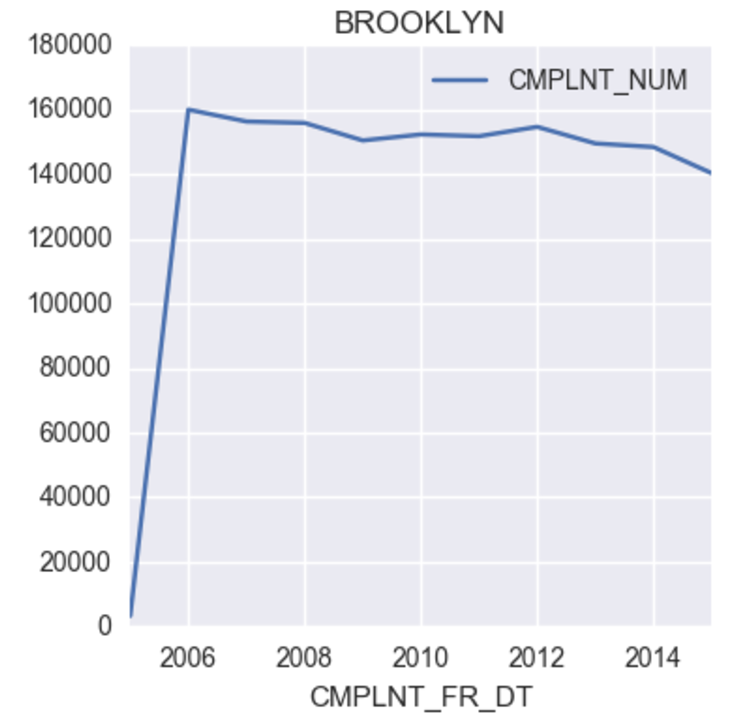
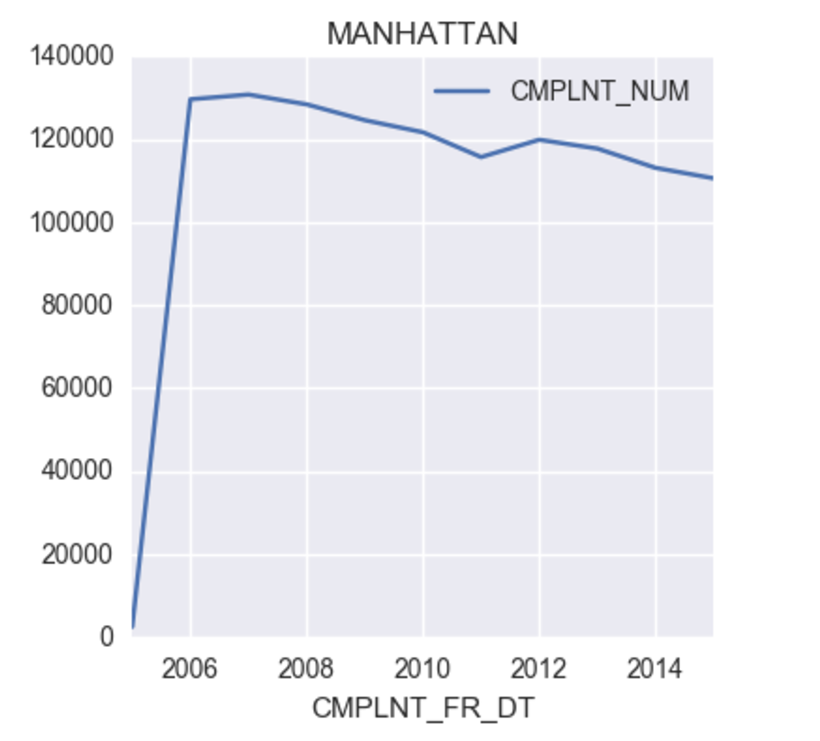
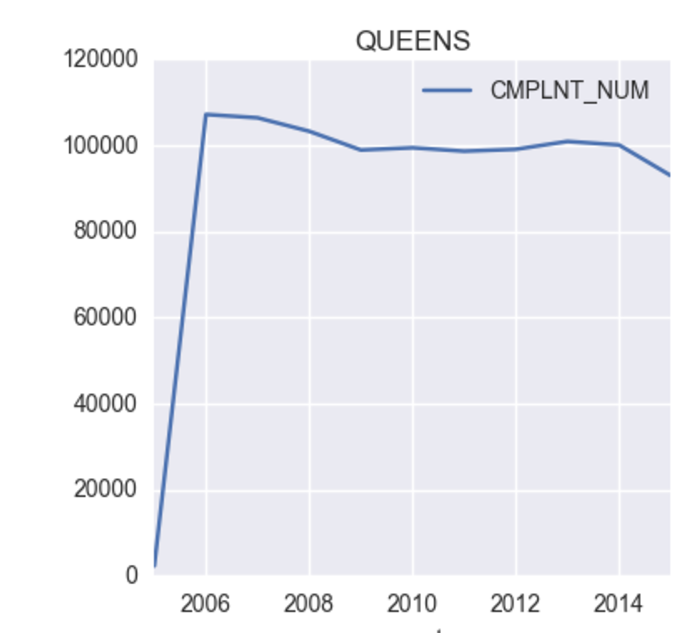
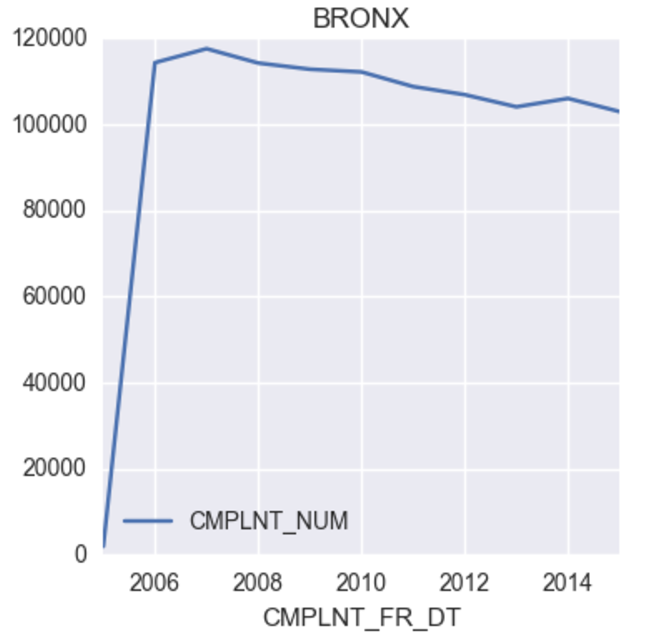
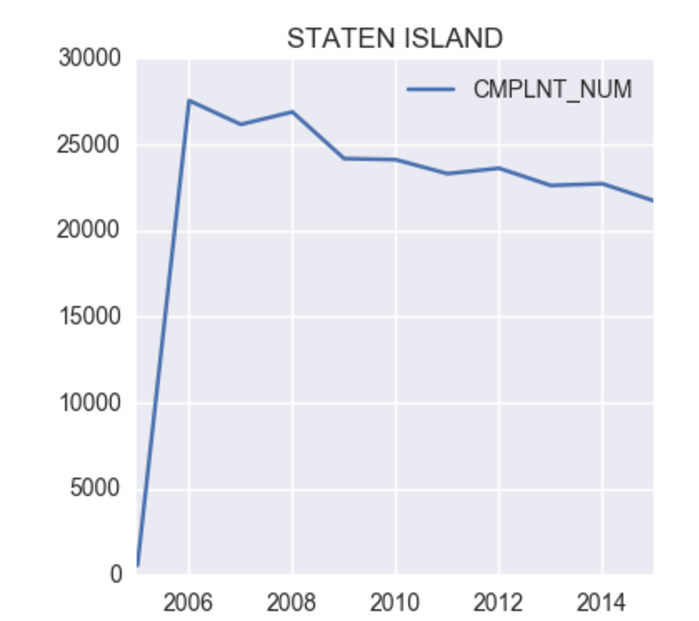
**MOTIVATION:**

