

QUESTION 1:

"Question 1: Which type of complaint should the Department of Housing Preservation and Development of New York City focus on first?"

"Notes on Problem 1: In the first question, we should look the 311 data set. 311 service is for New York citizens where citizens can report non-emergency requests from the city. Important note: The actual dataset is much bigger than this one and can be downloaded from the link I've provided in README file, however I selected the columns that I am going to use. I didn't choose the all recommended columns because of size issues. The dataset is in data.csv file and transferred here as df, using pandas. For answering question 1, we should find the column that contains complaint type information and find the complaint type that has occurred most."

```
import pandas as pd
```

```
#local path of data from my computer
```

```
body1 = r"C:\Users\PRADYUM\Downloads\fhrw-4uyv.csv"
```

```
df = pd.read_csv(body1)
```

```
df.head()
```

	created_date	unique_key	complaint_type
incident_zip \			
0 2020-08-23T09:51:03.000	47338119	HEAT/HOT WATER	
11234.0			
1 2020-08-23T14:52:28.000	47339485	PLUMBING	
11694.0			
2 2020-08-23T07:21:11.000	47338136	UNSANITARY CONDITION	
10011.0			
3 2020-08-23T01:40:51.000	47343253	UNSANITARY CONDITION	
11355.0			
4 2020-08-23T09:45:44.000	47336184	ELEVATOR	
10039.0			

	incident_address	street_name
0	6614 VETERANS AVENUE	VETERANS AVENUE
1	193 BEACH 112 STREET	BEACH 112 STREET
2	103 WEST 14 STREET	WEST 14 STREET
3	137-27 HOLLY AVENUE	HOLLY AVENUE
4	2890 FREDERICK DOUGLASS BOULEVARD	FREDERICK DOUGLASS BOULEVARD

	address_type	city
0	ADDRESS	BROOKLYN
1	ADDRESS	Rockaway Park
2	ADDRESS	NEW YORK

3	ADDRESS	Flushing
4	ADDRESS	NEW YORK

	resolution_description	borough
latitude \		
0	The following complaint conditions are still o...	BROOKLYN
40.619789		
1	The following complaint conditions are still o...	QUEENS
40.580445		
2	The following complaint conditions are still o...	MANHATTAN
40.737564		
3	The following complaint conditions are still o...	QUEENS
40.749927		
4	The following complaint conditions are still o...	MANHATTAN
40.828110		

	longitude	closed_date	location_type	status
0	-73.913164	NaN	RESIDENTIAL BUILDING	Open
1	-73.833540	NaN	RESIDENTIAL BUILDING	Open
2	-73.997301	NaN	RESIDENTIAL BUILDING	Open
3	-73.820885	NaN	RESIDENTIAL BUILDING	Open
4	-73.938052	NaN	RESIDENTIAL BUILDING	Open

```
dff = pd.read_csv(r"C:\Users\PRADYUM\Desktop\PLUTO_for_WEB\
BX_18v122.csv")
```

```
C:\Users\PRADYUM\Anaconda3\lib\site-packages\IPython\core\
interactiveshell.py:3049: DtypeWarning: Columns (19,20,22,23,64,65,80)
have mixed types. Specify dtype option on import or set
low_memory=False.
```

```
interactivity=interactivity, compiler=compiler, result=result)
```

```
dff.head()
```

	Borough	Block	Lot	CD	CT2010	CB2010	SchoolDist	Council
ZipCode \								
0	BX	2260	1	201	19.0	1022.0	7.0	8.0
10454.0								
1	BX	2260	4	201	19.0	1022.0	7.0	8.0
10454.0								
2	BX	2260	10	201	19.0	1022.0	7.0	8.0
10454.0								
3	BX	2260	17	201	19.0	1022.0	7.0	8.0
10454.0								
4	BX	2260	18	201	19.0	1022.0	7.0	8.0
10454.0								

