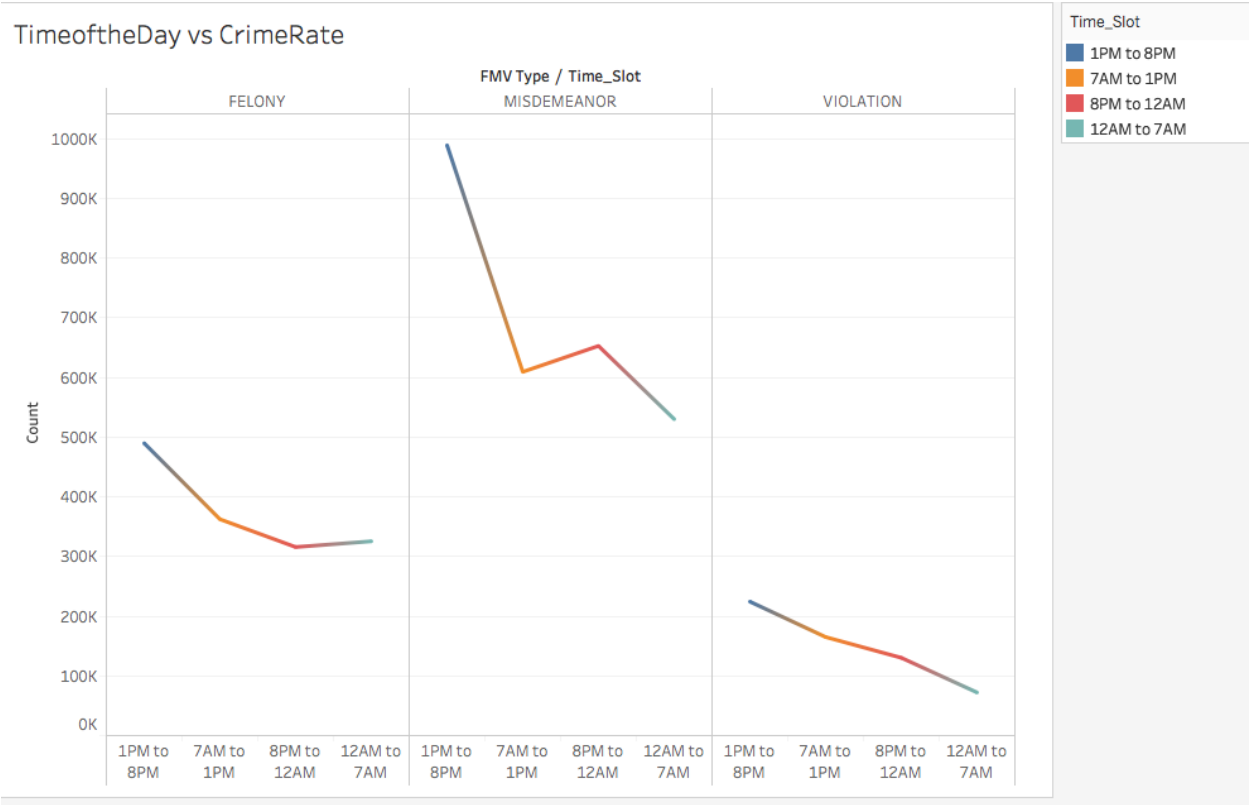
2. **Major Crimes vs Borough:**
   Below graph is give an idea about major crimes happened with respect to different boroughs and major crimes are such as robbery, murder, rape and forgery. Our analysis indicates that Brooklyn has significant major crime rates, which strengthens the above analysis.



Script Used:

MajorCrimeRate_vs_Boroughs_map.py
MajorCrimeRate_vs_Boroughs_reduce.py

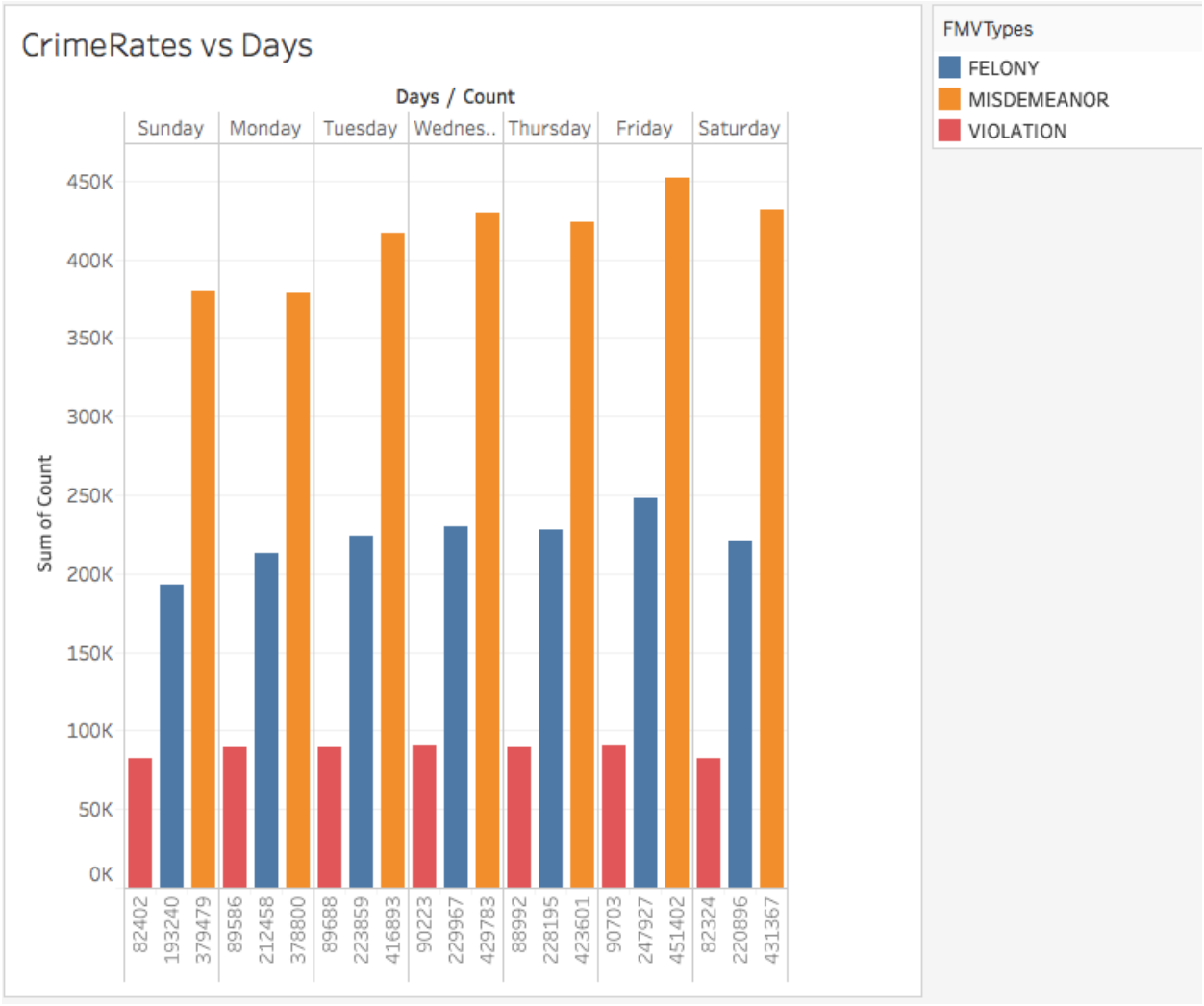
3. **Time of the Day Analysis vs Crimes:**
Below graph throws the light upon crime occurring during different hours of the day. Our assumption was it must be higher in night time /evening time but it shows that for all the crime types, crimes happened in day time i.e. 1PM to 8PM.



Script Used:

FMVcount_vs_Time_map.py
FMVcount_vs_Time_reduce.py

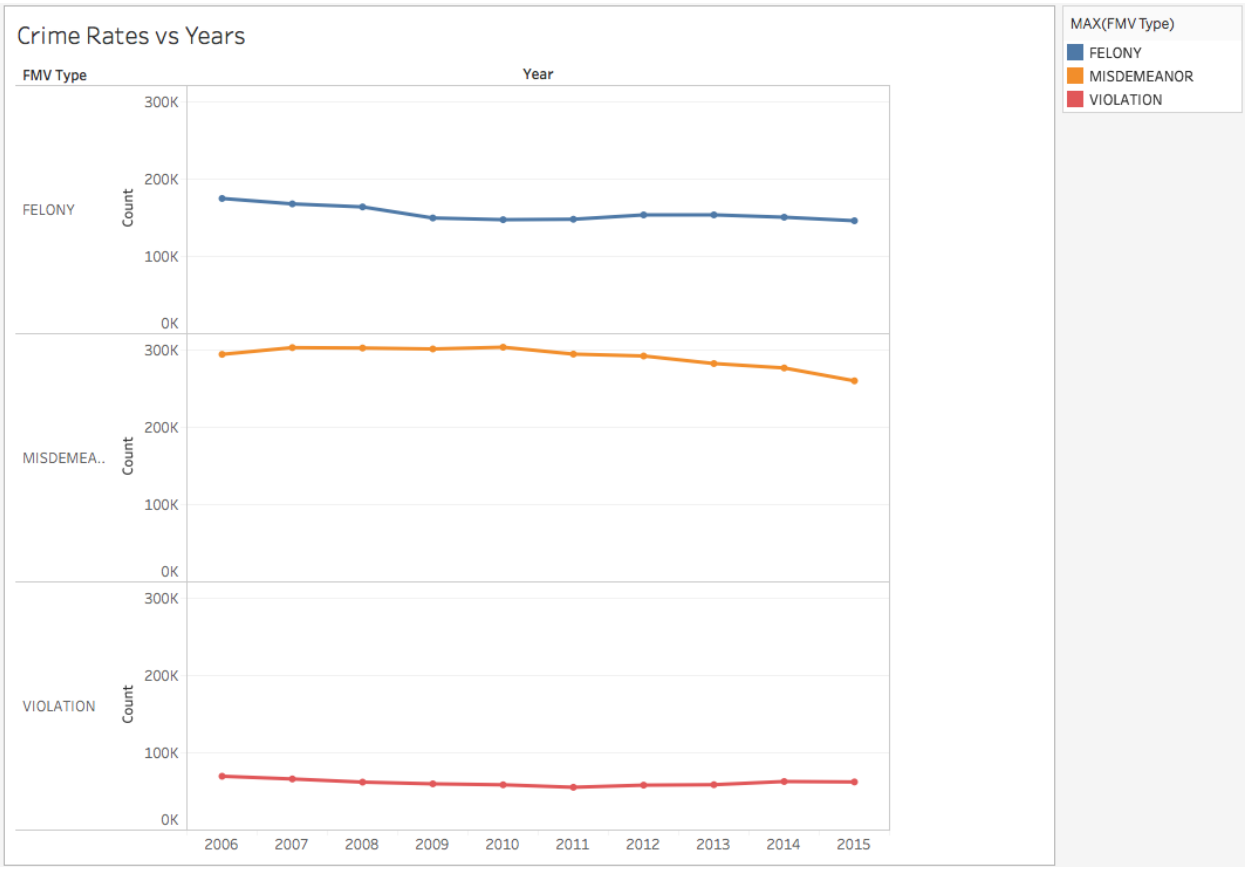
4. **Days of Week vs Crimes Analysis:**
   Below graph indicates that most of crimes have occurred at the beginning of the weekend .i.e. Wednesday, Thursday, Friday and Saturday.



Script Used:

FMVcount_vs_Day_map.py
FMVcount_vs_Day_reduce.py

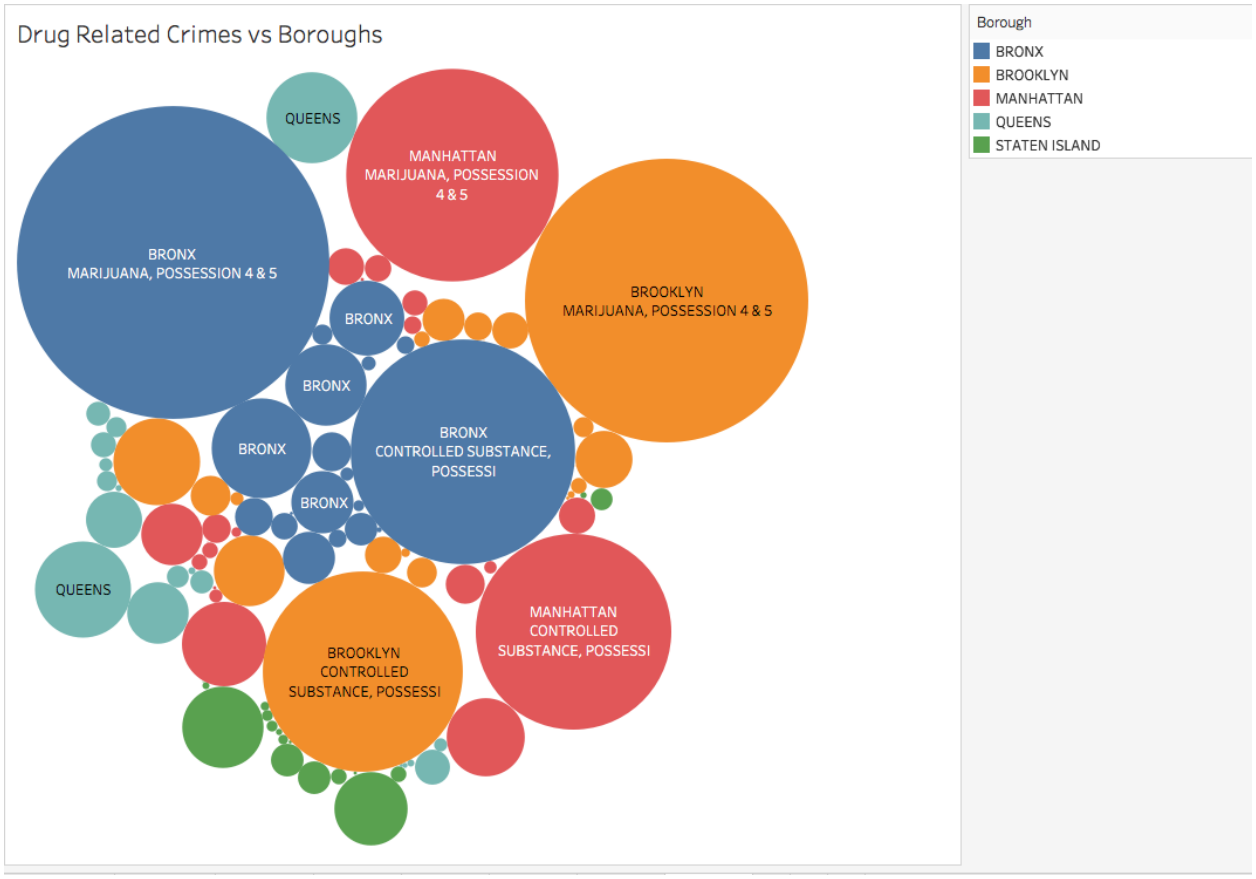
5. **Crimes per Year Analysis:**
   Below graph shows that Violation and Felony has been steady throughout years from 2006-2015.
   We observed that Misdemeanor has been reduced slightly over the period of type.