Consider the following graphs demonstrating crime rate across different years.







The crime in New York seems to be following a very linear trend across years from 2006 - 2015. We are not considering years before that, because the graph is highly skewed and looks suspiciously negligible for years before 2006. Since the crime is linearly going down, it certainly makes sense to model this performance with some of the inherent characteristics of the Borough. Some of them could be using features like, total population, Population under poverty, Population of Black people, Percentage of educated people, immigrant’s migration rate, etc.   
  
In this task, we will try to compare crime rate with some of the above features individually.   
  
Our proposed hypothesis is that, the crime rate positively correlates with the features mentioned above.

**DATA:**  
  
The census data was collected from American FactFinder site. The data being used was prepared by American Community Survey, which uses 1 year estimates to come up with census data.  
  
The 10 csv files (each for a given year), was downloaded individually and then merged into one csv file. This entire csv file was then loaded to pyspark to check for data quality and issues. Note that the data has only 55 rows (5 boroughs \* 10 years ). The entire code for data quality and results have been included below.  
  
You could see that the data is clean and no issues have been found. We will use this data for testing our hypothesis. Note that the code is attached in github repo.

**Results from Data Quality issues:**

# Data exploration using Pyspark

# PART I

# Validity per column & integrity.

demographic\_csv.count()

# >>> 56

# The count should be 55 (11 years data for 5 boroughs). The extra count is for the column names

##### 1. DATA QUALITY CHECK

### COLUMN 1 - YEAR:

# All the distinct values should be between 2005 and 2015. Hence it should be 11

+----+

|Year|

+----+

|2012|

|2014|

|2013|

|2005|

|2009|

|2006|

|2011|

|2008|

|2007|

|2015|

|2010|

+----+

+--------------------+

|count(DISTINCT Year)|

+--------------------+

| 11|

+--------------------+

### COLUMN 2 - Borough

# there should be 5 distinct values

+-------------+

| Borough|

+-------------+

| Queens|

| Brooklyn|

|Staten Island|

| Manhattan|

| Bronx|

+-------------+

spark.sql("SELECT COUNT (DISTINCT Borough) FROM df").show()

+-----------------------+

|count(DISTINCT Borough)|

+-----------------------+

| 5|

+-----------------------+

### COLUMN 3 - Total population

# check count - should be 55

spark.sql("SELECT COUNT (Total\_population) FROM df").show()

+-----------------------+

|count(Total\_population)|

+-----------------------+

| 55|

+-----------------------+

# Check if it is integer or not

spark.sql("SELECT Total\_population ,'integer' AS base\_type ,'total population' AS semantic\_type, CASE WHEN Total\_population = '' THEN 'null' WHEN Total\_population not LIKE '%[^0-9]%' THEN 'valid' ELSE 'invalid' END AS is\_valid from df").show()

+----------------+---------+----------------+--------+

|Total\_population|base\_type| semantic\_type|is\_valid|

+----------------+---------+----------------+--------+

| 1300331| integer|total population| valid|

| 2440413| integer|total population| valid|

| 1526982| integer|total population| valid|

| 2208518| integer|total population| valid|

| 454610| integer|total population| valid|

| 1320591| integer|total population| valid|

| 2485425| integer|total population| valid|

| 1565250| integer|total population| valid|

| 2230432| integer|total population| valid|

| 468646| integer|total population| valid|

| 1337923| integer|total population| valid|

| 2508282| integer|total population| valid|

| 1584449| integer|total population| valid|

| 2248820| integer|total population| valid|

| 469575| integer|total population| valid|

| 1349428| integer|total population| valid|

| 2538724| integer|total population| valid|

| 1592904| integer|total population| valid|

| 2269435| integer|total population| valid|

| 478849| integer|total population| valid|

+----------------+---------+----------------+--------+

spark.sql("SELECT is\_valid, count(is\_valid) from (SELECT Total\_population ,'integer' AS base\_type ,'total population' AS semantic\_type, CASE WHEN Total\_population = '' THEN 'null' WHEN Total\_population not LIKE '%[^0-9]%' THEN 'valid' ELSE 'invalid' END AS is\_valid from df) group by is\_valid").show()

+--------+---------------+

|is\_valid|count(is\_valid)|

+--------+---------------+

| valid| 55|

+--------+---------------+

### COLUMN 4 - Below poverty level

# Check count

spark.sql("SELECT COUNT (Below\_poverty\_level) FROM df").show()

+--------------------------+

|count(Below\_poverty\_level)|

+--------------------------+

| 55|

+--------------------------+

# Check it is integer or not

spark.sql("SELECT Below\_poverty\_level ,'integer' AS base\_type ,'Below\_poverty\_level' AS semantic\_type, CASE WHEN Below\_poverty\_level = '' THEN 'null' WHEN Below\_poverty\_level not LIKE '%[^0-9]%' THEN 'valid' ELSE 'invalid' END AS is\_valid from df").show()

+-------------------+---------+-------------------+--------+

|Below\_poverty\_level|base\_type| semantic\_type|is\_valid|

+-------------------+---------+-------------------+--------+

| 379959| integer|Below\_poverty\_level| valid|

| 545611| integer|Below\_poverty\_level| valid|

| 272890| integer|Below\_poverty\_level| valid|

| 263701| integer|Below\_poverty\_level| valid|

| 49951| integer|Below\_poverty\_level| valid|

| 383788| integer|Below\_poverty\_level| valid|

| 561548| integer|Below\_poverty\_level| valid|

| 286800| integer|Below\_poverty\_level| valid|

| 271980| integer|Below\_poverty\_level| valid|

| 43036| integer|Below\_poverty\_level| valid|

| 362062| integer|Below\_poverty\_level| valid|

| 550169| integer|Below\_poverty\_level| valid|

| 279522| integer|Below\_poverty\_level| valid|

| 270066| integer|Below\_poverty\_level| valid|

| 45877| integer|Below\_poverty\_level| valid|

| 371971| integer|Below\_poverty\_level| valid|

| 536474| integer|Below\_poverty\_level| valid|

| 268635| integer|Below\_poverty\_level| valid|

| 275652| integer|Below\_poverty\_level| valid|

| 47752| integer|Below\_poverty\_level| valid|

+-------------------+---------+-------------------+--------+

# Check for invalid values

spark.sql("SELECT is\_valid, count(is\_valid) from (SELECT Below\_poverty\_level ,'integer' AS base\_type ,'Below\_poverty\_level' AS semantic\_type, CASE WHEN Below\_poverty\_level = '' THEN 'null' WHEN Below\_poverty\_level not LIKE '%[^0-9]%' THEN 'valid' ELSE 'invalid' END AS is\_valid from df) group by is\_valid").show()