	FireComp	...	ZMCode	Sanborn	TaxMap	EDesignNum	APPBBL
APPDate \							
0	L029	...	NaN	209S016	20901.0	E-143	0.0
							NaN

1	L029	...	NaN	209S016	20901.0	E-143	0.0	NaN
2	L029	...	NaN	209S016	20901.0	E-143	0.0	NaN
3	L029	...	NaN	209S016	20901.0	E-143	0.0	NaN
4	L029	...	NaN	209S016	20901.0	E-143	0.0	NaN

	PLUTOMapID	FIRM07_FLAG	PFIRM15_FLAG	Version
0	1	NaN	NaN	18V1
1	1	NaN	NaN	18V1
2	1	NaN	NaN	18V1
3	1	NaN	NaN	18V1
4	1	NaN	NaN	18V1

[5 rows x 87 columns]

```
dff.dropna(subset = ['ZipCode'], inplace=True)
```

```
dff.ZipCode.unique()
```

```
array([10454., 10455., 10451., 10456., 10452., 10453., 10465., 10474.,
       11370., 10459., 10472., 10457., 10460., 10458., 10468., 10463.,
       10467., 10470., 10466., 10473., 10462., 10461., 10469., 10475.,
       10464., 10471.])
```

Therefor 26 unique zipcodes in Bronx PLUTO dataset

```
dff2 = pd.read_csv(r"C:\Users\PRADYUM\Desktop\PLUTO_for_WEB\
QN_18v1.csv")
```

```
dff2.dropna(subset = ['ZipCode'], inplace=True)
```

```
array([11101., 11109., 11104., 11377., 11106., 11102., 11103., 11105.,
       11370., 11369., 11372., 11373., 11385., 11368., 11421., 11355.,
       11374., 11375., 11367., 11415., 11378., 11379., 11418., 11432.,
       11356., 11420., 11357., 11354., 11697., 11693., 11358., 11361.,
       11365., 11364., 11360., 11359., 11435., 11366., 11423., 11363.,
       11362., 11427., 11426., 11428., 11004., 11005., 11040., 11001.,
       11416., 11417., 11419., 11433., 11413., 11434., 11412., 11429.,
       11411., 11414., 11430., 11436., 11422., 11691., 11692., 11694.,
       11695.])
```

```
len(dff2.ZipCode.unique())
```

65

Therefor 65 unique zipcodes in QUEENS PLUTO dataset

Coming back to main dataset df

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4220047 entries, 0 to 4220046
Data columns (total 15 columns):
created_date          object
unique_key            int64
complaint_type        object
incident_zip          float64
incident_address      object
street_name           object
address_type          object
city                  object
resolution_description object
borough               object
latitude              float64
longitude              float64
closed_date           object
location_type         object
status                object
dtypes: float64(3), int64(1), object(11)
memory usage: 482.9+ MB

df["complaint_type"].unique()

array(['HEAT/HOT WATER', 'PLUMBING', 'UNSANITARY CONDITION',
      'ELEVATOR',
      'WATER LEAK', 'PAINT/PLASTER', 'GENERAL', 'FLOORING/STAIRS',
      'SAFETY', 'APPLIANCE', 'DOOR/WINDOW', 'ELECTRIC',
      'OUTSIDE BUILDING', 'Appliance', 'Unsanitary Condition',
      'Safety',
      'Electric', 'HPD Literature Request', 'HEATING',
      'GENERAL CONSTRUCTION', 'PAINT - PLASTER', 'NONCONST',
      'CONSTRUCTION', 'General', 'AGENCY', 'VACANT APARTMENT',
      'STRUCTURAL', 'Outside Building', 'Plumbing', 'Mold'],
      dtype=object)

df.incident_address.isnull().sum()

52821

df["complaint_type"].isnull().sum()