Script Used:

FMVcount_vs_Year_map.py
FMVcount_vs_Year_reduce.py

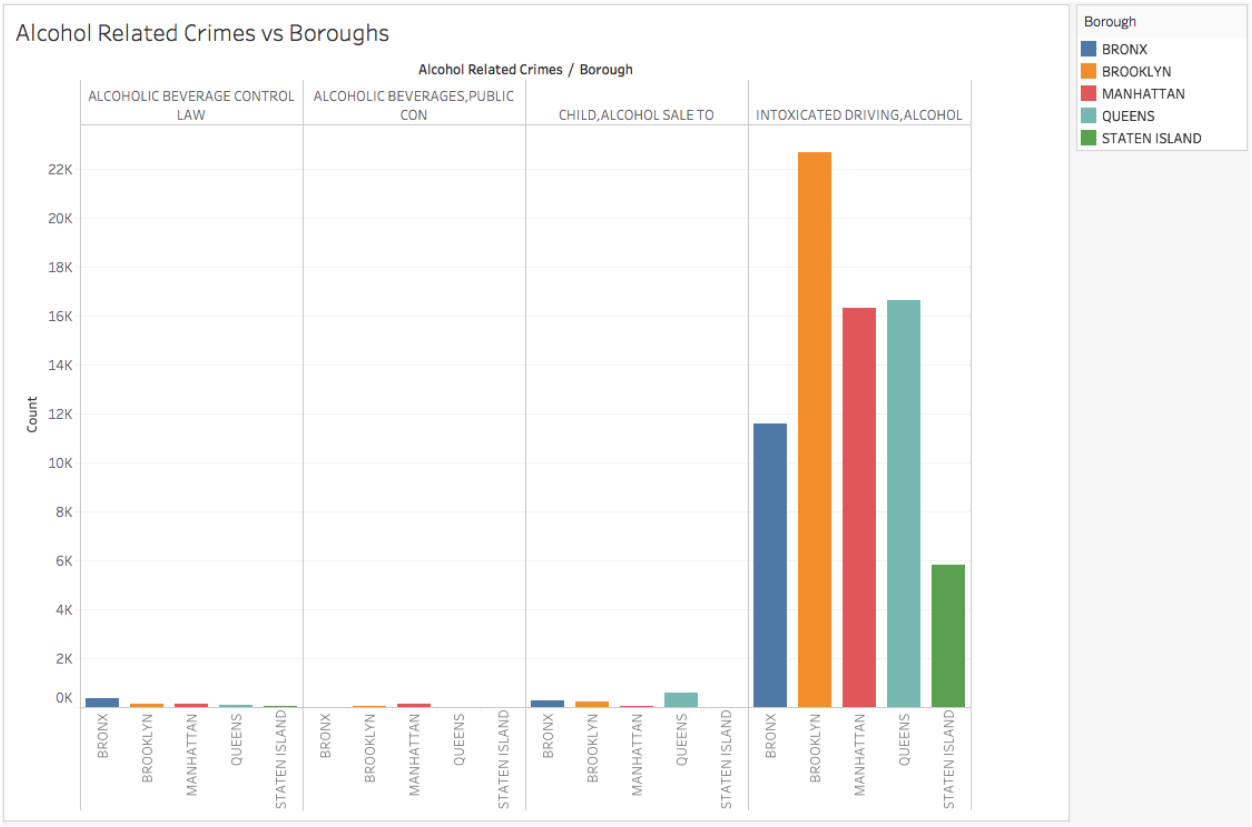
6. **Drug related Crimes per Boroughs:**
   Below graph is to figure out which borough shows maximum number of drugs related crimes i.e. sale and possession with respect to other boroughs. Brooklyn and Bronx are involved in drugs related activity, followed by Manhattan.



Script Used:

DangerousDrugs_vs_Boroughs_map.py
DangerousDrugs_vs_Boroughs_reduce.py

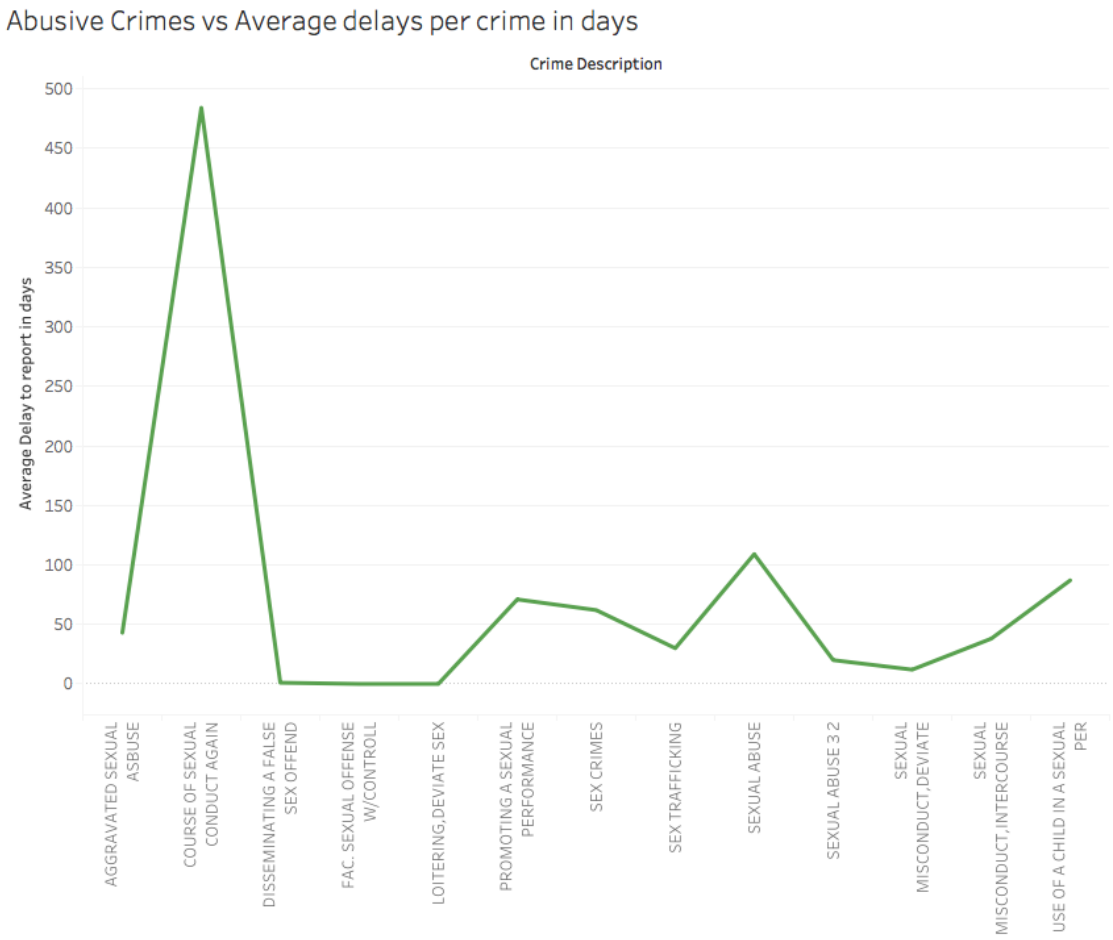
7. **Alcohol related Crimes per Boroughs:**
   This graph brings shows that intoxicated driving is most occurring offense compared to any other alcohol related crimes and again maximum occurrence in Brooklyn.



Script Used:

AlcoholCrimes_vs_Boroughs_map.py
AlcoholCrimes_vs_Boroughs_reduce.py

8. **Average delays to report abusive crimes per days:**
   Below graph brings specifically draws our attention to the sexual crimes happened and the average time taken to report them per days per type of crime. It surprises us with the fact that there are crimes which were reported after an year too.



Script Used:

SexualCrimeRates_vs_Timedelay_map.py
SexualCrimeRates_vs_Timedelay_reduce.py

PART I: DATA SUMMARY AND QUALITY ISSUES

COLUMNS ANALYSIS and ISSUES:

1.**CMPLNT_NUM** – Unique value for the column
We looked for blank spaces and NULL, INVALID values. Also, we considered this as unique value so did not consider the condition to check the duplicates.
Data Issue: Received '0' as value but we considered it as "Unique" value.
Mapper Script: CMPLNT_NUM_Validator_map.py
Reducer Script: Columns_Validator_reduce.py

Count of Valid Values : 5101222
Count of Invalid Values : 0
Count of Null Values : 0

2. **CMPLNT_FR_DT** – Start Date of Incident occurrence
We looked for blank spaces, format of the date and if it falls within range of year 2006 to 2017 considered as "VALID" else considered as "INVALID".
Mapper Script: CMPLNT_FR_DT_Validator_map.py
Reducer Script: Columns_Validator_reduce.py
Count of Valid Values: 5081794
Count of Invalid Values: 18783
Count of Null Values: 655

3. **CMPLNT_FR_TM** – Start Time of Incident occurrence
We looked for blank spaces, format of the time and if it falls within correct range of time considered as "VALID" else considered as "INVALID".
Mapper Script: CMPLNT_FR_TM_Validator_map.py
Reducer Script: Columns_Validator_reduce.py

Count of Valid Values: 5101222
Count of Invalid Values: 0
Count of Null Values: 0

4. **CMPLNT_TO_DT** - End Date of Incident occurrence
We looked for blank spaces, format of the date and if it falls within range of year 2006 to 2017 considered as "VALID" else considered as "INVALID".
Mapper Script: CMPLNT_TO_DT_Validator_map.py
Reducer Script: Columns_Validator_reduce.py

Count of Valid Values: 3704869
Count of Invalid Values: 4885
Count of Null Values: 1391478

5. **CMPLNT_TO_TM** – End Time of Incident occurrence
We looked for blank spaces, format of the time and if it falls within correct range of time considered as "VALID" else considered as "INVALID".
Mapper Script: CMPLNT_TO_TM_Validator_map.py

Reducer Script: Columns_Validator_reduce.py
Count of Valid Values: 5101222
Count of Invalid Values: 0
Count of Null Values: 0

6. **RPT_DT** – Report Date
We looked for blank spaces, format of the date and if it falls within range of year 2006 to 2017 considered as “VALID” else considered as “INVALID”.
Mapper Script: RPT_DT_Validator_map.py
Reducer Script: Columns_Validator_reduce.py

Count of Valid Values: 5101222
Count of Invalid Values: 0
Count of Null Values: 0

7. **KY_CD** – 3 digit Classification code
We looked for the length of code and it’s value as integer, not greater than 999 and not 000 as we have already looked for anomalies by writing a Map and Reducer function to get unique possible values.
Mapper Script: KY_CD_Validator_map.py
Reducer Script: Columns_Validator_reduce.py
Count of Valid Values: 5101231
Count of Invalid Values: 1
Count of Null Values: 0

8. **OFNS_DESC** - Offence Description code
We looked for only blank spaces as this is a description as we have already looked for anomalies by writing a Map and Reducer function to get unique possible values.
Mapper Script: OFNS_DESC_Validator_map.py
Reducer Script: Columns_Validator_reduce.py

Count of Valid Values: 5082392
Count of Invalid Values: 0
Count of Null Values: 18840

9. **PD_CD** - 3 digit Classification code
We looked for the length of code and its value as integer, not greater than 999 and not 000 as we have already looked for anomalies by writing a Map and Reducer function to get unique possible values.
Mapper Script: PD_CD_Validator_map.py
Reducer Script: Columns_Validator_reduce.py

Count of Valid Values: 5096648
Count of Invalid Values: 0
Count of Null Values: 4574

10. **PD_DESC** – PD Code Description

We looked for only blank spaces as this is a description as we have already looked for anomalies by writing a Map and Reducer function to get unique possible values.
Mapper Script: PD_DESC_Validator_map.py
Reducer Script: Columns_Validator_reduce.py

Count of Valid Values: 5101225
Count of Invalid Values: 0
Count of Null Values: 7

11. **CRM_ATPT_CPTD_CD** – Crime Status
As it has only two values we got from a Map and Reduce function, we looked for blank values and "COMPLETED" and "ATTEMPTED" values, considered other as "INVALID".
Mapper Script: CRM_ATPT_CPTD_CD_Validator_map.py
Reducer Script: Columns_Validator_reduce.py

Count of Valid Values: 5101224
Count of Invalid Values: 1
Count of Null Values: 7

