PRIYA:

My part for the project is Data Munging.To begin with,I was given the data collected by my teammate for the year of 2010. The data collected was there, but like common problems during matching data, they were attributed in different divisions. But as it is said,”Your analysis is as good as your data”, I wanted to be sure of what I was providing my other team mates with.

DATA COLLECTED :

1. For the **CRIME** Analysis of New York city, the data available was only **PRECINCT WISE**. No other divisions were possible, viz. Borough wise or Zip code wise, or Census Tracts wise.
2. As we wanted to see the correlation between Crime and Education, we can needed NUMBER OF SCHOOLS, in the city which was available ZIP CODE WISE for the year 2010.
3. The regressors /independent variables we considered for multivariate regression were:

* **DEMOGRAPHY** : Available in the form of **CENSUS TRACTS** division
* **MEDIAN/MEAN INCOME** : Available in the form of **CENSUS TRACTS** division
* **POPULATION BY GENDER AND AGE**: Available in the of **ZIP CODE** division

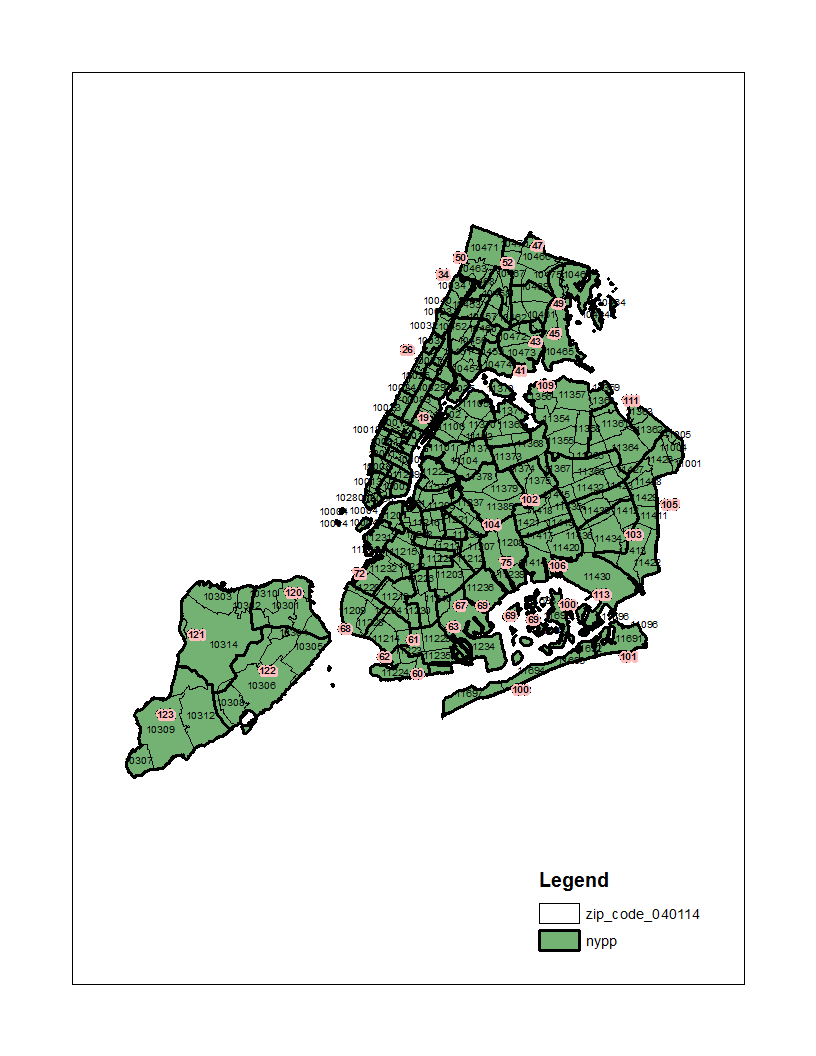
With the data available in all these formats, the most efficient way of bringing it down to a common division for me was using ARCGis, for the following two reasons:

1. The shapefiles(which constitute 3 combined files with extentions: SHP(shape),DBF(Database),SHX(Support file)) for all the data was available.My user experience with this type of file has been good as it has can be opened on a variety of softwares and languages. But working on Python with geopandas for this needs additional bunch of packages while ESRI combines all the data files at once, smoothing the work flow.
2. We all had consented to bring the data down to the **PRECINCT** division as the data might be helpful for NYPD if we ever intend on sharing it to help the city agency. To be absolutely sure of it, I needed visual confirmation on the various divisions of zip codes and census tracts overlaying on precinct, for which ArcGis was the obvious choice due to it’s easy management of shapefiles segregated into superimposed layers . In Python it slows the process down the entire process and also needs additional plotting packages to see.