+--------+---------------+

|is\_valid|count(is\_valid)|

+--------+---------------+

| valid| 55|

+--------+---------------+

### COLUMN 5 - Percent below poverty level

# Check count

spark.sql("SELECT COUNT (Percent\_Below\_poverty\_level) FROM df").show()

+----------------------------------+

|count(Percent\_Below\_poverty\_level)|

+----------------------------------+

| 55|

+----------------------------------+

# check the min and max. The min and max should be between 0 and 100 (because it is percentage calculation)

spark.sql("SELECT MIN(int(Percent\_Below\_poverty\_level)) FROM df").show()

+--------------------------------------------------------------------+

|min(CAST(CAST(Percent\_Below\_poverty\_level AS DECIMAL(20,0)) AS INT))|

+--------------------------------------------------------------------+

| 9|

+--------------------------------------------------------------------+

spark.sql("SELECT MAX(int(Percent\_Below\_poverty\_level)) FROM df").show()

+--------------------------------------------------------------------+

|max(CAST(CAST(Percent\_Below\_poverty\_level AS DECIMAL(20,0)) AS INT))|

+--------------------------------------------------------------------+

| 32|

+--------------------------------------------------------------------+

# Check it is integer or not

spark.sql("SELECT Percent\_Below\_poverty\_level ,'integer' AS base\_type ,'Percent Below poverty level' AS semantic\_type, CASE WHEN Percent\_Below\_poverty\_level = '' THEN 'null' WHEN Percent\_Below\_poverty\_level not LIKE '%[^0-9]%' THEN 'valid' ELSE 'invalid' END AS is\_valid from df").show()

+---------------------------+---------+--------------------+--------+

|Percent\_Below\_poverty\_level|base\_type| semantic\_type|is\_valid|

+---------------------------+---------+--------------------+--------+

| 29.2| integer|Percent Below pov...| valid|

| 22.4| integer|Percent Below pov...| valid|

| 17.9| integer|Percent Below pov...| valid|

| 11.9| integer|Percent Below pov...| valid|

| 11.0| integer|Percent Below pov...| valid|

| 29.1| integer|Percent Below pov...| valid|

| 22.6| integer|Percent Below pov...| valid|

| 18.3| integer|Percent Below pov...| valid|

| 12.2| integer|Percent Below pov...| valid|

| 9.2| integer|Percent Below pov...| valid|

| 27.1| integer|Percent Below pov...| valid|

| 21.9| integer|Percent Below pov...| valid|

| 17.6| integer|Percent Below pov...| valid|

| 12.0| integer|Percent Below pov...| valid|

| 9.8| integer|Percent Below pov...| valid|

| 27.6| integer|Percent Below pov...| valid|

| 21.1| integer|Percent Below pov...| valid|

| 16.9| integer|Percent Below pov...| valid|

| 12.1| integer|Percent Below pov...| valid|

| 10.0| integer|Percent Below pov...| valid|

+---------------------------+---------+--------------------+--------+

# Check for invalid values

spark.sql("SELECT is\_valid, count(is\_valid) from (SELECT Percent\_Below\_poverty\_level ,'integer' AS base\_type ,'Percent\_Below\_poverty\_level' AS semantic\_type, CASE WHEN Percent\_Below\_poverty\_level = '' THEN 'null' WHEN Percent\_Below\_poverty\_level not LIKE '%[^0-9]%' THEN 'valid' ELSE 'invalid' END AS is\_valid from df) group by is\_valid").show()

+--------+---------------+

|is\_valid|count(is\_valid)|

+--------+---------------+

| valid| 55|

+--------+---------------+

### COLUMN 5 - Percent Below poverty level Black

# Check count

spark.sql("SELECT COUNT (Percent\_Below\_poverty\_level\_Black) FROM df").show()

+----------------------------------------+

|count(Percent\_Below\_poverty\_level\_Black)|

+----------------------------------------+

| 55|

+----------------------------------------+

# check the min and max. The min and max should be between 0 and 100 (because it is percentage calculation)

spark.sql("SELECT MAX(int(Percent\_Below\_poverty\_level\_Black)) FROM df").show()

+--------------------------------------------------------------------------+

|max(CAST(CAST(Percent\_Below\_poverty\_level\_Black AS DECIMAL(20,0)) AS INT))|

+--------------------------------------------------------------------------+

| 35|

+--------------------------------------------------------------------------+

spark.sql("SELECT MIN(int(Percent\_Below\_poverty\_level\_Black)) FROM df").show()

+--------------------------------------------------------------------------+

|min(CAST(CAST(Percent\_Below\_poverty\_level\_Black AS DECIMAL(20,0)) AS INT))|

+--------------------------------------------------------------------------+

| 11|

+--------------------------------------------------------------------------+

# Check it is integer or not

spark.sql("SELECT Percent\_Below\_poverty\_level\_Black ,'integer' AS base\_type ,'Percent\_Below\_poverty\_level\_Black' AS semantic\_type, CASE WHEN Percent\_Below\_poverty\_level\_Black = '' THEN 'null' WHEN Percent\_Below\_poverty\_level\_Black not LIKE '%[^0-9]%' THEN 'valid' ELSE 'invalid' END AS is\_valid from df").show()

+---------------------------------+---------+--------------------+--------+

|Percent\_Below\_poverty\_level\_Black|base\_type| semantic\_type|is\_valid|

+---------------------------------+---------+--------------------+--------+

| 26.3| integer|Percent\_Below\_pov...| valid|

| 21.1| integer|Percent\_Below\_pov...| valid|

| 30.0| integer|Percent\_Below\_pov...| valid|

| 11.9| integer|Percent\_Below\_pov...| valid|

| 30.6| integer|Percent\_Below\_pov...| valid|

| 27.1| integer|Percent\_Below\_pov...| valid|

| 23.2| integer|Percent\_Below\_pov...| valid|

| 31.7| integer|Percent\_Below\_pov...| valid|

| 11.9| integer|Percent\_Below\_pov...| valid|

| 22.7| integer|Percent\_Below\_pov...| valid|

| 24.2| integer|Percent\_Below\_pov...| valid|

| 20.9| integer|Percent\_Below\_pov...| valid|

| 29.8| integer|Percent\_Below\_pov...| valid|

| 11.3| integer|Percent\_Below\_pov...| valid|

| 27.2| integer|Percent\_Below\_pov...| valid|

| 25.1| integer|Percent\_Below\_pov...| valid|

| 20.9| integer|Percent\_Below\_pov...| valid|

| 28.7| integer|Percent\_Below\_pov...| valid|

| 12.3| integer|Percent\_Below\_pov...| valid|

| 31.9| integer|Percent\_Below\_pov...| valid|

+---------------------------------+---------+--------------------+--------+

# Check for invalid values

spark.sql("SELECT is\_valid, count(is\_valid) from (SELECT Percent\_Below\_poverty\_level\_Black ,'integer' AS base\_type ,'Percent\_Below\_poverty\_level\_Black' AS semantic\_type, CASE WHEN Percent\_Below\_poverty\_level\_Black = '' THEN 'null' WHEN Percent\_Below\_poverty\_level\_Black not LIKE '%[^0-9]%' THEN 'valid' ELSE 'invalid' END AS is\_valid from df) group by is\_valid").show()