12. **LAW_CAT_CD** – Level of offence
As it has only 3 correct values, blank spaces were assigned as "NULL" and all others as "INVALID" values.
Mapper Script: LAW_CAT_CD _Validator_map.py
Reducer Script: Columns_Validator_reduce.py

Count of Valid Values: 5101231
Count of Invalid Values: 1
Count of Null Values: 0
13. **JURIS_DESC** – Responsible Jurisdiction
We got the all unique possible values of this column by writing a Map and Reducer. Other than these values were considered as "INVALID" and blanks as "NULL".
Mapper Script: JURIS_DESC_Validator_map
Reducer Script: Columns_Validator_reduce.py

Count of Valid Values: 5101231
Count of Invalid Values: 1
Count of Null Values: 0

14. **BORO_NM** – Borough Name
We just checked if this column contains one of the value from 5 boroughs, otherwise "INVALID", and "NULL" for blank spaces.
We also checked the Latitude and Longitude of the location with respect to NYC shape file and came to conclusion that few of the borough names were not according to the respective location. So, we extracted the borough names from shape files and corrected in csv data.
Mapper Script: BORO_NM_Validator_map.py
Reducer Script: Columns_Validator_reduce.py

Count of Valid Values: 5100759
Count of Invalid Values: 0

Count of Null Values: 463

15. **ADDR_PCT_CD** – Precinct Code

We looked for the length of code and its value as integer, not greater than 999 and not 000 as we have already looked for anomalies by writing a Map and Reducer function to get unique possible values.
Mapper Script: ADDR_PCT_CD_Validator_map.py
Reducer Script: Columns_Validator_reduce.py

Count of Valid Values: 5100841
Count of Invalid Values: 1
Count of Null Values: 390

16. **LOC_OF_OCCUR_DESC**- Location of incident occurrence

We looked for only blank spaces as this is a description as we have already looked for anomalies by writing a Map and Reducer function to get unique possible values.
Mapper Script: LOC_OF_OCCUR_DESC_map.py
Reducer Script: Columns_Validator_reduce.py

Count of Valid Values: 3974096
Count of Invalid Values: 0
Count of Null Values: 1127126

17. **PREM_TYP_DESC** – Premises of incident occurrence

We looked for only blank spaces as this is a description as we have already looked for anomalies by writing a Map and Reducer function to get unique possible values.
Mapper Script PREM_TYP_DESC_Validator_map.py
Reducer Script: Columns_Validator_reduce.py

Count of Valid Values: 5067944
Count of Inalid Values: 0
Count of Null Values: 33278

18. **PARKS_NM** – Name of the Parks

We looked for only blank spaces as this is a description as we have already looked for anomalies by writing a Map and Reducer function to get unique possible values.
Mapper Script: PARKS_NM_Validator_map .py
Reducer Script: Columns_Validator_reduce.py

Count of Valid Values: 7600
Count of Invalid Values: 0
Count of Null Values: 5093632

19. **HADEVELOPT**- NYCHA Housing

We looked for only blank spaces as this is a description as we have already looked for anomalies by writing a Map and Reducer function to get unique possible values.
Mapper Script: HADEVELOPT_Validator_map.py
Reducer Script: Columns_Validator_reduce.py

Count of Valid Values: 253206

Count of Invalid Values: 0
Count of Null Values: 4848026

20. **X_COORD_CD** – X coordinate of location
We have checked if the value of column resides in the X coordinates of New York city else considered as “INVALID”, in case of blank considered as “NULL”.
Mapper Script: X_COORD_CD_Validator_map.py
Reducer Script: Columns_Validator_reduce.py
Count of Valid Values: 4913076
Count of Invalid Values: 0
Count of Null Values: 188146

21. **Y_COORD_CD** - Y coordinate of location
We have checked if the value of column resides in the Y coordinates of New York city else considered as “INVALID”, in case of blank considered as “NULL”.
Mapper Script Y_COORD_CD_Validator_map.py
Reducer Script: Columns_Validator_reduce.py
Count of Valid Values: 4913076
Count of Invalid Values: 0
Count of Null Values: 188146

22. **Latitude**
We have checked if the value of column lies in the latitude range of New York city else considered as “INVALID”, in case of blank considered as “NULL”.
Mapper Script: Latitude_Validator_map.py
Reducer Script: Columns_Validator_reduce.py

Count of Valid Values: 4913076
Count of Inalid Values: 0
Count of Null Values: 188146

23. **Longitude**
We have checked if the value of column lies in the longitude range of New York city else considered as “INVALID”, in case of blank considered as “NULL”.
Mapper Script: Longitude_Validator_map.py
Reducer Script: Columns_Validator_reduce.py

Count of Valid Values: 4913076
Count of Inalid Values: 0
Count of Null Values: 188146

24. **Lat_Lon**- (Latitude, Longitude)
We check if this latitude and longitude exists within the polygon of shape file of New York city and considered it as “VALID” else, considered it as “INVALID”, and blank spaces as “NULL” respectively.
Mapper Script: Lat_Lon_Validator_map.py
Reducer Script: Columns_Validator_reduce.py

Other data quality challenges:

1.Collecting External (High Quality) Data Sources:

Every value in JURIS_DESC belongs to one set. Identifying that set is was challenging.
Solution: Columns “JURIS_DESC” has had no set of defined set of values. However, by inspection, we could see pattern that, values belong to certain set. We collected names of department available for with map -reduce approach with maximum occurrence method. This data source helped us to remove unforced errors people make.

2. Repair by value modification / Resolve contradicting information

Used latitude and longitude of the data file “Lat_Lon” to check if the borough name “BORO_NM” mentioned in data file is right or not. If not, replace the “BORO_NM” with the respective latitude and longitude Borough name from json file.

In this we used an external json file of New York City that has five borough name and has their respective continuous boundary coordinate. So given the latitude and longitude , it first tried to search whether this coordinate lies in one of the borough. If the latitude and longitude lies in one of the borough , we find out the respective borough name from the json file. So the borough name from json file is compared with the borough name “BORO_NM” in data file. If it does not match, then the “BORO_NM” in data file is replaced with the borough name corresponding to its respective latitude and longitude.

3. Decide if two records represent the same entity / String Similarity functions

Values of three columns, CRM_ATPT_CPTD_CD, LAW_CAT_CD belong to set of values. This highly defined set of values can be used to correct any spelling mistakes. Given no values in sets are very similar, we have used 0.8 ratio match and correct any spelling mistakes in above mentioned columns. We have made use of fuzzy string comparison to find out if two records represent same entity as correct as possible

4. Use of Regular Expression

We have made sure that below columns follow the specific format of date and time using regular expression CMPLNT_FR_DT, CMPLNT_FR_TM, CMPLNT_TO_DT,CMPLNT_TO_TM, RPT_DT

5. Modify data to improve quality

We have checked all the possible valid values for all columns, specified invalid values as “INVALID” and empty values as “NULL”. Using this data we created a new CSV file which will make analysis easier.

PART II : DATA EXPLORATION
Data quality issues faced

1. Collection of neighborhoods:
In previously cleaned data of Crimes neighborhood data was not available. So based on the latitude and longitude we collected neighborhood data i.e. NTACode from NYC's shape file.

2. Collection of Population, Poverty rate, Unemployment rate and Median income per year per neighborhood:
We did not find the above data on NYC Open data, we had to gather the data from Furman center's pdf files and generate new csv. Also, this data is from 2006 to 2014, so the crimes from 2015 and 2016 we have not considered while drawing below hypothesis.
CSV : CombinedProperties_vs_Neighbourhoods.csv

3. Complaints Data Cleaning :
We have also used the Complaints data set listed above in order to find the safest and better neighbourhood place to live in each borough. In that case, we had to clean all the columns such that check for valid, invalid data and use "NULL" where data is not present.
Script used for cleaning:

map311.py
reduce311.py

Experimental Setup Description:

We used Map Reduce to clean NYC 311 Complaint data that had 52 columns. So this is the map reduce code:
map311.py
reduce311.py

Initially we ran with 10 cluster nodes but it would take lot of time to run. So to reduce the running time, we optimized it by changing the number of reducers to 1000 to clean complaint data. We had to include a wrapper mod.sh so that we could run shapefile in hadoop.
This is the command we used to run the code:

hjs -D mapred.reduce.tasks=1000 -files "mod.sh,map311.py,reduce311.py,query.txt" -mapper "mod.sh map311.py" -reducer "mod.sh reduce311.py" -input /user/aks629/311.csv -output /user/aks629/final_complaint.out

We then used hfs -getmerge to combine those 1000 output part file and get that file locally. This txt file was converted to csv file by using a simple python code convert_txt_to_csv.py .

We used map reduce code to perform JOIN (set) operation by NTA code . We had two files CombinedProperties_vs_Neighbourhoods.csv and Crimerate.csv. So we had to merge this two file to one by their NTA code , so that it would be easier to plot the data and prove our hypothesis by Neighbourhood. Following are the map reduce code:
CombinedRatesperYear_vs_CrimeRate_map.py
CombinedRatesperYear_vs_CrimeRate_reduce.py

Similar JOIN operation was performed for CombinedProperties_vs_Neighbourhoods.csv and

311_Complaintrate.csv. We used one mapper and one reducer.
f1_f2_map.py
f1_f2_reduce.py

We wrote the Map Reduce code to assemble all the coordinates where crime was committed in crime data to one list by Neighbourhood name. We used one mapper and 4 reducers.
map_NTA_coordinates.py
reduce_NTA_coordinates.py

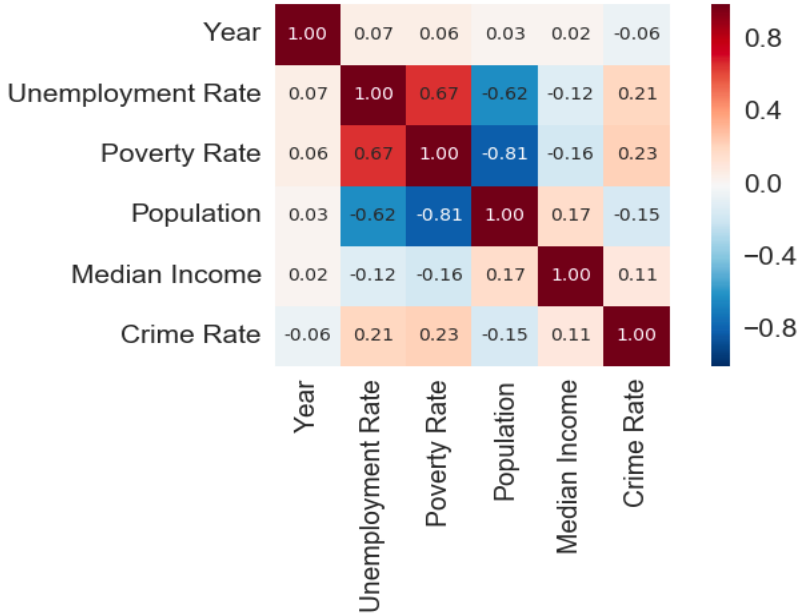
Analysis:

We have generated New_Combined_Properties_vs_Neighbourhood.csv file using, below scripts where all the properties such as Population, Crime rate, Unemployment rate, Poverty rate, Median Income, Neighborhood name and year. This data we have used to generate the below pearson coefficient matrix.

Scripts used -
CombinedRatesperYear_vs_CrimeRate_map.py
CombinedRatesperYear_vs_CrimeRate_reduce.py
Input - Properties_vs_Neighbourhoods.csv and CrimeRate_vs_Year.csv

Attributes taken into consideration:

- Crime Rate
- Unemployment Rate
- Poverty Rate
- Median Income
- Population
- Complaint rate



Hypothesis 1:

Finding top 3 best places to live and worst place in new york city borough wise.

Here hypothesis is investigate to between Crime rate, Complaint rate and neighbourhood

Techniques used for Analysis: Principal component analysis.

.

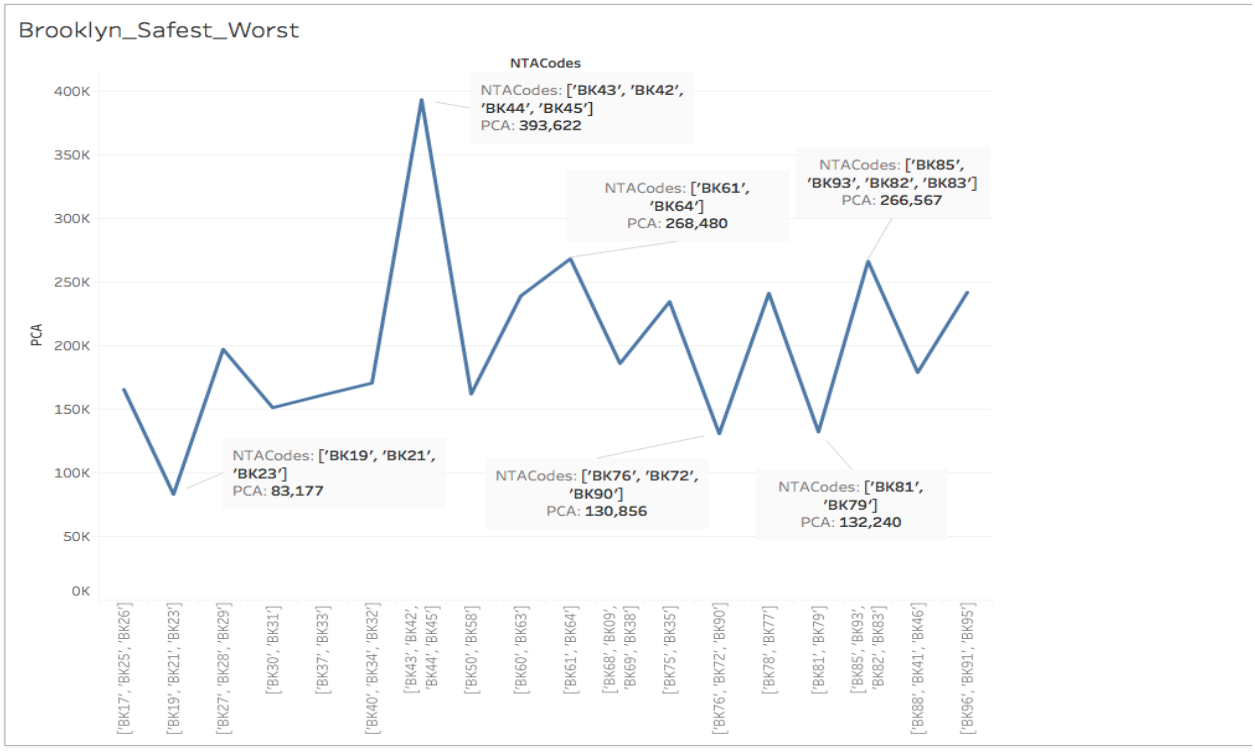
Analysis:

- Collected number of crime and complaint for each neighbourhood. Analysed crime and complaint rate, found out that correlation coefficient between complaint and crime is x which is quite high. This coefficient signifies that crime and complaint can be assumed to be strongly related.
- Hence we used Principal component analysis to reduce the dimension of complaint and crime.
- To get better insightful visualization, graph has been plot with neighbourhood and new reduced data from complaint and crime which can be found below.
- Also, from the graphs we can conclude that higher the PCA value, higher number of crimes and complaints, and lesser PCA value suggests the better place to live.

