


## Import Dataset

jupyter CSR\_311 Last Checkpoint: 42 minutes ago (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

In [3]: ## Understand the dataset

# Import the dataset

dataset=pd.read_csv('D:\Simplilearn\Assessments\Data Science with Python\Customer Service Requests-311\Dataset\311_csr.csv',low_n

#Visualize the dataset


dataset.head()
```

Out[3]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	...	Bridge Highway Name	Bridge Highway Direction	Road Ramp	Bri High Segn
0	32310363	12/31/2015 11:59:45 PM	01/01/2016 12:55:15 AM	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk	10034.0	71 VERMILYEA AVENUE	...	NaN	NaN	NaN	I
1	32309934	12/31/2015 11:59:44 PM	01/01/2016 01:26:57 AM	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	11105.0	27-07 23 AVENUE	...	NaN	NaN	NaN	I
2	32309159	12/31/2015 11:59:29 PM	01/01/2016 04:51:03 AM	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	10458.0	2897 VALENTINE AVENUE	...	NaN	NaN	NaN	I
3	32305098	12/31/2015 11:57:46 PM	01/01/2016 07:43:13 AM	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Sidewalk	10461.0	2940 BAISLEY AVENUE	...	NaN	NaN	NaN	I
4	32306529	12/31/2015 11:56:58 PM	01/01/2016 03:24:42 AM	NYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street/Sidewalk	11373.0	87-14 57 ROAD	...	NaN	NaN	NaN	I

5 rows x 53 columns

## Identify Shape and check null values

jupyter CSR\_311 Last Checkpoint: 43 minutes ago (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
In [5]: #Identify the shape of the dataset

dataset.shape

Out[5]: (364558, 53)

In [6]: #Identify the variables with null values

dataset.isna()

Out[6]:
```

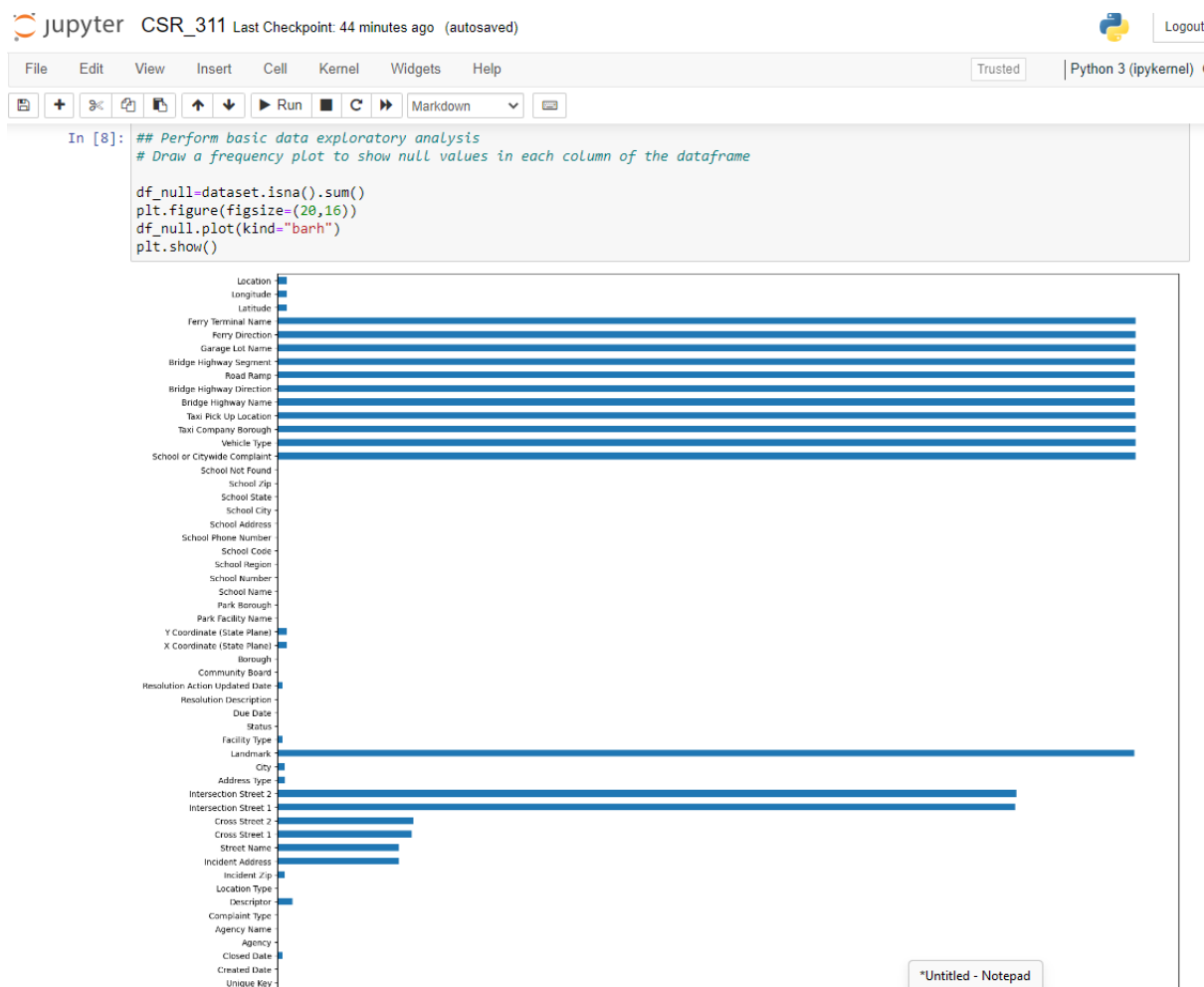
	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	...	Bridge Highway Name	Bridge Highway Direction	Road Ramp	Bridge Highway Segment	Garage Lot Name	F Direc
0	False	False	False	False	False	False	False	False	False	False	...	True	True	True	True	True	
1	False	False	False	False	False	False	False	False	False	False	...	True	True	True	True	True	
2	False	False	False	False	False	False	False	False	False	False	...	True	True	True	True	True	
3	False	False	False	False	False	False	False	False	False	False	...	True	True	True	True	True	
4	False	False	False	False	False	False	False	False	False	False	...	True	True	True	True	True	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
364553	False	False	False	False	False	False	False	False	False	False	...	True	True	True	True	True	
364554	False	False	False	False	False	False	False	False	False	False	...	True	True	True	True	True	
364555	False	False	False	False	False	False	False	False	False	False	...	True	True	True	True	True	
364556	False	False	False	False	False	False	False	False	False	False	...	True	True	True	True	True	
364557	False	False	False	False	False	False	False	False	False	False	...	True	True	True	True	True	