Hence I used ESRI’s ARCGis for mapping different attributes of the data.

1. MUNGING ZIPCODES, CENSUS TRACTS on PRECINCTS :

When you see the shapefiles of zipcodes and precincts superimposed on each other, it looks something like this [Figure 1]



Image

You can see the that multiple zipcodes fall in a precinct. What is more complicated is that, the same zipcode falls under two/three different zipcodes. I looked for dataset where I could get the number,and the zipcode number of zipcodes in each precinct. But there is no dataset available for that. In my view, I could even segregate out zipcodes in each precinct using Python but because the parent dataset was not available, I manually made a zipcode and precinct dataset from the following webpage[<http://nyc.pediacities.com/Resource/PolicePrecinct/32>],where webscraping was not possible, and filled in the missing values of Precincts 33,121 using Google Maps.

But the problem then remained, which could have been only discovered on visually seeing the image, is that sometimes even if there is only a minute percentage of zipcode area in a precinct it includes it in that precinct, like below [Figure 2]. This would seriously hamper our analysis as the values of regressors available for zipcodes would get duplicated over two different precincts.

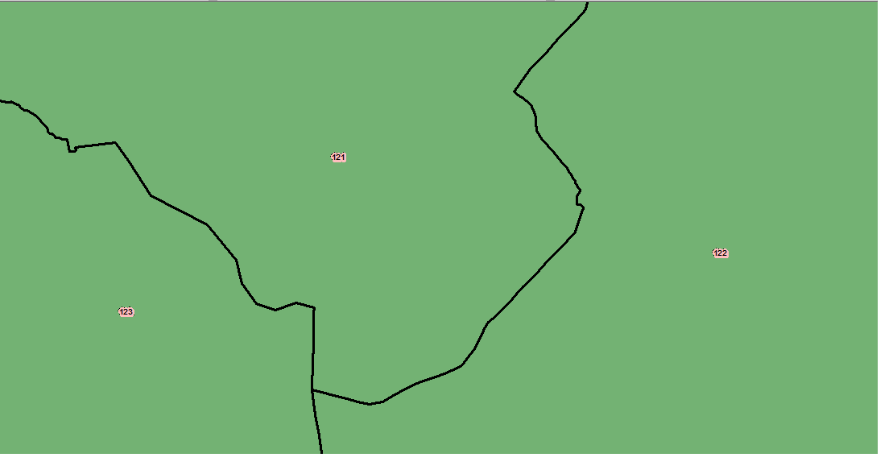


Image : Precint 121 has zipcodes [10312](http://nyc.pediacities.com/Resource/PostalCode/10312), [10314](http://nyc.pediacities.com/Resource/PostalCode/10314), [10306](http://nyc.pediacities.com/Resource/PostalCode/10306), [10308](http://nyc.pediacities.com/Resource/PostalCode/10308), [10302](http://nyc.pediacities.com/Resource/PostalCode/10302), [10303](http://nyc.pediacities.com/Resource/PostalCode/10303) and [10310](http://nyc.pediacities.com/Resource/PostalCode/10310).

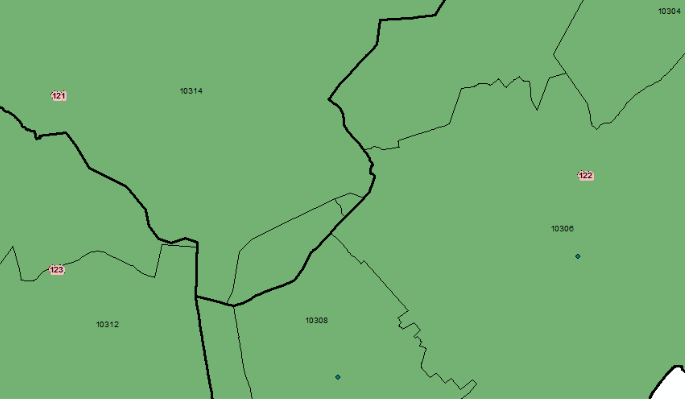


Image : While the area of Zipcode is 10306 and more so 13012 is negligible in the Precinct

**APPROACH 1** : So the first approach I followed was a Geoprocessing tool called *Intersection* which can gives a new layer of intersected regions, of the two layers (Zipcodes and Precincts), and decided to calculate to remove the zipcodes from the precinct number if the intersected area between them is less than a customizable value. Like below, the value chosen is less than 0.0005(areas are normalized on the map).