  Scripts Used - best_place_live.py

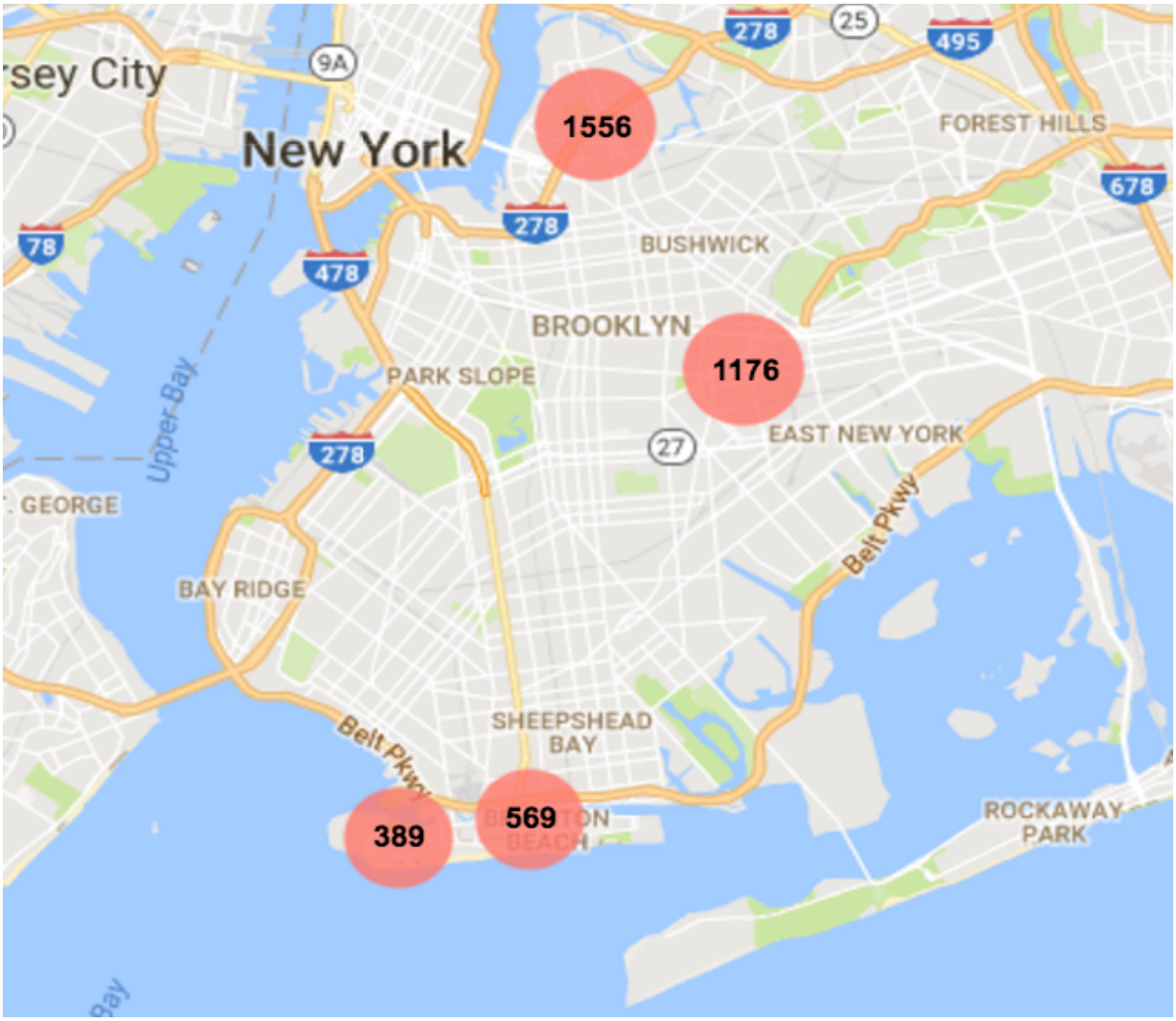
PCA's for each neighbourhood as below:

1. **Brooklyn**

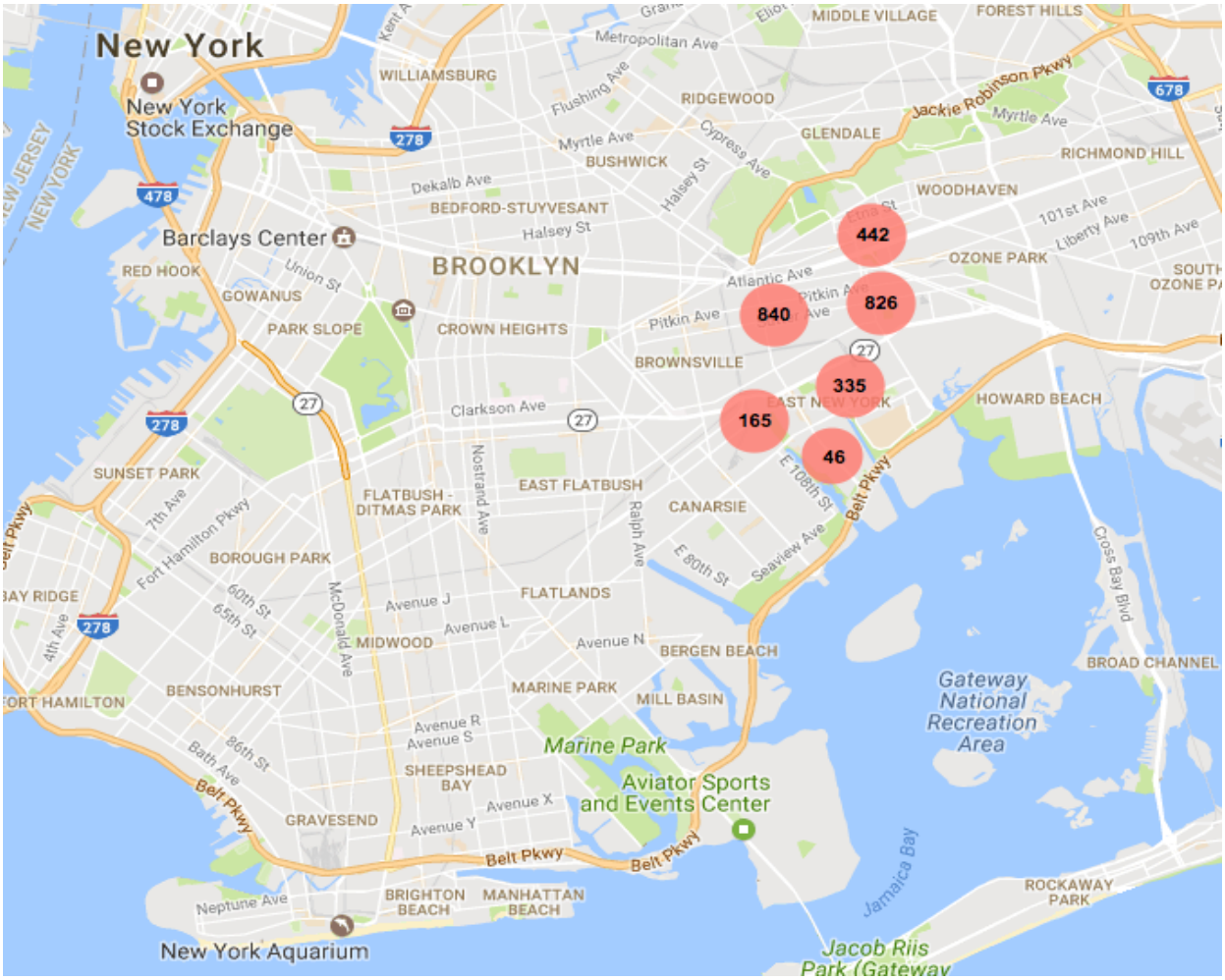


From the graph above and list of coordinates from Crime data, we have plotted the top 3 safe and worst locations in google map as below.

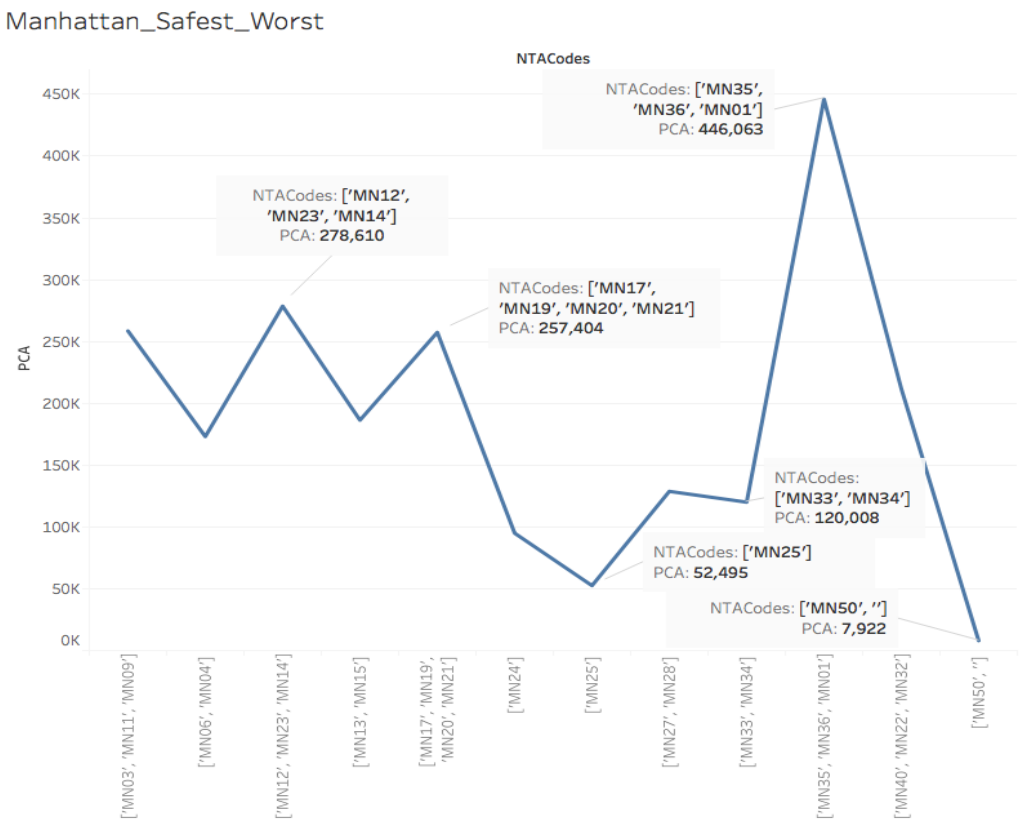
Safer places:



Worst Places:

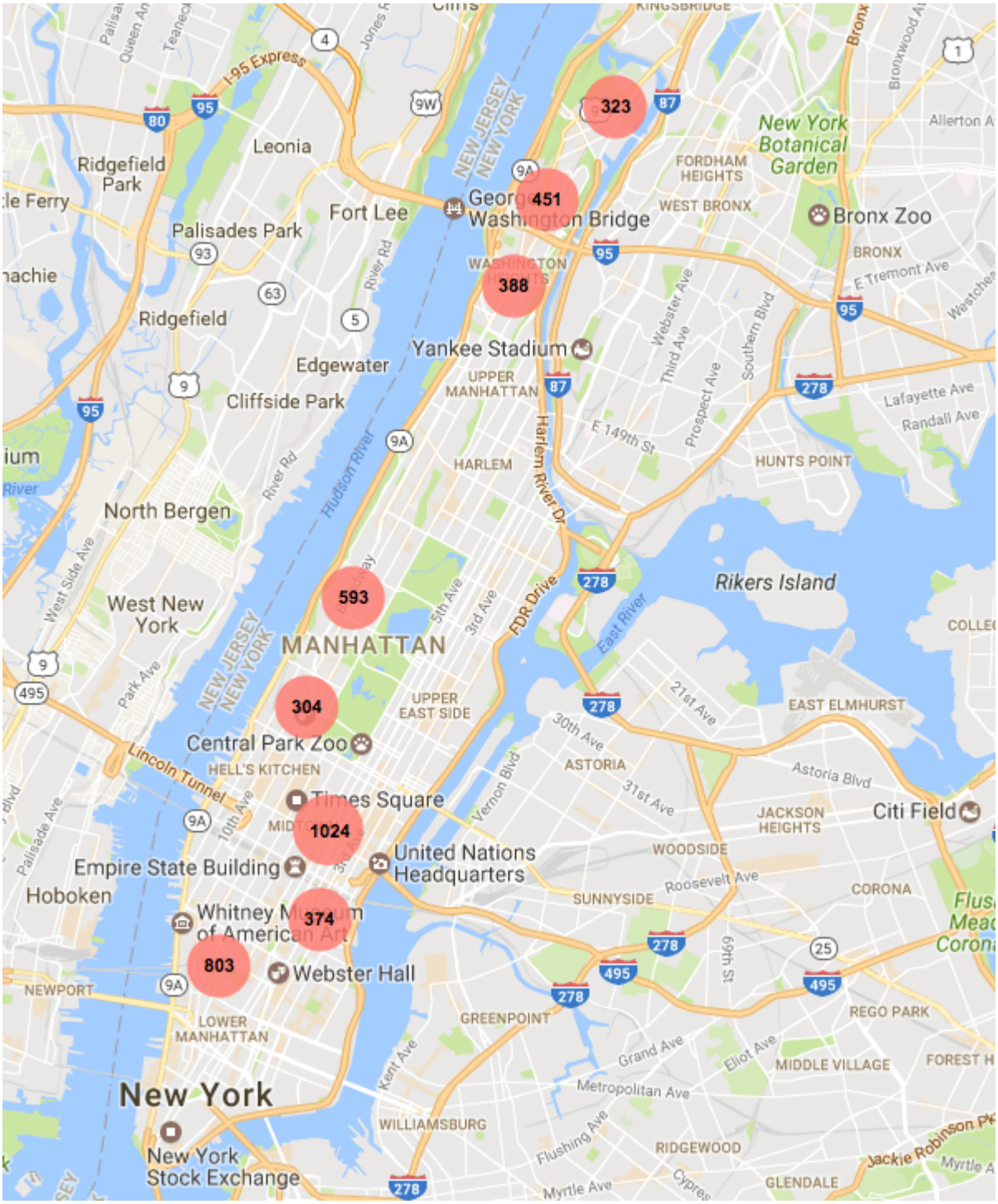


2. Manhattan

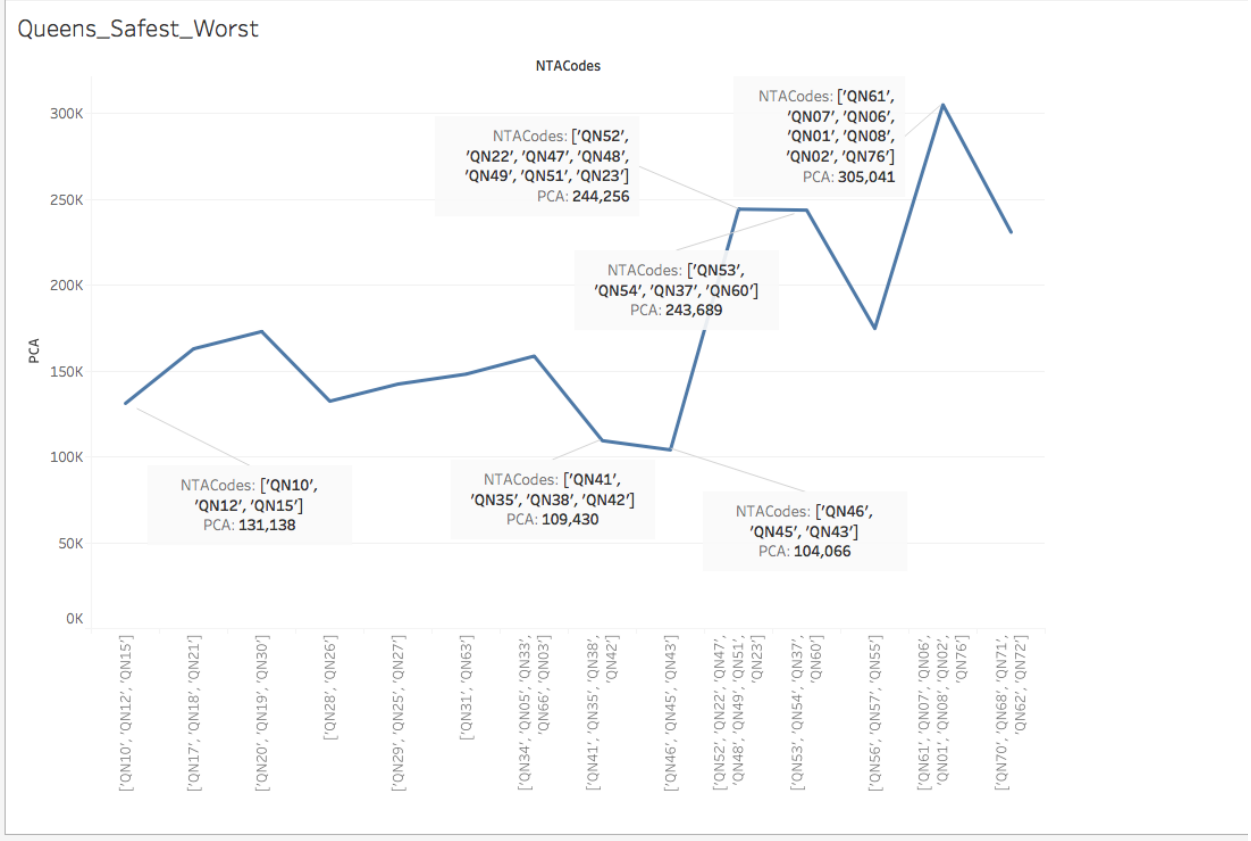

Manhattan_Safest_Worst
NTACodes
NTACodes: ['MN35', 'MN36', 'MN01']
PCA: 446,063
NTACodes: ['MN12', 'MN23', 'MN14']
PCA: 278,610
NTACodes: ['MN17', 'MN19', 'MN20', 'MN21']
PCA: 257,404
NTACodes: ['MN33', 'MN34']
PCA: 120,008
NTACodes: ['MN25']
PCA: 52,495
NTACodes: ['MN50', '']
PCA: 7,922
PCA
450K
400K
350K
300K
250K
200K
150K
100K
50K
0K
['MN03', 'MN11', 'MN09']
['MN06', 'MN04']
['MN12', 'MN23', 'MN14']
['MN13', 'MN15']
['MN17', 'MN19', 'MN20', 'MN21']
['MN24']
['MN25']
['MN27', 'MN28']
['MN33', 'MN34']
['MN35', 'MN36', 'MN01']
['MN40', 'MN22', 'MN32']
['MN50', '']

Safer places:

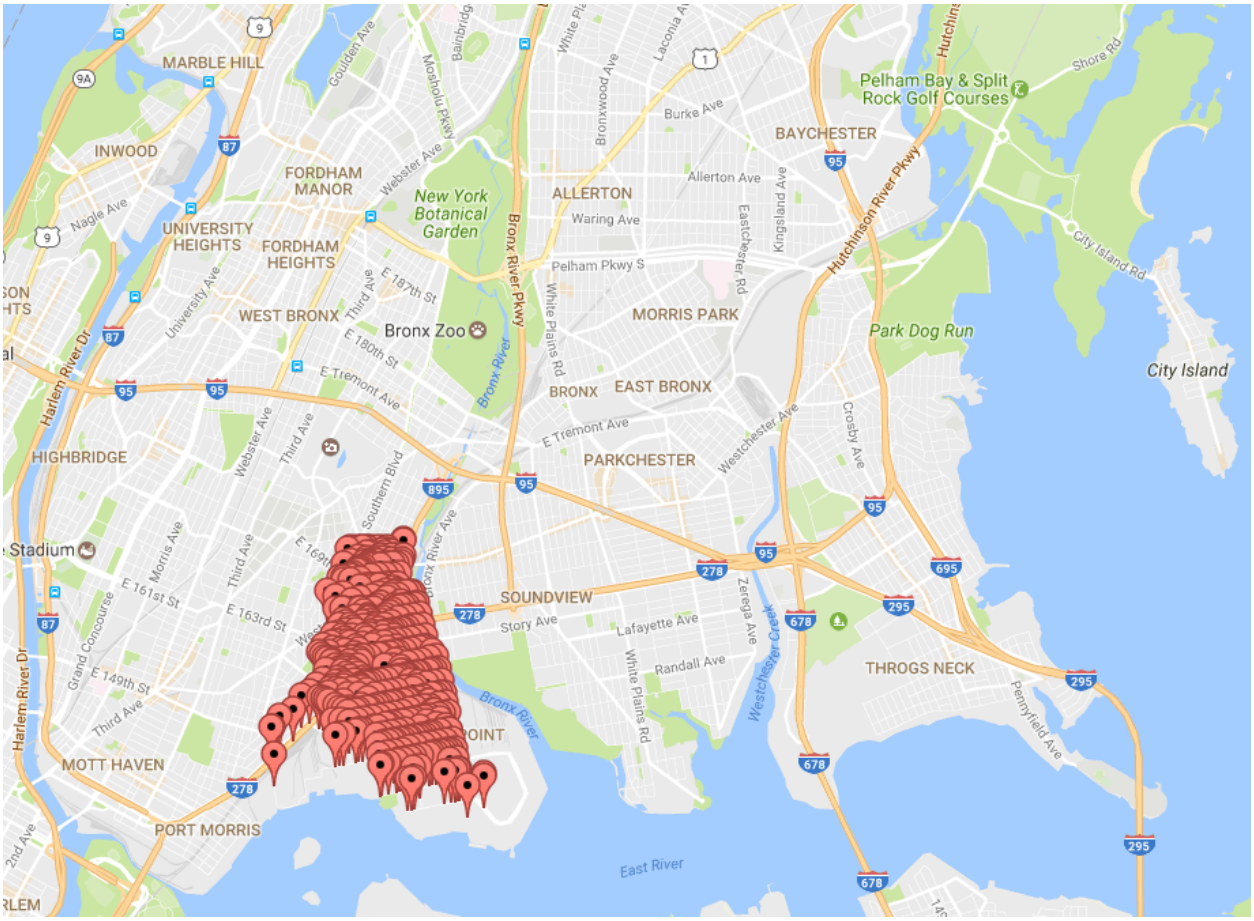
Worst Places in Manhattan:



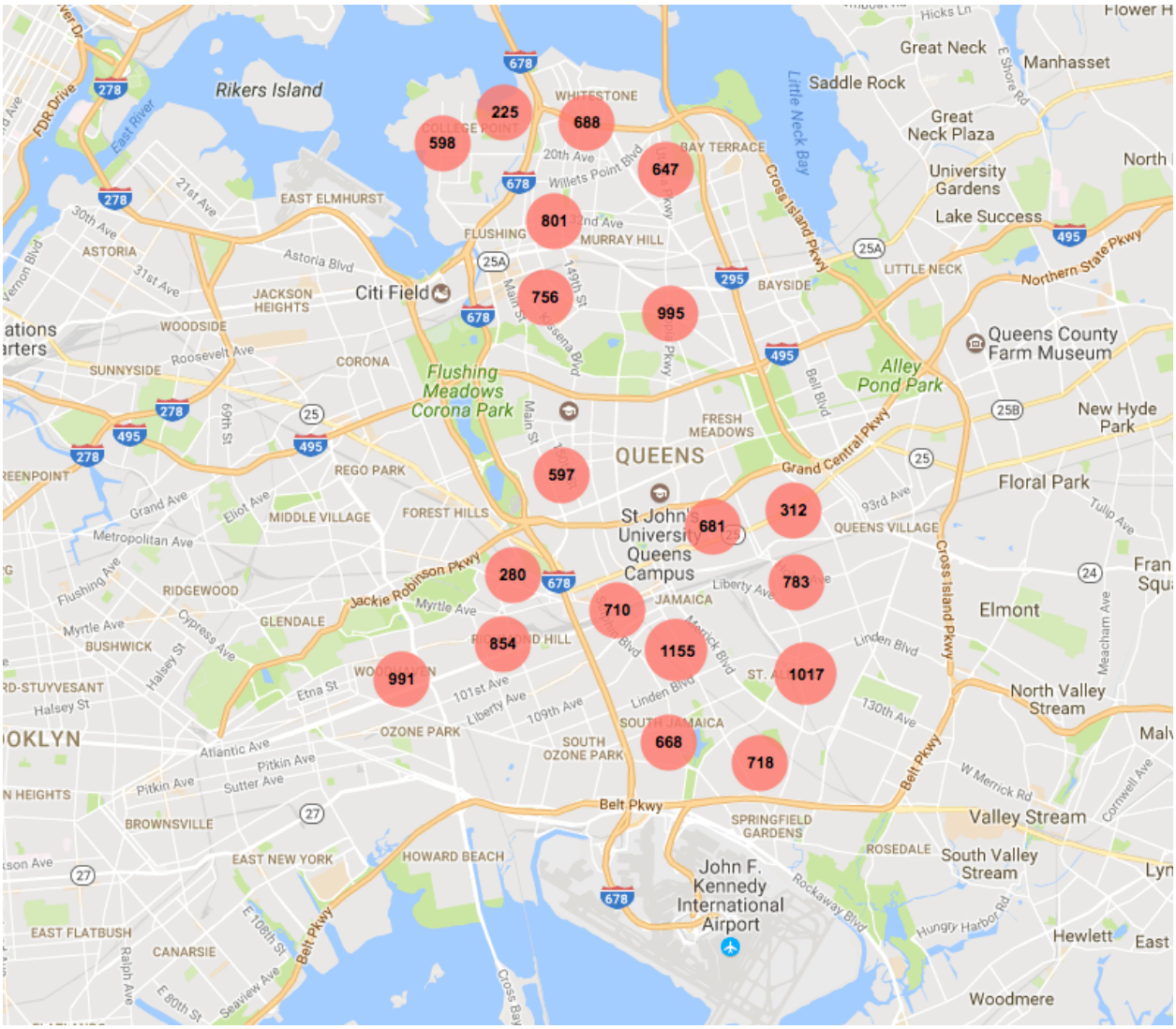
3. Queens


Queens_Safest_Worst
NTACodes
PCA
300K
250K
200K
150K
100K
50K
0K
NTACodes: ['QN61', 'QN07', 'QN06', 'QN01', 'QN08', 'QN02', 'QN76']
PCA: 305,041
NTACodes: ['QN52', 'QN22', 'QN47', 'QN48', 'QN49', 'QN51', 'QN23']
PCA: 244,256
NTACodes: ['QN53', 'QN54', 'QN37', 'QN60']
PCA: 243,689
NTACodes: ['QN10', 'QN12', 'QN15']
PCA: 131,138
NTACodes: ['QN41', 'QN35', 'QN38', 'QN42']
PCA: 109,430
NTACodes: ['QN46', 'QN45', 'QN43']
PCA: 104,066
['QN10', 'QN12', 'QN15']
['QN17', 'QN18', 'QN21']
['QN20', 'QN19', 'QN30']
['QN28', 'QN26']
['QN29', 'QN25', 'QN27']
['QN31', 'QN63']
['QN34', 'QN05', 'QN33', 'QN66', 'QN03']
['QN41', 'QN35', 'QN38', 'QN42']
['QN46', 'QN45', 'QN43']
['QN52', 'QN22', 'QN47', 'QN48', 'QN49', 'QN51', 'QN23']
['QN53', 'QN54', 'QN37', 'QN60']
['QN56', 'QN57', 'QN55']
['QN61', 'QN07', 'QN06', 'QN01', 'QN08', 'QN02', 'QN76']
['QN70', 'QN68', 'QN71', 'QN62', 'QN72']

