School of Information Studies

FINAL PROJECT
IST – 652
SCRIPTING FOR DATA ANALYSIS
FALL 2022

AIRBNB AND CRIME RATE IN NYC

PREPARED BY : **HENDI KUSHTA**

Table of Contents

1. Introduction....3
2. Data Exploratory and Preprocessing....4
    2.1 NYPD_Complaint_Data_Historic....4
    2.2 Air Bnb....7
    2.3 Question 1 Analysis....7
    2.4 Question 2 Analysis....10
    2.5 Question 3 Analysis....12
    2.6 Question 4 Analysis....15
3. Output Files....17
    3.1 Analysis 1....17
    3.2 Analysis 2....18
    3.3 Analysis 3....19
    3.4 Analysis 4....20

1. Introduction

Since 2008, visitors and hosts have used Airbnb to expand travel options and provide more unique, individualized ways of seeing the world.

Property owners may rent out their rooms to travelers looking for a place to stay through a service called Airbnb, which stands for "Air Bed and Breakfast." Travelers can choose to rent a shared space with individual rooms, a large space for a group, or the full property for themselves.

Brian Chesky and Joe Gebbia, two industrial designers who had just relocated to San Francisco, founded Airbnb. The couple decided to make up the money they needed by renting out their apartment to people who couldn't find motels to stay at while attending local trade exhibits because they couldn't afford the rent for their loft at the time. They provided air mattresses for their visitors to sleep on in the apartment's living room, and they prepared a fresh breakfast each morning. Since then, Airbnb has emerged as a pioneer in the peer-to-peer leasing of real estate.

I will examine the reasons for differences in costs, reviews, and the types of rooms in each location to determine which neighborhood is the most desirable. I will compare airbnb prices also by checking the crime rate impact in the prices. I will provide analysis about the prices and what might be the main factors that play a key role in assigning them.

I will give answer to questions below but not only while I work on the project:

- 1- What is the number of each room type in each of the neighbourhood group? Which are the neighbourhoods with the highest numbers of properties?
- 2- What is the average room type price in each of the neighbourhood_group?
- 3- Which are top neighbourhoods with the highest number of airbnb properties in NYC? What are the average prices in the 5 neighbourhood_groups? What is the number of bookings and average price/night in the top 2 neighbourhood groups?
- 4- Which are the safest neighborhood_groups and which are the least safier? Which type of offensive level occurs mostly in the neighbourhood_groups?

The first dataset that I have chosen for this project can be found in insideAirBnb.com named: “listings.csv” and it’s link is http://insideairbnb.com/get-the-data/. This dataset details the metrics and listing activity in 2019 in NYC. This data file contains all the details required to learn more about hosts, their geographic accessibility, and the relevant metrics to generate analysis and reach conclusions. It has 48895 rows and 16 attributes.

The second dataset that I have chosen for this project can be found in New York City Open Data named: “NYPD_Complaint_Data_Historic” and it’s link is https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i.This dataset has data from 2006 to 2021. It represents all criminal activities in NYC. It has 35 columns and more than 7.83 million records.

The project's direction and the many data gathering and analysis methods employed will be highly influenced by the preliminary results, but the approaches listed below are some of the ones I will most likely to use.

Due to the many diverse ways the datasets are arranged, I will need to do data munging, manipulation, grouping, and merging. I might combine the datasets using latitude and longitude.

I might determine a linear relationship between the cost of AirBnBs and the crime rate and forecast the value of properties in various NYC neighborhoods.

Make comparisons using basic statistics on crime rate and airbnb prices in different neighborhoods, but not only.

Potential development tasks:

I will need to group airbnbs based on the neighborhoods, type of the properties, hosts etc which will produce different tables.

A new dataset that keeps only records of crimes during 2021 will be needed to created since the airbnb dataset is with properties only for 2021. I will use historical data on the crime rates to make predictions on the properties prices.

Development of a recommendation system for the neighborhoods with low crime rates and good airbnb prices.

2. Data Exploratory and Preprocessing

I started this project by importing the necessary libraries I needed in the project. Pandas, numpy, matplotlib and seaborn which I used to plot graphs.

2.1 NYPD_Complaint_Data_Historic

Firstly I read the NYPD complaint historical data as below:

```
# read our first dataset
listing_df = pd.read_csv('NYPD_Complaint_Data_Historic.csv')
```

Then I started exploring and cleaning it. I checked for the number of records in the dataset. Than I printed the first records just to make sure that the data was properly imported. My next step was to check the data types for each of the variables.

```
# check our dataset length
len(listing_df)
```

```
7825499
```

```
# show first rows
listing_df.head()
```

	CMPLNT_NUM	CMPLNT_FR_DT	CMPLNT_FR_TM	CMPLNT_TO_DT	CMPLNT_TO_TM	ADDR_PCT_CD	RPT_DT	KY_CD	OFNS_DESC	PD_CD	...	S
0	506547392	03/29/2015	20:30:00	NaN	NaN	32.0	03/30/2015	351	CRIMINAL MISCHIEF & RELATED OF	254.0	...	
1	629632533	02/06/2015	23:15:00	NaN	NaN	52.0	02/07/2015	341	PETIT LARCENY	333.0	...	
2	757203902	11/21/2015	00:15:00	11/21/2015	00:20:00	75.0	11/21/2015	341	PETIT LARCENY	321.0	...	
3	250364015	06/09/2015	21:42:00	06/09/2015	21:43:00	10.0	06/10/2015	361	OFF. AGNST PUB ORD SENSBLTY &	639.0	...	
4	955500320	11/10/2015	19:40:00	11/10/2015	19:45:00	19.0	11/10/2015	341	PETIT LARCENY	333.0	...	

5 rows × 35 columns

```
# data type for each attribute in our first dataset
listing_df.dtypes
```

```
CMPLNT_NUM              int64
CMPLNT_FR_DT           object
CMPLNT_FR_TM           object
CMPLNT_TO_DT           object
CMPLNT_TO_TM           object
ADDR_PCT_CD           float64
RPT_DT                 object
KY_CD                   int64
OFNS_DESC              object
PD_CD                 float64
PD_DESC                object
CRM_ATPT_CPTD_CD       object
LAW_CAT_CD             object
BORO_NM                object
LOC_OF_OCCUR_DESC      object
PREM_TYP_DESC          object
JURIS_DESC             object
JURISDICTION_CODE     float64
PARKS_NM               object
HADEVELOPT             object
HOUSING_PSA            object
X_COORD_CD            float64
Y_COORD_CD            float64
SUSP_AGE_GROUP         object
SUSP_RACE              object
SUSP_SEX               object
TRANSIT_DISTRICT      float64
Latitude              float64
Longitude             float64
Lat_Lon                object
PATROL_BORO            object
STATION_NAME           object
VIC_AGE_GROUP          object
VIC_RACE               object
VIC_SEX                object
dtype: object
```

