IBM Data Science Capstone Project

Yaroslav Aulin

Problem Statement

The people of New York use the 311 system to report complaints about the non-emergency problems to local authorities. Various agencies in New York are assigned these problems. The Department of Housing Preservation and Development of New York City is the agency that processes 311 complaints that are related to housing and buildings.

In the last few years, the number of 311 complaints coming to the Department of Housing Preservation and Development has increased significantly. Although these complaints are not necessarily urgent, the large volume of complaints and the sudden increase is impacting the overall efficiency of operations of the agency.

Therefore, the Department of Housing Preservation and Development has approached your organization to help them manage the large volume of 311 complaints they are receiving every year.

The agency needs answers to several questions. The answers to those questions must be supported by data and analytics. These are their questions:

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

Should the Department of Housing Preservation and Development of New York City focus on any particular set of boroughs, ZIP codes, or street (where the complaints are severe) for the specific type of complaints you identified in response to Question 1?

Does the Complaint Type that you identified in response to question 1 have an obvious relationship with any particular characteristic or characteristics of the houses or buildings?

Can a predictive model be built for a future prediction of the possibility of complaints of the type that you have identified in response to question 1?

Your organization has assigned you as the lead data scientist to provide the answers to these questions. You need to work on getting answers to them in this Capstone Project by following the standard approach of data science and machine learning.

In [1]:

```
# import pandas library
import pandas as pd
```

In [2]:

```
# Read the NYC 311 Dataset

#https://data.cityofnewyork.us/resource/fhrw-4uyv.csv?$limit=100000000&Agency=HPD&$select=created_date,unique_key,complaint_type,incident_zip,incident_address,street_name,address_type,city,resolution_description,borough,latitude,longitude,closed_date,location_type,status
#https://cocl.us/311_NYC_Dataset


#path = "https://s3-api.us-geo.objectstorage.softlayer.net/cf-courses-data/CognitiveClass/DA0101EN/auto.csv"

path_NYC311="https://cocl.us/311_NYC_Dataset?$limit=100000000&Agency=HPD&$select=created_date,unique_key,complaint_type,incident_zip,incident_address,street_name,address_type,city,resolution_description,borough,latitude,longitude,closed_date,location_type,status"


df_NYC311 = pd.read_csv(path_NYC311,parse_dates=True)
```

In [5]:

```
# Create Credential and Bucket Variables

import types
import pandas as pd
from botocore.client import Config
import ibm_boto3


# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
client_cred = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='F2S4IZJuCCwvKymK8226lFl1UP5QUCDGEDy0AWmb_Bn7',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3-api.us-geo.objectstorage.service.networklayer.com')



bucket = 'ibmdatasciencecapstoneproject-donotdelete-pr-pvvc664o5el5ua'
```

In [6]:

```
# create a pickle file

df_NYC311.to_pickle('./df_NYC311_raw.pkl')

# upload a pickle file to Cloud Object Store

client_cred.upload_file('./df_NYC311_raw.pkl',bucket,'df_NYC311_raw_cos.pkl')
```

In [31]:

```
# download a pickle file from Cloud Object Store


client_cred.download_file(Bucket=bucket,Key='df_NYC311_raw_cos.pkl',Filename='./
df_NYC311_raw_local.pkl')


# create a dataframe out of pickle file


df_NYC311 = pd.read_pickle('./df_NYC311_raw_local.pkl')
```

In [9]:

```
# get the NYC PLUTO Dataset
!wget https://www1.nyc.gov/assets/planning/download/zip/data-maps/open-data/nyc_
pluto_18v1.zip
```

```
--2020-07-28 05:49:44--  https://www1.nyc.gov/assets/planning/downlo
ad/zip/data-maps/open-data/nyc_pluto_18v1.zip
Resolving www1.nyc.gov (www1.nyc.gov)... 104.94.79.44, 2600:1404:680
0:3a1::1500, 2600:1404:6800:3aa::1500
Connecting to www1.nyc.gov (www1.nyc.gov)|104.94.79.44|:443... conne
cted.
HTTP request sent, awaiting response... 200 OK
Length: 48263311 (46M) [application/zip]
Saving to: ‘nyc_pluto_18v1.zip’

100%[=======================================>] 48,263,311  15.8MB/s
in 2.9s

2020-07-28 05:49:48 (15.8 MB/s) - ‘nyc_pluto_18v1.zip’ saved [482633
11/48263311]
```

In [10]:

```
!ls
```

```
df_NYC311_raw_local.pkl  df_NYC311_raw.pkl  nyc_pluto_18v1.zip
```

In [11]:

```
# extract the file from archive
!unzip nyc_pluto_18v1.zip
```

```
Archive:  nyc_pluto_18v1.zip
  inflating: PLUTO_for_WEB/BK_18v1.csv
  inflating: PLUTO_for_WEB/BX_18v1.csv
  inflating: PLUTO_for_WEB/MN_18v1.csv
  inflating: PLUTO_for_WEB/PLUTODD18v1.pdf
  inflating: PLUTO_for_WEB/PlutoReadme18v1.pdf
  inflating: PLUTO_for_WEB/QN_18v1.csv
  inflating: PLUTO_for_WEB/SI_18v1.csv
```

In [12]:

```
!ls
```

```
df_NYC311_raw_local.pkl  df_NYC311_raw.pkl  nyc_pluto_18v1.zip  PLUT
O_for_WEB
```

In [218]:

```
# read .csv files to dataframes

df_BK = pd.read_csv('PLUTO_for_WEB/BK_18v1.csv',parse_dates=True)  # Brooklyn
df_BX = pd.read_csv('PLUTO_for_WEB/BX_18v1.csv',parse_dates=True)  # Bronx
df_MN = pd.read_csv('PLUTO_for_WEB/MN_18v1.csv',parse_dates=True)  # Manhattan
df_QN = pd.read_csv('PLUTO_for_WEB/QN_18v1.csv',parse_dates=True)  # Queens
df_SI = pd.read_csv('PLUTO_for_WEB/SI_18v1.csv',parse_dates=True)  # Staten Isla
nd
```

```
/opt/conda/envs/Python36/lib/python3.6/site-packages/IPython/core/in
teractiveshell.py:3020: DtypeWarning: Columns (19,20,22,23,24,25,26,
64,65,80) have mixed types. Specify dtype option on import or set lo
w_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
/opt/conda/envs/Python36/lib/python3.6/site-packages/IPython/core/in
teractiveshell.py:3020: DtypeWarning: Columns (19,20,22,23,64,65,80)
have mixed types. Specify dtype option on import or set low_memory=F
alse.
  interactivity=interactivity, compiler=compiler, result=result)
/opt/conda/envs/Python36/lib/python3.6/site-packages/IPython/core/in
teractiveshell.py:3020: DtypeWarning: Columns (20,23,24,26,64) have
mixed types. Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
/opt/conda/envs/Python36/lib/python3.6/site-packages/IPython/core/in
teractiveshell.py:3020: DtypeWarning: Columns (19,20,22,23,64,65,77)
have mixed types. Specify dtype option on import or set low_memory=F
alse.
  interactivity=interactivity, compiler=compiler, result=result)
/opt/conda/envs/Python36/lib/python3.6/site-packages/IPython/core/in
teractiveshell.py:3020: DtypeWarning: Columns (19,20,22,23,24,64,65,
80) have mixed types. Specify dtype option on import or set low_memo
ry=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

In [15]:

```
# create pickle files

df_BK.to_pickle('./df_BK_raw.pkl')
df_BX.to_pickle('./df_BX_raw.pkl')
df_MN.to_pickle('./df_MN_raw.pkl')
df_QN.to_pickle('./df_QN_raw.pkl')
df_SI.to_pickle('./df_SI_raw.pkl')
```

In [16]:

```
# upload pickle files to Cloud Object Store

client_cred.upload_file('./df_BK_raw.pkl',bucket,'df_BK_raw_cos.pkl')
client_cred.upload_file('./df_BX_raw.pkl',bucket,'df_BX_raw_cos.pkl')
client_cred.upload_file('./df_MN_raw.pkl',bucket,'df_MN_raw_cos.pkl')
client_cred.upload_file('./df_QN_raw.pkl',bucket,'df_QN_raw_cos.pkl')
client_cred.upload_file('./df_SI_raw.pkl',bucket,'df_SI_raw_cos.pkl')
```

Exploring NYC 311 dataset

In [19]:

```
df_NYC311.head()
```

Out[19]:

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip	Incic Addr
0	0	45531130	02/02/2020 06:09:17 AM	NaN	HEAT/HOT WATER	RESIDENTIAL BUILDING	10019.0	426 WI 52 STRI
1	1	45529784	02/02/2020 02:15:24 PM	NaN	UNSANITARY CONDITION	RESIDENTIAL BUILDING	11204.0	175: STRI
2	2	45527528	02/02/2020 02:27:41 AM	NaN	HEAT/HOT WATER	RESIDENTIAL BUILDING	11372.0	87-1! AVEI
3	3	45530329	02/02/2020 12:13:18 PM	NaN	HEAT/HOT WATER	RESIDENTIAL BUILDING	10458.0	2 SOUTHE BOULEV/
4	4	45528814	02/02/2020 01:59:44 PM	NaN	APPLIANCE	RESIDENTIAL BUILDING	11209.0	22: STRI

In [41]:

```
df_NYC311.tail()
```

Out[41]:

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip	
6019838	6019838	44063685	2019-10-15 10:35:45	2019-10-16 17:55:02	HEAT/HOT WATER	RESIDENTIAL BUILDING	10029.0	
6019839	6019839	44063692	2019-10-15 06:09:28	2019-10-15 16:12:58	HEAT/HOT WATER	RESIDENTIAL BUILDING	10461.0	E
6019840	6019840	44063724	2019-10-15 10:54:45	2019-10-16 18:14:29	HEAT/HOT WATER	RESIDENTIAL BUILDING	10034.0	
6019841	6019841	44063726	2019-10-15 14:07:34	2019-10-16 19:10:19	HEAT/HOT WATER	RESIDENTIAL BUILDING	10467.0	
6019842	6019842	44063737	2019-10-15 11:29:01	2019-10-15 21:19:45	PAINT/PLASTER	RESIDENTIAL BUILDING	10009.0	1

In [27]:

```
df_NYC311.dtypes
```

Out[27]:

```
Unnamed: 0                  int64
Unique Key                  int64
Created Date               object
Closed Date                object
Complaint Type             object
Location Type              object
Incident Zip              float64
Incident Address           object
Street Name                object
Address Type               object
City                       object
Status                     object
Resolution Description     object
Borough                    object
Latitude                  float64
Longitude                 float64
dtype: object
```

In [34]:

```
# convert date columns from object to datetime

df_NYC311['Created Date']=pd.to_datetime(df_NYC311['Created Date'],format="%m/%d/%Y  %I:%M:%S %p")
df_NYC311['Closed Date']=pd.to_datetime(df_NYC311['Closed Date'],format="%m/%d/%Y  %I:%M:%S %p")
```

In [35]:

```
df_NYC311.dtypes
```

Out[35]:

```
Unnamed: 0                         int64
Unique Key                         int64
Created Date              datetime64[ns]
Closed Date               datetime64[ns]
Complaint Type                    object
Location Type                     object
Incident Zip                     float64
Incident Address                  object
Street Name                       object
Address Type                      object
City                              object
Status                            object
Resolution Description            object
Borough                           object
Latitude                         float64
Longitude                        float64
dtype: object
```

The dataset contains complaints logged since what date?

In [36]:

```
df_NYC311['Created Date'].min()
```

Out[36]:

```
Timestamp('2010-01-01 00:00:00')
```

The dataset contains complaints logged till what date?

In [37]:

```
df_NYC311['Created Date'].max()
```

Out[37]:

```
Timestamp('2020-02-02 23:58:57')
```

How many incidents have a missing Incident Address?

In [40]:

```
df_NYC311['Incident Address'].isnull().sum()
```

Out[40]:

52825

Exploring PLUTO dataset

How many valid ZIP Codes exist in the Bronx PLUTO dataset?

Exploring Bronx dataset

In [42]:

```
df_BX.head()
```

Out[42]:

	Borough	Block	Lot	CD	CT2010	CB2010	SchoolDist	Council	ZipCode	FireComp	...
0	BX	2260	1	201	19.0	1022.0	7.0	8.0	10454.0	L029	...
1	BX	2260	4	201	19.0	1022.0	7.0	8.0	10454.0	L029	...
2	BX	2260	10	201	19.0	1022.0	7.0	8.0	10454.0	L029	...
3	BX	2260	17	201	19.0	1022.0	7.0	8.0	10454.0	L029	...
4	BX	2260	18	201	19.0	1022.0	7.0	8.0	10454.0	L029	...

5 rows × 87 columns

In [59]:

```
import numpy as np

zipcodes_BX=df_BX['ZipCode'].unique()
zipcodes_BX = zipcodes_BX[~np.isnan(zipcodes_BX)] #exclude NaN value
zipcodes_BX
```

Out[59]:

```
array([10454., 10455., 10451., 10456., 10452., 10453., 10465., 1047
4.,
       11370., 10459., 10472., 10457., 10460., 10458., 10468., 1046
3.,
       10467., 10470., 10466., 10473., 10462., 10461., 10469., 1047
5.,
       10464., 10471.])
```

In [60]:

```
len(zipcodes_BX)
```

Out[60]:

```
26
```

How many valid ZIP Codes exist in the Queens PLUTO dataset?

In [62]:

```
zipcodes_QN=df_QN['ZipCode'].unique()
zipcodes_QN = zipcodes_QN[~np.isnan(zipcodes_QN)] #exclude NaN value
zipcodes_QN
```

Out[62]:

```
array([11101., 11109., 11104., 11377., 11106., 11102., 11103., 1110
5.,
       11370., 11369., 11372., 11373., 11385., 11368., 11421., 1135
5.,
       11374., 11375., 11367., 11415., 11378., 11379., 11418., 1143
2.,
       11356., 11420., 11357., 11354., 11697., 11693., 11358., 1136
1.,
       11365., 11364., 11360., 11359., 11435., 11366., 11423., 1136
3.,
       11362., 11427., 11426., 11428., 11004., 11005., 11040., 1100
1.,
       11416., 11417., 11419., 11433., 11413., 11434., 11412., 1142
9.,
       11411., 11414., 11430., 11436., 11422., 11691., 11692., 1169
4.,
       11695.])
```

In [63]:

```
len(zipcodes_QN)
```

Out[63]:

```
65
```

Exploring NYC 311 dataset

What is the total number of complaints that exist in the dataset?

In [77]:

```
len(df_NYC311)
```

Out[77]:

```
6019843
```

How many differnt Complaint Types can you find in the dataset, including duplicates entries of the same type?

In [78]:

```
df_NYC311['Complaint Type'].unique()
```

Out[78]:

```
array(['HEAT/HOT WATER', 'UNSANITARY CONDITION', 'APPLIANCE', 'GENER
AL',
       'ELECTRIC', 'PLUMBING', 'PAINT/PLASTER', 'WATER LEAK', 'SAFET
Y',
       'DOOR/WINDOW', 'OUTSIDE BUILDING', 'ELEVATOR', 'FLOORING/STAI
RS',
       'Unsanitary Condition', 'HPD Literature Request', 'HEATING',
       'PAINT - PLASTER', 'Safety', 'Electric', 'General', 'Applianc
e',
       'GENERAL CONSTRUCTION', 'NONCONST', 'CONSTRUCTION', 'AGENCY',
       'STRUCTURAL', 'VACANT APARTMENT', 'Outside Building', 'Plumbi
ng',
       'Mold'], dtype=object)
```

In [79]:

```
len(df_NYC311['Complaint Type'].unique())
```

Out[79]:

```
30
```

How many Elevator complaints can you find in the dataset? How many Electric complaints can you find in the dataset?

In [87]:

```
df_NYC311.groupby(by='Complaint Type').count()
```

Out[87]:

Complaint Type	Unnamed: 0	Unique Key	Created Date	Closed Date	Location Type	Incident Zip	Incident Address	S N
AGENCY	9	9	9	9	9	8	9	
APPLIANCE	112831	112831	112831	109163	112831	112677	112831	11
Appliance	4	4	4	2	4	4	4	
CONSTRUCTION	5078	5078	5078	4821	5078	5044	5078	
DOOR/WINDOW	205278	205278	205278	204059	205278	205133	205278	20
ELECTRIC	307310	307310	307310	298849	307310	306447	307310	30
ELEVATOR	6725	6725	6725	6631	6725	6720	6725	
Electric	1	1	1	1	1	1	1	
FLOORING/STAIRS	137402	137402	137402	136595	137402	137313	137402	13
GENERAL	151308	151308	151308	150047	151308	151176	151308	15
GENERAL CONSTRUCTION	500863	500863	500863	473118	500863	498752	500863	50
General	1163	1163	1163	1067	1163	1157	1163	
HEAT/HOT WATER	1261574	1261574	1261574	1256876	1261574	1254458	1261574	126
HEATING	887850	887850	887850	876553	887850	875942	887849	88
HPD Literature Request	52824	52824	52824	51745	0	0	0	
Mold	1	1	1	0	1	1	1	
NONCONST	260890	260890	260890	246901	260890	259999	260890	26
OUTSIDE BUILDING	7142	7142	7142	7086	7142	7133	7142	
Outside Building	6	6	6	6	6	6	6	
PAINT - PLASTER	361257	361257	361257	340937	361257	359741	361257	36
PAINT/PLASTER	346438	346438	346438	344810	346438	346166	346438	34
PLUMBING	711130	711130	711130	686531	711130	709126	711130	71
Plumbing	11	11	11	11	11	11	11	
SAFETY	51529	51529	51529	51261	51529	51495	51529	5
STRUCTURAL	16	16	16	16	16	16	16	
Safety	424	424	424	405	424	424	424	
UNSANITARY CONDITION	451643	451643	451643	448143	451643	451236	451643	45
Unsanitary Condition	5499	5499	5499	5253	5499	5486	5499	
VACANT APARTMENT	6	6	6	1	6	6	6	
WATER LEAK	193631	193631	193631	192289	193631	193468	193631	19

rename values

In [105]:

```
df_NYC311['Complaint Type']=df_NYC311['Complaint Type'].replace(['General'],'GEN
ERAL')
df_NYC311['Complaint Type']=df_NYC311['Complaint Type'].replace(['Appliance'],'A
PPLIANCE')
df_NYC311['Complaint Type']=df_NYC311['Complaint Type'].replace(['Electric'],'EL
ECTRIC')
df_NYC311['Complaint Type']=df_NYC311['Complaint Type'].replace(['Outside Buildi
ng'],'OUTSIDE BUILDING')
df_NYC311['Complaint Type']=df_NYC311['Complaint Type'].replace(['PAINT - PLASTE
R'],'PAINT/PLASTER')
df_NYC311['Complaint Type']=df_NYC311['Complaint Type'].replace(['Plumbing'],'PL
UMBING')
df_NYC311['Complaint Type']=df_NYC311['Complaint Type'].replace(['Safety'],'SAFE
TY')
df_NYC311['Complaint Type']=df_NYC311['Complaint Type'].replace(['Unsanitary Con
dition'],'UNSANITARY CONDITION')
```

Using 80,000 as a threshold, what complaint type(s) do you recommend the Department of Housing Preservation and Development of New York City address first? Select all that apply.

In [134]:

```
grouped=df_NYC311.groupby(by='Complaint Type').count().loc[:, ['Unique Key']]
grouped.rename(columns={'Unique Key':'Cases'}, inplace=True)
grouped
```

Out[134]:

Complaint Type	Cases
AGENCY	9
APPLIANCE	112835
CONSTRUCTION	5078
DOOR/WINDOW	205278
ELECTRIC	307311
ELEVATOR	6725
FLOORING/STAIRS	137402
GENERAL	152471
GENERAL CONSTRUCTION	500863
HEAT/HOT WATER	1261574
HEATING	887850
HPD Literature Request	52824
Mold	1
NONCONST	260890
OUTSIDE BUILDING	7148
PAINT/PLASTER	707695
PLUMBING	711141
SAFETY	51953
STRUCTURAL	16
UNSANITARY CONDITION	457142
VACANT APARTMENT	6
WATER LEAK	193631

In [119]:

```
top_complaints=grouped[grouped>80000].dropna()
top_complaints
```

Out[119]:

Complaint Type	Cases
APPLIANCE	112835.0
DOOR/WINDOW	205278.0
ELECTRIC	307311.0
FLOORING/STAIRS	137402.0
GENERAL	152471.0
GENERAL CONSTRUCTION	500863.0
HEAT/HOT WATER	1261574.0
HEATING	887850.0
NONCONST	260890.0
PAINT/PLASTER	707695.0
PLUMBING	711141.0
UNSANITARY CONDITION	457142.0
WATER LEAK	193631.0

In [135]:

```
# sort by number of complaints

top_complaints.sort_values(by=['Cases'],ascending = False)
```

Out[135]:

Complaint Type	Cases
HEAT/HOT WATER	1261574.0
HEATING	887850.0
PLUMBING	711141.0
PAINT/PLASTER	707695.0
GENERAL CONSTRUCTION	500863.0
UNSANITARY CONDITION	457142.0
ELECTRIC	307311.0
NONCONST	260890.0
DOOR/WINDOW	205278.0
WATER LEAK	193631.0
GENERAL	152471.0
FLOORING/STAIRS	137402.0
APPLIANCE	112835.0

For the complaint types that you selected in the previous module that had at least 80,000 complaints logged, which borough had the highest number of complaints submitted?

In [136]:

```
tc_array=top_complaints.index.values
tc_array
```

Out[136]:

```
array(['APPLIANCE', 'DOOR/WINDOW', 'ELECTRIC', 'FLOORING/STAIRS',
       'GENERAL', 'GENERAL CONSTRUCTION', 'HEAT/HOT WATER', 'HEATIN
G',
       'NONCONST', 'PAINT/PLASTER', 'PLUMBING', 'UNSANITARY CONDITIO
N',
       'WATER LEAK'], dtype=object)
```

In [137]:

```
df_NYC311
```

Out[137]:

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip
0	0	45531130	2020-02-02 06:09:17	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10019.0
1	1	45529784	2020-02-02 14:15:24	NaT	UNSANITARY CONDITION	RESIDENTIAL BUILDING	11204.0
2	2	45527528	2020-02-02 02:27:41	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11372.0
3	3	45530329	2020-02-02 12:13:18	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10458.0
4	4	45528814	2020-02-02 13:59:44	NaT	APPLIANCE	RESIDENTIAL BUILDING	11209.0
5	5	45530344	2020-02-02 07:52:08	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10456.0
6	6	45530969	2020-02-02 23:13:42	NaT	GENERAL	RESIDENTIAL BUILDING	11226.0
7	7	45527939	2020-02-02 09:00:54	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11372.0
8	8	45530238	2020-02-02 11:38:34	NaT	ELECTRIC	RESIDENTIAL BUILDING	11230.0
9	9	45529416	2020-02-02 10:59:58	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11207.0
10	10	45530349	2020-02-02 05:38:39	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10459.0

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip
11	11	45530831	2020-02-02 10:13:57	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10039.0
12	12	45529310	2020-02-02 09:12:29	NaT	PLUMBING	RESIDENTIAL BUILDING	11433.0
13	13	45528385	2020-02-02 13:53:34	NaT	UNSANITARY CONDITION	RESIDENTIAL BUILDING	11238.0
14	14	45529877	2020-02-02 16:09:57	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11226.0
15	15	45530802	2020-02-02 15:50:43	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10011.0
16	16	45528878	2020-02-02 04:11:19	NaT	UNSANITARY CONDITION	RESIDENTIAL BUILDING	10456.0
17	17	45528911	2020-02-02 04:32:03	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10026.0
18	18	45528877	2020-02-02 11:10:08	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10031.0
19	19	45529362	2020-02-02 13:38:25	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10473.0
20	20	45529853	2020-02-02 10:35:22	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11204.0
21	21	45528464	2020-02-02 23:46:12	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11210.0
22	22	45530266	2020-02-02 16:32:27	NaT	UNSANITARY CONDITION	RESIDENTIAL BUILDING	10457.0

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip
23	23	45529440	2020-02-02 12:43:22	NaT	PAINT/PLASTER	RESIDENTIAL BUILDING	11216.0
24	24	45528481	2020-02-02 11:13:18	NaT	PLUMBING	RESIDENTIAL BUILDING	10014.0
25	25	45532031	2020-02-02 11:19:40	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10034.0
26	26	45529887	2020-02-02 09:43:11	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11238.0
27	27	45530787	2020-02-02 19:20:04	NaT	WATER LEAK	RESIDENTIAL BUILDING	10458.0
28	28	45529874	2020-02-02 21:26:43	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11203.0
29	29	45528454	2020-02-02 22:25:32	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10463.0
...	...	...	...	...	...	...	...
6019813	6019813	44063460	2019-10-15 06:04:28	2019-10-16 19:12:28	HEAT/HOT WATER	RESIDENTIAL BUILDING	11213.0
6019814	6019814	44063463	2019-10-15 23:02:37	2019-10-16 19:09:53	HEAT/HOT WATER	RESIDENTIAL BUILDING	11226.0
6019815	6019815	44063467	2019-10-15 08:15:08	2019-10-16 21:55:24	HEAT/HOT WATER	RESIDENTIAL BUILDING	10466.0
6019816	6019816	44063469	2019-10-15 10:47:49	2019-10-16 21:57:50	HEAT/HOT WATER	RESIDENTIAL BUILDING	10023.0

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip	
6019817	6019817	44063470	2019-10-15 07:49:50	2019-10-16 09:15:08	HEAT/HOT WATER	RESIDENTIAL BUILDING	11233.0	
6019818	6019818	44063493	2019-10-15 14:55:34	2019-10-16 16:18:30	HEAT/HOT WATER	RESIDENTIAL BUILDING	11423.0	1
6019819	6019819	44063499	2019-10-15 16:57:42	2019-10-16 14:04:54	HEAT/HOT WATER	RESIDENTIAL BUILDING	10467.0	I
6019820	6019820	44063561	2019-10-15 07:24:44	2019-10-16 21:29:49	HEAT/HOT WATER	RESIDENTIAL BUILDING	11414.0	1
6019821	6019821	44063574	2019-10-15 06:23:40	2019-10-16 21:55:24	HEAT/HOT WATER	RESIDENTIAL BUILDING	10466.0	
6019822	6019822	44063580	2019-10-15 11:15:53	2019-10-16 18:50:00	HEAT/HOT WATER	RESIDENTIAL BUILDING	10303.0	
6019823	6019823	44063592	2019-10-15 06:45:03	2019-10-15 16:05:35	HEAT/HOT WATER	RESIDENTIAL BUILDING	10029.0	
6019824	6019824	44063603	2019-10-15 07:40:34	2019-10-16 02:07:17	HEAT/HOT WATER	RESIDENTIAL BUILDING	10474.0	8
6019825	6019825	44063604	2019-10-15 08:44:42	2019-10-16 20:25:11	HEAT/HOT WATER	RESIDENTIAL BUILDING	11204.0	
6019826	6019826	44063628	2019-10-15 18:18:33	2019-10-16 10:44:28	HEAT/HOT WATER	RESIDENTIAL BUILDING	10024.0	
6019827	6019827	44063632	2019-10-15 19:27:28	2019-10-16 17:01:05	HEAT/HOT WATER	RESIDENTIAL BUILDING	10472.0	(
6019828	6019828	44063636	2019-10-15 08:03:00	2019-10-16 17:10:34	HEAT/HOT WATER	RESIDENTIAL BUILDING	10036.0	

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip
6019829	6019829	44063643	2019-10-15 18:56:11	2019-10-16 19:19:47	HEAT/HOT WATER	RESIDENTIAL BUILDING	11234.0
6019830	6019830	44063644	2019-10-15 22:27:15	2019-10-16 10:44:29	HEAT/HOT WATER	RESIDENTIAL BUILDING	10470.0
6019831	6019831	44063645	2019-10-15 07:50:19	2019-10-16 17:14:33	HEAT/HOT WATER	RESIDENTIAL BUILDING	10458.0
6019832	6019832	44063654	2019-10-15 12:04:24	NaT	PAINT/PLASTER	RESIDENTIAL BUILDING	11102.0
6019833	6019833	44063661	2019-10-15 22:13:11	2019-10-16 02:07:21	HEAT/HOT WATER	RESIDENTIAL BUILDING	10452.0
6019834	6019834	44063670	2019-10-15 08:25:25	2019-10-16 21:55:24	HEAT/HOT WATER	RESIDENTIAL BUILDING	10470.0
6019835	6019835	44063671	2019-10-15 09:32:54	2019-10-16 19:02:48	HEAT/HOT WATER	RESIDENTIAL BUILDING	10458.0
6019836	6019836	44063680	2019-10-15 18:43:56	2019-10-16 17:55:02	HEAT/HOT WATER	RESIDENTIAL BUILDING	10029.0
6019837	6019837	44063681	2019-10-15 11:35:19	NaT	UNSANITARY CONDITION	RESIDENTIAL BUILDING	10472.0
6019838	6019838	44063685	2019-10-15 10:35:45	2019-10-16 17:55:02	HEAT/HOT WATER	RESIDENTIAL BUILDING	10029.0
6019839	6019839	44063692	2019-10-15 06:09:28	2019-10-15 16:12:58	HEAT/HOT WATER	RESIDENTIAL BUILDING	10461.0
6019840	6019840	44063724	2019-10-15 10:54:45	2019-10-16 18:14:29	HEAT/HOT WATER	RESIDENTIAL BUILDING	10034.0

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip
6019841	6019841	44063726	2019-10-15 14:07:34	2019-10-16 19:10:19	HEAT/HOT WATER	RESIDENTIAL BUILDING	10467.0
6019842	6019842	44063737	2019-10-15 11:29:01	2019-10-15 21:19:45	PAINT/PLASTER	RESIDENTIAL BUILDING	10009.0

6019843 rows × 16 columns

In [152]:

```
df_tclist=df_NYC311[['Complaint Type','Borough']]
df_tclist
```

Out[152]:

	Complaint Type	Borough
0	HEAT/HOT WATER	MANHATTAN
1	UNSANITARY CONDITION	BROOKLYN
2	HEAT/HOT WATER	QUEENS
3	HEAT/HOT WATER	BRONX
4	APPLIANCE	BROOKLYN
5	HEAT/HOT WATER	BRONX
6	GENERAL	BROOKLYN
7	HEAT/HOT WATER	QUEENS
8	ELECTRIC	BROOKLYN
9	HEAT/HOT WATER	BROOKLYN
10	HEAT/HOT WATER	BRONX
11	HEAT/HOT WATER	MANHATTAN
12	PLUMBING	QUEENS
13	UNSANITARY CONDITION	BROOKLYN
14	HEAT/HOT WATER	BROOKLYN
15	HEAT/HOT WATER	MANHATTAN
16	UNSANITARY CONDITION	BRONX
17	HEAT/HOT WATER	MANHATTAN
18	HEAT/HOT WATER	MANHATTAN
19	HEAT/HOT WATER	BRONX
20	HEAT/HOT WATER	BROOKLYN
21	HEAT/HOT WATER	BROOKLYN
22	UNSANITARY CONDITION	BRONX
23	PAINT/PLASTER	BROOKLYN
24	PLUMBING	MANHATTAN
25	HEAT/HOT WATER	MANHATTAN
26	HEAT/HOT WATER	BROOKLYN
27	WATER LEAK	BRONX
28	HEAT/HOT WATER	BROOKLYN
29	HEAT/HOT WATER	BRONX
...	...	...
6019813	HEAT/HOT WATER	BROOKLYN
6019814	HEAT/HOT WATER	BROOKLYN
6019815	HEAT/HOT WATER	BRONX
6019816	HEAT/HOT WATER	MANHATTAN
6019817	HEAT/HOT WATER	BROOKLYN
6019818	HEAT/HOT WATER	QUEENS

	Complaint Type	Borough
6019819	HEAT/HOT WATER	BRONX
6019820	HEAT/HOT WATER	QUEENS
6019821	HEAT/HOT WATER	BRONX
6019822	HEAT/HOT WATER	STATEN ISLAND
6019823	HEAT/HOT WATER	MANHATTAN
6019824	HEAT/HOT WATER	BRONX
6019825	HEAT/HOT WATER	BROOKLYN
6019826	HEAT/HOT WATER	MANHATTAN
6019827	HEAT/HOT WATER	BRONX
6019828	HEAT/HOT WATER	MANHATTAN
6019829	HEAT/HOT WATER	BROOKLYN
6019830	HEAT/HOT WATER	BRONX
6019831	HEAT/HOT WATER	BRONX
6019832	PAINT/PLASTER	QUEENS
6019833	HEAT/HOT WATER	BRONX
6019834	HEAT/HOT WATER	BRONX
6019835	HEAT/HOT WATER	BRONX
6019836	HEAT/HOT WATER	MANHATTAN
6019837	UNSANITARY CONDITION	BRONX
6019838	HEAT/HOT WATER	MANHATTAN
6019839	HEAT/HOT WATER	BRONX
6019840	HEAT/HOT WATER	MANHATTAN
6019841	HEAT/HOT WATER	BRONX
6019842	PAINT/PLASTER	MANHATTAN

6019843 rows × 2 columns

In [153]:

```
df_tclist = df_tclist[df_tclist['Complaint Type'].isin(tc_array)]
```

In [154]:

```
df_tclist
```

Out[154]:

	Complaint Type	Borough
0	HEAT/HOT WATER	MANHATTAN
1	UNSANITARY CONDITION	BROOKLYN
2	HEAT/HOT WATER	QUEENS
3	HEAT/HOT WATER	BRONX
4	APPLIANCE	BROOKLYN
5	HEAT/HOT WATER	BRONX
6	GENERAL	BROOKLYN
7	HEAT/HOT WATER	QUEENS
8	ELECTRIC	BROOKLYN
9	HEAT/HOT WATER	BROOKLYN
10	HEAT/HOT WATER	BRONX
11	HEAT/HOT WATER	MANHATTAN
12	PLUMBING	QUEENS
13	UNSANITARY CONDITION	BROOKLYN
14	HEAT/HOT WATER	BROOKLYN
15	HEAT/HOT WATER	MANHATTAN
16	UNSANITARY CONDITION	BRONX
17	HEAT/HOT WATER	MANHATTAN
18	HEAT/HOT WATER	MANHATTAN
19	HEAT/HOT WATER	BRONX
20	HEAT/HOT WATER	BROOKLYN
21	HEAT/HOT WATER	BROOKLYN
22	UNSANITARY CONDITION	BRONX
23	PAINT/PLASTER	BROOKLYN
24	PLUMBING	MANHATTAN
25	HEAT/HOT WATER	MANHATTAN
26	HEAT/HOT WATER	BROOKLYN
27	WATER LEAK	BRONX
28	HEAT/HOT WATER	BROOKLYN
29	HEAT/HOT WATER	BRONX
...	...	...
6019813	HEAT/HOT WATER	BROOKLYN
6019814	HEAT/HOT WATER	BROOKLYN
6019815	HEAT/HOT WATER	BRONX
6019816	HEAT/HOT WATER	MANHATTAN
6019817	HEAT/HOT WATER	BROOKLYN
6019818	HEAT/HOT WATER	QUEENS

	Complaint Type	Borough
6019819	HEAT/HOT WATER	BRONX
6019820	HEAT/HOT WATER	QUEENS
6019821	HEAT/HOT WATER	BRONX
6019822	HEAT/HOT WATER	STATEN ISLAND
6019823	HEAT/HOT WATER	MANHATTAN
6019824	HEAT/HOT WATER	BRONX
6019825	HEAT/HOT WATER	BROOKLYN
6019826	HEAT/HOT WATER	MANHATTAN
6019827	HEAT/HOT WATER	BRONX
6019828	HEAT/HOT WATER	MANHATTAN
6019829	HEAT/HOT WATER	BROOKLYN
6019830	HEAT/HOT WATER	BRONX
6019831	HEAT/HOT WATER	BRONX
6019832	PAINT/PLASTER	QUEENS
6019833	HEAT/HOT WATER	BRONX
6019834	HEAT/HOT WATER	BRONX
6019835	HEAT/HOT WATER	BRONX
6019836	HEAT/HOT WATER	MANHATTAN
6019837	UNSANITARY CONDITION	BRONX
6019838	HEAT/HOT WATER	MANHATTAN
6019839	HEAT/HOT WATER	BRONX
6019840	HEAT/HOT WATER	MANHATTAN
6019841	HEAT/HOT WATER	BRONX
6019842	PAINT/PLASTER	MANHATTAN

5896083 rows × 2 columns

In [160]:

```
grouped1 = df_tclist.groupby(by='Borough').count()
grouped1.rename(columns={'Complaint Type':'Number of Complaints'}, inplace=True)
grouped1.sort_values(by=['Number of Complaints'],ascending = False, inplace=True
)
grouped1
```

Out[160]:

Borough	Number of Complaints
BROOKLYN	1714713
BRONX	1599894
MANHATTAN	1041994
Unspecified	818871
QUEENS	634703
STATEN ISLAND	85908

For the complaint types that you selected in the previous module that had at least 80,000 complaints logged, which ZIP code had the highest number of complainted submitted?

In [161]:

```
df_tclist1=df_NYC311[['Complaint Type','Incident Zip']]
df_tclist1 = df_tclist1[df_tclist1['Complaint Type'].isin(tc_array)]
df_tclist1
```

Out[161]:

	Complaint Type	Incident Zip
0	HEAT/HOT WATER	10019.0
1	UNSANITARY CONDITION	11204.0
2	HEAT/HOT WATER	11372.0
3	HEAT/HOT WATER	10458.0
4	APPLIANCE	11209.0
5	HEAT/HOT WATER	10456.0
6	GENERAL	11226.0
7	HEAT/HOT WATER	11372.0
8	ELECTRIC	11230.0
9	HEAT/HOT WATER	11207.0
10	HEAT/HOT WATER	10459.0
11	HEAT/HOT WATER	10039.0
12	PLUMBING	11433.0
13	UNSANITARY CONDITION	11238.0
14	HEAT/HOT WATER	11226.0
15	HEAT/HOT WATER	10011.0
16	UNSANITARY CONDITION	10456.0
17	HEAT/HOT WATER	10026.0
18	HEAT/HOT WATER	10031.0
19	HEAT/HOT WATER	10473.0
20	HEAT/HOT WATER	11204.0
21	HEAT/HOT WATER	11210.0
22	UNSANITARY CONDITION	10457.0
23	PAINT/PLASTER	11216.0
24	PLUMBING	10014.0
25	HEAT/HOT WATER	10034.0
26	HEAT/HOT WATER	11238.0
27	WATER LEAK	10458.0
28	HEAT/HOT WATER	11203.0
29	HEAT/HOT WATER	10463.0
...	...	...
6019813	HEAT/HOT WATER	11213.0
6019814	HEAT/HOT WATER	11226.0
6019815	HEAT/HOT WATER	10466.0
6019816	HEAT/HOT WATER	10023.0
6019817	HEAT/HOT WATER	11233.0
6019818	HEAT/HOT WATER	11423.0

	Complaint Type	Incident Zip
6019819	HEAT/HOT WATER	10467.0
6019820	HEAT/HOT WATER	11414.0
6019821	HEAT/HOT WATER	10466.0
6019822	HEAT/HOT WATER	10303.0
6019823	HEAT/HOT WATER	10029.0
6019824	HEAT/HOT WATER	10474.0
6019825	HEAT/HOT WATER	11204.0
6019826	HEAT/HOT WATER	10024.0
6019827	HEAT/HOT WATER	10472.0
6019828	HEAT/HOT WATER	10036.0
6019829	HEAT/HOT WATER	11234.0
6019830	HEAT/HOT WATER	10470.0
6019831	HEAT/HOT WATER	10458.0
6019832	PAINT/PLASTER	11102.0
6019833	HEAT/HOT WATER	10452.0
6019834	HEAT/HOT WATER	10470.0
6019835	HEAT/HOT WATER	10458.0
6019836	HEAT/HOT WATER	10029.0
6019837	UNSANITARY CONDITION	10472.0
6019838	HEAT/HOT WATER	10029.0
6019839	HEAT/HOT WATER	10461.0
6019840	HEAT/HOT WATER	10034.0
6019841	HEAT/HOT WATER	10467.0
6019842	PAINT/PLASTER	10009.0

5896083 rows × 2 columns

In [164]:

```
grouped2 = df_tclist1.groupby(by='Incident Zip').count()
grouped2.rename(columns={'Complaint Type':'Number of Complaints'}, inplace=True)
grouped2.sort_values(by=['Number of Complaints'],ascending = False, inplace=True
)
grouped2
```

Out[164]:

Incident Zip	Number of Complaints
11226.0	213855
10467.0	172285
10458.0	168025
10453.0	161053
10468.0	146893
10457.0	144731
10452.0	144729
10456.0	131366
10031.0	122500
11225.0	119250
11213.0	117040
10032.0	109087
11207.0	106760
11212.0	105476
11233.0	104845
11221.0	100305
10460.0	95702
10472.0	92423
10040.0	88223
11203.0	87060
11216.0	87003
11208.0	85326
10033.0	82718
10462.0	78894
10463.0	78051
10034.0	72278
10459.0	71135
10466.0	68546
11230.0	64022
10451.0	62433
...	...
10006.0	435
10005.0	407
10280.0	337
10004.0	325
11001.0	271

Incident Zip	Number of Complaints
11697.0	266
11040.0	111
10069.0	87
11109.0	71
10129.0	40
11005.0	14
10162.0	14
10020.0	5
11430.0	4
10803.0	4
10158.0	4
10112.0	4
10041.0	3
10103.0	3
10169.0	2
11241.0	2
10106.0	1
10045.0	1
11243.0	1
10118.0	1
10121.0	1
10177.0	1
10271.0	1
10279.0	1
12345.0	1

201 rows × 1 columns

For the complaint types that you selected in the previous module that had at least 80,000 complaints logged, the address 89-21 Elmhurst Avenue had the highest number of complainted submitted?

In [167]:

```
df_tclist2=df_NYC311[['Complaint Type','Incident Address']]
df_tclist2 = df_tclist2[df_tclist2['Complaint Type'].isin(tc_array)]


grouped2 = df_tclist2.groupby(by='Incident Address').count()
grouped2.rename(columns={'Complaint Type':'Number of Complaints'}, inplace=True)
grouped2.sort_values(by=['Number of Complaints'],ascending = False, inplace=True
)
grouped2
```

Out[167]:

Incident Address	Number of Complaints
34 ARDEN STREET	14294
89-21 ELMHURST AVENUE	12681
1025 BOYNTON AVENUE	9716
3810 BAILEY AVENUE	7174
9511 SHORE ROAD	5062
2913 FOSTER AVENUE	4885
750 GRAND CONCOURSE	4506
1711 FULTON STREET	4343
888 GRAND CONCOURSE	4295
3555 BRUCKNER BOULEVARD	4071
2090 EAST TREMONT AVENUE	3986
3230 CRUGER AVENUE	3895
72-10 41 AVENUE	3794
1589 OCEAN AVENUE	3303
2856 WEBB AVENUE	3279
2750 HOMECREST AVENUE	3203
2968 PERRY AVENUE	3148
1030 BOYNTON AVENUE	3099
957 WOODYCREST AVENUE	3099
3605 SEDGWICK AVENUE	3070
2074 WALLACE AVENUE	3034
3451 GILES PLACE	2977
1515 SELWYN AVENUE	2940
446 WEST 58 STREET	2922
97 SHERMAN AVENUE	2892
2176 TIEBOUT AVENUE	2860
2181 BARNES AVENUE	2799
2040 BRONXDALE AVENUE	2728
48 JEFFERSON STREET	2701
223 LINDEN BOULEVARD	2647
...	...
188-36 LINDEN BOULEVARD	1
188-35 71 CRESCENT	1
1881 NEW YORK AVENUE	1
535 WEST 160 STREET	1
1885 WEST 9 STREET	1

Incident Address	Number of Complaints
1881 WEST 6 STREET	1
535 HALSEY STREET	1
535 HAVEMEYER AVENUE	1
1885 WEST 12 STREET	1
535 JEROME STREET	1
535 KENT AVENUE	1
535 MAC DONOUGH STREET	1
535 MACON STREET	1
1885 TROUTMAN STREET	1
1884 MC DONALD AVENUE	1
1883 EAST 52 STREET	1
535 NOSTRAND AVENUE	1
535 ODER AVENUE	1
535 PARK AVENUE	1
535 QUINCY STREET	1
1883 58 STREET	1
535 SNEDIKER AVENUE	1
1882 WEST 9 STREET	1
535 TAYLOR AVENUE	1
1882 MADISON STREET	1
535 WEST 111 STREET	1
535 WEST 112 STREET	1
1882 BLEECKER STREET	1
1881 WOODBINE STREET	1
UNKNOWN	1

181486 rows × 1 columns

For the complaint types that you selected in the previous module that had at least 80,000 complaints logged, how many of the submitted tickets were closed?

In [177]:

```
df_tclist3=df_NYC311[['Complaint Type','Closed Date']]
df_tclist3=df_tclist3[df_tclist3['Complaint Type'].isin(tc_array)].dropna()
df_tclist3
```

Out[177]:

	Complaint Type	Closed Date
33	HEAT/HOT WATER	2020-02-02 18:53:37
82	HEAT/HOT WATER	2020-02-02 18:06:59
85	HEAT/HOT WATER	2020-02-02 18:53:37
168	HEAT/HOT WATER	2020-02-02 18:53:37
179	HEAT/HOT WATER	2020-02-02 18:07:00
203	HEAT/HOT WATER	2020-02-02 18:06:59
208	HEAT/HOT WATER	2020-02-02 16:33:48
238	HEAT/HOT WATER	2020-02-02 18:53:37
244	HEAT/HOT WATER	2020-02-02 18:53:37
251	HEAT/HOT WATER	2020-02-02 10:19:04
265	HEAT/HOT WATER	2020-02-02 18:53:37
297	HEAT/HOT WATER	2020-02-02 21:35:59
309	HEAT/HOT WATER	2020-02-02 18:30:07
312	HEAT/HOT WATER	2020-02-02 18:30:07
314	HEAT/HOT WATER	2020-02-02 17:27:33
361	HEAT/HOT WATER	2020-02-02 18:07:00
370	HEAT/HOT WATER	2020-02-02 18:06:59
382	HEAT/HOT WATER	2020-02-02 18:53:37
384	HEAT/HOT WATER	2020-02-02 20:05:34
403	HEAT/HOT WATER	2020-02-02 18:53:37
406	HEAT/HOT WATER	2020-02-02 18:06:59
419	HEAT/HOT WATER	2020-02-02 19:58:59
423	HEAT/HOT WATER	2020-02-02 18:16:04
430	HEAT/HOT WATER	2020-02-02 18:53:37
431	HEAT/HOT WATER	2020-02-02 18:53:37
452	HEAT/HOT WATER	2020-02-02 16:33:49
454	HEAT/HOT WATER	2020-02-02 18:53:37
455	HEAT/HOT WATER	2020-02-02 16:33:31
457	HEAT/HOT WATER	2020-02-02 18:53:37
478	HEAT/HOT WATER	2020-02-02 17:54:04
...	...	...
6019811	HEAT/HOT WATER	2019-10-15 19:01:42
6019812	HEAT/HOT WATER	2019-10-16 08:49:35
6019813	HEAT/HOT WATER	2019-10-16 19:12:28
6019814	HEAT/HOT WATER	2019-10-16 19:09:53
6019815	HEAT/HOT WATER	2019-10-16 21:55:24
6019816	HEAT/HOT WATER	2019-10-16 21:57:50

	Complaint Type	Closed Date
6019817	HEAT/HOT WATER	2019-10-16 09:15:08
6019818	HEAT/HOT WATER	2019-10-16 16:18:30
6019819	HEAT/HOT WATER	2019-10-16 14:04:54
6019820	HEAT/HOT WATER	2019-10-16 21:29:49
6019821	HEAT/HOT WATER	2019-10-16 21:55:24
6019822	HEAT/HOT WATER	2019-10-16 18:50:00
6019823	HEAT/HOT WATER	2019-10-15 16:05:35
6019824	HEAT/HOT WATER	2019-10-16 02:07:17
6019825	HEAT/HOT WATER	2019-10-16 20:25:11
6019826	HEAT/HOT WATER	2019-10-16 10:44:28
6019827	HEAT/HOT WATER	2019-10-16 17:01:05
6019828	HEAT/HOT WATER	2019-10-16 17:10:34
6019829	HEAT/HOT WATER	2019-10-16 19:19:47
6019830	HEAT/HOT WATER	2019-10-16 10:44:29
6019831	HEAT/HOT WATER	2019-10-16 17:14:33
6019833	HEAT/HOT WATER	2019-10-16 02:07:21
6019834	HEAT/HOT WATER	2019-10-16 21:55:24
6019835	HEAT/HOT WATER	2019-10-16 19:02:48
6019836	HEAT/HOT WATER	2019-10-16 17:55:02
6019838	HEAT/HOT WATER	2019-10-16 17:55:02
6019839	HEAT/HOT WATER	2019-10-15 16:12:58
6019840	HEAT/HOT WATER	2019-10-16 18:14:29
6019841	HEAT/HOT WATER	2019-10-16 19:10:19
6019842	PAINT/PLASTER	2019-10-15 21:19:45

5771205 rows × 2 columns

In [180]:

```
df_tclist3.count()
```

Out[180]:

```
Complaint Type    5771205
Closed Date       5771205
dtype: int64
```

Can you determine the age of the building from the PLUTO dataset?

In [185]:

```
df_BK.columns   # YearBuilt column
```

Out[185]:

```
Index(['Borough', 'Block', 'Lot', 'CD', 'CT2010', 'CB2010', 'SchoolD
ist',
       'Council', 'ZipCode', 'FireComp', 'PolicePrct', 'HealthCenter
District',
       'HealthArea', 'SanitBoro', 'SanitDistrict', 'SanitSub', 'Addr
ess',
       'ZoneDist1', 'ZoneDist2', 'ZoneDist3', 'ZoneDist4', 'Overlay
1',
       'Overlay2', 'SPDist1', 'SPDist2', 'SPDist3', 'LtdHeight', 'Sp
litZone',
       'BldgClass', 'LandUse', 'Easements', 'OwnerType', 'OwnerNam
e',
       'LotArea', 'BldgArea', 'ComArea', 'ResArea', 'OfficeArea', 'R
etailArea',
       'GarageArea', 'StrgeArea', 'FactryArea', 'OtherArea', 'AreaSo
urce',
       'NumBldgs', 'NumFloors', 'UnitsRes', 'UnitsTotal', 'LotFron
t',
       'LotDepth', 'BldgFront', 'BldgDepth', 'Ext', 'ProxCode', 'Irr
LotCode',
       'LotType', 'BsmtCode', 'AssessLand', 'AssessTot', 'ExemptLan
d',
       'ExemptTot', 'YearBuilt', 'YearAlter1', 'YearAlter2', 'HistDi
st',
       'Landmark', 'BuiltFAR', 'ResidFAR', 'CommFAR', 'FacilFAR', 'B
oroCode',
       'BBL', 'CondoNo', 'Tract2010', 'XCoord', 'YCoord', 'ZoneMap',
'ZMCode',
       'Sanborn', 'TaxMap', 'EDesigNum', 'APPBBL', 'APPDate', 'PLUTO
MapID',
       'FIRM07_FLAG', 'PFIRM15_FLAG', 'Version'],
      dtype='object')
```

Top complaint type - HEAT/HOT WATER

Should the Department of Housing Preservation and Development of New York City focus on any particular set of boroughs, ZIP codes, or street (where the complaints are severe) for the specific type of complaints you identified in response to Question 1?

In this exercise, you will use 311 Dataset to determine whether to focus on any particular borough, ZIP code, or street (where the complaints are severe) for the specific Complaint Type you decided to focus at the end of the last exercise.

In [191]:

```
df_NYC311_01=df_NYC311[df_NYC311['Complaint Type']=='HEAT/HOT WATER']
df_NYC311_01
```

Out[191]:

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip	Incid
0	0	45531130	2020-02-02 06:09:17	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10019.0	4
2	2	45527528	2020-02-02 02:27:41	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11372.0	87-15
3	3	45530329	2020-02-02 12:13:18	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10458.0	2405 E
5	5	45530344	2020-02-02 07:52:08	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10456.0	11
7	7	45527939	2020-02-02 09:00:54	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11372.0	37-20
9	9	45529416	2020-02-02 10:59:58	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11207.0	
10	10	45530349	2020-02-02 05:38:39	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10459.0	1
11	11	45530831	2020-02-02 10:13:57	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10039.0	2819 E
14	14	45529877	2020-02-02 16:09:57	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11226.0	297 LE
15	15	45530802	2020-02-02 15:50:43	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10011.0	3
17	17	45528911	2020-02-02 04:32:03	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10026.0	22 P/

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip	Incid
18	18	45528877	2020-02-02 11:10:08	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10031.0	52
19	19	45529362	2020-02-02 13:38:25	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10473.0	88
20	20	45529853	2020-02-02 10:35:22	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11204.0	2270
21	21	45528464	2020-02-02 23:46:12	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11210.0	165
25	25	45532031	2020-02-02 11:19:40	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10034.0	50
26	26	45529887	2020-02-02 09:43:11	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11238.0	50
28	28	45529874	2020-02-02 21:26:43	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11203.0	2
29	29	45528454	2020-02-02 22:25:32	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10463.0	KI
30	30	45531430	2020-02-02 23:26:24	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10462.0	2040 E
31	31	45531874	2020-02-02 11:44:22	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10462.0	2040 E
32	32	45530782	2020-02-02 20:34:37	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11210.0	390
33	33	45530303	2020-02-02 08:24:04	2020-02-02 18:53:37	HEAT/HOT WATER	RESIDENTIAL BUILDING	10457.0	21

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip	Incid
34	34	45527571	2020-02-02 15:41:33	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10463.0	16
35	35	45528925	2020-02-02 19:23:01	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10460.0	91
36	36	45531281	2020-02-02 22:55:29	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10458.0	WA
37	37	45531738	2020-02-02 15:10:21	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10031.0	62
38	38	45528436	2020-02-02 13:41:10	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11385.0	5
40	40	45530808	2020-02-02 18:12:52	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11219.0	1454
44	44	45528926	2020-02-02 23:30:11	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	11373.0	89-21
...	...	...	...	...	...	...	...	
6019810	6019810	44062627	2019-10-15 14:35:03	2019-10-16 13:57:35	HEAT/HOT WATER	RESIDENTIAL BUILDING	10457.0	3(
6019811	6019811	44062634	2019-10-15 09:41:59	2019-10-15 19:01:42	HEAT/HOT WATER	RESIDENTIAL BUILDING	10457.0	
6019812	6019812	44062648	2019-10-15 13:33:21	2019-10-16 08:49:35	HEAT/HOT WATER	RESIDENTIAL BUILDING	11206.0	165
6019813	6019813	44063460	2019-10-15 06:04:28	2019-10-16 19:12:28	HEAT/HOT WATER	RESIDENTIAL BUILDING	11213.0	94

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip	Incid
6019814	6019814	44063463	2019-10-15 23:02:37	2019-10-16 19:09:53	HEAT/HOT WATER	RESIDENTIAL BUILDING	11226.0	262
6019815	6019815	44063467	2019-10-15 08:15:08	2019-10-16 21:55:24	HEAT/HOT WATER	RESIDENTIAL BUILDING	10466.0	S(
6019816	6019816	44063469	2019-10-15 10:47:49	2019-10-16 21:57:50	HEAT/HOT WATER	RESIDENTIAL BUILDING	10023.0	3
6019817	6019817	44063470	2019-10-15 07:49:50	2019-10-16 09:15:08	HEAT/HOT WATER	RESIDENTIAL BUILDING	11233.0	3
6019818	6019818	44063493	2019-10-15 14:55:34	2019-10-16 16:18:30	HEAT/HOT WATER	RESIDENTIAL BUILDING	11423.0	100-11
6019819	6019819	44063499	2019-10-15 16:57:42	2019-10-16 14:04:54	HEAT/HOT WATER	RESIDENTIAL BUILDING	10467.0	PARK\
6019820	6019820	44063561	2019-10-15 07:24:44	2019-10-16 21:29:49	HEAT/HOT WATER	RESIDENTIAL BUILDING	11414.0	149-30
6019821	6019821	44063574	2019-10-15 06:23:40	2019-10-16 21:55:24	HEAT/HOT WATER	RESIDENTIAL BUILDING	10466.0	73
6019822	6019822	44063580	2019-10-15 11:15:53	2019-10-16 18:50:00	HEAT/HOT WATER	RESIDENTIAL BUILDING	10303.0	2524
6019823	6019823	44063592	2019-10-15 06:45:03	2019-10-15 16:05:35	HEAT/HOT WATER	RESIDENTIAL BUILDING	10029.0	4(
6019824	6019824	44063603	2019-10-15 07:40:34	2019-10-16 02:07:17	HEAT/HOT WATER	RESIDENTIAL BUILDING	10474.0	838 HU
6019825	6019825	44063604	2019-10-15 08:44:42	2019-10-16 20:25:11	HEAT/HOT WATER	RESIDENTIAL BUILDING	11204.0	1953

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip	Incid
6019826	6019826	44063628	2019-10-15 18:18:33	2019-10-16 10:44:28	HEAT/HOT WATER	RESIDENTIAL BUILDING	10024.0	2301
6019827	6019827	44063632	2019-10-15 19:27:28	2019-10-16 17:01:05	HEAT/HOT WATER	RESIDENTIAL BUILDING	10472.0	COMM
6019828	6019828	44063636	2019-10-15 08:03:00	2019-10-16 17:10:34	HEAT/HOT WATER	RESIDENTIAL BUILDING	10036.0	5
6019829	6019829	44063643	2019-10-15 18:56:11	2019-10-16 19:19:47	HEAT/HOT WATER	RESIDENTIAL BUILDING	11234.0	1
6019830	6019830	44063644	2019-10-15 22:27:15	2019-10-16 10:44:29	HEAT/HOT WATER	RESIDENTIAL BUILDING	10470.0	7
6019831	6019831	44063645	2019-10-15 07:50:19	2019-10-16 17:14:33	HEAT/HOT WATER	RESIDENTIAL BUILDING	10458.0	W
6019833	6019833	44063661	2019-10-15 22:13:11	2019-10-16 02:07:21	HEAT/HOT WATER	RESIDENTIAL BUILDING	10452.0	WC
6019834	6019834	44063670	2019-10-15 08:25:25	2019-10-16 21:55:24	HEAT/HOT WATER	RESIDENTIAL BUILDING	10470.0	7(
6019835	6019835	44063671	2019-10-15 09:32:54	2019-10-16 19:02:48	HEAT/HOT WATER	RESIDENTIAL BUILDING	10458.0	2786 E
6019836	6019836	44063680	2019-10-15 18:43:56	2019-10-16 17:55:02	HEAT/HOT WATER	RESIDENTIAL BUILDING	10029.0	3
6019838	6019838	44063685	2019-10-15 10:35:45	2019-10-16 17:55:02	HEAT/HOT WATER	RESIDENTIAL BUILDING	10029.0	3
6019839	6019839	44063692	2019-10-15 06:09:28	2019-10-15 16:12:58	HEAT/HOT WATER	RESIDENTIAL BUILDING	10461.0	3555 E

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip	Incid
6019840	6019840	44063724	2019-10-15 10:54:45	2019-10-16 18:14:29	HEAT/HOT WATER	RESIDENTIAL BUILDING	10034.0	16
6019841	6019841	44063726	2019-10-15 14:07:34	2019-10-16 19:10:19	HEAT/HOT WATER	RESIDENTIAL BUILDING	10467.0	66

1261574 rows × 16 columns

In [203]:

```
grouped3=df_NYC311_01[['Unique Key','Borough']].groupby(by='Borough').count()
grouped3.rename(columns={'Unique Key':'Number of Complaints'}, inplace=True)
grouped3
```

Out[203]:

Borough	Number of Complaints
BRONX	410853
BROOKLYN	384523
MANHATTAN	285526
QUEENS	168318
STATEN ISLAND	12353
Unspecified	1

In [208]:

```
grouped4=df_NYC311_01[['Unique Key','Incident Zip']].groupby(by='Incident Zip').
count()
grouped4.rename(columns={'Unique Key':'Number of Complaints'}, inplace=True)
grouped4.sort_values(by='Number of Complaints',ascending=False,inplace=True)
grouped4
```

Out[208]:

Incident Zip	Number of Complaints
11226.0	41786
10458.0	38864
10467.0	38110
10468.0	34507
10453.0	34241
10457.0	28295
10452.0	28118
10031.0	27274
10456.0	25660
10462.0	25523
11225.0	25194
11213.0	22916
10040.0	22214
10472.0	21537
10032.0	20624
10033.0	20419
10460.0	19956
10463.0	19751
11212.0	19260
11373.0	17585
11233.0	17446
10451.0	16603
11203.0	15731
10034.0	15353
10025.0	15024
11221.0	14744
10466.0	14700
11207.0	14277
11216.0	14142
10027.0	13913
...	...
10464.0	219
10312.0	203
11426.0	201
10309.0	181
10007.0	162

Incident Zip	Number of Complaints
10006.0	158
10282.0	143
11004.0	136
10308.0	121
11362.0	109
10307.0	102
11001.0	67
10005.0	61
10004.0	57
11697.0	47
11040.0	46
10280.0	40
10069.0	20
11109.0	18
11005.0	7
10162.0	5
10020.0	3
10041.0	3
10169.0	2
10045.0	1
10103.0	1
10112.0	1
10121.0	1
10129.0	1
12345.0	1

191 rows × 1 columns

In [209]:

```
grouped4=df_NYC311_01[['Unique Key','Street Name']].groupby(by='Street Name').co
unt()
grouped4.rename(columns={'Unique Key':'Number of Complaints'}, inplace=True)
grouped4.sort_values(by='Number of Complaints',ascending=False,inplace=True)
grouped4
```

Out[209]:

Street Name	Number of Complaints
GRAND CONCOURSE	22287
BROADWAY	15368
ELMHURST AVENUE	11377
OCEAN AVENUE	10110
MORRIS AVENUE	9727
BOYNTON AVENUE	8725
ST NICHOLAS AVENUE	8557
AMSTERDAM AVENUE	7288
DR M L KING JR BOULEVARD	6727
OCEAN PARKWAY	6276
LINDEN BOULEVARD	6071
ARDEN STREET	5754
DECATUR AVENUE	5698
SEDGWICK AVENUE	5643
WALTON AVENUE	5554
CRESTON AVENUE	5465
BEDFORD AVENUE	5186
SHORE ROAD	5177
NOSTRAND AVENUE	5134
EAST TREMONT AVENUE	5128
EASTERN PARKWAY	5066
FULTON STREET	5001
RIVERSIDE DRIVE	4893
2 AVENUE	4683
WASHINGTON AVENUE	4445
DAVIDSON AVENUE	4440
SHERIDAN AVENUE	4387
41 AVENUE	4334
3 AVENUE	4332
ADAM C POWELL BOULEVARD	4252
...	...
TYLER AVENUE	1
258 STREET	1
TWIN OAK DRIVE	1
BEACH 102 STREET	1
TUSKEGEE AIRMAN WAY	1

Street Name	Number of Complaints
REEDER STREET	1
SIMONSON STREET	1
RED HOOK LANE	1
JOHNSON STREET	1
MERIDIAN BOULEVARD	1
LUCILLE AVENUE	1
BEACH 148 STREET	1
COMMERCIAL STREET	1
BEACH 136 STREET	1
CASS PLACE	1
39 DRIVE	1
COMMODORE DRIVE	1
UNION COURT	1
GREENPOINT AV BR OVR NEWTOWN CRK	1
CARTER AVENUE	1
AUBURN PLACE	1
32 ROAD	1
CONDUIT BOULEVARD	1
30 PLACE	1
MELISSA STREET	1
GREAVES AVENUE	1
AUBURNDALE LANE	1
271 STREET	1
269 STREET	1
ST STEPHENS PLACE	1

5290 rows × 1 columns

In this exercise, use the 311 dataset.

You also need to read back the PLUTO dataset from Cloud Object Store that you saved previously in the course. Use the PLUTO dataset for the borough that you already identified to focus on the last exercise.Ensure that you use only a limited number of fields from the dataset so that you are not consuming too much memory during your analysis.

The recommended fields are Address, BldgArea, BldgDepth, BuiltFAR, CommFAR, FacilFAR, Lot, LotArea, LotDepth, NumBldgs, NumFloors, OfficeArea, ResArea, ResidFAR, RetailArea, YearBuilt, YearAlter1, ZipCode, YCoord, and XCoord.

At the end of this exercise, you should determine whether the type of complaint that you have identified as the response to Question 1 has an obvious relationship with any particular characteristic or characteristics of the houses.

In [219]:

```
df_BX.head()
```

Out[219]:

	Borough	Block	Lot	CD	CT2010	CB2010	SchoolDist	Council	ZipCode	FireComp	...
0	BX	2260	1	201	19.0	1022.0	7.0	8.0	10454.0	L029	...
1	BX	2260	4	201	19.0	1022.0	7.0	8.0	10454.0	L029	...
2	BX	2260	10	201	19.0	1022.0	7.0	8.0	10454.0	L029	...
3	BX	2260	17	201	19.0	1022.0	7.0	8.0	10454.0	L029	...
4	BX	2260	18	201	19.0	1022.0	7.0	8.0	10454.0	L029	...

5 rows × 87 columns

In [211]:

```
df_BX.columns
```

Out[211]:

```
Index(['Borough', 'Block', 'Lot', 'CD', 'CT2010', 'CB2010', 'SchoolD
ist',
       'Council', 'ZipCode', 'FireComp', 'PolicePrct', 'HealthCenter
District',
       'HealthArea', 'SanitBoro', 'SanitDistrict', 'SanitSub', 'Addr
ess',
       'ZoneDist1', 'ZoneDist2', 'ZoneDist3', 'ZoneDist4', 'Overlay
1',
       'Overlay2', 'SPDist1', 'SPDist2', 'SPDist3', 'LtdHeight', 'Sp
litZone',
       'BldgClass', 'LandUse', 'Easements', 'OwnerType', 'OwnerNam
e',
       'LotArea', 'BldgArea', 'ComArea', 'ResArea', 'OfficeArea', 'R
etailArea',
       'GarageArea', 'StrgeArea', 'FactryArea', 'OtherArea', 'AreaSo
urce',
       'NumBldgs', 'NumFloors', 'UnitsRes', 'UnitsTotal', 'LotFron
t',
       'LotDepth', 'BldgFront', 'BldgDepth', 'Ext', 'ProxCode', 'Irr
LotCode',
       'LotType', 'BsmtCode', 'AssessLand', 'AssessTot', 'ExemptLan
d',
       'ExemptTot', 'YearBuilt', 'YearAlter1', 'YearAlter2', 'HistDi
st',
       'Landmark', 'BuiltFAR', 'ResidFAR', 'CommFAR', 'FacilFAR', 'B
oroCode',
       'BBL', 'CondoNo', 'Tract2010', 'XCoord', 'YCoord', 'ZoneMap',
'ZMCode',
       'Sanborn', 'TaxMap', 'EDesigNum', 'APPBBL', 'APPDate', 'PLUTO
MapID',
       'FIRM07_FLAG', 'PFIRM15_FLAG', 'Version'],
      dtype='object')
```

In [221]:

```
#set address as index

df_BX1=df_BX.set_index('Address')
df_BX1.head()
```

Out[221]:

Address	Borough	Block	Lot	CD	CT2010	CB2010	SchoolDist	Council	ZipCode	Fi
122 BRUCKNER BOULEVARD	BX	2260	1	201	19.0	1022.0	7.0	8.0	10454.0	
126 BRUCKNER BOULEVARD	BX	2260	4	201	19.0	1022.0	7.0	8.0	10454.0	
138 BRUCKNER BOULEVARD	BX	2260	10	201	19.0	1022.0	7.0	8.0	10454.0	
144 BRUCKNER BOULEVARD	BX	2260	17	201	19.0	1022.0	7.0	8.0	10454.0	
148 BRUCKNER BOULEVARD	BX	2260	18	201	19.0	1022.0	7.0	8.0	10454.0	

5 rows × 86 columns

In [234]:

```
df_NYC311_01=df_NYC311[df_NYC311['Borough']=='BRONX']
df_NYC311_01
```

Out[234]:

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip	In
3	3	45530329	2020-02-02 12:13:18	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10458.0	24
5	5	45530344	2020-02-02 07:52:08	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10456.0	
10	10	45530349	2020-02-02 05:38:39	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10459.0	
16	16	45528878	2020-02-02 04:11:19	NaT	UNSANITARY CONDITION	RESIDENTIAL BUILDING	10456.0	
19	19	45529362	2020-02-02 13:38:25	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10473.0	
22	22	45530266	2020-02-02 16:32:27	NaT	UNSANITARY CONDITION	RESIDENTIAL BUILDING	10457.0	1
27	27	45530787	2020-02-02 19:20:04	NaT	WATER LEAK	RESIDENTIAL BUILDING	10458.0	
29	29	45528454	2020-02-02 22:25:32	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10463.0	
30	30	45531430	2020-02-02 23:26:24	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10462.0	204
31	31	45531874	2020-02-02 11:44:22	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10462.0	204
33	33	45530303	2020-02-02 08:24:04	2020-02-02 18:53:37	HEAT/HOT WATER	RESIDENTIAL BUILDING	10457.0	

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip	In
34	34	45527571	2020-02-02 15:41:33	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10463.0	
35	35	45528925	2020-02-02 19:23:01	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10460.0	
36	36	45531281	2020-02-02 22:55:29	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10458.0	
39	39	45530702	2020-02-02 20:13:14	NaT	ELECTRIC	RESIDENTIAL BUILDING	10456.0	
45	45	45527990	2020-02-02 14:21:43	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10463.0	
53	53	45529405	2020-02-02 07:46:11	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10453.0	
54	54	45530822	2020-02-02 07:02:35	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10453.0	
56	56	45530824	2020-02-02 11:06:39	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10471.0	66
58	58	45529433	2020-02-02 10:57:02	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10468.0	232
59	59	45527559	2020-02-02 13:22:28	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10462.0	204
63	63	45527566	2020-02-02 23:56:06	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10453.0	
64	64	45531577	2020-02-02 13:28:55	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10468.0	232

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip	In
65	65	45530717	2020-02-02 09:52:17	NaT	UNSANITARY CONDITION	RESIDENTIAL BUILDING	10466.0	
68	68	45528450	2020-02-02 22:39:56	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10458.0	
69	69	45530720	2020-02-02 16:27:54	NaT	PLUMBING	RESIDENTIAL BUILDING	10459.0	
70	70	45532008	2020-02-02 17:23:19	NaT	UNSANITARY CONDITION	RESIDENTIAL BUILDING	10458.0	187
74	74	45527972	2020-02-02 21:39:19	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10473.0	8
75	75	45530337	2020-02-02 22:46:04	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10452.0	1
77	77	45531736	2020-02-02 22:49:43	NaT	HEAT/HOT WATER	RESIDENTIAL BUILDING	10468.0	
...	...	...	...	...	...	...	...	
6019755	6019755	44060221	2019-10-15 09:52:19	2019-10-16 18:52:16	HEAT/HOT WATER	RESIDENTIAL BUILDING	10456.0	
6019759	6019759	44060251	2019-10-15 09:11:30	2019-10-15 13:27:02	HEAT/HOT WATER	RESIDENTIAL BUILDING	10467.0	
6019762	6019762	44060279	2019-10-15 20:47:38	2019-10-16 15:21:18	HEAT/HOT WATER	RESIDENTIAL BUILDING	10456.0	
6019763	6019763	44060297	2019-10-15 10:29:43	2019-10-16 17:14:33	HEAT/HOT WATER	RESIDENTIAL BUILDING	10457.0	1

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip	In
6019764	6019764	44060306	2019-10-15 11:11:36	2019-10-15 16:12:58	HEAT/HOT WATER	RESIDENTIAL BUILDING	10461.0	35
6019770	6019770	44061036	2019-10-15 11:31:59	NaT	APPLIANCE	RESIDENTIAL BUILDING	10468.0	2
6019775	6019775	44061292	2019-10-15 08:57:32	2019-10-15 13:25:37	HEAT/HOT WATER	RESIDENTIAL BUILDING	10462.0	
6019776	6019776	44061295	2019-10-15 15:03:39	2019-10-16 13:57:35	HEAT/HOT WATER	RESIDENTIAL BUILDING	10457.0	
6019779	6019779	44061320	2019-10-15 06:21:56	2019-10-15 16:12:58	HEAT/HOT WATER	RESIDENTIAL BUILDING	10461.0	35
6019784	6019784	44061354	2019-10-15 07:33:41	2019-10-16 17:31:49	HEAT/HOT WATER	RESIDENTIAL BUILDING	10468.0	28(
6019789	6019789	44061492	2019-10-15 18:56:29	NaT	UNSANITARY CONDITION	RESIDENTIAL BUILDING	10467.0	:
6019790	6019790	44061526	2019-10-15 21:19:33	NaT	UNSANITARY CONDITION	RESIDENTIAL BUILDING	10453.0	
6019791	6019791	44062509	2019-10-15 11:16:15	NaT	GENERAL	RESIDENTIAL BUILDING	10467.0	:
6019798	6019798	44062547	2019-10-15 12:07:55	2019-10-16 10:45:31	HEAT/HOT WATER	RESIDENTIAL BUILDING	10452.0	9
6019803	6019803	44062585	2019-10-15 06:37:25	2019-10-16 19:02:47	HEAT/HOT WATER	RESIDENTIAL BUILDING	10458.0	29
6019810	6019810	44062627	2019-10-15 14:35:03	2019-10-16 13:57:35	HEAT/HOT WATER	RESIDENTIAL BUILDING	10457.0	

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip	In
6019811	6019811	44062634	2019-10-15 09:41:59	2019-10-15 19:01:42	HEAT/HOT WATER	RESIDENTIAL BUILDING	10457.0	
6019815	6019815	44063467	2019-10-15 08:15:08	2019-10-16 21:55:24	HEAT/HOT WATER	RESIDENTIAL BUILDING	10466.0	
6019819	6019819	44063499	2019-10-15 16:57:42	2019-10-16 14:04:54	HEAT/HOT WATER	RESIDENTIAL BUILDING	10467.0	PAI
6019821	6019821	44063574	2019-10-15 06:23:40	2019-10-16 21:55:24	HEAT/HOT WATER	RESIDENTIAL BUILDING	10466.0	
6019824	6019824	44063603	2019-10-15 07:40:34	2019-10-16 02:07:17	HEAT/HOT WATER	RESIDENTIAL BUILDING	10474.0	838
6019827	6019827	44063632	2019-10-15 19:27:28	2019-10-16 17:01:05	HEAT/HOT WATER	RESIDENTIAL BUILDING	10472.0	COI
6019830	6019830	44063644	2019-10-15 22:27:15	2019-10-16 10:44:29	HEAT/HOT WATER	RESIDENTIAL BUILDING	10470.0	
6019831	6019831	44063645	2019-10-15 07:50:19	2019-10-16 17:14:33	HEAT/HOT WATER	RESIDENTIAL BUILDING	10458.0	
6019833	6019833	44063661	2019-10-15 22:13:11	2019-10-16 02:07:21	HEAT/HOT WATER	RESIDENTIAL BUILDING	10452.0	\
6019834	6019834	44063670	2019-10-15 08:25:25	2019-10-16 21:55:24	HEAT/HOT WATER	RESIDENTIAL BUILDING	10470.0	
6019835	6019835	44063671	2019-10-15 09:32:54	2019-10-16 19:02:48	HEAT/HOT WATER	RESIDENTIAL BUILDING	10458.0	278
6019837	6019837	44063681	2019-10-15 11:35:19	NaT	UNSANITARY CONDITION	RESIDENTIAL BUILDING	10472.0	

	Unnamed: 0	Unique Key	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip	In
6019839	6019839	44063692	2019-10-15 06:09:28	2019-10-15 16:12:58	HEAT/HOT WATER	RESIDENTIAL BUILDING	10461.0	35
6019841	6019841	44063726	2019-10-15 14:07:34	2019-10-16 19:10:19	HEAT/HOT WATER	RESIDENTIAL BUILDING	10467.0	

1617956 rows × 16 columns

In [235]:

```
df_NYC311_01.columns
```

Out[235]:

```
Index(['Unnamed: 0', 'Unique Key', 'Created Date', 'Closed Date',
       'Complaint Type', 'Location Type', 'Incident Zip', 'Incident
Address',
       'Street Name', 'Address Type', 'City', 'Status',
       'Resolution Description', 'Borough', 'Latitude', 'Longitud
e'],
      dtype='object')
```

In [236]:

```
df_NYC311_02=df_NYC311_01[['Unique Key','Incident Address']].groupby('Incident A
ddress').count().sort_values(by='Unique Key',ascending=False).rename(columns={'U
nique Key':'Number of Complaints'})
df_NYC311_02.head()
```

Out[236]:

Incident Address	Number of Complaints
1025 BOYNTON AVENUE	9737
3810 BAILEY AVENUE	7171
750 GRAND CONCOURSE	4183
3555 BRUCKNER BOULEVARD	4158
888 GRAND CONCOURSE	3994

In [243]:

```
df_BX1['Water Incidents']=0   #add column

df_BX1.columns
```

Out[243]:

```
Index(['Borough', 'Block', 'Lot', 'CD', 'CT2010', 'CB2010', 'SchoolD
ist',
       'Council', 'ZipCode', 'FireComp', 'PolicePrct', 'HealthCenter
District',
       'HealthArea', 'SanitBoro', 'SanitDistrict', 'SanitSub', 'Zone
Dist1',
       'ZoneDist2', 'ZoneDist3', 'ZoneDist4', 'Overlay1', 'Overlay
2',
       'SPDist1', 'SPDist2', 'SPDist3', 'LtdHeight', 'SplitZone', 'B
ldgClass',
       'LandUse', 'Easements', 'OwnerType', 'OwnerName', 'LotArea',
'BldgArea',
       'ComArea', 'ResArea', 'OfficeArea', 'RetailArea', 'GarageAre
a',
       'StrgeArea', 'FactryArea', 'OtherArea', 'AreaSource', 'NumBld
gs',
       'NumFloors', 'UnitsRes', 'UnitsTotal', 'LotFront', 'LotDept
h',
       'BldgFront', 'BldgDepth', 'Ext', 'ProxCode', 'IrrLotCode', 'L
otType',
       'BsmtCode', 'AssessLand', 'AssessTot', 'ExemptLand', 'ExemptT
ot',
       'YearBuilt', 'YearAlter1', 'YearAlter2', 'HistDist', 'Landmar
k',
       'BuiltFAR', 'ResidFAR', 'CommFAR', 'FacilFAR', 'BoroCode', 'B
BL',
       'CondoNo', 'Tract2010', 'XCoord', 'YCoord', 'ZoneMap', 'ZMCod
e',
       'Sanborn', 'TaxMap', 'EDesigNum', 'APPBBL', 'APPDate', 'PLUTO
MapID',
       'FIRM07_FLAG', 'PFIRM15_FLAG', 'Version', 'Water Incidents'],
      dtype='object')
```

In [271]:

```
for address1 in df_NYC311_02.index:
    if address1 in df_BX1.index:
        df_BX1.loc[address1,'Water Incidents']=df_NYC311_02.loc[address1,'Number
of Complaints']
```

In [272]:

```
df_BX1.sort_values(by='Water Incidents',ascending=False,inplace=True)
df_BX1
```

Out[272]:

Address	Borough	Block	Lot	CD	CT2010	CB2010	SchoolDist	Council	ZipCode
1025 BOYNTON AVENUE	BX	3714	54	209	50.01	2000.0	8.0	18.0	10472.0
3810 BAILEY AVENUE	BX	3263	7	208	279.00	6000.0	10.0	11.0	10463.0
750 GRAND CONCOURSE	BX	2458	26	204	59.02	2001.0	7.0	17.0	10451.0
3555 BRUCKNER BOULEVARD	BX	4178	10	210	266.02	4000.0	8.0	13.0	10461.0
888 GRAND CONCOURSE	BX	2459	34	204	59.02	1000.0	7.0	16.0	10451.0
3230 CRUGER AVENUE	BX	4597	21	212	374.00	1001.0	11.0	15.0	10467.0
2856 WEBB AVENUE	BX	3250	71	208	267.02	1002.0	10.0	14.0	10468.0
957 WOODYCREST AVENUE	BX	2511	68	204	189.00	3002.0	9.0	8.0	10452.0
2968 PERRY AVENUE	BX	3292	19	207	415.00	1001.0	10.0	11.0	10458.0
1030 BOYNTON AVENUE	BX	3715	17	209	50.01	3000.0	8.0	18.0	10472.0
1515 SELWYN AVENUE	BX	2819	51	204	225.00	1000.0	9.0	16.0	10457.0
2176 TIEBOUT AVENUE	BX	3143	130	205	383.02	2001.0	10.0	15.0	10457.0
2074 WALLACE AVENUE	BX	4292	1	211	228.00	4000.0	11.0	13.0	10462.0
3605 SEDGWICK AVENUE	BX	3258	190	208	279.00	3000.0	10.0	11.0	10463.0
2040 BRONXDALE AVENUE	BX	4285	5	211	224.04	3001.0	11.0	13.0	10462.0
2181 BARNES AVENUE	BX	4321	50	211	228.00	3000.0	11.0	13.0	10462.0
3451 GILES PLACE	BX	3258	222	208	279.00	3000.0	10.0	11.0	10463.0
2410 WASHINGTON AVENUE	BX	3057	11	206	387.00	1006.0	10.0	15.0	10458.0
1967 MARMION AVENUE	BX	3107	36	206	365.01	3001.0	12.0	17.0	10460.0
1749 GRAND CONCOURSE	BX	2822	19	205	227.01	3000.0	9.0	14.0	10453.0

Address	Borough	Block	Lot	CD	CT2010	CB2010	SchoolDist	Council	ZipCode
1080 ANDERSON AVENUE	BX	2505	6	204	193.00	1000.0	9.0	8.0	10452.0
30 BUCHANAN PLACE	BX	3196	10	205	251.00	1002.0	10.0	14.0	10453.0
1225 SHERIDAN AVENUE	BX	2457	26	204	179.01	3001.0	9.0	16.0	10456.0
2490 TIEBOUT AVENUE	BX	3023	12	205	399.02	1002.0	10.0	15.0	10458.0
709 FAIRMOUNT PLACE	BX	2950	30	206	369.01	1000.0	12.0	17.0	10457.0
975 WALTON AVENUE	BX	2476	22	204	195.00	2002.0	9.0	8.0	10452.0
530 EAST 169 STREET	BX	2610	12	203	145.00	1000.0	9.0	16.0	10456.0
1892 MORRIS AVENUE	BX	2805	31	205	233.01	2000.0	9.0	14.0	10453.0
1221 SHERIDAN AVENUE	BX	2457	29	204	179.01	3001.0	9.0	16.0	10456.0
1381 CASTLE HILL AVENUE	BX	3935	53	209	222.00	4000.0	11.0	18.0	10462.0
...	...	...	...	...	...	...	...	...	...
1317 FTELEY AVENUE	BX	3870	62	209	64.00	3000.0	12.0	18.0	10472.0
1319 FTELEY AVENUE	BX	3870	61	209	64.00	3000.0	12.0	18.0	10472.0
1321 FTELEY AVENUE	BX	3870	60	209	64.00	3000.0	12.0	18.0	10472.0
1325 FTELEY AVENUE	BX	3870	58	209	64.00	3000.0	12.0	18.0	10472.0
1318 FTELEY AVENUE	BX	3871	12	209	64.00	2000.0	12.0	18.0	10472.0
1329 FTELEY AVENUE	BX	3870	56	209	64.00	3000.0	12.0	18.0	10472.0
1331 FTELEY AVENUE	BX	3870	55	209	64.00	3000.0	12.0	18.0	10472.0
1335 FTELEY AVENUE	BX	3870	53	209	64.00	3000.0	12.0	18.0	10472.0
1339 FTELEY AVENUE	BX	3870	51	209	64.00	3000.0	12.0	18.0	10472.0
1345 FTELEY AVENUE	BX	3870	48	209	64.00	3000.0	12.0	18.0	10472.0
1347 FTELEY AVENUE	BX	3870	47	209	64.00	3000.0	12.0	18.0	10472.0
1316 FTELEY AVENUE	BX	3871	11	209	64.00	2000.0	12.0	18.0	10472.0

Address	Borough	Block	Lot	CD	CT2010	CB2010	SchoolDist	Council	ZipCode
1320 FTELEY AVENUE	BX	3871	13	209	64.00	2000.0	12.0	18.0	10472.0
1329 CROES AVENUE	BX	3871	57	209	64.00	2000.0	12.0	18.0	10472.0
1348 FTELEY AVENUE	BX	3871	26	209	64.00	2000.0	12.0	18.0	10472.0
1333 CROES AVENUE	BX	3871	55	209	64.00	2000.0	12.0	18.0	10472.0
1337 CROES AVENUE	BX	3871	53	209	64.00	2000.0	12.0	18.0	10472.0
1339 CROES AVENUE	BX	3871	52	209	64.00	2000.0	12.0	18.0	10472.0
1345 CROES AVENUE	BX	3871	49	209	64.00	2000.0	12.0	18.0	10472.0
1347 CROES AVENUE	BX	3871	48	209	64.00	2000.0	12.0	18.0	10472.0
1352 FTELEY AVENUE	BX	3871	28	209	64.00	2000.0	12.0	18.0	10472.0
1346 FTELEY AVENUE	BX	3871	25	209	64.00	2000.0	12.0	18.0	10472.0
1322 FTELEY AVENUE	BX	3871	14	209	64.00	2000.0	12.0	18.0	10472.0
1344 FTELEY AVENUE	BX	3871	24	209	64.00	2000.0	12.0	18.0	10472.0
1342 FTELEY AVENUE	BX	3871	23	209	64.00	2000.0	12.0	18.0	10472.0
1340 FTELEY AVENUE	BX	3871	22	209	64.00	2000.0	12.0	18.0	10472.0
1332 FTELEY AVENUE	BX	3871	19	209	64.00	2000.0	12.0	18.0	10472.0
1330 FTELEY AVENUE	BX	3871	18	209	64.00	2000.0	12.0	18.0	10472.0
1328 FTELEY AVENUE	BX	3871	17	209	64.00	2000.0	12.0	18.0	10472.0
NaN	BX	0	8900	208	NaN	NaN	NaN	NaN	NaN

89854 rows × 87 columns

The recommended fields are Address, BldgArea, BldgDepth, BuiltFAR, CommFAR, FacilFAR, Lot, LotArea, LotDepth, NumBldgs, NumFloors, OfficeArea, ResArea, ResidFAR, RetailArea, YearBuilt, YearAlter1, ZipCode, YCoord, and XCoord.

In [276]:

```
df_BX2=df_BX1[['BldgArea', 'BldgDepth', 'BuiltFAR', 'CommFAR', 'FacilFAR', 'Lot'
, 'LotArea', 'LotDepth', 'NumBldgs', 'NumFloors', 'OfficeArea', 'ResArea', 'Resi
dFAR', 'RetailArea', 'YearBuilt', 'YearAlter1', 'ZipCode', 'YCoord','XCoord','Wa
ter Incidents']]
```

In [277]:

```
df_BX2.head()
```

Out[277]:

Address	BldgArea	BldgDepth	BuiltFAR	CommFAR	FacilFAR	Lot	LotArea	LotDepth
1025 BOYNTON AVENUE	61500	87.00	4.99	0.0	4.8	54	12319	100.00
3810 BAILEY AVENUE	54001	92.00	2.53	0.0	4.8	7	21320	164.00
750 GRAND CONCOURSE	123000	110.00	5.91	0.0	6.5	26	20800	120.70
3555 BRUCKNER BOULEVARD	112000	53.33	6.40	0.0	4.8	10	17500	100.00
888 GRAND CONCOURSE	122800	178.00	4.32	0.0	6.5	34	28444	188.55

In [ ]:

```

```

In [ ]:

```

```