364558 rows x 53 columns

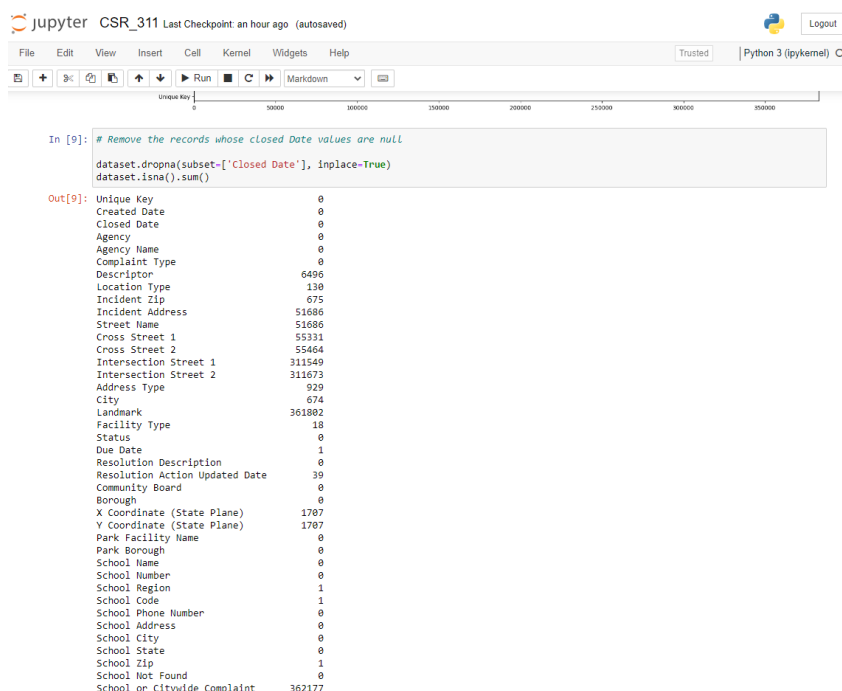
```
In [7]: dataset.isna().sum()

Out[7]: Unique Key          0
Created Date          0
Closed Date        2381
Agency              0
Agency Name         0
Complaint Type       0
Descriptor         6501
Location Type        133
Incident Zip        2998
```