Since the data has more than 7.83 million records, so I wont need to read the data again, I have assigned the data to another variable, just in case I mess them up and need to start work from the beginning. The next step was to convert the event date to a date time type in order for us to use it and gave it a format of year, month and day, where year has 4 digits, month and day 2. Next, I filtered data and selected only events that happened on 2021. From 35 columns, I chose only 8 columns that I think are most important for analysis.

```
# since our data is very big, have assigned to another
# variable so we do not need to run it from the beginning in case of some problems
listing_df1 = listing_df
# convert rpt_dt in a datetime datatype
listing_df['RPT_DT'] = pd.to_datetime(listing_df['RPT_DT'])
# change the format of event date
listing_df['RPT_DT'] = pd.to_datetime(listing_df['RPT_DT'], format='%Y-%m-%d')
# filter only rows greater than 2020-12-31 since the data is very big
listing_df = listing_df.loc[(listing_df['RPT_DT'] > '2020-12-31')]
len(listing_df)
```

```
449506
```

```
# select only some of the columns from our dataset
nyc_crime_rates_filtered = listing_df[["RPT_DT", "OFNS_DESC", "CRM_ATPT_CPTD_CD", "LAW_CAT_CD", "BORO_NM", "PREM_
```

```
nyc_crime_rates_filtered.head()
```

	RPT_DT	OFNS_DESC	CRM_ATPT_CPTD_CD	LAW_CAT_CD	BORO_NM	PREM_TYP_DESC	Latitude	Longitude
681129	2021-12-23	MISCELLANEOUS PENAL LAW	COMPLETED	FELONY	BRONX	DEPARTMENT STORE	40.830443	-73.871349
681150	2021-12-31	FELONY ASSAULT	COMPLETED	FELONY	BRONX	RESIDENCE-HOUSE	40.817577	-73.865994
681170	2021-12-22	ARSON	COMPLETED	FELONY	BRONX	STREET	40.859744	-73.826259
681171	2021-12-31	FORGERY	COMPLETED	FELONY	QUEENS	STREET	40.746775	-73.750567
681230	2021-12-31	PETIT LARCENY	COMPLETED	MISDEMEANOR	QUEENS	STREET	40.754364	-73.912557

The I checked the data for null values and duplicated values. I dropped them. The last step I did in this dataset, was to change the names of the variables I used to make the data analysis.

```
# check information about the filtered dataset.
nyc_crime_rates_filtered.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 449506 entries, 681129 to 1677434
Data columns (total 8 columns):
 #   Column            Non-Null Count   Dtype         
---  ------            --------------   -----         
 0   RPT_DT            449506 non-null  datetime64[ns]
 1   OFNS_DESC         449497 non-null  object        
 2   CRM_ATPT_CPTD_CD  449345 non-null  object        
 3   LAW_CAT_CD        449506 non-null  object        
 4   BORO_NM           448355 non-null  object        
 5   PREM_TYP_DESC     448236 non-null  object        
 6   Latitude          449506 non-null  float64       
 7   Longitude         449506 non-null  float64       
dtypes: datetime64[ns](1), float64(2), object(5)
memory usage: 30.9+ MB
```

```
# drop null values
nyc_crime_rates_filtered = nyc_crime_rates_filtered.dropna()
len(nyc_crime_rates_filtered)
```

```
447496
```

```
# drop duplicate values
nyc_crime_rates_filtered = nyc_crime_rates_filtered.drop_duplicates()
len(nyc_crime_rates_filtered)
```

```
428192
```

```
# check unique values in boro_nm column
nyc_crime_rates_filtered.BORO_NM.unique()
```

```
array(['BRONX', 'QUEENS', 'BROOKLYN', 'MANHATTAN', 'STATEN ISLAND'],
      dtype=object)
```

```
# rename columns
nyc_crime_rates_filtered.columns = ['event_date', 'offense_desc', 'crime_status', 'offense_level', 'neighbourhoo
nyc_crime_rates_filtered.head()
```

In the end I just saved a filtered csv file of NYPD complaint data.

2.2 Air Bnb

Secondly, I read the AirBnb data as below:

```
# Read second dataset
airbnb_listings_df = pd.read_csv('listings.csv')
```

I followed the same logic for the second dataset. But in this dataset, apart from removing null values and duplicates, I also changed to upped case the values for the neighborhood_group attribute.

```
# convert to upper case values in neighbourhood_group
airbnb_listings_df['neighbourhood_group'] = airbnb_listings_df['neighbourhood_group'].str.upper()
```

```
# check unique values
airbnb_listings_df.neighbourhood_group.unique()
```

```
array(['BROOKLYN', 'QUEENS', 'BRONX', 'MANHATTAN', 'STATEN ISLAND'],
      dtype=object)
```

```
# drop license column
airbnb_listings_df = airbnb_listings_df.drop('license', axis=1)
# drop null values
airbnb_listings_df = airbnb_listings_df.dropna()
# drop duplicate values
airbnb_listings_df = airbnb_listings_df.drop_duplicates()
len(airbnb_listings_df)
```

```
31505
```

In the end I save a csv file for the filtered and processed data.

```
# save the filtered and processed dataset
airbnb_listings_df.to_csv('airbnb_listings_df.csv')
```

2.3 Question 1 Analysis

1. What is the number of each room type in each of the neighborhood group? Which are the neighborhoods with the highest numbers of properties?

As my first question I have found what are the neighborhoods that have the highest number of properties and the type of the most frequent rooms in this neighborhood groups.

First of all I created a new data frame only with the necessary attributes for the first questions. I than used the count function to find the number of each room type.

```
# create a dataset only with 2 columns
nr_roomtype_in_neighbourhood = airbnb_listings_df[["neighbourhood_group", "room_type"]]
nr_roomtype_in_neighbourhood.head()
```

	neighbourhood_group	room_type
0	BROOKLYN	Hotel room
1	BROOKLYN	Private room
2	QUEENS	Entire home/apt
4	MANHATTAN	Private room
5	MANHATTAN	Private room

```
# count number of room types in total
nr_roomtype_in_neighbourhood['room_type'].value_counts()
```

```
Entire home/apt    18373
Private room       12560
Shared room          416
Hotel room           156
Name: room_type, dtype: int64
```