+--------+---------------+

|is\_valid|count(is\_valid)|

+--------+---------------+

| valid| 55|

+--------+---------------+

### COLUMN 6 - Percent with education less than high school level

# Check count

spark.sql("SELECT COUNT (Percent\_Below\_poverty\_level\_Black) FROM df").show()

+----------------------------------------+

|count(Percent\_Below\_poverty\_level\_Black)|

+----------------------------------------+

| 55|

+----------------------------------------+

# check the min and max. The min and max should be between 0 and 100 (because it is percentage calculation)

spark.sql("SELECT MIN(int(percent\_with\_education\_below\_high\_school)) FROM df").show()

spark.sql("SELECT MAX(int(percent\_with\_education\_below\_high\_school) FROM df").show()

# Check it is integer or not

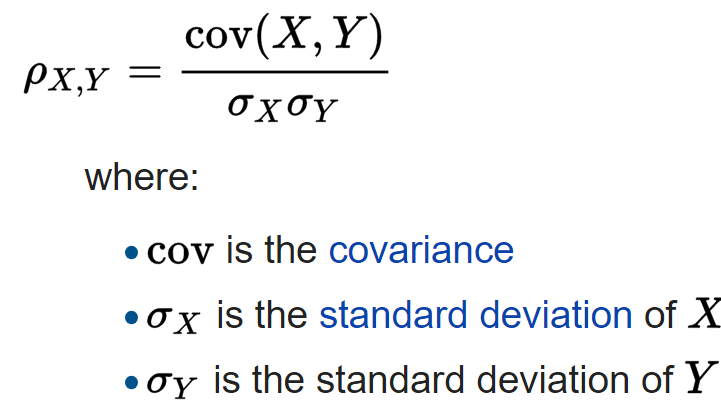
spark.sql("SELECT percent\_with\_education\_below\_high\_school ,'integer' AS base\_type ,'percent\_with\_education\_below\_high\_school' AS semantic\_type, CASE WHEN percent\_with\_education\_below\_high\_school = '' THEN 'null' WHEN percent\_with\_education\_below\_high\_school not LIKE '%[^0-9]%' THEN 'valid' ELSE 'invalid' END AS is\_valid from df").show()

# Check for invalid values

spark.sql("SELECT is\_valid, count(is\_valid) from (SELECT percent\_with\_education\_below\_high\_school ,'integer' AS base\_type ,'percent\_with\_education\_below\_high\_school' AS semantic\_type, CASE WHEN percent\_with\_education\_below\_high\_school = '' THEN 'null' WHEN percent\_with\_education\_below\_high\_school not LIKE '%[^0-9]%' THEN 'valid' ELSE 'invalid' END AS is\_valid from df) group by is\_valid").show()

**EXPERIMENTAL TECHNIQUES:**  
  
To prove our hypothesis we need to check the dependence of features with the crime rate in New York. Hence, we will be testing the correlation of each feature with the crime rate. For this we will be using the **"Statistics"** library of pyspark.

For checking correlation, we would be using the **Pearson** correlation measure. It is calculated as coefficient is the [covariance](https://en.wikipedia.org/wiki/Covariance) of the two variables divided by the product of their [standard deviations](https://en.wikipedia.org/wiki/Standard_deviations). The form of the definition involves a "product moment", that is, the mean (the first moment about the origin) of the product of the mean-adjusted random variables. It is equal to



Apart from this, we will be doing visualization of these features across years, and compare them the trend of crime rate.

The results of our analysis is shown below.

**RESULTS (HYPOTHESIS TESTING)**

**Part – I: (Checking correlation)**

Pearson correlation of mllib was used. The code and results of this is (using pyspark) :

# Finding correlation

v1 = demographic\_csv.map(lambda x: x[3]) # vector for total population

v2 = demographic\_csv.map(lambda x: x[4]) # vector for population under poverty level

v3 = demographic\_csv.map(lambda x: x[14]) # vector for black population

v4 = demographic\_csv.map(lambda x: x[20]) # vector for under-educated population

v5 = demographic\_csv.map(lambda x: x[19]) # Total crime count

# Correlation between total population and crime count

print("Correlation is: " + str(Statistics.corr(v1, v5, method="pearson")))

**# Correlation is: 0.6767118210127591**

# Correlation between population under poverty level and crime count

print("Correlation is: " + str(Statistics.corr(v2, v5, method="pearson")))

**# Correlation is: 0.7305919332530579**

# Correlation between black population and crime count

print("Correlation is: " + str(Statistics.corr(v3, v5, method="pearson")))

**# Correlation is: 0.6712880230552578**

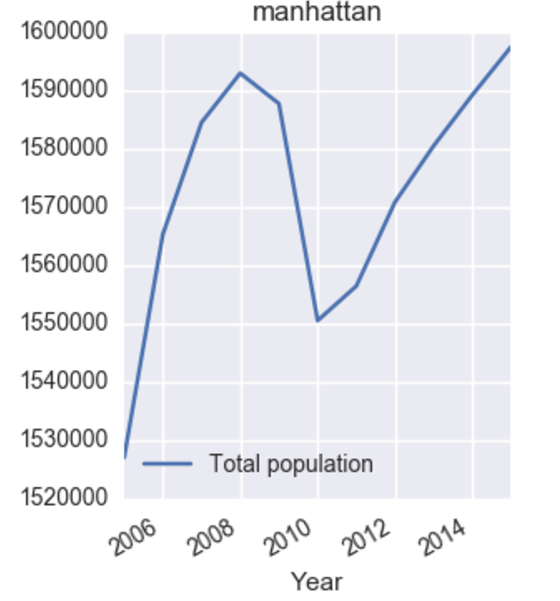
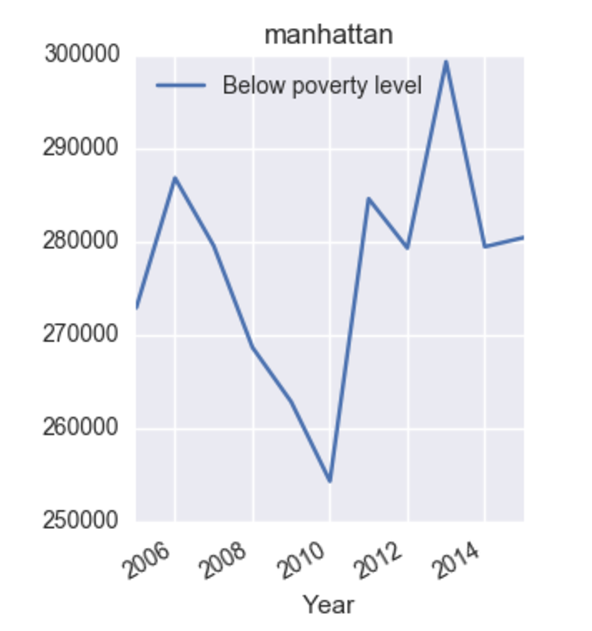
# Correlation between under-educated population and crime count

