NYPD Incident Level Data Exploration

Minda Fang NetID: mf3308

Qiming Zhang NetID: qz718

Mingdi Mao NetID: mm8688

April 17, 2017

Contents

1 Abstract

The purpose of this report is for sum-up of NYPD incident level dataset released in 2016. The report includes detecting data issue, data cleaning and data summary. Analysis principle could be seen in the first part and crime distribution graphs could be seen at the last part.

2 Introduction

This is NYU Tandon CS-9223B Big data final project part I report. We select NYPD Complaint Data Historic dataset(update Dec. 2016) to investigate. To analyze the dataset , we hope to find where is wrong and how we could get rid of them then illustrate the analysis result in graphs, so the work done so far includes three parts, which are detecting issue data, cleaning dataset and delivering summary of dataset. At the very first part, we analyzed the dataset in five major dimensionalities,null value problem, range problem, unexpected type, mapping problem and other issues. We fixed the error and build a clean dataset based on results of detected issues. Lastly, we summarized the dataset in 9 aspects to show what we learned from this dataset. Java was used in the first part and Python2 for other work.

3 Part 1: data issues and data summary

Data issues

3.1 Null values

Column null restriction

A lot of fields has null values. Some of them are considered valid, some of them are not.

Column	Allowed to be Null (How to judge if valid or not)
CMPLNT_NUM	Cannot be null and must be a number string
CMPLNT_FR_DT	Can be null (But if this is null and CMPLNT_FR_TM is not null, then this Null value will be regarded as invalid.)
CMPLNT_FR_TM	Can be null (But if this is null and CMPLNT_FR_DT is not null, then this Null value will be regarded as invalid.)
CMPLNT_TO_DT	Can be null (But if this is null and CMPLNT_FR_DT is not null, then this Null value will be regarded as invalid.)
CMPLNT_TO_TM	Can be null (But if this is null and CMPLNT_FR_DT is not null, then this Null value will be regarded as invalid.)
RPT_DT	Cannot be null
KY_CD	Cannot be null
OFNS_DESC	Cannot be null
PD_CD	Cannot be null
PD_DESC	Cannot be null
CRM_ATPT_CPTD_CD	Cannot be null
LAW_CAT_CD	Cannot be null
JURIS_DESC	Cannot be null
BORO_NM	Cannot be null
ADDR_PCT_CD	Cannot be null
LOC_OF_OCCUR_DESC	Can be null
PREM_TYP_DESC	Cannot be null
PARKS_NM	Can be null
HADEVELOPT	Can be null
X_COORD_CD	Can be null but only if OFNS_DESC is Rape or Sex Crime (But if this is null and any one of other Geo info fields is not null, then this Null value will be regarded as invalid.)
Y_COORD_CD	Can be null but only if OFNS_DESC is Rape or Sex Crime (But if this is null and any one of other Geo info fields is not null, then this Null value will be regarded as invalid.)
Latitude	Can be null but only if OFNS_DESC is Rape or Sex Crime (But if this is null and any one of other Geo info fields is not null, then this Null value will be regarded as invalid.)
Longitude	Can be null but only if OFNS_DESC is Rape or Sex Crime (But if this is null and any one of other Geo info fields is not null, then this Null value will be regarded as invalid.)
Lat_Lon	Can be null but only if OFNS_DESC is Rape or Sex Crime (But if this is null and any one of other Geo info fields is not null, then this Null value will be regarded as invalid.)

We deal with all these null value checking Basic_Issue and DataClean.py

List of issues we found in null value checking (all of them lead to the whole row to be invalid data)

- 4574 rows of data missing three digit internal classification code(PD_CD). These 4574 rows of data contains all the offense classification code of "101"(MURDER & NON-NEGL) MANSLAUGHTER.
- 463 rows of data has no borough record.
- 390 rows of data has no precinct record.
- 7 rows of data contains empty crime completion record.
- 5415 rows of record contains incomplete date error problem. We first judge if each row of data has incomplete start/end time record. Incomplete means that this start/end date combination has only date or time(one of them is missing)

CMPLNT_NUM	CMPLNT_FR_DT	CMPLNT_FR_FM	CMPLNT_TO_DT	CMPLNT_TO_FM
494598165	12/29/2015	06:00:00	(missing)	12:29:00
259351246	(missing)	09:00:00		
546886673	09/18/2014	15:00:00	07/14/2015	(missing)

3.2 Range problem

Field range problem is very important in data integrity. Here is how we check if certain field may have range problem

- 4574 rows of data missing three digit internal classification code(PD_CD). These 4574 rows of data contains all the offense classification code of "101"(MURDER & NON-NEGL) MANSLAUGHTER.
- 463 rows of data has no borough record.
- 390 rows of data has no precinct record.
- 7 rows of data contains empty crime completion record.
- 5415 rows of record contains incomplete date error problem. We first judge if each row of data has incomplete start/end time record. Incomplete means that this start/end date combination has only date or time(one of them is missing)