Next step was to group by the data frame I initially created with the attributes I took in consideration and I used the aggregate function to count the room type number in each of the neighborhood groups. I gave a name to the newly created column and reset the index. Finally I save the data in a csv file.

```
# groupby the new dataset using 2 columns type and year.
# find the number of movies shown through years using count
# aggregate function.
# give a name to the column of the aggregate function
# reset index so the dataset can appear as below.

nr_roomtype_in_neighbourhood = nr_roomtype_in_neighbourhood.groupby(['room_type', 'neighbourhood_group']).agg({'
nr_roomtype_in_neighbourhood.columns = ['nr_of_each_roomtype_in_neighbourhoods']
nr_roomtype_in_neighbourhood = nr_roomtype_in_neighbourhood.reset_index()
print(nr_roomtype_in_neighbourhood)
```

```
          room_type neighbourhood_group  nr_of_each_roomtype_in_neighbourhoods
0   Entire home/apt               BRONX                                      1
1   Entire home/apt            BROOKLYN                                      1
2   Entire home/apt           MANHATTAN                                      1
3   Entire home/apt              QUEENS                                      1
4   Entire home/apt       STATEN ISLAND                                      1
5        Hotel room            BROOKLYN                                      1
6        Hotel room           MANHATTAN                                      1
7        Hotel room              QUEENS                                      1
8      Private room               BRONX                                      1
9      Private room            BROOKLYN                                      1
10     Private room           MANHATTAN                                      1
11     Private room              QUEENS                                      1
12     Private room       STATEN ISLAND                                      1
13      Shared room               BRONX                                      1
14      Shared room            BROOKLYN                                      1
15      Shared room           MANHATTAN                                      1
16      Shared room              QUEENS                                      1
17      Shared room       STATEN ISLAND                                      1
```

```
# save table as a csv file
nr_roomtype_in_neighbourhood.to_csv('nr_roomtype_in_neighbourhood.csv')
```

```
# pivot dataset
nr_roomtype_in_neighbourhood_pivot = pd.pivot_table(nr_roomtype_in_neighbourhood, values='nr_of_each_roomtype_ir
                                    index='room_type', columns='neighbourhood_group')

# replace null values in the pivoted table with 0
nr_roomtype_in_neighbourhood_pivot = nr_roomtype_in_neighbourhood_pivot.replace(np.nan,0)
# converted to type integer
nr_roomtype_in_neighbourhood_pivot = nr_roomtype_in_neighbourhood_pivot.astype(int)
nr_roomtype_in_neighbourhood_pivot
```

neighbourhood_group	BRONX	BROOKLYN	MANHATTAN	QUEENS	STATEN ISLAND
room_type					
Entire home/apt	1	1	1	1	1
Hotel room	0	1	1	1	0
Private room	1	1	1	1	1
Shared room	1	1	1	1	1

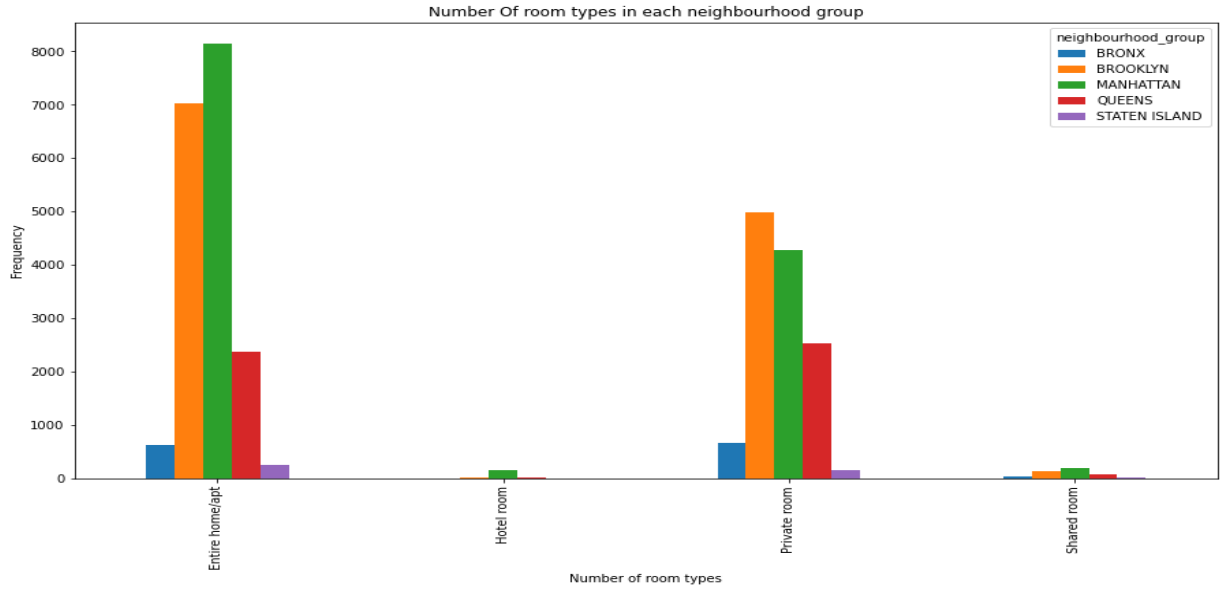
```
# save pivoted table as a csv file
nr_roomtype_in_neighbourhood_pivot.to_csv('nr_roomtype_in_neighbourhood_pivot.csv')
```

I then pivot the table, replace all the null values with zero and use the numpy library to convert the values in integers. I create a new csv file with the pivoted table.

Lastly I plot the graph. As I see, apart from private room that ts the highest in Brooklyn, the highest number for all the other types of rooms are in

```
# plot the graph of number Of room types in each neighbourhood group

nr_roomtype_in_neighbourhood_pivot.plot(kind='bar', figsize=(15,8))
plt.xlabel("Number of room types")
plt.ylabel("Frequency")
plt.title("Number Of room types in each neighbourhood group")
plt.savefig('Analysis1.png', dpi=300)
plt.show()
```



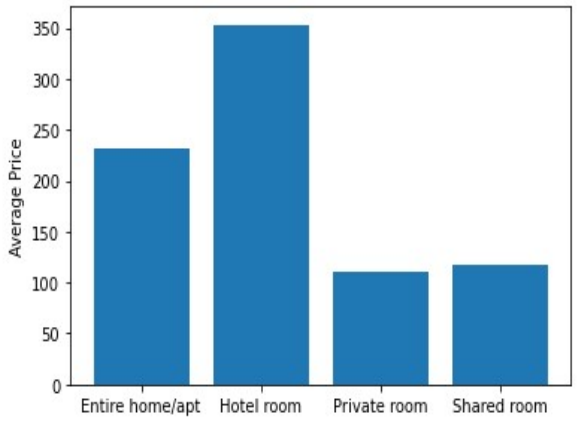
2.4 Question 2 Analysis

2. What is the average room type price in each of the neighborhood group?

To give answer to the second question, I firstly find the average price for each of the room type by using the mean function. Then I have plotted a simple chart for the means.

```
# find the average price for each room type and plot
price_by_room = airbnb_listings_df.groupby('room_type',as_index=False)['price'].mean()
print(price_by_room)
plt.bar(price_by_room['room_type'], price_by_room['price'])
plt.ylabel("Average Price")
plt.show()
```

```
         room_type       price
0  Entire home/apt  232.166766
1       Hotel room  353.141026
2     Private room  111.346656
3      Shared room  117.290865
```