## Draw Frequency plot to show null values



## Clean Closed Date NA values



## Remove columns with more than 80% of null values

Jupyter CSR\_311 Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [12]: `missing_percentage.mean()`

Out[12]: 27.087349816083353

In [13]: `## Since the percentage of missing vlaues in the dataset is less than 30% of the industry practice allowed for Null values  
## Hence we can drop the columns with null values having more than 80% from the dataset`

```
remove_cols=pd.DataFrame(dataset.columns.to_list()).set_index(0)
remove_cols=remove_cols[dataset.isna().sum()/dataset.shape[0]*100 < 80].reset_index()
remove_cols
```

Out[13]:

	0
0	Unique Key
1	Created Date
2	Closed Date
3	Agency
4	Agency Name
5	Complaint Type
6	Descriptor
7	Location Type
8	Incident Zip
9	Incident Address
10	Street Name
11	Cross Street 1
12	Cross Street 2
13	Address Type
14	City
15	Facility Type
16	Status
17	Due Date
18	Resolution Description
19	Resolution Action Updated Date
20	Community Board

## Removed columns with Unspecified values

Jupyter CSR\_311 Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [15]: `## The School related columns contains most of entries as unspecified  
dataset['School Name'].value_counts()`

Out[15]:

Unspecified	362176
Alley Pond Park - Nature Center	1

Name: School Name, dtype: int64

In [17]: `dataset['School Code'].value_counts()`

Out[17]:

Unspecified	362176
-------------	--------

Name: School Code, dtype: int64

In [18]: `## Lets remove all school related columns as most of the entries are Unspecified  
dataset=dataset.drop(dataset.filter(regex='School').columns,axis=1)  
dataset`

Out[18]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	Resolution Action Updated Date	Community Board
0	32310363	12/31/2015 11:59:45 PM	01/01/2016 12:55:15 AM	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk	10034.0	VERMILYEA AVENUE	01/01/2016 12:55:15 AM	12 MANHATTAN
1	32309934	12/31/2015 11:59:44 PM	01/01/2016 01:26:57 AM	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	11105.0	27-07 23 AVENUE	01/01/2016 01:26:57 AM	01 QUEENS
2	32309159	12/31/2015 11:59:29 PM	01/01/2016 04:51:03 AM	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	10458.0	2897 VALENTINE AVENUE	01/01/2016 04:51:03 AM	07 BRONX
3	32305098	12/31/2015 11:57:46 PM	01/01/2016 07:43:13 AM	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Sidewalk	10461.0	2940 BAISLEY AVENUE	01/01/2016 07:43:13 AM	10 BRONX
4	32306529	12/31/2015 11:56:58 PM	01/01/2016 03:24:42 AM	NYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street/Sidewalk	11373.0	87-14 57 ROAD	01/01/2016 03:24:42 AM	04 QUEENS
...	...	...	...	...	...	...	...	...	...	...	...	...
364553	29609918	01/01/2015 12:04:44 AM	01/01/2015 10:22:31 AM	NYPD	New York City Police Department	Illegal Parking	Blocked Hydrant	Street/Sidewalk	11421.0	84-25 85 ROAD	01/01/2015 10:22:31 AM	09 QUEENS
364554	29608392	01/01/2015 12:04:28 AM	01/01/2015 02:25:02 AM	NYPD	New York City Police Department	Noise - Vehicle	Car/Truck Horn	Street/Sidewalk	10468.0	2555 SEDGWICK AVENUE	01/01/2015 02:25:02 AM	07 BRONX
364555	29607589	01/01/2015 12:01:30 AM	01/01/2015 12:20:33 AM	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk	10031.0	508 WEST 139 STREET	01/01/2015 12:20:33 AM	09 MANHATTAN

## Convert dtype values of Date columns from object to dtype = datetime64[ns]

Jupyter CSR\_311 Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted | Python 3 (ipykernel)