Column	Range
CRM_ATPT_CPTD_CD	COMPLETED or ATTEMPTED
LAW_CAT_CD	FELONY, MISDEMEANOR or VIOLATION
BORO_NM	"MANHATTAN", "BRONX", "BROOKLYN", "QUEENS" or "STATEN ISLAND"
LOC_OF_OCCUR_DESC	"INSIDE", "OPPOSITE OF", "FRONT OF", "REAR OF" or NULL value

We deal with all these range problem DataClean.py We also check if each element is valid or not in these columns, we check this all in Basic_Issue
We discover that all data dont have range problem. They are either null value or exactly in the range.

3.3 Unexpected type problem

Every field has its own base type. So we also check some fields whether they have expected type

Column	Type
CMPLNT_NUM	Integer
CMPLNT_FR_DT	DateTime(MM(M)/dd(d)/yyyy)
CMPLNT_FR_TM	DateTime(HH:mm:ss)
CMPLNT_TO_DT	DateTime(MM(M)/dd(d)/yyyy)
CMPLNT_TO_TM	DateTime(HH:mm:ss)
RPT_DT	DateTime(MM(M)/dd(d)/yyyy)
KY_CD	Three Digit Number
OFNS_DESC	String
PD_CD	Three Digit Number
PD_DESC	String
CRM_ATPT_CPTD_CD	String
LAW_CAT_CD	String
JURIS_DESC	String
BORO_NM	String
ADDR_PCT_CD	String
LOC_OF_OCCUR_DESC	String
PREM_TYP_DESC	String
PARKS_NM	String
HADEVELOPT	String
X_COORD_CD	Float Number
Y_COORD_CD	Float Number
Latitude	Float Number
Longitude	Float Number
Lat_Lon	Tuple of Float Number

We deal with all these Unexpected type checking Basic_Issue

List of issues we found in base type checking: (All of them lead to the whole row of data to be invalid)

- **Invalid datetime** We detect if each data combination could be interpreted as real time.

```
def ifIsNotValidDateString(str):
    if not re.match(r"^(0?[1-9]|1[012])/(0?[1-9]|[12][0-9]|3[01])/\d{4}$", str):
        return True;
    array = str.split("/")
    ifCorrectDate = True
    try:
        newDate = datetime.datetime(int(array[2]),int(array[0]),int(array[1]))
    except ValueError:
        ifCorrectDate = False
    finally:
        return not ifCorrectDate

def ifIsNotValidTimeString(str):
    if re.match(r"^(0[0-9]|1[0-9]|2[0-3]):([0-5][0-9]):([0-5][0-9])$", str):
        return False
    else :
        return True
```

Figure 1: Date and time detection

```
24:00:00    TEXT    Complaint_Starting_Time Invalid
24:00:00    TEXT    Complaint_Starting_Time Invalid
24:00:00    TEXT    Complaint_Starting_Time Invalid
24:00:00    TEXT    Complaint_Starting_Time Invalid
24:00:00    TEXT    Complaint_Starting_Time Invalid
24:00:00    TEXT    Complaint_Starting_Time Invalid
24:00:00    TEXT    Complaint_Starting_Time Invalid
24:00:00    TEXT    Complaint_Starting_Time Invalid
24:00:00    TEXT    Complaint_Starting_Time Invalid
24:00:00    TEXT    Complaint_Starting_Time Invalid
24:00:00    TEXT    Complaint_Starting_Time Invalid
24:00:00    TEXT    Complaint_Starting_Time Invalid
```

Figure 2: Invalid date time discovered

3.4 Mapping problem

One column may be dependent on any other columns. Dependences we have found are below:

- Start time of occurrence must be earlier than end time of occurrence. That is (CMPLNT_FR_DT,CMPLNT_FR_TM) must be earlier than (CMPLNT_TO_DT, CMPLNT_TO_TM).
  No such problem detected.

- To further protect victim identities, rape and sex crime offenses are not geocoded.So when OFNS_DESC is Rape or Sex Crime, X_COORD_CD, Y_COORD_CD, Longitude, Latitude and tuple of (Longitude, Latitude) must all be null values.