I group by the dataset depending on 2 columns room type and neighborhood group and fund the average price for each room type in each of the neighborhood groups. I join grouped with all the data after that.

```
# group by dataset depending on 2 columns and find average prices
price_by_room_place = airbnb_listings_df.groupby(['room_type', 'neighbourhood_group'],as_index=False)['price'].m

price_by_room_place
```

:

	room_type	neighbourhood_group	price
0	Entire home/apt	BRONX	158.586430
1	Entire home/apt	BROOKLYN	207.449786
2	Entire home/apt	MANHATTAN	275.080556
3	Entire home/apt	QUEENS	184.461020
4	Entire home/apt	STATEN ISLAND	161.691667
5	Hotel room	BROOKLYN	228.285714
6	Hotel room	MANHATTAN	369.878571
7	Hotel room	QUEENS	189.888889
8	Private room	BRONX	86.668721
9	Private room	BROOKLYN	84.431321
10	Private room	MANHATTAN	166.477475
11	Private room	QUEENS	79.383670
12	Private room	STATEN ISLAND	81.978417
13	Shared room	BRONX	36.346154
14	Shared room	BROOKLYN	68.093750
15	Shared room	MANHATTAN	178.239362
16	Shared room	QUEENS	76.219178
17	Shared room	STATEN ISLAND	59.000000

```
# join grouped by with all data
with_average = pd.merge(airbnb_listings_df,price_by_room_place,
         left_on=['room_type', 'neighbourhood_group'],
         right_on=['room_type', 'neighbourhood_group'])

with_average['residual_price'] = with_average['price_y'] - with_average['price_x']
```

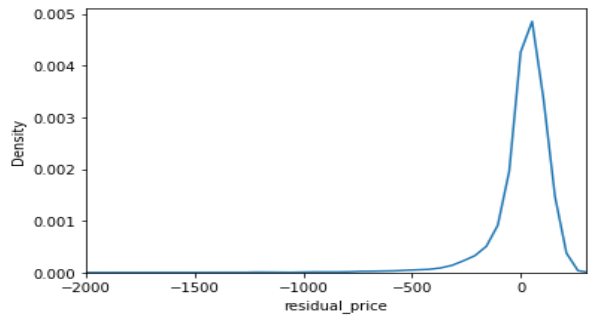
I find the smallest residuals and draw the distribution of the prices among the property types.

```
# find greatest residuals
small = with_average[['name', 'neighbourhood_group', 'room_type',
              'price_x', 'price_y', 'residual_price']].sort_values('residual_price').head()

small
```

	name	neighbourhood_group	room_type	price_x	price_y	residual_price
30417	WELCOME HOME 15 MINUTES TO MANHATTAN BOOK TODAY	BRONX	Private room	9994	86.668721	-9907.331279
9805	The Gregory Hotel, Tailored King with Sofa Bed	MANHATTAN	Private room	10000	166.477475	-9833.522525
9802	The Gregory Hotel, Tailored Double (2 Double B...	MANHATTAN	Private room	10000	166.477475	-9833.522525
9803	The Gregory Hotel, Tailored King	MANHATTAN	Private room	10000	166.477475	-9833.522525
9804	The Gregory Hotel, Tailored Double Queen	MANHATTAN	Private room	10000	166.477475	-9833.522525

```
sns.kdeplot(with_average['residual_price'])
plt.xlim([-2000, 300])
plt.show()
```



2.5 Question 3 Analysis

3. Which are top neighborhoods with the highest number of airbnb properties in NYC? What are the average prices in the 5 neighborhood_groups? What is the number of bookings and average price/night in the top 2 neighborhood groups?

In this question I find which of the neighborhoods have the highest number of properties in NYC. I also show the average prices of properties in top 5 neighborhoods in NYC and the average price per night in top 2 neighborhood groups.

I found firstly the number of properties in each neighborhood in NYC. Then I have provided a plot for the number of properties. As I see Bedford-Stuyvesant has the highest number of airbnb properties.

```
#top 10 neighbourhoods sub groups
airbnb_listings_df.neighbourhood.value_counts().head(10)
```

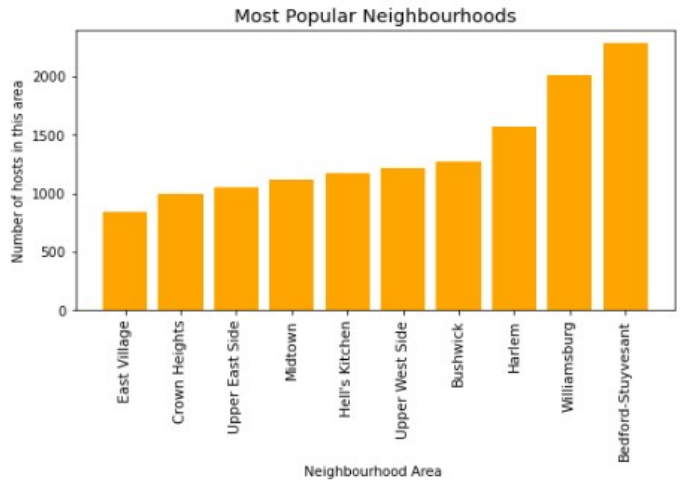
```
Bedford-Stuyvesant    2284
Williamsburg          2006
Harlem                1566
Bushwick              1277
Upper West Side       1218
Hell's Kitchen        1177
Midtown               1117
Upper East Side       1048
Crown Heights          992
East Village           845
Name: neighbourhood, dtype: int64
```

```
# bar graph using Matplotlib to the the top 10 neighbourhood sub groups

top_10 = airbnb_listings_df.neighbourhood.value_counts().head(10)
plt.figure(figsize=(8, 4))
x = list(top_10.index)
y = list(top_10.values)
x.reverse()
y.reverse()

plt.title('Most Popular Neighbourhoods', size=14)
plt.ylabel('Number of hosts in this area')
plt.xlabel('Neighbourhood Area ')
plt.xticks( rotation='vertical', size=11)

plt.bar(x, y , color='orange')
plt.savefig('Most Popular Neighbourhoods.png', dpi=300)
plt.show()
```