Run Markdown

```
In [25]: dataset.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 362177 entries, 0 to 364557
Data columns (total 26 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   Unique Key            362177 non-null int64
 1   Created Date           362177 non-null object
 2   Closed Date            362177 non-null object
 3   Agency                 362177 non-null object
 4   Agency Name            362177 non-null object
 5   Complaint Type         362177 non-null object
 6   Descriptor             355681 non-null object
 7   Location Type          362047 non-null object
 8   Incident Zip           361502 non-null float64
 9   Incident Address       310491 non-null object
10   Street Name            310491 non-null object
11   Cross Street 1         306846 non-null object
12   Cross Street 2         306713 non-null object
13   Address Type           361248 non-null object
14   City                   361503 non-null object
15   Facility Type          362159 non-null object
16   Status                 362177 non-null object
17   Due Date               362176 non-null object
18   Resolution Description  362177 non-null object
19   Resolution Action Updated Date 362138 non-null object
20   Borough               362177 non-null object
21   X Coordinate (State Plane) 360470 non-null float64
22   Y Coordinate (State Plane) 360470 non-null float64
23   Latitude               360470 non-null float64
24   Longitude              360470 non-null float64
25   Location               360470 non-null object
dtypes: float64(5), int64(1), object(20)
memory usage: 74.6+ MB
```

```
In [26]: ## Created and Closed date are in object type. Hence, need to change it to datetime type
```

```
dataset["Created Date"] = pd.to_datetime(dataset["Created Date"])
dataset["Closed Date"] = pd.to_datetime(dataset["Closed Date"])
dataset["Due Date"] = pd.to_datetime(dataset["Due Date"])
dataset.head()
```

Out[26]:

Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	Status	Due Date	Resolution Description	Resolution Action Updated Date
------------	--------------	-------------	--------	-------------	----------------	------------	---------------	--------------	------------------	--------	----------	------------------------	--------------------------------

## Insert new column for response time of Open and closed complaints

Jupyter CSR\_311 Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted | Python 3 (ipykernel)

Run Markdown

```
In [28]: ## Calculate the time elapsed in closed and created date for Response and Closure
```

```
dataset["Elapsed_Time"] = dataset["Closed Date"] - dataset["Created Date"]
Elapsed_Time = []
for x in dataset["Closed Date"] - dataset["Created Date"]:
    close = x.total_seconds()
    Elapsed_Time.append(close)
dataset["Elapsed_Time"] = Elapsed_Time
```

```
In [29]: ## Print the column of Elapsed_Time from the dataset to check if it is converted into secs
dataset.head()
```