0

df["complaint_type"].value_counts()
```

HEAT/HOT WATER	1306687
UNSANITARY CONDITION	474493
PLUMBING	425976
PAINT/PLASTER	353335
DOOR/WINDOW	213033
HEATING	205564
ELECTRIC	202684
WATER LEAK	200478
GENERAL	153518
FLOORING/STAIRS	141572
GENERAL CONSTRUCTION	139580
PAINT - PLASTER	101832
APPLIANCE	91452
NONCONST	80239
SAFETY	53618
HPD Literature Request	52820
OUTSIDE BUILDING	7299
ELEVATOR	7259
Unsanitary Condition	5499
CONSTRUCTION	1481
General	1163
Safety	424
Plumbing	11
AGENCY	9
VACANT APARTMENT	8
Outside Building	6
Appliance	4
Mold	1
Electric	1
STRUCTURAL	1

Name: complaint_type, dtype: int64

It is clear that the Heat/Hot Water problem is the one that NYC should focus on first! However, it's tricky. You may notice that there is big similarity between two groups. HEAT/HOT WATER and HEATING. Before 2014, this dataset is using HEATING label but after 2014, the label changed as "HEATING/HOT WATER". So, we need to change "HEATING" labels as "HEATING/HOT WATER" and analyze them together.

```
import numpy as np
import datetime as dt
df['complaint_type'] =
np.where(df['complaint_type']=='HEATING', 'HEAT/HOT
WATER', df['complaint_type'])

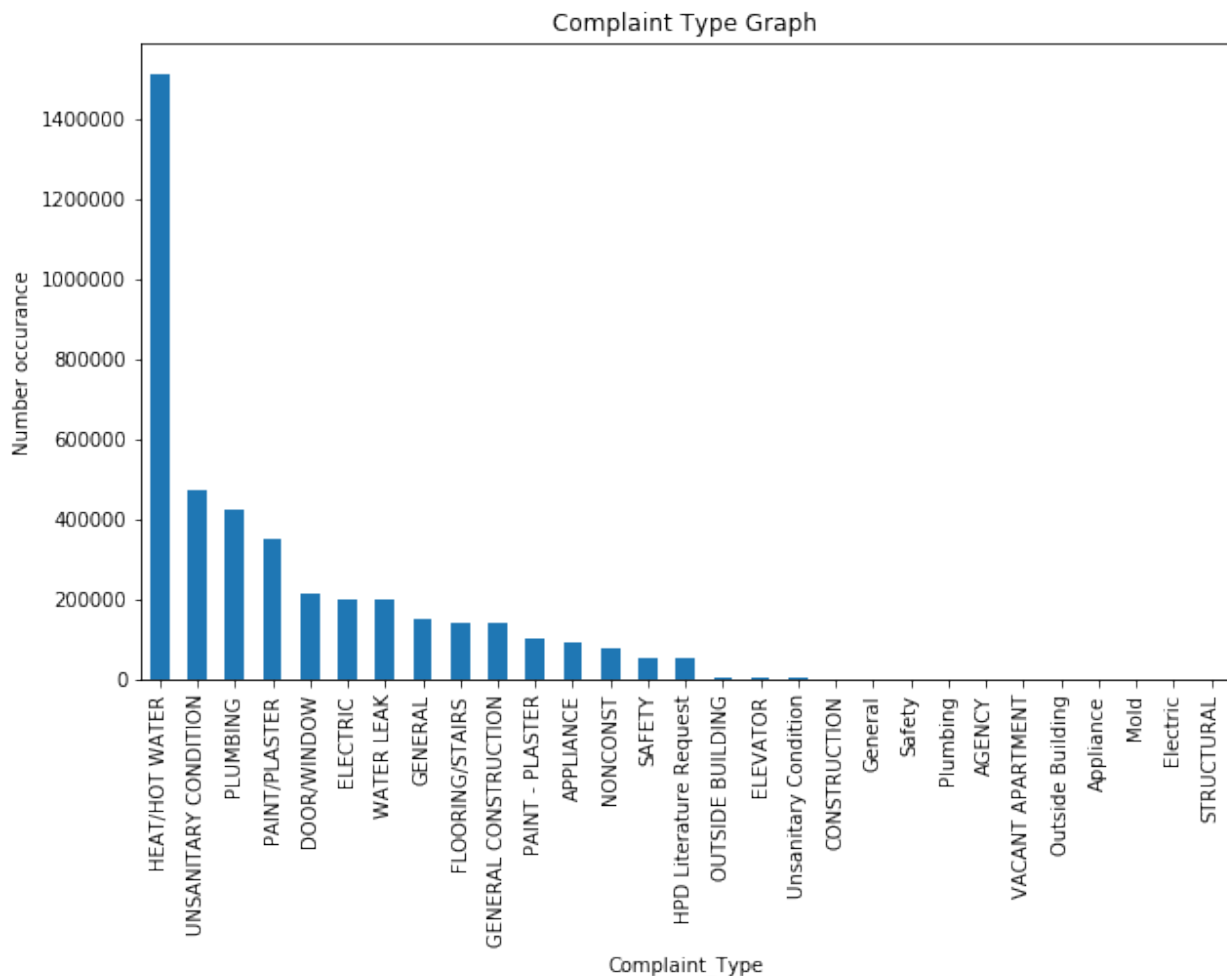
df["complaint_type"].value_counts().head()
```

HEAT/HOT WATER	1512251
UNSANITARY CONDITION	474493
PLUMBING	425976