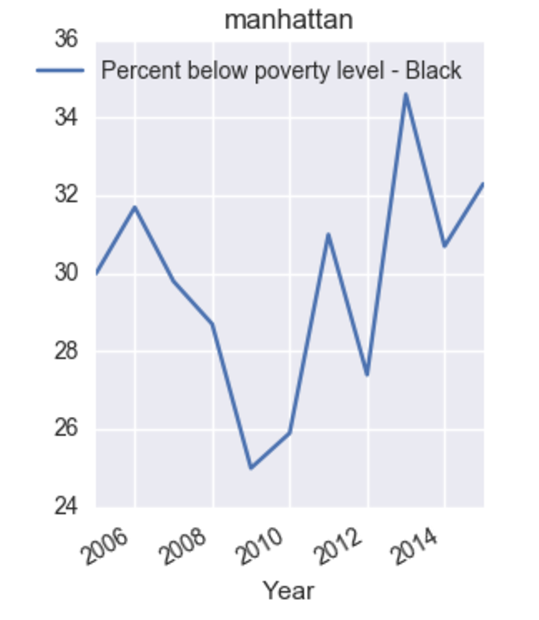
print("Correlation is: " + str(Statistics.corr(v4, v5, method="pearson")))

**# Correlation is: 0.6321731661199721**

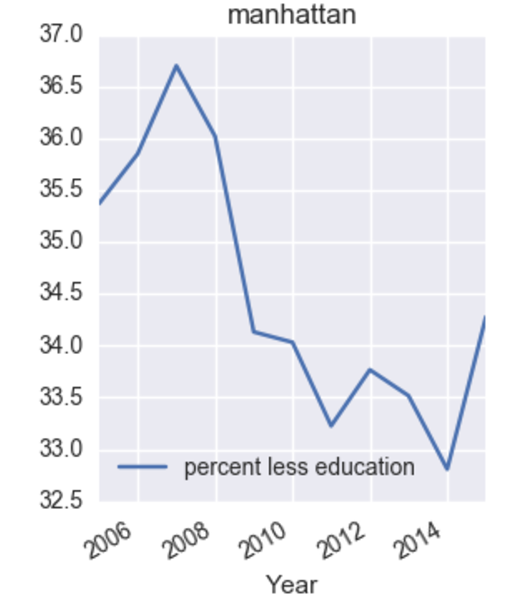
From the above results, we see that every feature seem to be statistically significant since they have more than **60%** of correlation with the crime rate. This proves a part of our hypothesis. Next we will visualize each feature and see how it compares with the distribution of crime.

**Part – II: Visualizing the graphs for features and comparing them**

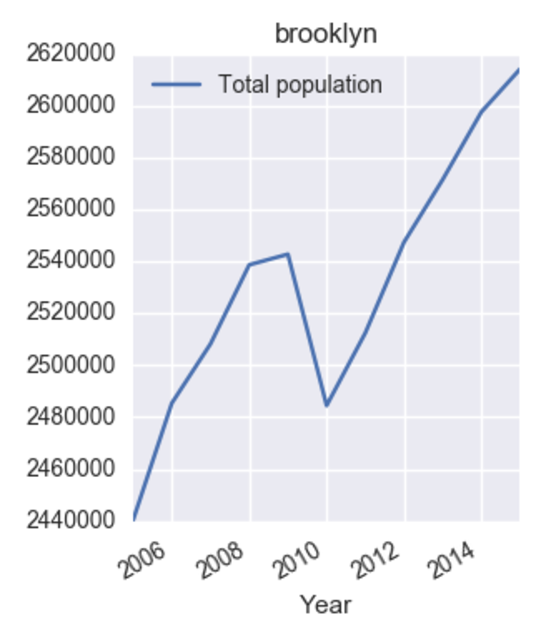
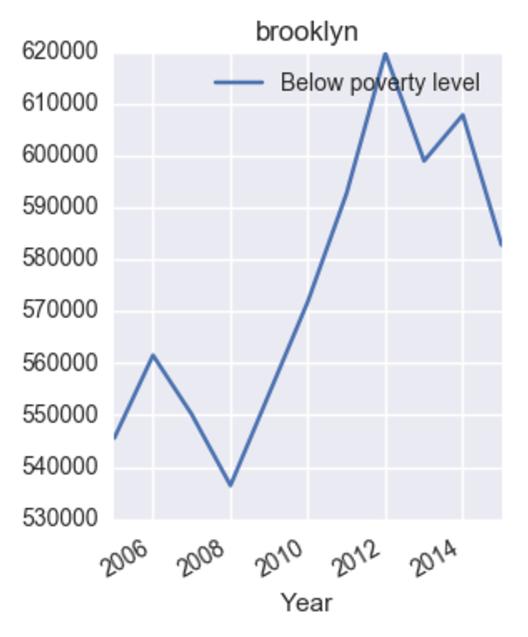
**(a) Manhattan (New York County):**  
  
The visualization graphs for the features are given below.   
  
**(i) POPULATION**  
  
  
  
We see that the population of Manhattan drops substantially in the year 2010, and then starts to increase slowly. Also, we know that the crime data calculated for the year 2011 uses the data uptil 2010. Hence, population seems to have a direct correlation with crime.   
  
Lets also analyze this behavior with other two important attributes  
  
**(ii) POPULATION UNDER POVERTY LEVEL**  
  
  
  
We see that the percent of population of people living under poverty is less in the year 2011. This might be clear from the fact that population in 2011 was less (as shown in the graph above).   
  
This measure also has a positive correlation for predicting crime rate in manhattan.  
  