I created a filter to further analyze data. From the top 5 neighborhoods which are Williamsburg,Bedford-Stuyvesant,Harlem, Bush wick & Upper west Side, all of them are located in Brooklyn and Manhattan, as I also check from the print of the newly created data frame.

```
# create a filter to further analyse the data from the top 5 neighbourhoods
# the top 5 neighbourhoods -Williamsburg,Bedford-Stuyvesant,Harlem, Bushwick & Upperwest Side- are
# situated either in Manhattan or Brooklyn.
top_5_neighbourhood =airbnb_listings_df.loc[(airbnb_listings_df['neighbourhood'] == 'Williamsburg') |
                                (airbnb_listings_df['neighbourhood'] == 'Bedford-Stuyvesant') |
                                (airbnb_listings_df['neighbourhood'] == 'Harlem') |
                                (airbnb_listings_df['neighbourhood'] == 'Bushwick') |
                                (airbnb_listings_df['neighbourhood'] == 'Upper West Side')]
```

```
top_5_neighbourhood
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minim
7	49048	B and B Style Rooms for Rent w bath	35935	Angela	BROOKLYN	Bedford-Stuyvesant	40.68290	-73.95701	Private room	90	
9	5121	BlissArtsSpace!	7356	Garon	BROOKLYN	Bedford-Stuyvesant	40.68535	-73.95512	Private room	60	
17	82928	BEAUTIFUL 2 BEDROOM APARTMENT	451545	Ruthven	BROOKLYN	Bedford-Stuyvesant	40.68433	-73.94469	Entire home/apt	150	
18	5203	Cozy Clean Guest Room - Family Apt	7490	MaryEllen	MANHATTAN	Upper West Side	40.80380	-73.96751	Private room	75	
20	6848	Only 2 stops to Manhattan studio	15991	Allen & Irina	BROOKLYN	Williamsburg	40.70935	-73.95342	Entire home/apt	84	
...	...	...	...	...	...	...	...	...	...	...	
39820	17662555	AMAZING CITY VIEWS 15 min Times Sq 30Day minimum	23156390	Emily	MANHATTAN	Upper West Side	40.78965	-74.00601	Entire home/apt	171	

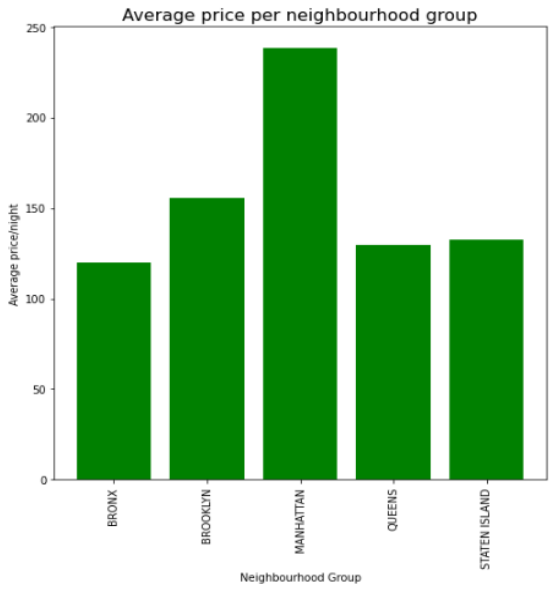
Find and plot the mean price of the listings in each neighborhood groups.

```
# find mean price for each neighbourhood_group
neighb_mean= airbnb_listings_df.groupby('neighbourhood_group').mean()

plt.figure(figsize =(8,8))

neighbourhood_group = [neighbourhood_group for neighbourhood_group, airbnb_listings_df in airbnb_listings_df.gro

plt.bar(neighbourhood_group, neighb_mean['price'], color='green')
plt.xticks(neighbourhood_group, rotation='vertical', size=10)
plt.xlabel('Neighbourhood Group')
plt.title('Average price per neighbourhood group' , size=16)
plt.ylabel('Average price/night')
plt.savefig('Average price per night in each neighbourhood group.png', dpi=300)
plt.show()
```



Lastly, I have created another filter to compare only Brooklyn and Manhattan average prices per night.

Then I found the number of properties in each of these neighborhood groups and the mean. I have created a small data frame with the values that I find.

```
# comparison of the top 2 neighbourhood groups, Brooklyn and Manhattan

# get the number of bookings by neighbourhood group for the 2 top groups
count_man=airbnb_listings_df.loc[airbnb_listings_df['neighbourhood_group'] == 'MANHATTAN'].host_id.count()
count_brook=airbnb_listings_df.loc[airbnb_listings_df['neighbourhood_group'] == 'BROOKLYN'].host_id.count()

mean_pr_man=airbnb_listings_df.loc[airbnb_listings_df['neighbourhood_group'] == 'MANHATTAN'].price.mean()
mean_pr_brook=airbnb_listings_df.loc[airbnb_listings_df['neighbourhood_group'] == 'BROOKLYN'].price.mean()



test = [count_brook, mean_pr_brook, count_man,mean_pr_man]
# since the mean price to rent a place in Brooklyn almost 2 times lower than in manhattam,
# the number of booked rooms is almost the same as in manhattam
test
```

```
[12132, 155.42334322453016, 12721, 238.30665828158163]
```

```
# create a dew dataframe that shows the number of bookings
# and average price/night in the top 2 neighbourhood groups
d = {'name': ['Brooklyn', 'Manhattan'],'mean_price': [155.42, 238.30], 'bookings': [12132, 12721]}

test_df = pd.DataFrame(d)
test_df.set_index('name', inplace=True)
test_df
```

name	mean_price	bookings
Brooklyn	155.42	12132
Manhattan	238.30	12721

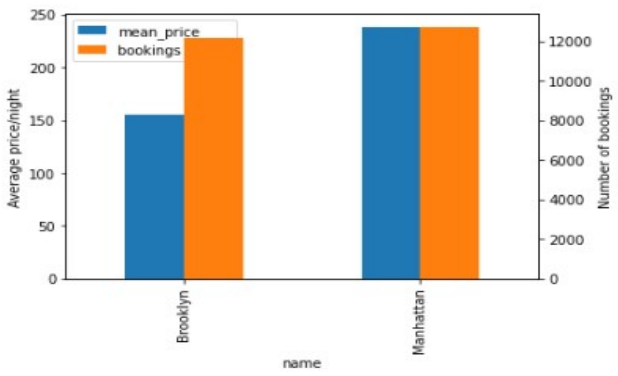
Plot the graph to show visually. As we see from the graph, the number of bookings in Brooklyn is almost as high as in Manhattan. The reason might be because the average price per night is lower than in Manhattan.