And meanwhile, when X_COORD_CD, Y_COORD_CD, Longitude, Latitude and tuple of (Longitude, Latitude) are null values, OFNS_DESC should only be Rape or Sex Crime.
119461 rows of data has such problem. Script for detecting such problem is SexRape_Coordinate_mapping.py

- One offense classification code(KY_CD) can only be mapped to one description(OFNS_DESC). **15 pairs of KY_CD & OFNS_DESC has conflict mapping problem.** Script for detecting such problem is OffenseCodeMapping.py
- One internal offense classification code(PD_CD) can only be mapped to one description(PD_DESC). **No such conflict mapping problem is found.** Script for detecting such problem is PDCodeMapping.py
- One internal offense classification code(PD_CD) can only be mapped to one offense classification code(KY_CD). **183 pairs of PD_CD & KY_CD has conflict mapping problem.** Script for detecting such problem is PD_OFNS_Mapping.py
- The tuple of Latitude and Longitude must be corresponding to the individual value of latitude and longitude. **No such problem is found.** Script for detecting such problem is LATITUDE_LONGITUDE.py
- One precinct(ADDR_PCT_CD) can only be mapping to one particular borough(BORO_NM). **27 pairs of ADDR_PCT_CD & BORO_NM has conflict mapping problem.** Script for detecting such problem is Precinct_BORO_mapping.py

3.5 Other issues

- No wrong number of fields record is found.All rows of data contains 24 columns of data.
- No duplicate compliant number record is found. (CMPLNT_NUM.py) No duplicate index
- No wrong report date problem is found. Ex. date time like 2/31/2017 is invalid.

Data Cleaning

After detecting and finding all possible issues, we need to get rid of these invalid data rows.

3.6 Handling conflicting data

One important aspect in data cleaning is how we decide whether some data should be considered invalid or not when it has conflicts with other data. For example, we can see conflicts in Precinct Num- Borough Name
If we see such problem, we will look through all the conflicting data and decide a threshold number for invalid number. Here we decide the threshold number to be 100. So if one mapping relationship has appeared less than 100, it is highly possible that it is wrong recorded information.

28 lines (27 sloc) | 697 Bytes

```
(('104', 'BROOKLYN'), 1)
(('104', 'MANHATTAN'), 1)
(('104', 'QUEENS'), 75114)
(('106', 'BROOKLYN'), 1)
(('106', 'QUEENS'), 61494)
(('114', 'BRONX'), 2)
(('114', 'QUEENS'), 91694)
(('121', 'BROOKLYN'), 1)
(('121', 'STATEN ISLAND'), 17374)
(('13', 'BROOKLYN'), 1)
(('13', 'MANHATTAN'), 74442)
(('14', 'BROOKLYN'), 1)
(('14', 'MANHATTAN'), 119414)
(('23', 'BRONX'), 3)
(('23', 'MANHATTAN'), 66690)
(('25', 'BRONX'), 1)
(('25', 'MANHATTAN'), 67479)
(('26', 'BROOKLYN'), 1)
(('26', 'MANHATTAN'), 34274)
(('6', 'BRONX'), 1)
(('6', 'MANHATTAN'), 54904)
(('7', 'BROOKLYN'), 1)
(('7', 'MANHATTAN'), 40885)
(('71', 'BRONX'), 1)
(('71', 'BROOKLYN'), 72088)
(('9', 'BROOKLYN'), 1)
(('9', 'MANHATTAN'), 62164)
```

Figure 3: all borough and precinct mapping

104	BROOKLYN
104	MANHATTAN
106	BROOKLYN
114	BRONX
121	BROOKLYN
13	BROOKLYN
14	BROOKLYN
23	BRONX
25	BRONX
26	BROOKLYN
6	BRONX
7	BROOKLYN
71	BRONX
9	BROOKLYN

Figure 4: Invalid mapping

3.7 Collecting key information of potential invalid row of data

For the each mapping problem, we output one .out file illustrating how one row of data will be regarded as invalid.For example:

120	ENDAN WELFARE INCOMP
124	KIDNAPPING
124	KIDNAPPING AND RELATED OFFENSES
345	ENDAN WELFARE INCOMP
364	AGRICULTURE & MRKTS LAW-UNCLASSIFIED
364	OTHER STATE LAWS (NON PENAL LAW)
677	NYS LAWS-UNCLASSIFIED VIOLATION

Figure 5: Invalid mapping

This is the .out file recording all invalid offense code mapping data. In this file, each row has one key-value pair. So as long as KY_CD equals to that key and OFNS_DESC equals to value, that row of data is considered as invalid. This is pretty much like where clause in sql query.
So in the data cleaning script, we load these .out file data.

```
if os.path.isfile('./Other_Issue/InvalidOffenseCodeMapping.out/part-00000'):
    file = open("./Other_Issue/InvalidOffenseCodeMapping.out/part-00000")
    while 1:
        line = file.readline().rstrip('\n')
        if not line:
            break
        [key, value] = line.split("\t")
        if not invalidOffenseCodeDetailRecord.has_key(key):
            invalidOffenseCodeDetailRecord[key] = []
        if value not in invalidOffenseCodeDetailRecord.get(key):
            invalidOffenseCodeDetailRecord.get(key).append(value)
```

Figure 6: data cleaning script

We load these out file in python dictionary data(HashMap) structure or set data structure(HashSet). Then we decide if certain row of data is invalid or not, we will check in these hashmap and hashset to see if key info has certain matches.