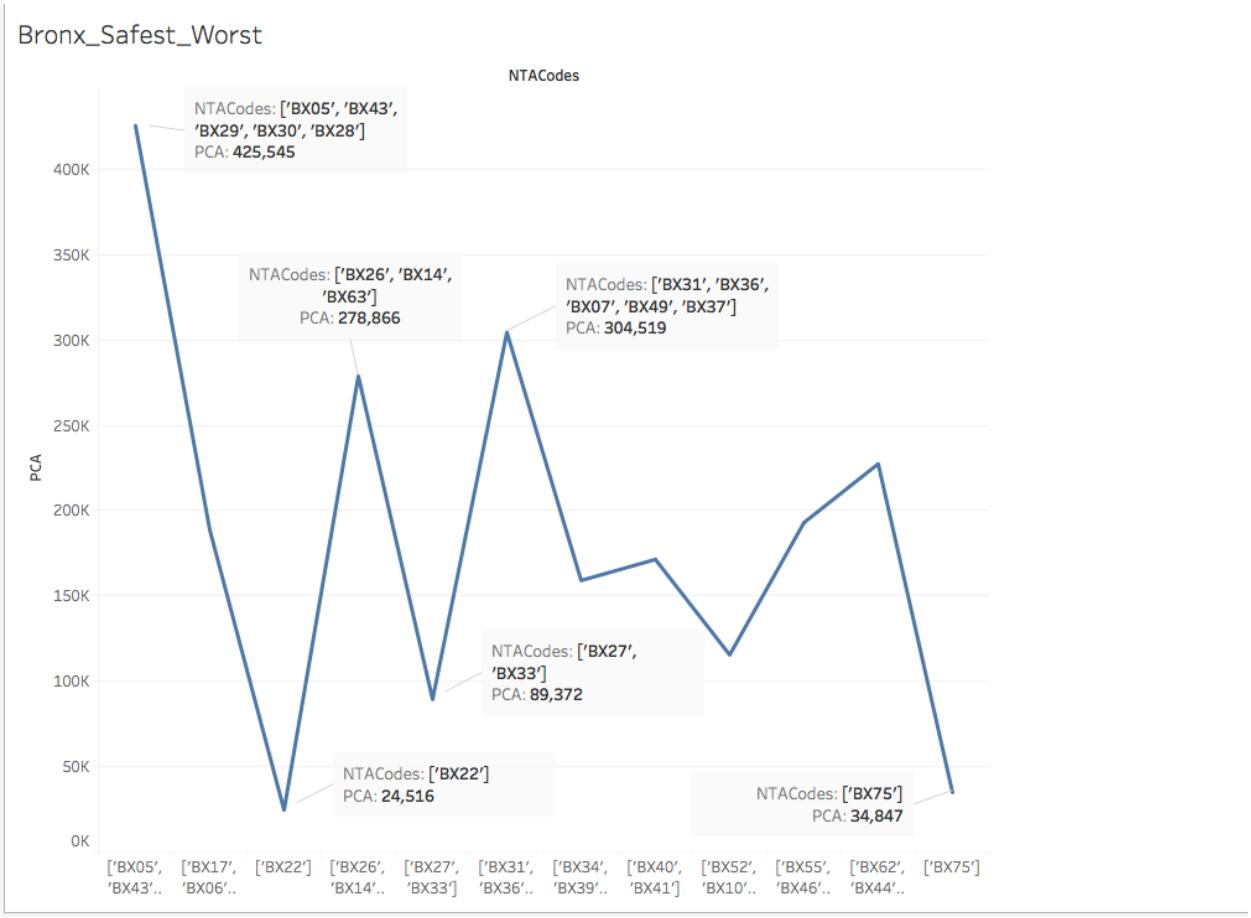
Safer Places:



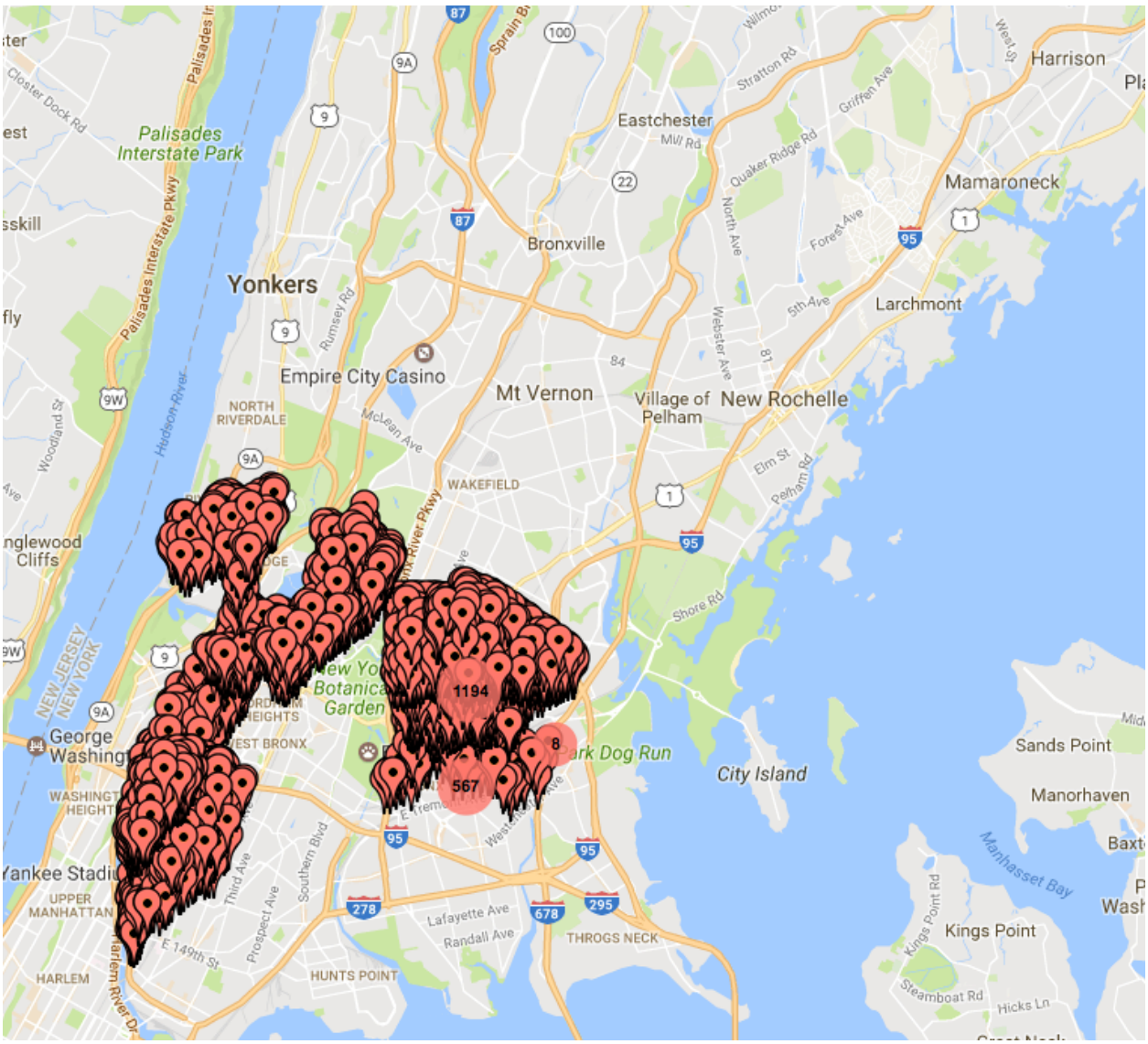
Worst Places:



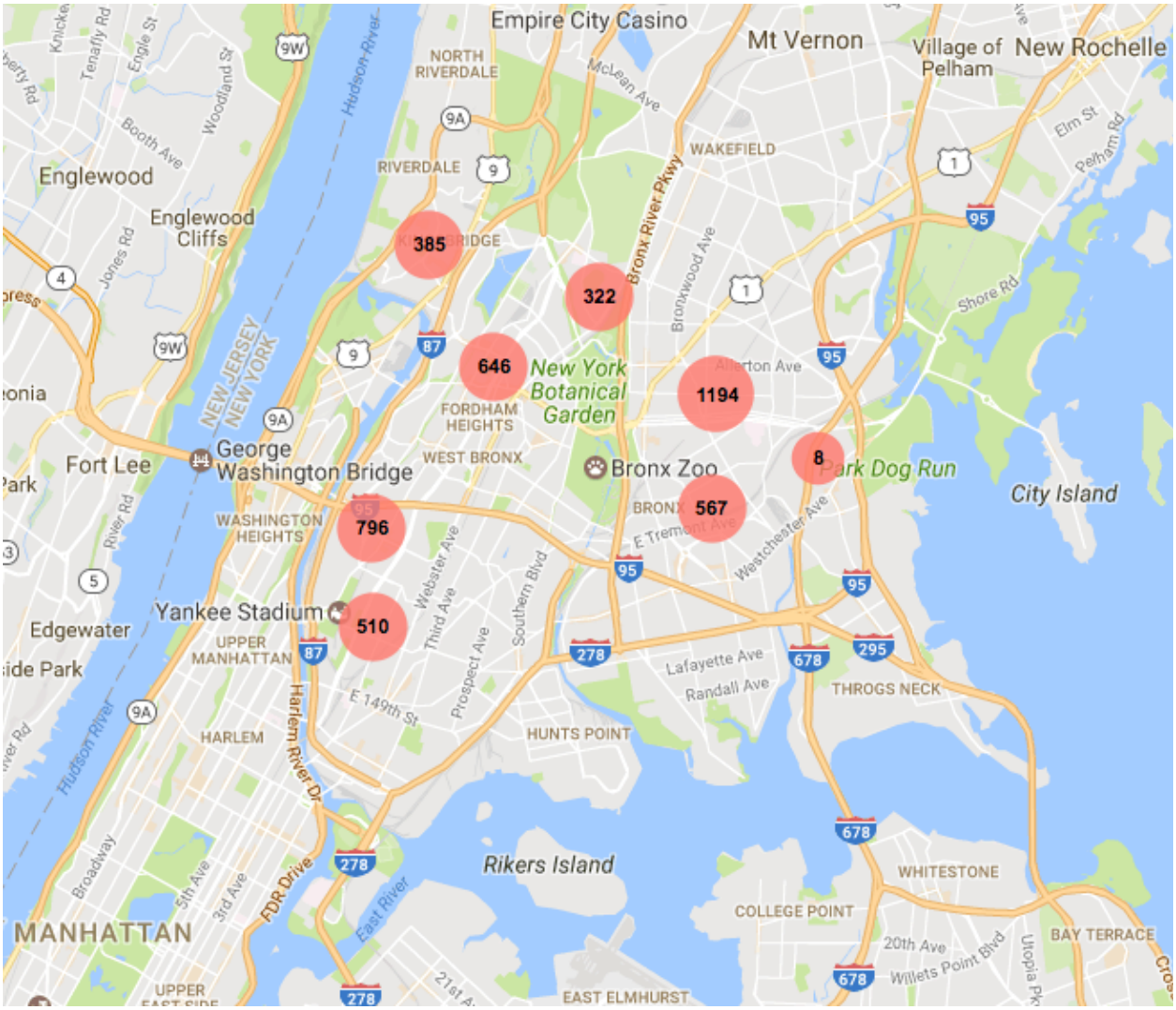
4. Bronx


Bronx_Safest_Worst
NTACodes
NTACodes: ['BX05', 'BX43', 'BX29', 'BX30', 'BX28']
PCA: 425,545
NTACodes: ['BX26', 'BX14', 'BX63']
PCA: 278,866
NTACodes: ['BX31', 'BX36', 'BX07', 'BX49', 'BX37']
PCA: 304,519
NTACodes: ['BX27', 'BX33']
PCA: 89,372
NTACodes: ['BX22']
PCA: 24,516
NTACodes: ['BX75']
PCA: 34,847
PCA
400K
350K
300K
250K
200K
150K
100K
50K
0K
['BX05', 'BX43'..
['BX17', 'BX06'..
['BX22']
['BX26', 'BX14'..
['BX27', 'BX33']
['BX31', 'BX36'..
['BX34', 'BX39'..
['BX40', 'BX41']
['BX52', 'BX10'..
['BX55', 'BX46'..
['BX62', 'BX44'..
['BX75']