```
# a double bar chart to show the number of bookings
# and average price/night in Manhattan and Brooklyn

fig = plt.figure(figsize=(10,5))
test_df.plot.bar( secondary_y= 'bookings', label = 'Name')

ax1, ax2 = plt.gcf().get_axes()
ax1.set_ylabel('Average price/night')
ax2.set_ylabel('Number of bookings')
plt.savefig('Average price per night in Brooklyn and Manhattan.png', dpi=300)
plt.show()
```

```
<Figure size 720x360 with 0 Axes>
```



2.6 Question 4 Analysis

4. Which are the safest neighborhood_groups and which are the least safer? Which type of offensive level occurs mostly in the neighborhood_groups?

In the last question, I show the safest neighborhood groups, an the level of offenses that have occurred mostly in these groups.

In this part of the code, I create a new data frame with the necessary attributes for analysis. I have also shown the number of each type of offense and the number of offenses in each neighborhood group.

```
# select only neccessary columns
nr_offense_level_in_neighbourhood = nyc_crime_rates_filtered[["offense_level", "neighbourhood_group"]]
```

```
# count number of each offense level
nyc_crime_rates_filtered['offense_level'].value_counts()
```

```
MISDEMEANOR    211313
FELONY         144234
VIOLATION       72645
Name: offense_level, dtype: int64
```

```
# check how many offenses have occurred in each neighbourhood group
nyc_crime_rates_filtered['neighbourhood_group'].value_counts()
```

```
BROOKLYN         122015
MANHATTAN        103582
QUEENS            93832
BRONX             90452
STATEN ISLAND     18311
Name: neighbourhood_group, dtype: int64
```

Group by the new dataset 2 columns of data frame and find the number of offense levels that have occurred in each neighborhood group. Give a name to the newly created attribute and reset the index. Save the data frame as a csv file.

```
# groupby the new dataset using 2 columns offense_level and neighbourhood_group.
# find the number of offense levels shown in each neighborhood using count
# aggregate function.
# give a name to the column of the aggregate function
# reset index so the dataset can appear as below.

nr_offense_level_in_neighbourhood = nr_offense_level_in_neighbourhood.groupby(['offense_level', 'neighbourhood_g
nr_offense_level_in_neighbourhood.columns = ['nr_offense_level_in_neighbourhood']
nr_offense_level_in_neighbourhood = nr_offense_level_in_neighbourhood.reset_index()
print(nr_offense_level_in_neighbourhood)
```

	offense_level	neighbourhood_group	nr_offense_level_in_neighbourhood
0	FELONY	BRONX	30694
1	FELONY	BROOKLYN	42073
2	FELONY	MANHATTAN	35299
3	FELONY	QUEENS	31083
4	FELONY	STATEN ISLAND	5085
5	MISDEMEANOR	BRONX	43624
6	MISDEMEANOR	BROOKLYN	58363
7	MISDEMEANOR	MANHATTAN	53752
8	MISDEMEANOR	QUEENS	46377
9	MISDEMEANOR	STATEN ISLAND	9197
10	VIOLATION	BRONX	16134
11	VIOLATION	BROOKLYN	21579
12	VIOLATION	MANHATTAN	14531
13	VIOLATION	QUEENS	16372
14	VIOLATION	STATEN ISLAND	4029

```
nr_offense_level_in_neighbourhood.to_csv("nr_offense_level_in_neighbourhood.csv")
```

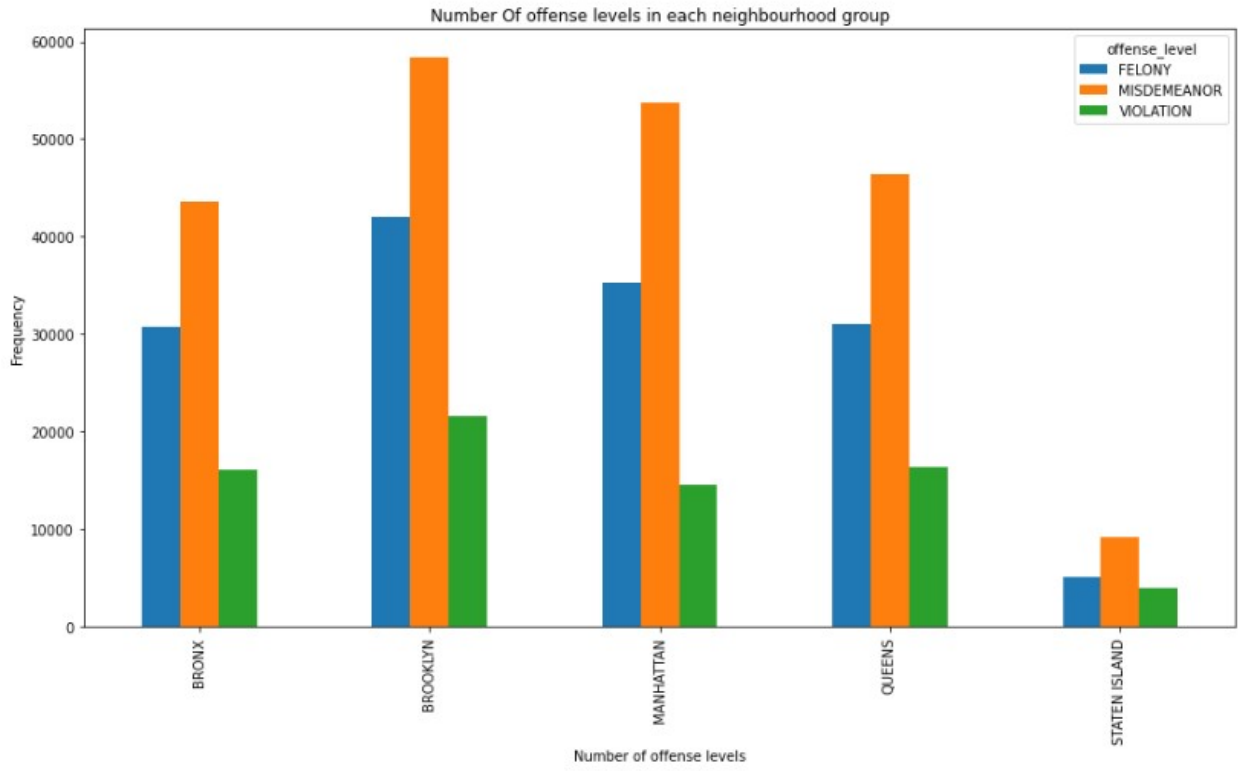
The next step was to create a pivot table for e better presentation which shows each offense level type in each of the neighborhood groups. I replaced the null values with 0 and converted the values to integer data types.