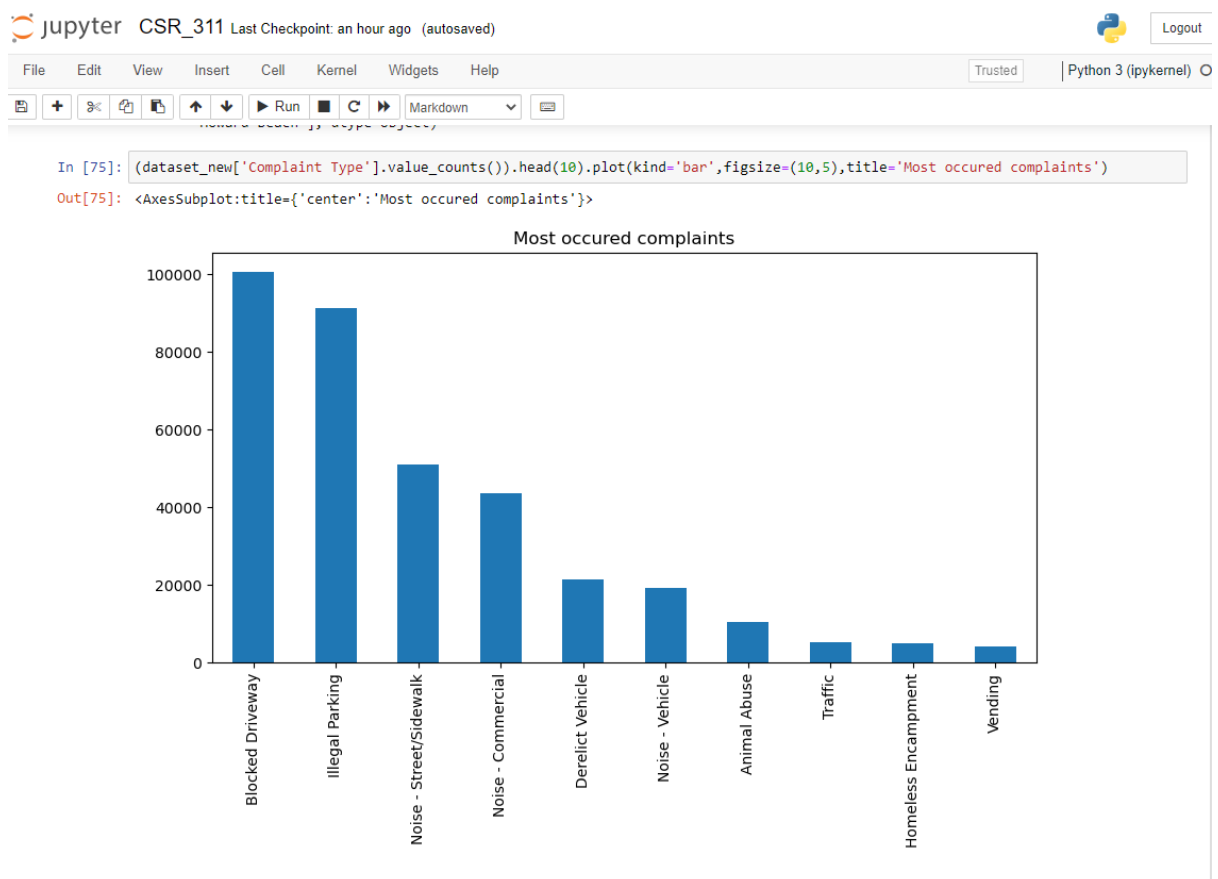
Out[29]:

pe	Incident Zip	Incident Address	Due Date	Resolution Description	Resolution Action Updated Date	Borough	X Coordinate (State Plane)	Y Coordinate (State Plane)	Latitude	Longitude	Location	Elapsed_Time
alk	10034.0	71 VERMILYEA AVENUE	2016-01-01 07:59:45	The Police Department responded and upon arriv...	01/01/2016 12:55:15 AM	MANHATTAN	1005409.0	254678.0	40.865682	-73.923501	(40.86568153633767, -73.923500995571744)	3330.0
alk	11105.0	27-07 23 AVENUE	2016-01-01 07:59:44	The Police Department responded to the complai...	01/01/2016 01:26:57 AM	QUEENS	1007766.0	221986.0	40.775945	-73.915094	(40.775945312321085, -73.9150938368005)	5233.0
alk	10458.0	2897 VALENTINE AVENUE	2016-01-01 07:59:29	The Police Department responded and upon arriv...	01/01/2016 04:51:03 AM	BRONX	1015081.0	256380.0	40.870325	-73.888525	(40.87032452211424, -73.88852464418646)	17494.0
alk	10461.0	2940 BAISLEY AVENUE	2016-01-01 07:57:46	The Police Department responded to the complai...	01/01/2016 07:43:13 AM	BRONX	1031740.0	243899.0	40.835994	-73.828379	(40.83599404063083, -73.82837939584206)	27927.0
alk	11373.0	87-14 57 ROAD	2016-01-01 07:56:58	The Police Department responded and upon arriv...	01/01/2016 03:24:42 AM	QUEENS	1019123.0	206375.0	40.733060	-73.874170	(40.733059618056015, -73.87416975810375)	12464.0

## Descriptive statistics of newly created column

```
jupyter CSR_311 Last Checkpoint: an hour ago (autosaved) Logout
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)
In [30]: ## View the descriptive statistics of newly created column i.e. Elapsed_Time
dataset_new=dataset
pd.options.display.float_format = "{:.2f}".format
dataset_new["Elapsed_Time"].describe()
Out[30]: count    362177.00
         mean     15113.30
         std      21102.55
         min         61.00
         25%      4533.00
         50%      9616.00
         75%     18878.00
         max     213432.00
         Name: Elapsed_Time, dtype: float64
```