**(iii) POPULATION OF BLACK PEOPLE**

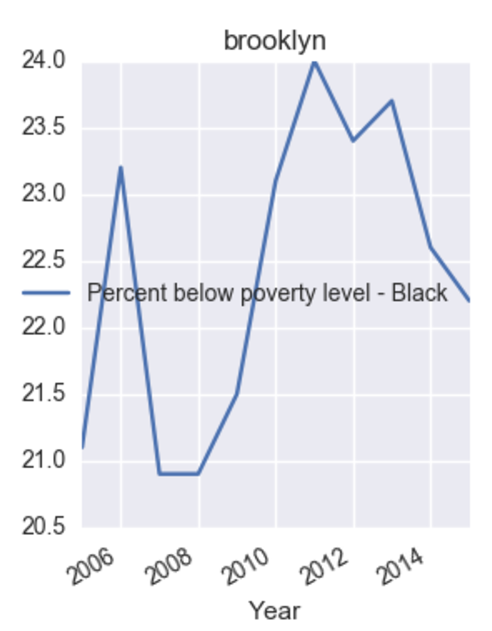
  
  
We did try to visualize graph of different races vs the crime rate. Only the population of black people seemed to have a positive correlation. The graph is attached below.   
  
We witness that this feature also seems to drop around the year 2010 and then increases. Almost similar trend is being followed by crime rate.

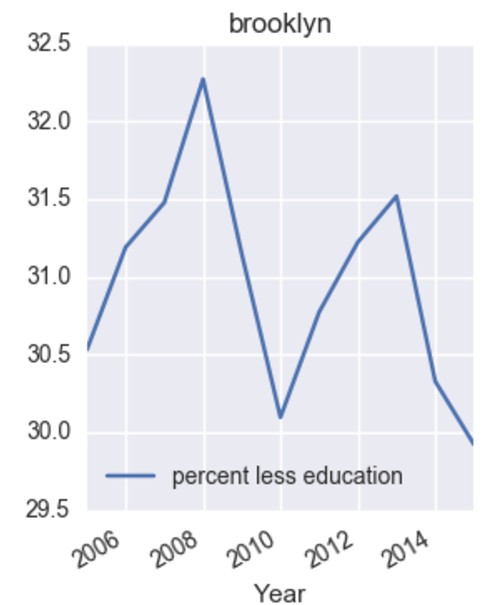
**(iv) EDUCATIONAL ATTAINMENT LEVEL FOR PEOPLE OVER 25 YEARS**

  
Even this complements the behavior of crime rates in manhattan. We see that the maximum number of people with less education level is in 2006 and minimum is during the years 2013 and 2014. Correspondingly, we will expect the crimes rates reported at the end of year 2006 to be high and those reported during the years 2013 and 2014 to be least. This is clearly true from the crime distribution graph of Manhattan.   
  
Hence the hypothesis framed for the relation between crime and general demographic data is clearly valid for Manhattan.

**(b) Brooklyn (King's County):**  
  
Brooklyn almost has a constant distribution. We can see that the crime rate is almost constant during the period 2006-2015, with slight variations. It decreases slightly for the years 2008 to 2009 and then increases and then decreases to its least value. This also can be explained using the three attributes.Hence, same kind of analysis can be performed here too.

**(i) POPULATION**  
  
  
  
We see that the population of Brooklyn increases linearly uptil the year 2008 and then starts dropping substantially for the years 2009 and 2010, which explains for the slight decrease in crime rate. It then increases sharply and becomes maximum in the year 2015. Unfortunately, crime rate is least as opposed to being high with increasing population. This might be due to other factors like decreased uneducated population level as shown below.  
  
**(ii) POPULATION UNDER POVERTY LEVEL**  
  
  
  
We see that the percent of population of people living under poverty is less in the years 2009 and 2015, which unsurprisingly are also the years with less crimes. Relatively, it is higher in the year 2012 and then in 2006. This can account for almost same number of crimes in those years. Also, this population starts decreasing towards the end of 2014, and so does the crime rate.  
  
**(iii) POPULATION OF BLACK PEOPLE**  
  
Almost similar trend is being followed by the percent of black population in Brooklyn. The trend starts to decrease (and so as crime rate) and then increases with increasing crime rate.

  
  
  
**(iv) EDUCATIONAL ATTAINMENT LEVEL FOR PEOPLE OVER 25 YEARS**

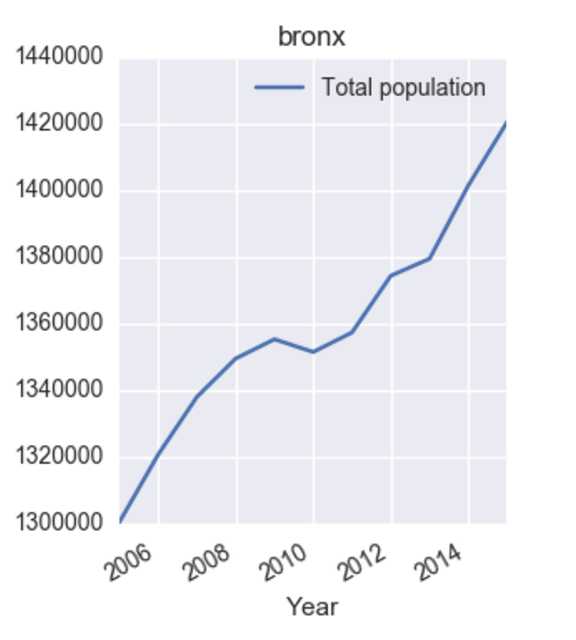
  
  
Even this measure complements the behavior of crime rates in Brooklyn. We see that the maximum number of people with education level is in 2009 and minimum is during the year 2015. Correspondingly, the crimes rates reported at the end of year 2009 to be high and those reported during the year 2014 to be least. This latter is clearly true from the crime distribution graph of brooklyn, but the former is not. This is because, the population in 2009 was very less as compared to other years in brooklyn. Hence, the total number of people with less education should also be less, which explains less crime.

The graph for other boroughs have also been shown below. We can see that the hypothesis seems to be valid for every one of them, which also explains the high correlation value of every feature with the crime rate.

**(c) BRONX (BRONX COUNTY)**

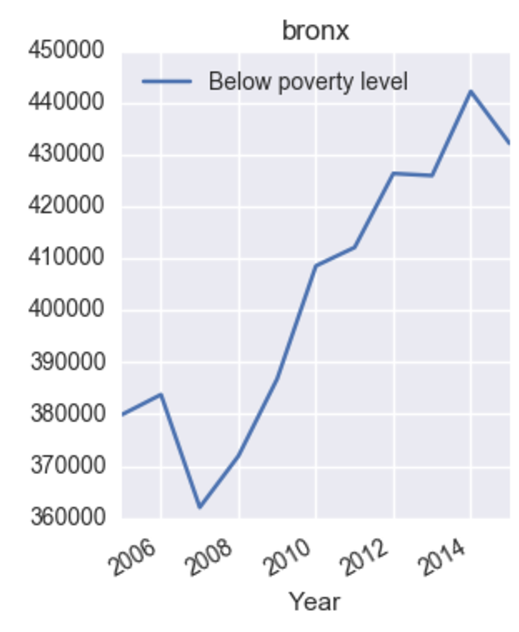
**(i) POPULATION**

Even though population seems to increasing, surprisingly crime rate decreases. Hence other features might be useful apart from this.