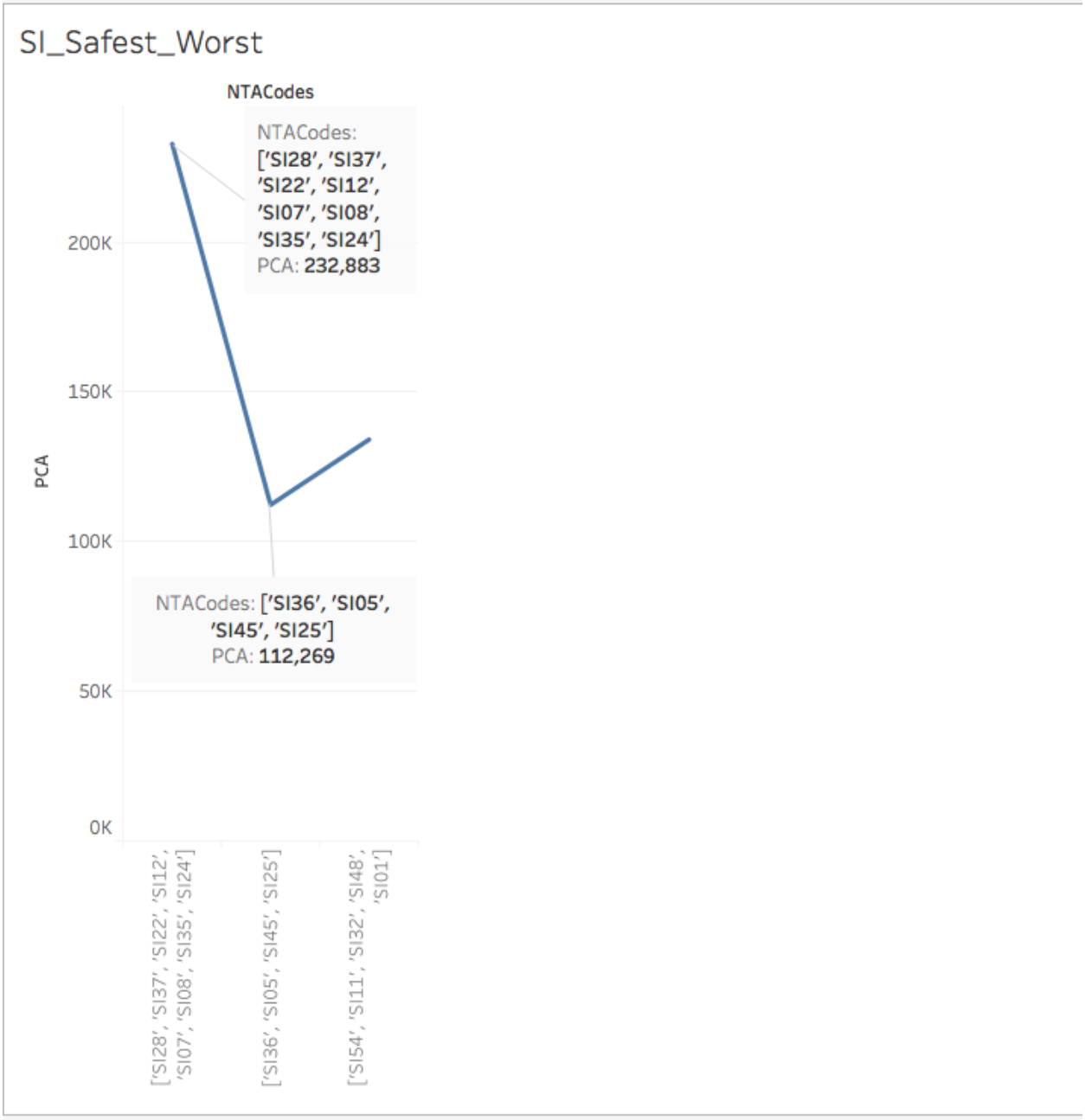
Safer Places:



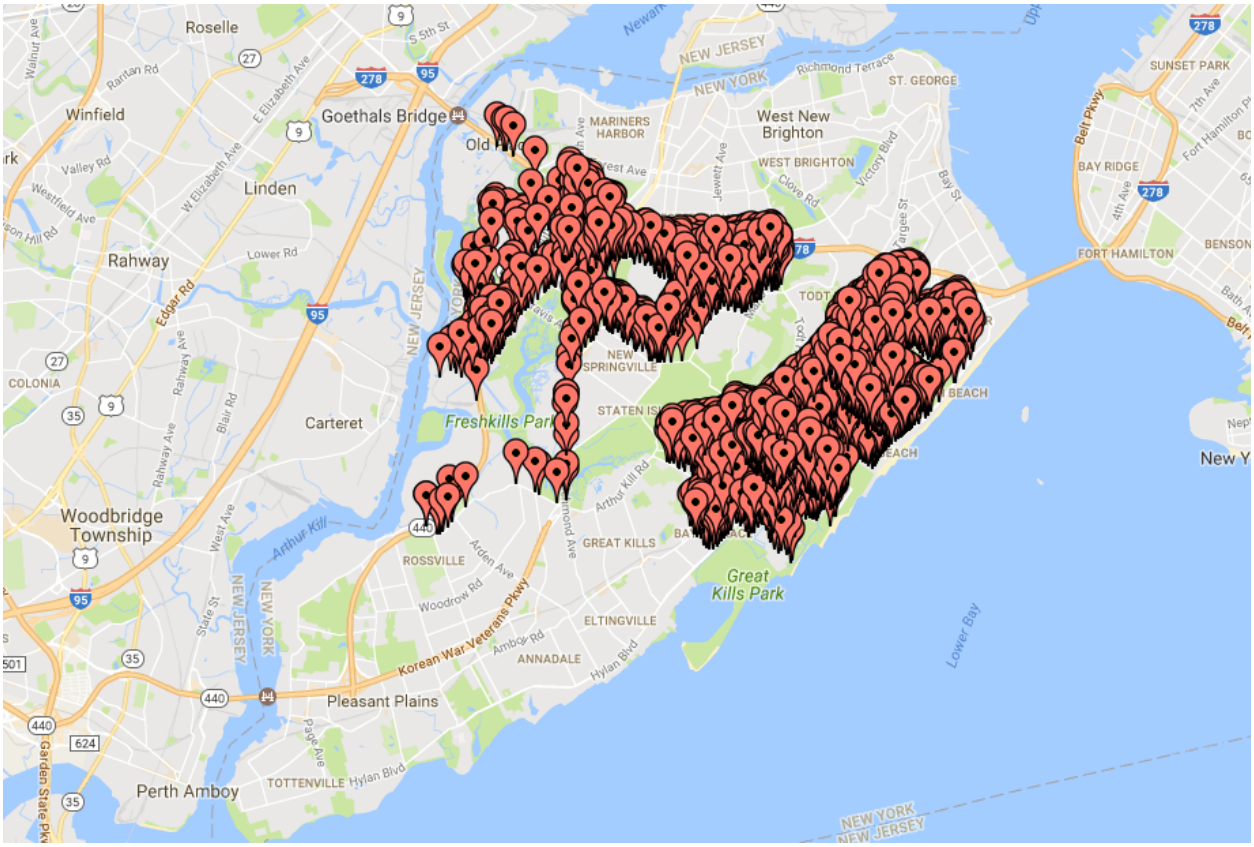
Worst Places:



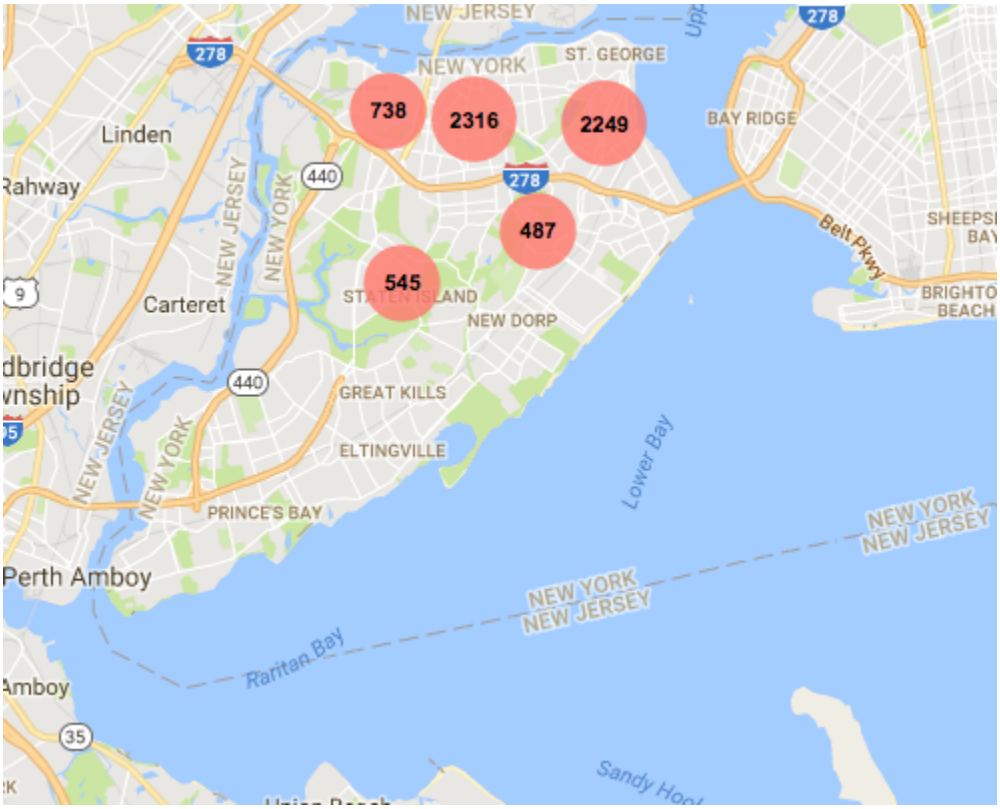
5. Staten Island



Safer Places:



Worst Places:



Hypothesis 2:

Crime Rate is directly dependent on unemployment rate.

We assumed that crime rate is related to unemployment rate but we found out that pearson coefficient between Crime and unemployment are is 0.21. which indicates that indicates that Crime and unemployment are not so much related. We can almost assume that they are unrelated.

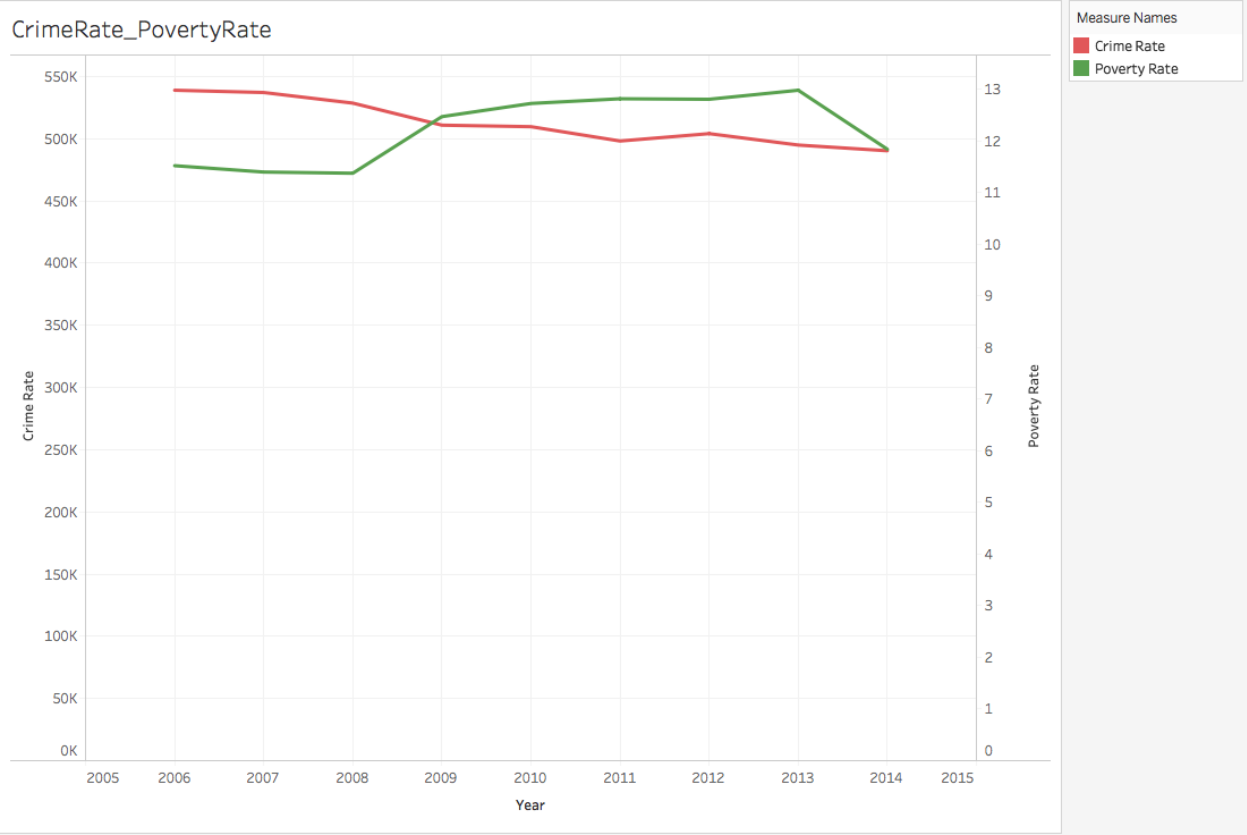
Below is the graph which also supports that these two attributes are unrelated.

Hypothesis 3:

Crime Rate is directly dependent on poverty rate.