## Top 10 complaints



[illegible]

Jupyter CSR\_311 Last Checkpoint an hour ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

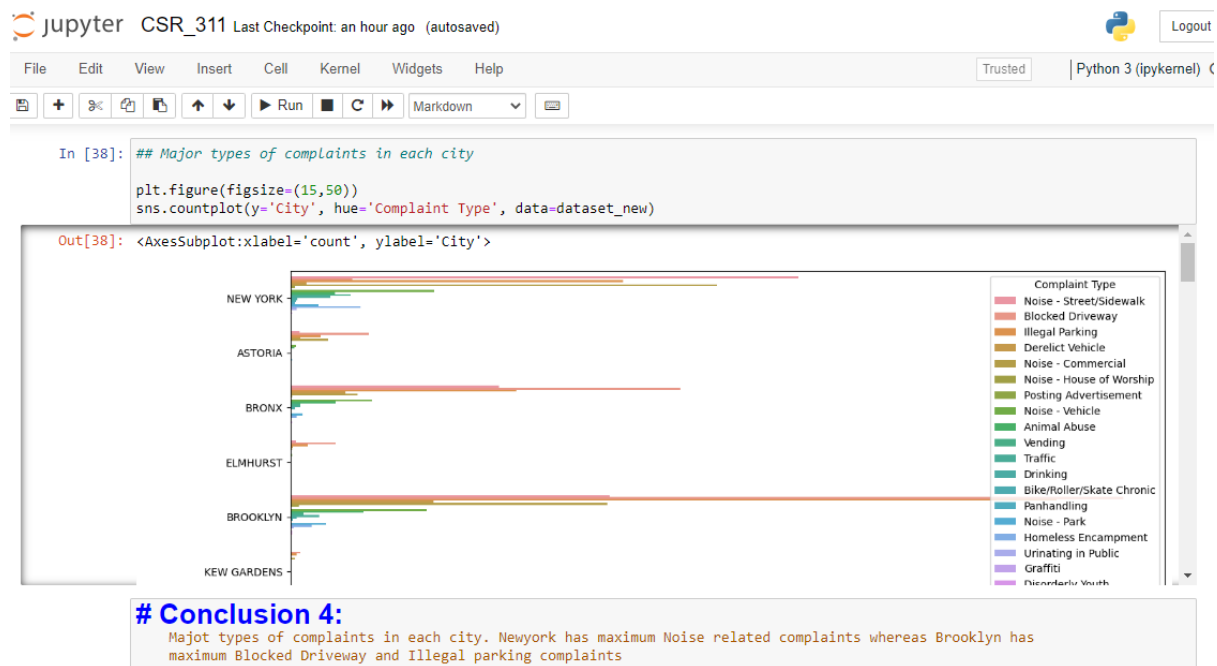
Python 3 (ipykernel)

```
In [37]: ## Frequency plot for City-wise complaints
plt.figure(figsize=(10,20))
sns.histplot(data=dataset_new,y='City')
```