```
PAINT/PLASTER          353335
DOOR/WINDOW            213033
Name: complaint_type, dtype: int64
```

```
import matplotlib.pyplot as plt
import seaborn as sns

df["complaint_type"].value_counts().plot(kind='bar', figsize=(10, 6))
#
plt.xlabel('Complaint_Type') # add to x-label to the plot
plt.ylabel('Number occurrence') # add y-label to the plot
plt.title('Complaint Type Graph') # add title to the plot
#
plt.show()
```



```
df["created_date"].tail()

4220042    2010-11-28T00:00:00.000
4220043    2010-11-28T00:00:00.000
4220044    2010-11-28T00:00:00.000
```

```
4220045    2010-11-28T00:00:00.000
4220046    2010-11-28T00:00:00.000
Name: created_date, dtype: object
```

it is not a datetime type column. Basically it contains dates as strings.

```
complaint_df = df[df.complaint_type == "HEAT/HOT WATER"]
complaint_df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1512251 entries, 0 to 4220046
Data columns (total 15 columns):
created_date            1512251 non-null object
unique_key              1512251 non-null int64
complaint_type          1512251 non-null object
incident_zip            1493218 non-null float64
incident_address        1512250 non-null object
street_name             1512250 non-null object
address_type            1494255 non-null object
city                    1493363 non-null object
resolution_description  1512086 non-null object
borough                 1512251 non-null object
latitude                1248318 non-null float64
longitude               1248318 non-null float64
closed_date             1507029 non-null object
location_type           1512251 non-null object
status                  1512251 non-null object
dtypes: float64(3), int64(1), object(11)
memory usage: 184.6+ MB

heat_year_sum_df = complaint_df[["complaint_type", "created_date"]]
heat_year_sum_df.head()

   complaint_type  created_date
0  HEAT/HOT WATER  2020-08-23T09:51:03.000
8  HEAT/HOT WATER  2020-08-23T10:09:29.000
11 HEAT/HOT WATER  2020-08-23T19:42:17.000
18 HEAT/HOT WATER  2020-08-23T07:30:14.000
20 HEAT/HOT WATER  2020-08-23T11:53:49.000

heat_year_sum_df["created_date"].head()

0      2020-08-23T09:51:03.000
8      2020-08-23T10:09:29.000
11     2020-08-23T19:42:17.000
18     2020-08-23T07:30:14.000
20     2020-08-23T11:53:49.000
Name: created_date, dtype: object
```

```
heat_year_sum_df["created_date"] =  
pd.to_datetime(heat_year_sum_df.created_date)
```