We suspected that poverty rate could play part in Crime rate. From above matrix we found that pearson coefficient between Crime and poverty rate is 0.23 which also indicates that Crime and poverty rate are are not so much related. We can almost assume that they are unrelated.

Below is the graph to support that they are much related.

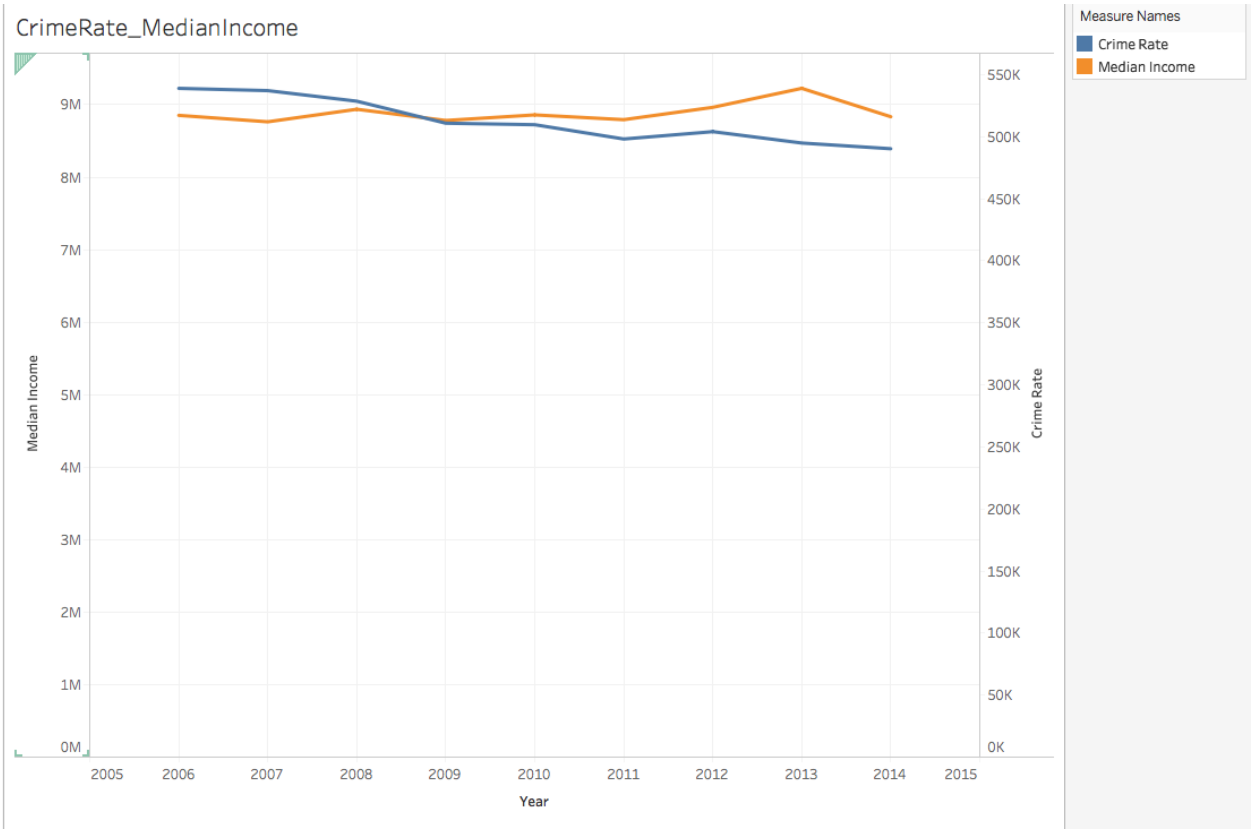


Hypothesis 4:

Crime Rate is indirectly / inversely proportional to Median income.

We suspected that median income could play part in Crime rate. From above matrix we found that pearson coefficient between Crime and median income is 0.11 which also indicates that Crime and median income are are not so much related

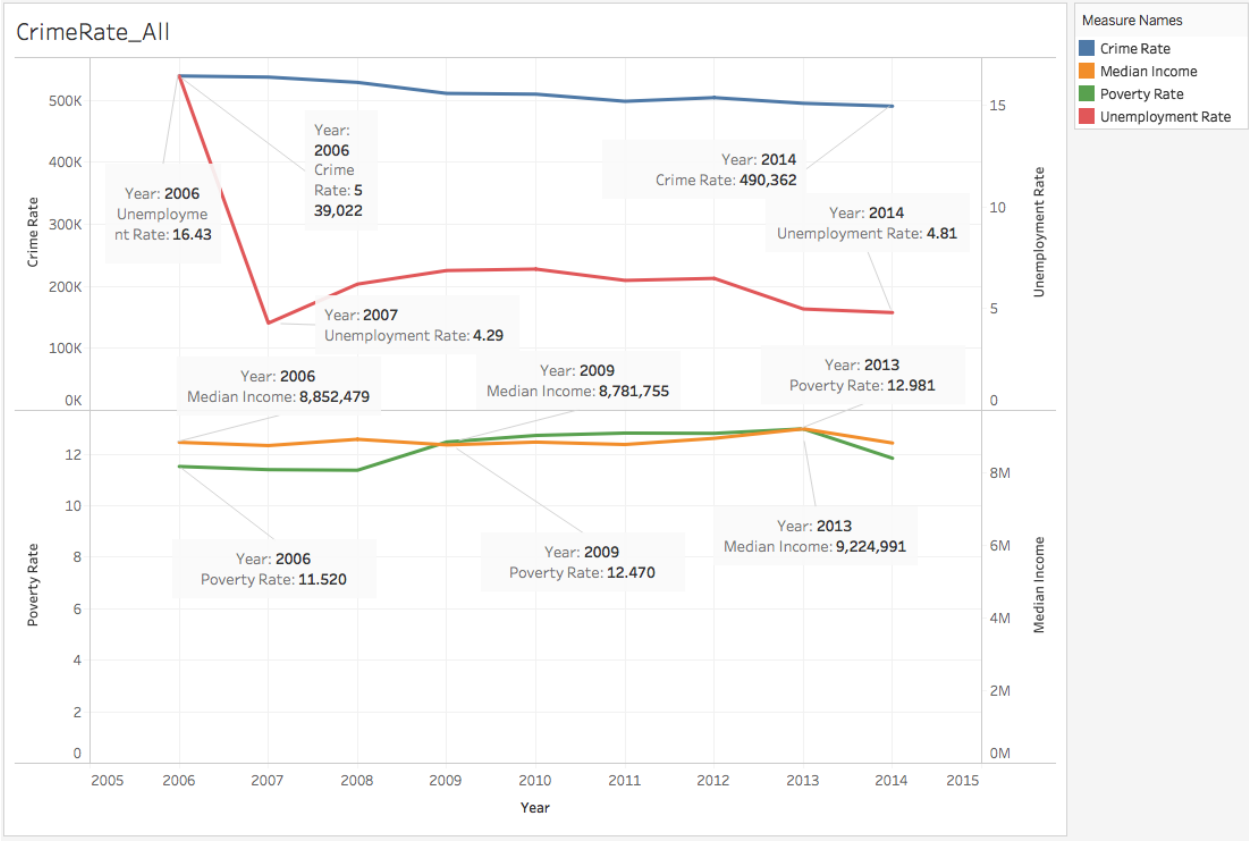
Below is the graph to support that they are not related. Although, at the starting of the graph it can be seen that it is inversely proportional as Crime rate is higher when median income is lower but, it it does not support the same rule till the end of graph.

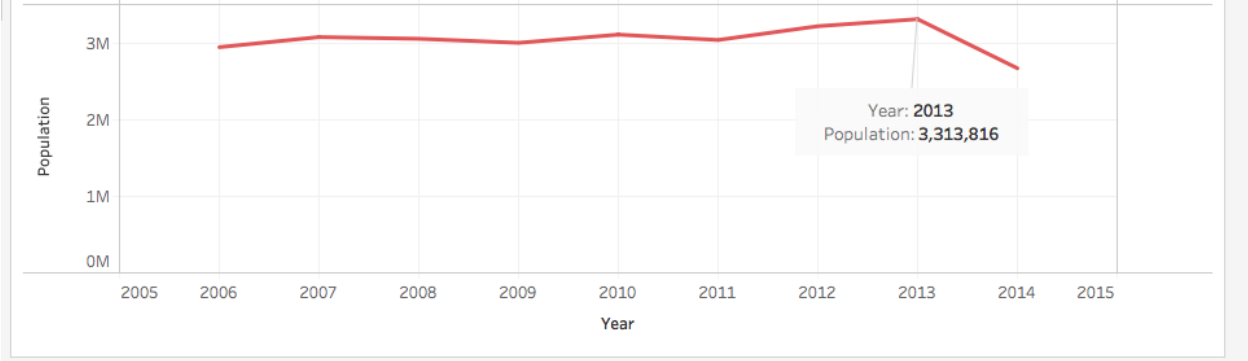


Unusual Cases:

We have plotted all the attributes i.e. crime rate, poverty rate, median income and unemployment rate in one graph in order to see any unusual cases.

1. In 2006, one can see unemployment rate was highest as well as crime rate so we can conclude higher crimes were because of higher unemployment.
2. Although in 2007, unemployment rate dropped to lowest i.e employment was much better still there was a slight drop in the crime rate. In 2007, it shows better employment did not affect the crime rate to a greater extent.
3. During 2008-09, unemployment rate seems to be increasing, but at the same time crime rate is decreasing. First 3 cases states that crime rate and unemployment rate does not hold any relation which supports our hypothesis 2.
4. In 2006, median income was higher so the poverty rate seems to be lowest, so we can conclude that they are obviously inversely proportional as this continous till 2013. After this we can see that both of them are decreasing which seems to be unusual. Possibility can be population was higher in 2013 which is proved with our population graph plotted below. Also, in spite of higher population, there was better employment and lesser crime rate which is almost equal to the lowest.




3M
2M
1M
0M
Population
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
Year
Year: 2013
Population: 3,313,816

INDIVIDUAL CONTRIBUTION:

Amitesh Sah (aks629)

- Responsible to validate and clean 8 columns of Crime data out of 24 columns by writing mapreduce code
- 3 out of 8 analysis in part 1 was carried i.e.Borough vs Crime Type,Time of the Day Analysis vs Crimes, Days of Week vs Crimes Analysis
- Used Shapefile to plot data and taught my teammates how to plot nyc data using geopandad

- Wrote and edited the report where necessary
- Collected the data of Population, Poverty rate, Unemployment rate and Median income per year per neighbourhood from Furman center's pdf files and generated new csv . I did for Queens Borough
- Responsible to clean 17 columns of 311 NYC Complaint data.
- Wrote code to perform JOIN operation between files CombinedProperties_vs_Neighbourhoods.csv and 311_Complaintrate.csv.
- Found out the top 3 best places to live and worst place in new york city for Queens Borough.
- Tested Hypothesis Crime Rate is directly dependent on unemployment rate.
- Discussed with the team mates and helped them if faced with with any problem

Prasad Bhagwat(pgb252)

- Responsible to validate and clean 8 columns of Crime data out of 24 columns by writing mapreduce code
- 3 out of 8 analysis in part 1 was carried i.e.Major Crimes vs Borough,Crimes per Year Analysis,Drug related Crimes per Boroughs
- Used Regex and even taught my teammates how to use it.
- Wrote and edited the report where necessary
- Collected the data of Population, Poverty rate, Unemployment rate and Median income per year per neighbourhood from Furman center's pdf files and generated new csv. I did for Brooklyn, Bronx Borough
- Responsible to clean 17 columns of 311 NYC Complaint data.
- Wrote code to perform JOIN operation between files CombinedProperties_vs_Neighbourhoods.csv and crimerate.csv.
- Find out the top 3 best places to live and worst place in new york city for Brooklyn, Bronx borough.
- Tested Hypothesis Crime Rate is directly dependent on poverty rate.
- Discussed with the team mates and helped them if faced with with any problem

Tejaswi Vinod (tvg226)

- Responsible to validate and clean 8 columns of Crime data out of 24 columns by writing mapreduce code
- 2 out of 8 analysis in part 1 was carried i.e.Alcohol related Crimes per Boroughs, Average delays to report abusive crimes per days,
- Wrote and edited the report where necessary
- Used everyones result and plotted those graphs in Tableau and taught my teammates how to use Tableau.
- Collected the data of Population, Poverty rate, Unemployment rate and Median income per year per neighbourhood from Furman center's pdf files and generated new csv. I did for Manhattan, Staten Island Borough
- Responsible to clean 18 columns of 311 NYC Complaint data.
- Wrote the Map Reduce code to assemble all the coordinates where crime was committed in crime data to one list by Neighbourhood name.
- Found out the top 3 best places to live and worst place in new york city for Manhattan, Staten Island borough.
- Tested Hypothesis Crime Rate is indirectly / inversely proportional to Median income.
- Discussed with the team mates and helped them if faced with with any problem

SUMMARY

At the end of this project, we are able to figure out the few best places to live in NYC city and which one of them to avoid by Neighbourhood wise within a Borough. Also, we learnt that there is hardly any relationship between the poverty rate or unemployment rate or population on crime rate. It is not based on any other factors.

Learnings:

We have learnt to use the map reduce framework, deal with big data quality issues and techniques to resolve them, different techniques to analyse the big data and find out correlations between them as well as new tools- Tableau, Pandas.

REFERENCES

- NYPD Crime Dataset
  https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i
- Complaints
  (2010-2016) https://data.cityofnewyork.us/Social-Services/311/wpe2-h2i5
  (2009) https://data.cityofnewyork.us/Social-Services/new-311/9s88-aed8.
- Population, Unemployment rate and Poverty rate neighborhood wise 2006-2014
  http://furmancenter.org/research/sonychan
- Lecture and Lab notes taught in class
- Mining of Massive Data Sets (version 2.1), by Anand Rajaraman, Jure Leskovec and Jeff Ullman