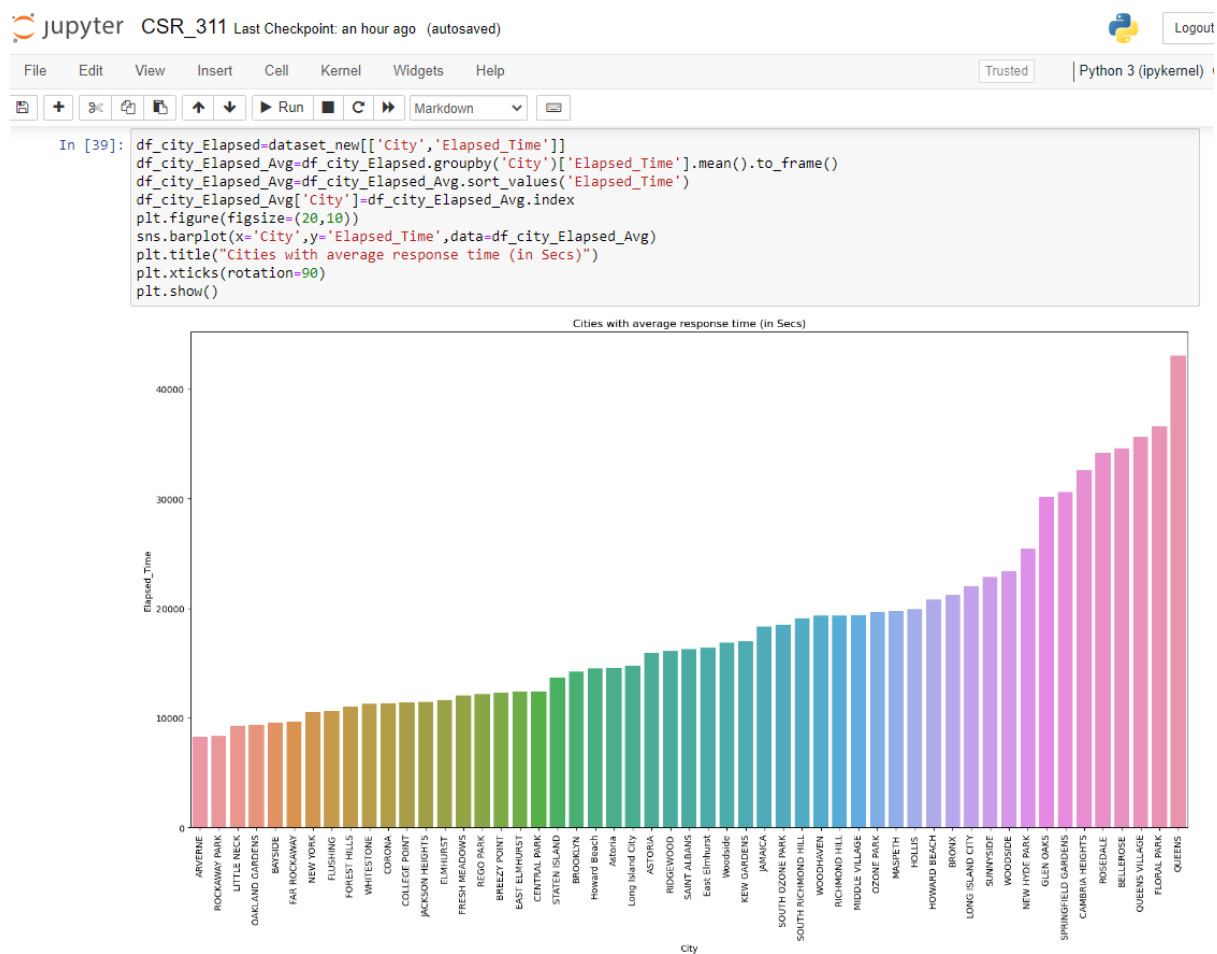
Out[37]: <AxesSubplot:xlabel='Count', ylabel='City'>

City	Count (approx.)
NEW YORK	15
ASTORIA	2
BRONX	10
ELMHURST	1
BROOKLYN	18
KEW GARDENS	0.5
JACKSON HEIGHTS	0.5
MIDDLE VILLAGE	0.5
REGO PARK	0.5
SAINT ALBANS	0.5
JAMAICA	2
SOUTH RICHMOND HILL	0.5
RIDGEWOOD	1.5
HOWARD BEACH	0.5
FOREST HILLS	0.5
STATEN ISLAND	4
OZONE PARK	0.5
RICHMOND HILL	0.5
WOODHAVEN	0.5
FLUSHING	2
CORONA	0.5
QUEENS VILLAGE	1
OAKLAND GARDENS	0.5