3.8 Script for data cleaning

We write a script for data cleaning. DataClean.py We have filters for the following issue: 1)If this field has invalid null value 2)If data in this field has invalid base type 3)If data in this field should be in certain range 4)If data in this field has mapping problem

```
cleanedData = lines.map(toStrip) \
                   .filter(isNullValue) \
                   .filter(dataTypeCheck) \
                   .filter(isInSpecificRange) \
                   .filter(mappingCheck)
```

Figure 7: filters for data clean

Results

After we do the cleaning work, we have our new cleanData.csv. All data summary work is based on clean data, not on raw data. The original data has size of 1.3GB. The clean data is 933MB left.

Data summary

3.9 Borough crime

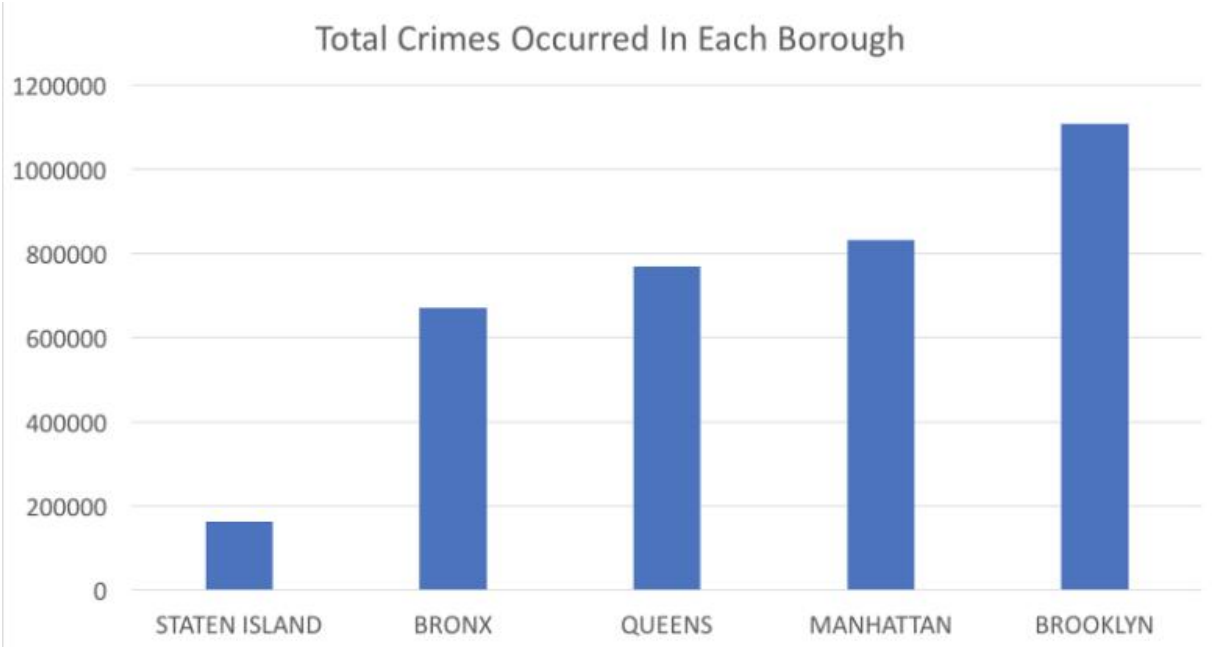

Figure 8: borough crime distribution

As we can see from figure , Brooklyn takes the first place for number of incidences occurred, which is 1,107,843. Manhattan takes the second place, which is 831,636. Queens and Bronx follow behind, and staten island has the least number of incidences occurred. However, as is shown in NYC distribution 2016[2], Brooklyn has the most population, and Queens has the second most population, and Manhattan takes the third place, which is contradicted the fact that manhattan has the second number of incidences occurred. (BoroughNameDistribution.py, PrecinctDistribution.py)