Image . The cyan regions indicate the selected intersected parts. You can see the blue line treading across the Precinct border, highlighting the problem that it takes even such small area from Zipcode 10312 into the precinct.

But this methodology was becoming complicated for each zipcode and precinct, and doing it manually did not seem suitable as we would have to judge the data we work on a customizable value that we would chose, which ran the risk of being biased.

**APPROACH 2 :** I decided to calculate the centroid of each of the zipcode (by this I mean, the centroid would be the intersection point of latitude and longitude of the zipcode area ) and overlay the point layer on precincts layer. Then using feature of ARCGis called ‘*Spatial Join’* I could find the number of zipcodes falling in each precinct. I had performed a spatial join of the zipcode layer and precinct layer before also for future verification. On verifying the new spatially joined data with the old one, it was observed that the count of zipcodes in each precinct had reduced, and the zipcode assignment to each precinct visually seemed justifiable.

Judging the two methods, I felt the second approach is better cleaning approach due to its practicality and neatness of the procedure .

Hence, the same method was followed while overlaying census tracts on the precincts .

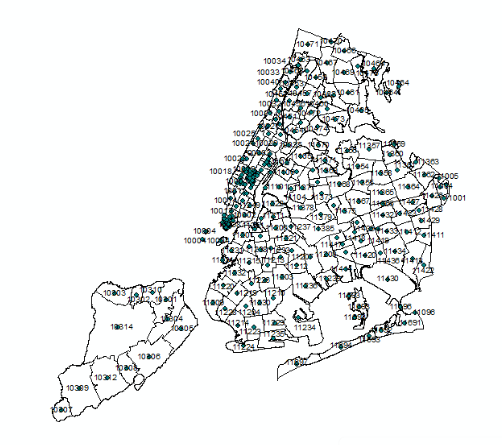


Image : Zipcodes with their centroids

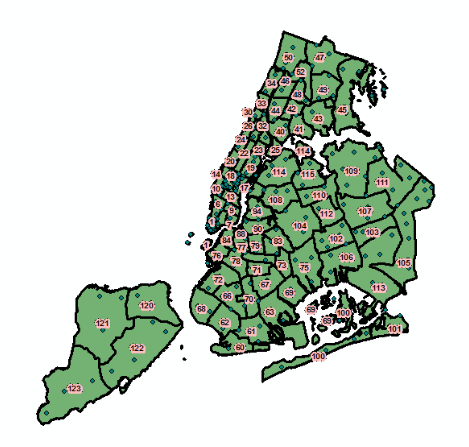


Image : The count in 121 precinct now reduced to 3 from 8. And visually it is better assignment.

1. Combining the regressors data.

To begin with, I would concisely state the further procedure , There are two joins performed here.

1. The joining of regressors dataset on to each point of the zipcode.
2. The joining of the zipcode points carrying the regressors data onto the precinct, so that ultimately we get every regressor data in a common division format of precinct.

Once the census tracts and zipcodes points were allotted to the precincts, the only part left was to add data to those points and then summing them up on the precinct basis.

For that, all the data that was collected was added into the map as table(only .xls works on ARCGis, which can be done by changing the .csv extenstion).( We only had gathered census tracts, zipcodes and precinct as shapefiles, as they pertain to geolocation more and rest data was in the .csv format. )

Once the tables were added, each had to be joined to its division base , i.e. income was to be joined to the data of census tracts, schools onto zipcode shapefile points. This was achieved by using *‘Join’*  feature. After adding each of the data, there was always a need for verification and checking for null values. For all the datasets, except Gender(which had inherent null values) we achieved a one hundred percent matching (for gender there was 85% matching).

Joining requires a common field in two datasets, which was available as were joining datsets with their own division. Joining was achieved in the following way for each of the table:

1.Demography was joined with based on CT2010 attribute on Census Tracts Data

2.Income was joined with Census Tracts shapefile using Census Label attribute

3.Schools were joined with Zipcode shapefile based on Zipcode attribute

4.Population by Gender and age(data cleaned by my teammate) was joined with Zipcode shapefile using Zipcode attribute

Lastly, these joined datasets were combined with the precinct shapefile. These values summed over the precinct using spatial join (100% matching here also), they gave valid values for all of them.