```
# pivot dataset
nr_offense_level_in_neighbourhood_pivot = pd.pivot_table(nr_offense_level_in_neighbourhood, values='nr_offense_l
                                          index='neighbourhood_group', columns='offense_level')

# replace null values in the pivoted table with 0
nr_offense_level_in_neighbourhood_pivot = nr_offense_level_in_neighbourhood_pivot.replace(np.nan,0)
# converted to type integer
nr_offense_level_in_neighbourhood_pivot = nr_offense_level_in_neighbourhood_pivot.astype(int)
nr_offense_level_in_neighbourhood_pivot
```

offense_level	FELONY	MISDEMEANOR	VIOLATION
neighbourhood_group			
BRONX	30694	43624	16134
BROOKLYN	42073	58363	21579
MANHATTAN	35299	53752	14531
QUEENS	31083	46377	16372
STATEN ISLAND	5085	9197	4029

Lastly I have plotted a chart showing also the distribution of each offense level in each of the neighborhood groups.



3. Output Files

3.1 Analysis 1

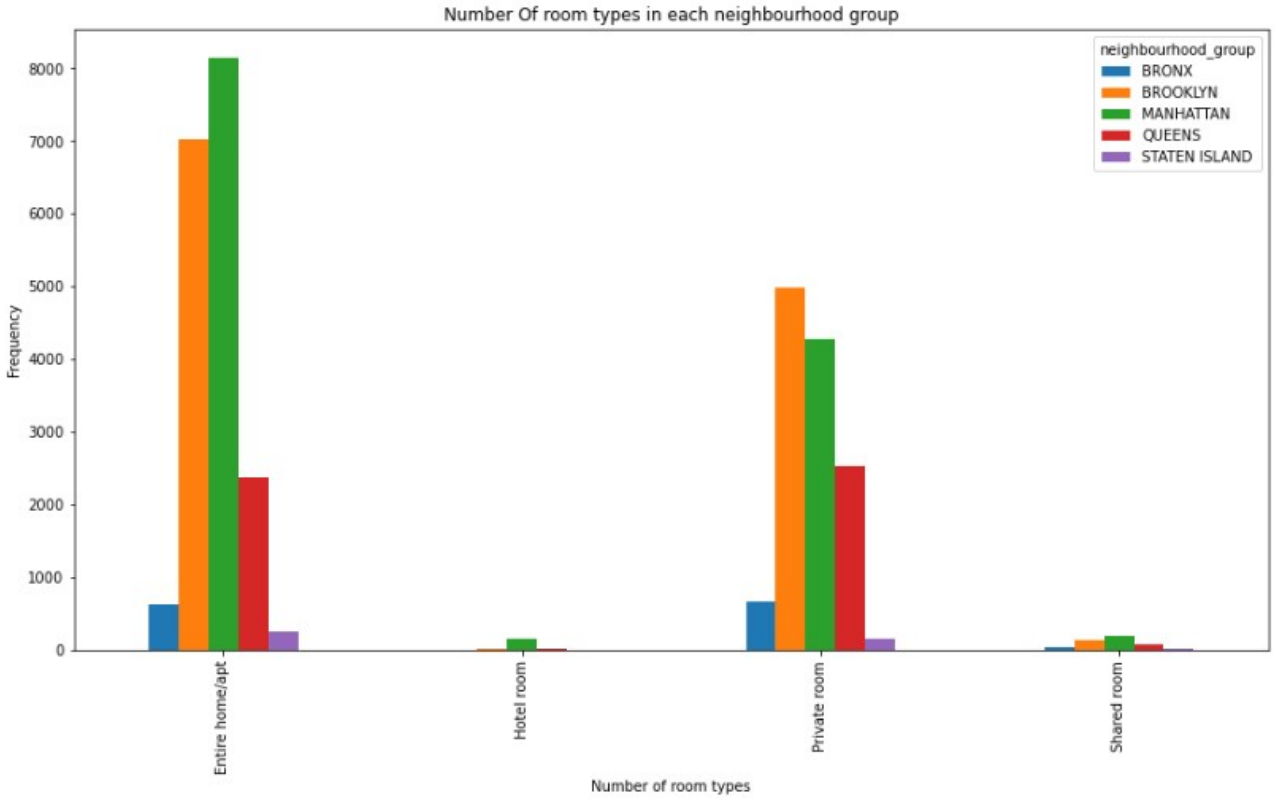
1- 2 csv files

		room_type	neighbourhood_group	nr_of_each_roomtype_in_neighbourhoods
2	0	Entire home/apt	BRONX	1
3	1	Entire home/apt	BROOKLYN	1
4	2	Entire home/apt	MANHATTAN	1
5	3	Entire home/apt	QUEENS	1
6	4	Entire home/apt	STATEN ISLAND	1
7	5	Hotel room	BROOKLYN	1
8	6	Hotel room	MANHATTAN	1

and

	room_type	BRONX	BROOKLYN	MANHATTAN	QUEENS	STATEN ISLAND
2	Entire home/apt	1	1	1	1	1
3	Hotel room	0	1	1	1	0
4	Private room	1	1	1	1	1
5	Shared room	1	1	1	1	1

2- 1 graph

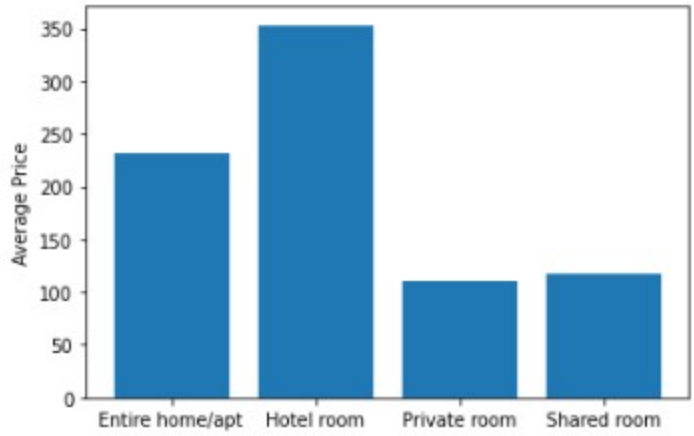


3.2 Analysis 2

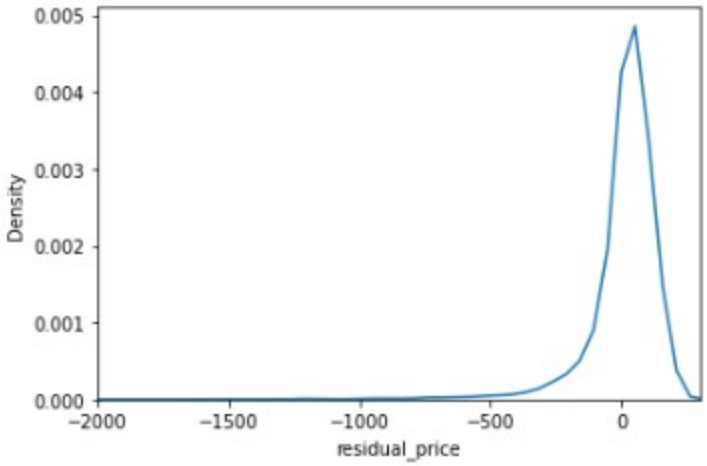
1- 1 csv file

		room_type	neighbourhood_group	price
1	0	Entire home/apt	BRONX	158.5864297253635
2	1	Entire home/apt	BROOKLYN	207.44978601997147
3	2	Entire home/apt	MANHATTAN	275.0805558971836
4	3	Entire home/apt	QUEENS	184.46101980615256