3.10 Precinct crime

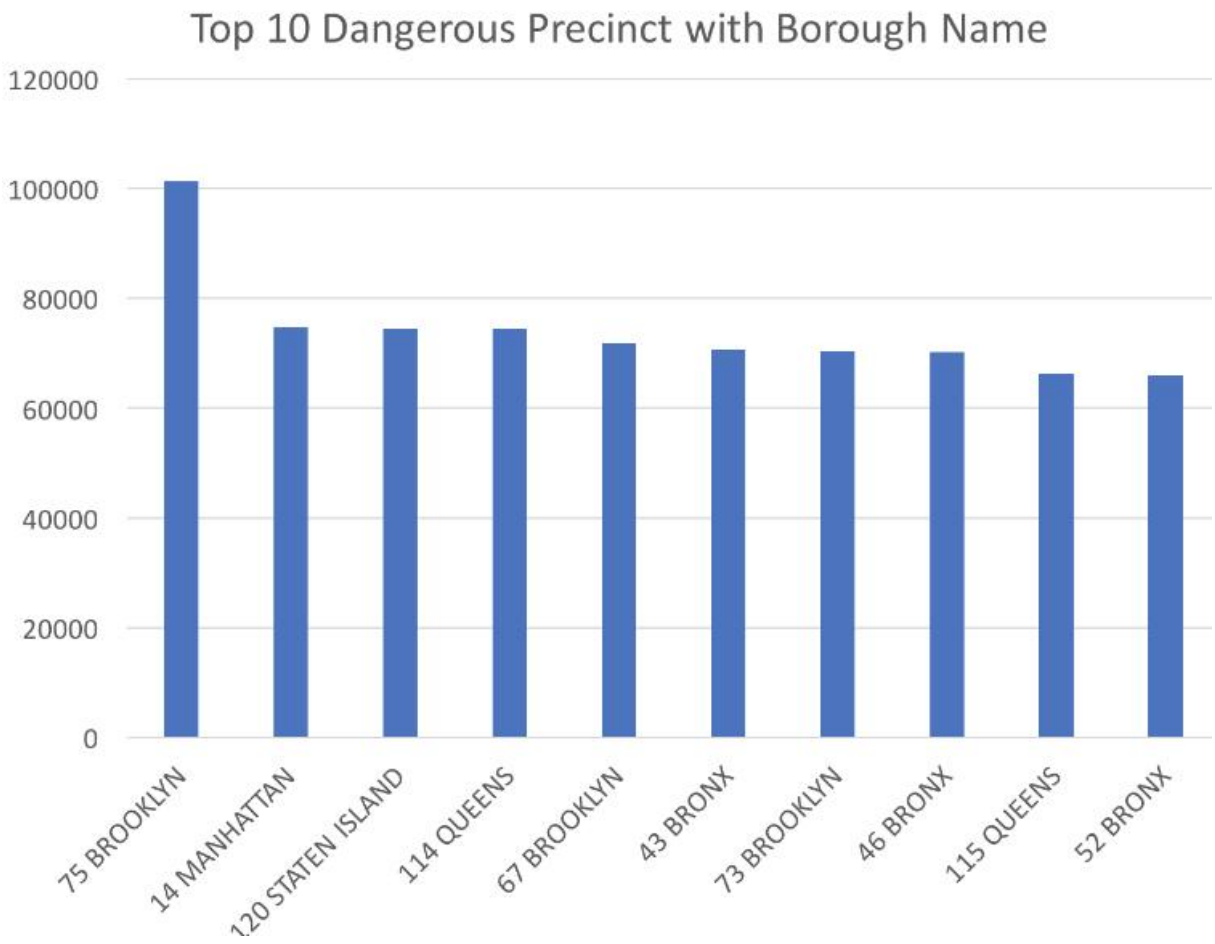

Figure 9: Precinct crime distribution

The figure below shows the top 10 precincts with borough name with respect to number of incidents. From the figure below, we can see the most dangerous precinct is 75 Brooklyn, over 100,000 times of crime happened here, which is far more than the other nine. Meanwhile, Brooklyn has 3 precincts in this top 10 list, which is the same as Bronx. The number of precincts in Manhattan, Queens and Staten Island is 1, 2 and 1 respectively. This distribution can also reflect the borough distribution above, in which Brooklyn takes the first place and Manhattan takes the second.

3.11 Date of occurrence and date reported to police

Figure 10: Date of occurrence and date reported to police distribution

In this diagram, crime happens least in February and peaks at August. In fact we can see there is an obvious raise from February to May and fluctuates during the next few months. The difference between the max one and the least one is about 70,000. We guess the reason is due to the temperature change, which means criminals also dont want to go outside in winter.