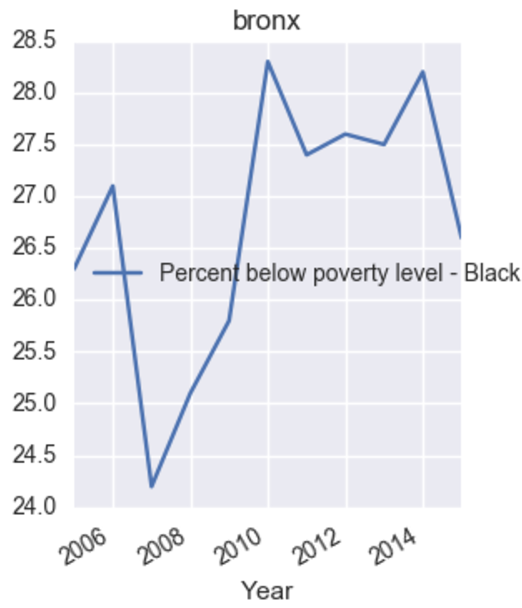
**(ii) POPULATION UNDER POVERTY LEVEL**

Here we see that the relationship is almost dissimilar. This can be because, population and population under poverty are directly correlated. Hence, still seeing other features is better



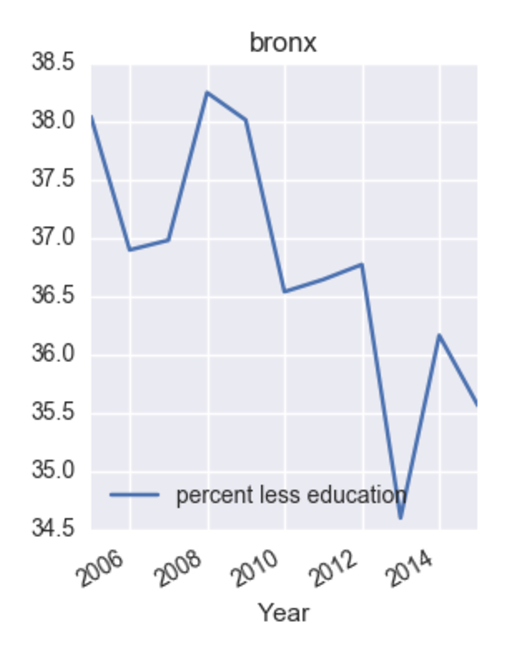
**(iii) POPULATION OF BLACK PEOPLE**

Here we see that relationship is is almost constant initially and then decreases. This is almost the distribution of crime in Bronx.



**(iv) EDUCATIONAL ATTAINMENT LEVEL FOR PEOPLE OVER 25 YEARS**

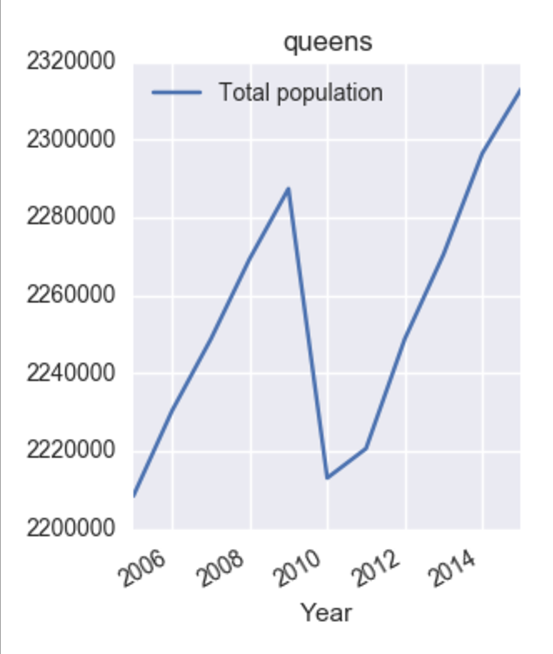
This is clearly 100 % correlated with crime in Bronx if you examine both these graphs. Hence, in Bronx, Education count rally plays a major role to deteremine crime rate.



**(d) QUEENS (QUEENS COUNTY)**

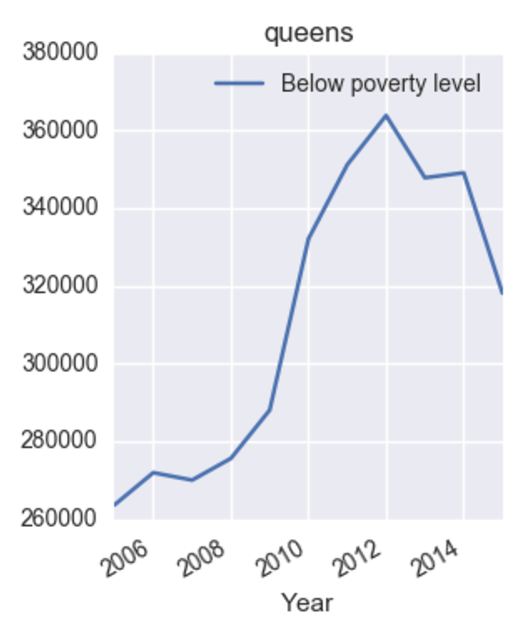
**(i) POPULATION**

The population is fairly the same with slight increases and decrease. It is also the case for crime distribution.



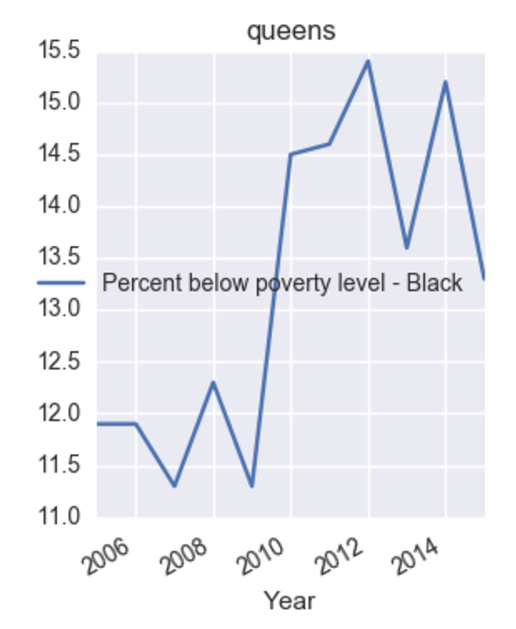
**(ii) POPULATION UNDER POVERTY LEVEL**

Here we see that poverty level increases towards the end and so does crime.