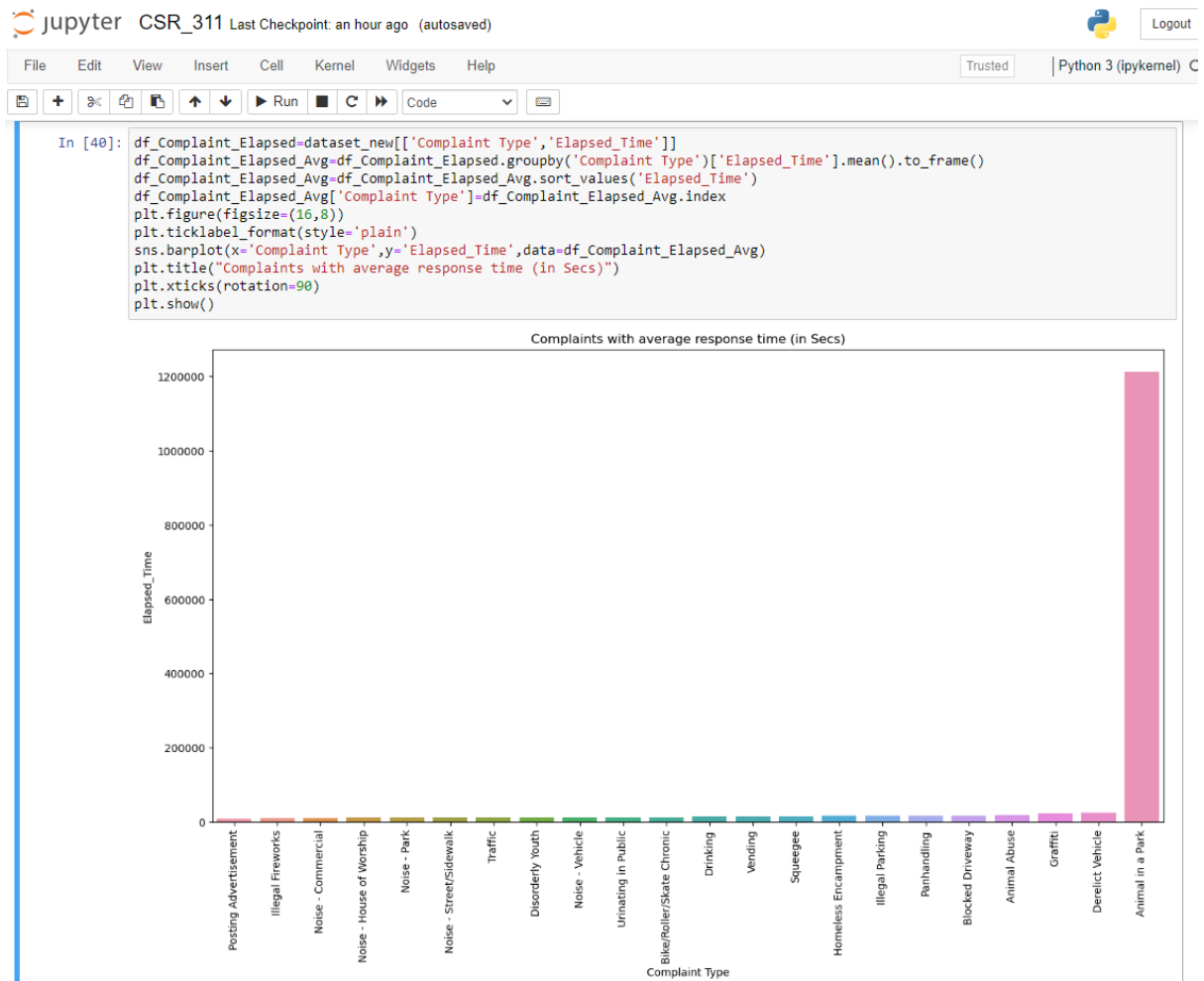
## Major types of complaints in each city



## Cities with average response time



## Complaints with average response time



## Separate dataset for complaints based on cities

Jupyter CSR\_311 Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) C

```
In [74]: df_cce=dataset_new[['City','Complaint Type','Elapsed_Time']].copy()
cce=pd.DataFrame({'count':df_cce.groupby(['City','Complaint Type']).size()})
cce.head(50)
```

Out[74]:

City	Complaint Type	Count
ARVERNE	Animal Abuse	46
	Blocked Driveway	50
	Derelict Vehicle	32
	Disorderly Youth	2
	Drinking	1
	Graffiti	1
	Homeless Encampment	4
	Illegal Parking	62
	Noise - Commercial	2
	Noise - House of Worship	14
	Noise - Park	2
	Noise - Street/Sidewalk	29
	Noise - Vehicle	10
	Panhandling	1
	Traffic	1
ASTORIA	Animal Abuse	170
	Bike/Roller/Skate Chronic	16
	Blocked Driveway	3436
	Derelict Vehicle	426
	Disorderly Youth	5
	Drinking	43
	Graffiti	4
	Homeless Encampment	32
	Illegal Fireworks	4
	Illegal Parking	4