3.12 Offense numbers in daytime

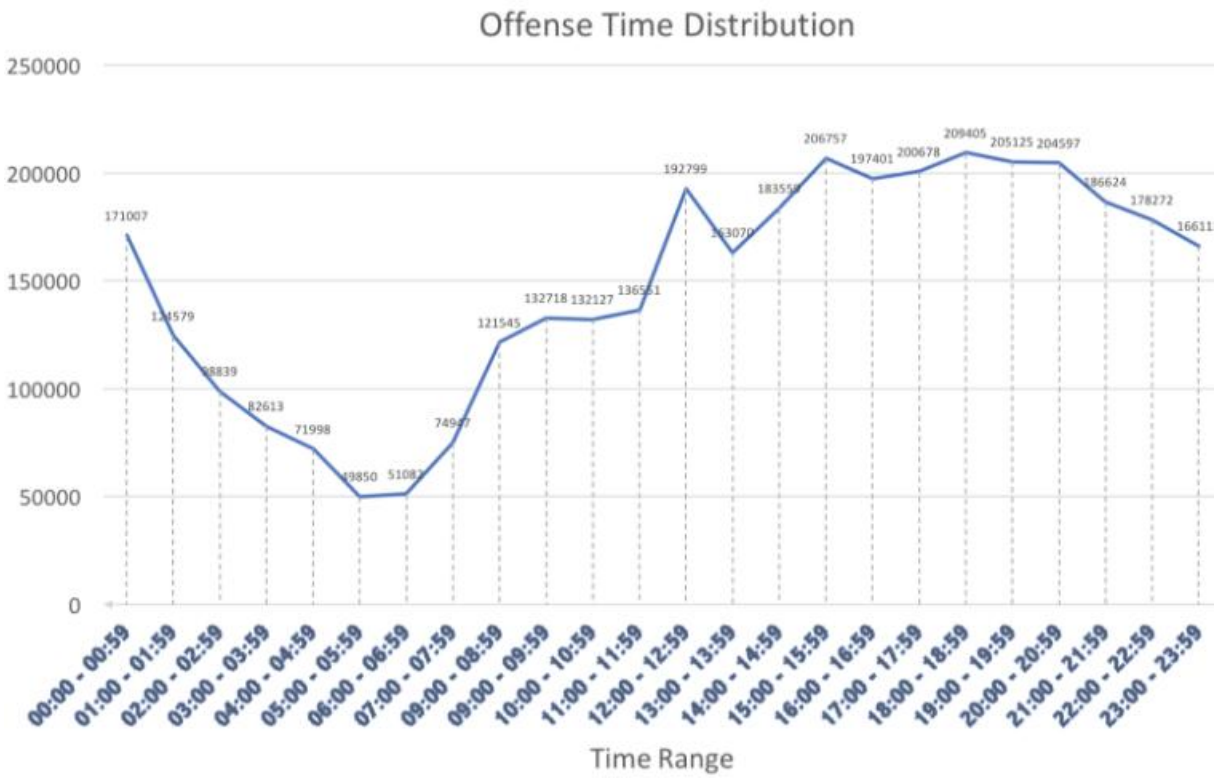

Figure 11: Offense numbers in daytime

In this picture, we can see offence most happen between 15:00 and 21:00, which reminds us to avoid hanging in dangerous area during this period. The crime number decreases sharply during 00:00 to 05:00 and raises significantly during 06:00 to 12:00. Its corresponded to peoples sleeping custom. The maximum occurrence between 15:00 to 15:59 is around 5 times of that minimum between 05:00 to 05:59. Whats interesting is some criminals are night-owls and they will commit offenses to those who are also night-owls.

3.13 Offense level

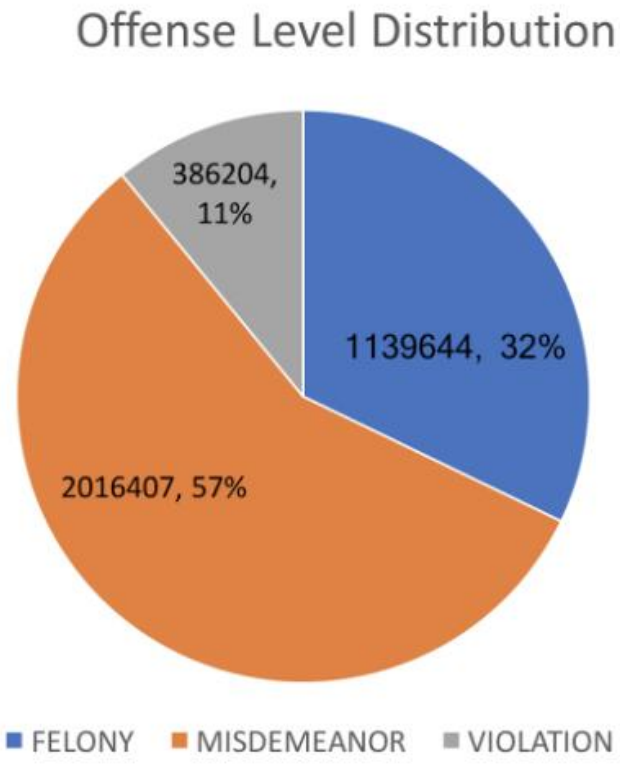

Figure 12: Offense level distribution

The analysis of offense level distribution is quite fitting our assumption. Felony is the most part which follows the misdemeanor, while the former one is about twice of the later one. Despite of felony and misdemeanor, violation seems to be few. We guess its because violation couldnt be counted as a crime and people would choose not to report if they have meet some not so much annoying offense.