```
C:\Users\PRADYUM\Anaconda3\lib\site-packages\ipykernel_launcher.py:1:  
SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
```

```
"""Entry point for launching an IPython kernel.
```

```
#now we group the incidents by every year
```

```
grp = heat_year_sum_df.groupby(heat_year_sum_df.created_date.dt.year)
```

```
grp.count()
```

	complaint_type	created_date
created_date		
2010	161611	161611
2011	6085	6085
2012	11971	11971
2013	21720	21720
2014	135043	135043
2015	223011	223011
2016	225267	225267
2017	213244	213244
2018	221035	221035
2019	208128	208128
2020	85136	85136

```
pd.DataFrame = grp.count()
```

```
grp_df = pd.DataFrame
```

```
grp_df
```

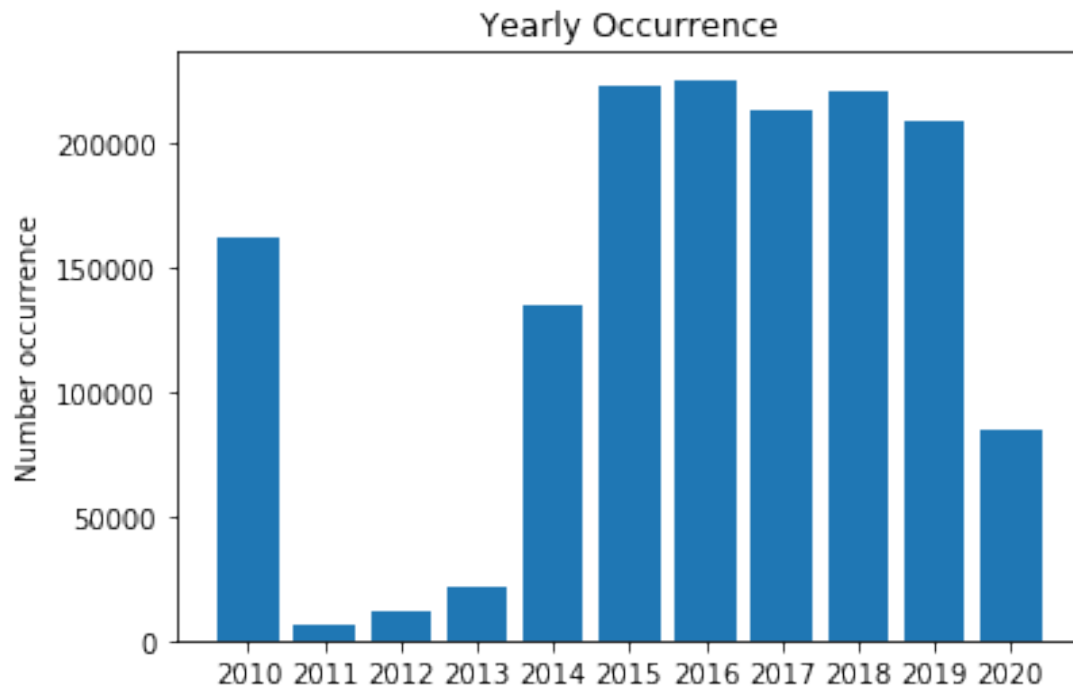
	complaint_type	created_date
created_date		
2010	161611	161611
2011	6085	6085
2012	11971	11971
2013	21720	21720
2014	135043	135043
2015	223011	223011
2016	225267	225267
2017	213244	213244
2018	221035	221035
2019	208128	208128
2020	85136	85136

```
plt.bar(grp_df.index, grp_df["complaint_type"])
```

```
plt.xticks(grp_df.index.values)
```



```
plt.ylabel('Number occurrence')  
plt.title('Yearly Occurrence')  
  
plt.show()
```



CONCLUDING REMARKS: Department of Housing Preservation and Development of New York City should address HEAT/HOT WATER problem first.