5	4	Entire home/apt	STATEN ISLAND	161.69166666666666
6	5	Hotel room	BROOKLYN	228.28571428571428
7	6	Hotel room	MANHATTAN	369.87857142857143

2- 2 graphs



and

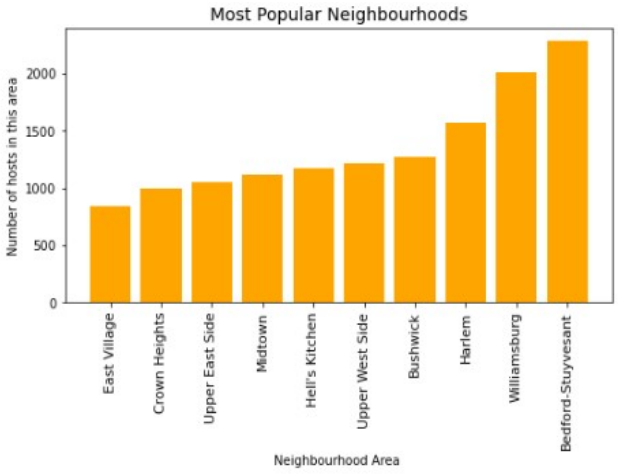


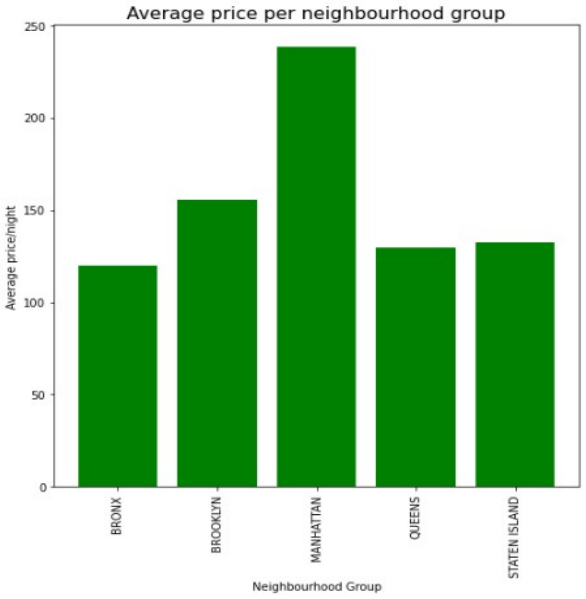
3.3 Analysis 3

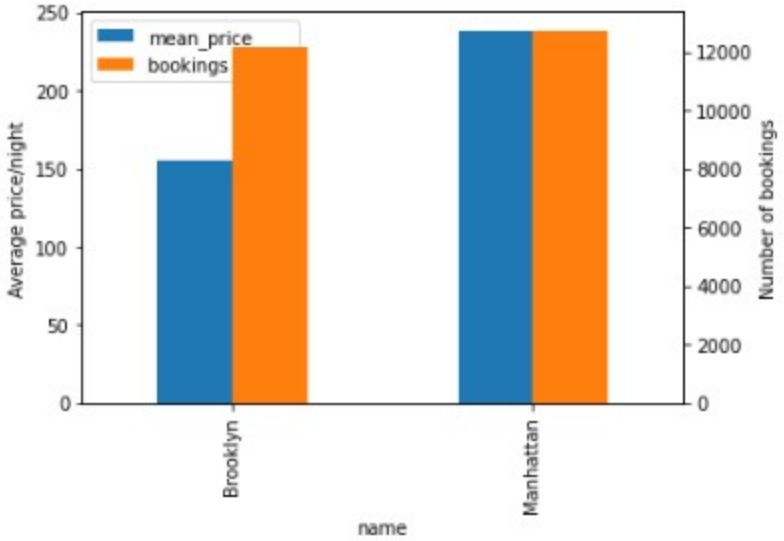
1- 1 csv file

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nigh
7	49048	B and B Style Rooms for Rent w bath	35935	Angela	BROOKLYN	Bedford-Stuyvesant	40.6829	-73.95701	Private room	90	30
9	5121	BlissArtsSpace!	7356	Garon	BROOKLYN	Bedford-Stuyvesant	40.68535	-73.95512	Private room	60	30
17	82928	BEAUTIFUL 2 BEDROOM APARTMENT	451545	Ruthven	BROOKLYN	Bedford-Stuyvesant	40.68433	-73.94469	Entire home/apt	150	30
18	5203	Cozy Clean Guest Room - Family Apt	7490	MaryEllen	MANHATTAN	Upper West Side	40.8038	-73.96751	Private room	75	2
20	6848	Only 2 stops to Manhattan studio	15991	Allen & Irina	BROOKLYN	Williamsburg	40.70935	-73.95342	Entire home/apt	84	30
22	83243	Brooklyn Cove 1 Br Apt w/ Garden In Bushwick!!	453519	Julian	BROOKLYN	Bushwick	40.68769	-73.91788	Entire home/apt	77	30
32	93313	MAISON DES SIRENES 2	25183	Nathalie	BROOKLYN	Bedford-Stuyvesant	40.68413	-73.93817	Entire home/apt	145	2

2- 3 graphs







3.4 Analysis 4

1- 2 csv files

		offense_level	neighbourhood_group	nr_offense_level_in_neighbourhood
1	0	FELONY	BRONX	30694
2	1	FELONY	BROOKLYN	42073
3	2	FELONY	MANHATTAN	35299
4	3	FELONY	QUEENS	31083
5	4	FELONY	STATEN ISLAND	5085
6	5	MISDEMEANOR	BRONX	43624
7	6	MISDEMEANOR	BROOKLYN	58363

and

	neighbourhood_group	FELONY	MISDEMEANOR	VIOLATION
1	BRONX	30694	43624	16134
2	BROOKLYN	42073	58363	21579
3	MANHATTAN	35299	53752	14531
4	QUEENS	31083	46377	16372
5	STATEN ISLAND	5085	9197	4029

2- 1 graph