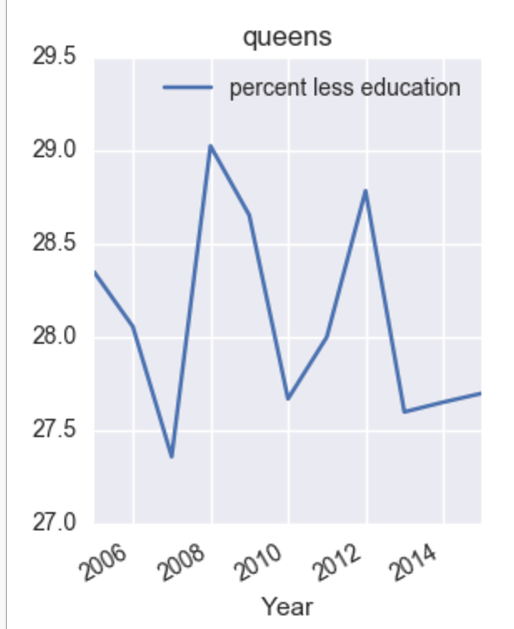
**(iii) POPULATION OF BLACK PEOPLE**

This again, increases towards the end with increasing crime rate. Hence even this measure plays a high significant role.



**(iv) EDUCATIONAL ATTAINMENT LEVEL FOR PEOPLE OVER 25 YEARS**

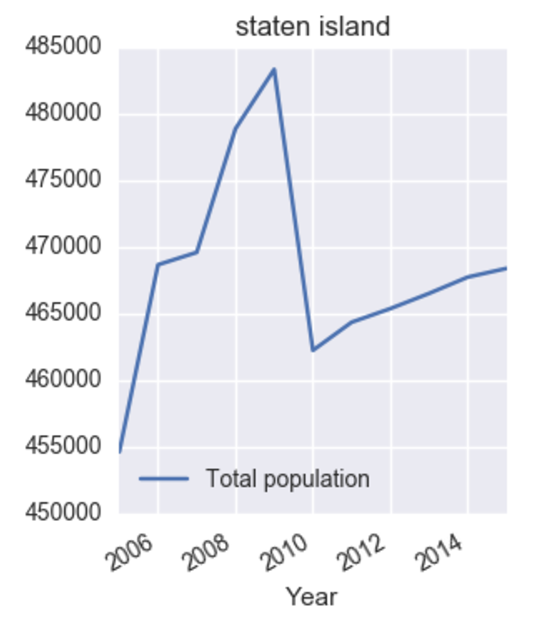
For Queens, percent with less education remained the same across years and it is difficult to measure crime rate with respect to this. But other measures like black population and poverty level population seem to do a pretty good job.



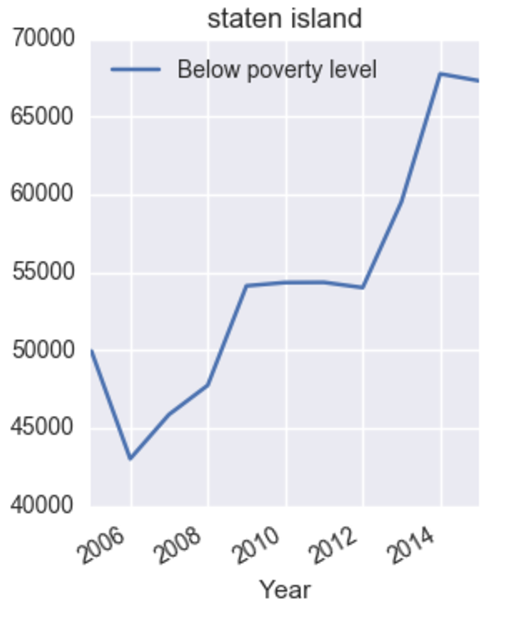
**(e) STATEN ISLAND (RICHMOND COUNTY)**

**(i) POPULATION**

Staten Island crime rate linearly decreases which also is somewhat the case for population across the years.

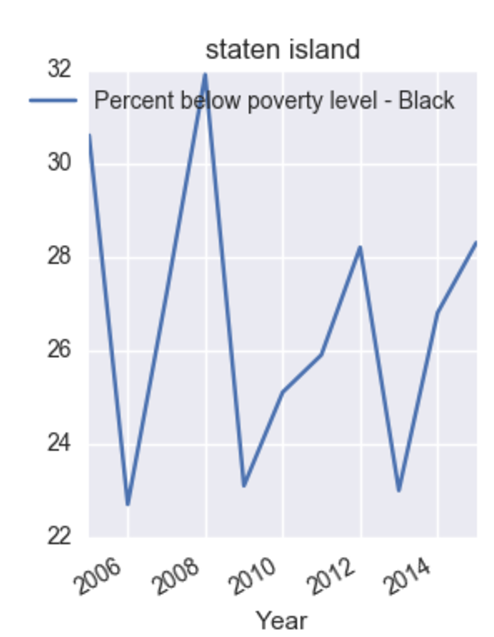


**(ii) POPULATION UNDER POVERTY LEVEL**

Surprisingly, this measure is highly negative with the crime rate in Staten Island. But other features like education level and black population seem to do better job for Staten Island.

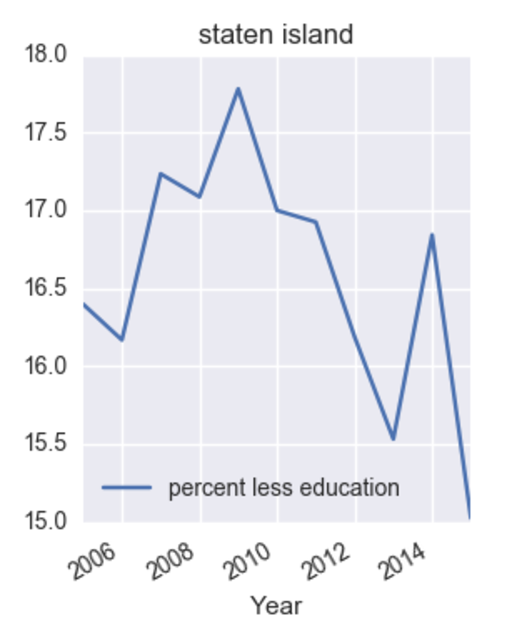
**(iii) POPULATION OF BLACK PEOPLE**

As seen from the graph, the population of black people decreases with time, which is also the trend or crime in Staten Island. Hence, it is positively correlated also.



**(iv) EDUCATIONAL ATTAINMENT LEVEL FOR PEOPLE OVER 25 YEARS**

Percentage for people with less education certainly decreases a lot with time in Staten Island. This is exactly the trend for crime wherein, the curve drops sharply over time and is least in 2016. Hence, this also has a high positive correlation.



**Conclusion from results:**

From the above described results, every borough had atleast 3 out 4 features to be directly positively correlated with the crime rate. This also explains why the Pearson correlation rate was so high. Hence, using these two results, we can say that the hypothesis that the crime rates vary a lot with general demographic data is sufficiently true.

**DISCUSSIONS:**

The analysis carried out above was based on borough level. The same problem could be done at a much smaller granularity like, observing the crime rate at zip code level, census block level and census tract level. Due to time constraints and data limitations, we could limit our analysis to borough level only. The data limitations was that the data was very inconsistent at these granularity levels. The inconsistencies included the time range of the data (it didn’t cover all the years from 2006 – 2015), and some features missing were missing in some of the years, which made it difficult for us to perform our analysis.