3.14 Offense level distribution corresponding to borough

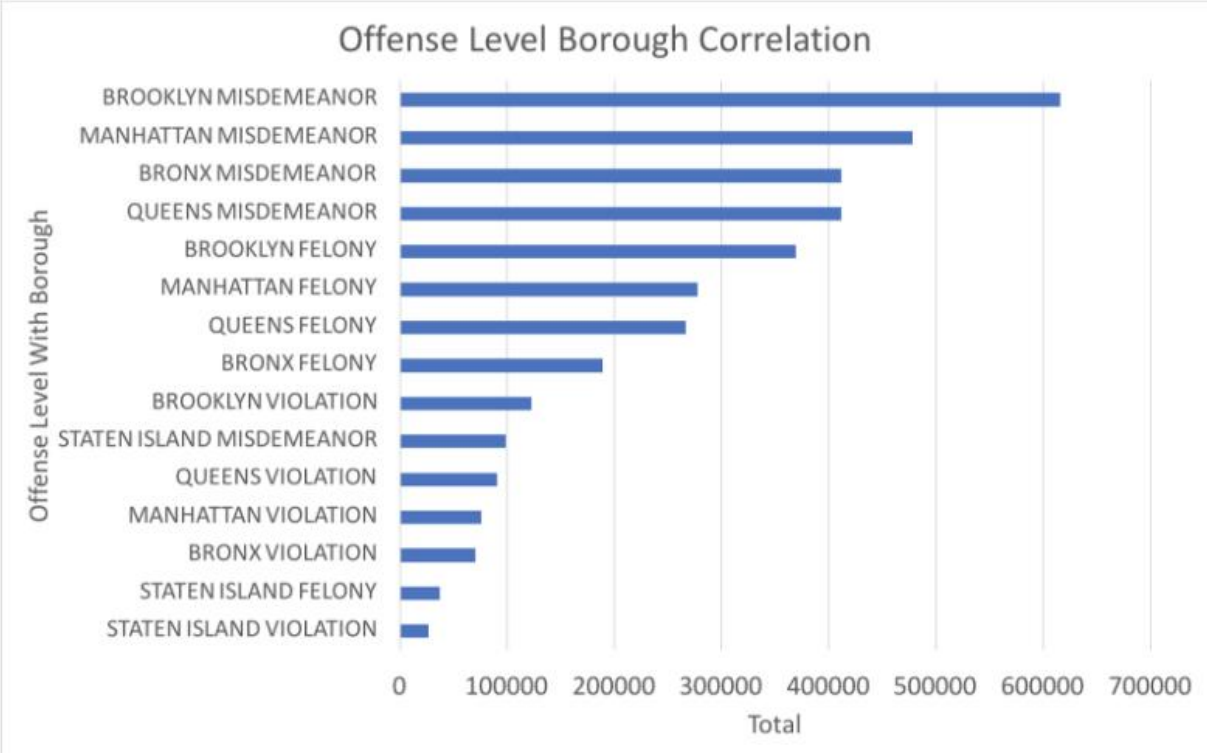

Figure 13: Offense level distribution corresponding to borough

Among the this chart, we can learn that Brooklyn is the most dangerous precinct, no matter comparing misdemeanor, felony or violation, more than 600,000 misdemeanor, 380,000 felony and 80,000 violation happening here. Whats attracting us is that the felony happened in staten island is pretty less than the other counterparts, even less than violation of other four precinct. The ratio among misdemeanor, felony and violation in all five precinct matches the former analysis, misdemeanor is the most and violation is the least no matter in which area.

3.15 Offense type

Top 20 Most Occurred Offense Event With Level			
Rank	Offense Detail	Level	Total
1	PETIT LARCENY	MISDEMEANOR	616025
2	HARRASSMENT 2	VIOLATION	378619
3	ASSAULT 3 & RELATED OFFENSES	MISDEMEANOR	354209
4	CRIMINAL MISCHIEF & RELATED OF	MISDEMEANOR	330679
5	GRAND LARCENY	FELONY	321773
6	OFF. AGNST PUB ORD SENSBLTY &	MISDEMEANOR	188292
7	DANGEROUS DRUGS	MISDEMEANOR	185416
8	BURGLARY	FELONY	170623
9	ROBBERY	FELONY	136197
10	FELONY ASSAULT	FELONY	126996
11	GRAND LARCENY OF MOTOR VEHICLE	FELONY	89563
12	MISCELLANEOUS PENAL LAW	FELONY	78140
13	OFFENSES AGAINST PUBLIC ADMINI	MISDEMEANOR	69570
14	CRIMINAL MISCHIEF & RELATED OF	FELONY	56334
15	INTOXICATED & IMPAIRED DRIVING	MISDEMEANOR	51255
16	DANGEROUS WEAPONS	MISDEMEANOR	49441
17	CRIMINAL TRESPASS	MISDEMEANOR	43877
18	DANGEROUS DRUGS	FELONY	40469
19	DANGEROUS WEAPONS	FELONY	34733
20	FORGERY	FELONY	34570

Figure 14: Offense type

3.16 Specific description of Premises

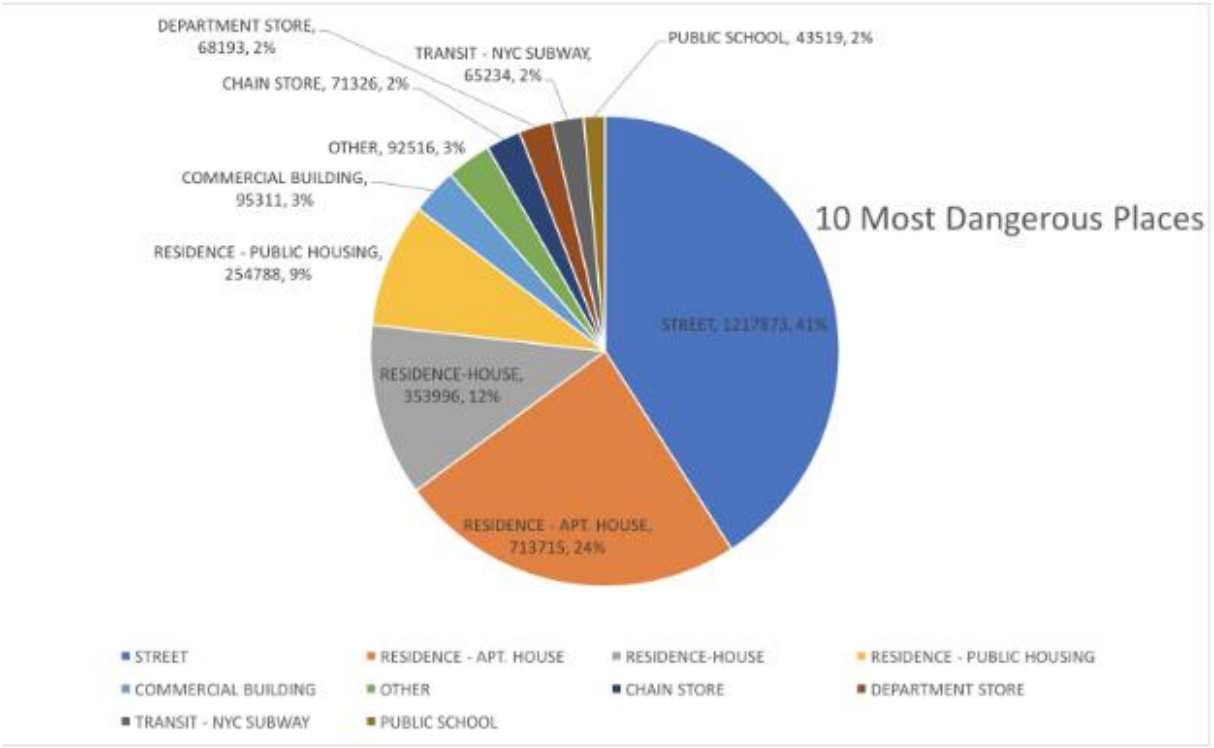

Figure 15: Specific description of Premises

As is shown in the chart above, most incidents happen on the streets and residence houses or apartments, which take up roughly 77%. Thats where people usually gather together or stay in. Some other places include commercial buildings, stores, public transportation, schools.

4 Individual contributions

Minda Fang: Participated in writing data issue detecting code. Participated in writing data summary code. Participated in writing the final report.
Qiming Zhang: Participated in writing data issue detecting code. Participated in writing data summary code. Participated in writing the final report.
Mindi Mao: Participated in writing data summary code. Accessed dumbo to generate the cleaned dataset. Participated in writing the final report.

5 Conclusions

After we finished all analysis, we found that Brooklyn is the most dangerous precinct and staten island is the safest one in whatever analysis dimensionality we use. As we are international students, we heard that Brooklyn is not as dangerous as twenty years ago, and we didnt meet any sort of crime during past days. Therefore, we guess the first reason is the dataset contains the data from 2006 to 2015, which is the past ten years, and Brooklyn is possible to improve itself that hasnt been reflected in the dataset till now;

the second reason is that we live in certain safe area and commute during safe period. The advice we could drive from this report is that if you dont know NYC pretty well and hope to live in safety, staten island and Roosevelt Island may be a good choice, if you live in some dangerous area like Brooklyn and Manhattan, you should avoid hanging around at dangerous area during 15:00 to 1:00. We wish people could enjoy their lives at NYC and benefit from our report.

6 Reference

1. Project Code link: https://github.com/mmdtoycar/NYU_Bigdata_Final/
2. NYC 2016 population
3. Dataset link: NYPD (2006-2016)