

```
In [1]: from google.colab import drive
drive.mount('/content/MyDrive/')
```

Drive already mounted at /content/MyDrive/; to attempt to forcibly remount, call drive.mount("/content/MyDrive/", force\_remount=True).

```
In [60]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
import calendar
import sys
import traceback
%matplotlib inline
```

1.Import a 311 NYC service request.

```
In [3]: data = pd.read_csv('/content/MyDrive/MyDrive/311_Service_Requests_from_20
10_to_Present.csv')
```

```
/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py:
2882: DtypeWarning: Columns (48,49) have mixed types.Specify dtype optio
n on import or set low_memory=False.
exec(code_obj, self.user_global_ns, self.user_ns)
```

In [4]: `data.info()`

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 300698 entries, 0 to 300697
```

```
Data columns (total 53 columns):
```

#	Column	Non-Null Count	Dtype
0	Unique Key	300698 non-null	int64
1	Created Date	300698 non-null	object
2	Closed Date	298534 non-null	object
3	Agency	300698 non-null	object
4	Agency Name	300698 non-null	object
5	Complaint Type	300698 non-null	object
6	Descriptor	294784 non-null	object
7	Location Type	300567 non-null	object
8	Incident Zip	298083 non-null	float64
9	Incident Address	256288 non-null	object
10	Street Name	256288 non-null	object
11	Cross Street 1	251419 non-null	object
12	Cross Street 2	250919 non-null	object
13	Intersection Street 1	43858 non-null	object
14	Intersection Street 2	43362 non-null	object
15	Address Type	297883 non-null	object
16	City	298084 non-null	object
17	Landmark	349 non-null	object
18	Facility Type	298527 non-null	object
19	Status	300698 non-null	object
20	Due Date	300695 non-null	object
21	Resolution Description	300698 non-null	object
22	Resolution Action Updated Date	298511 non-null	object
23	Community Board	300698 non-null	object
24	Borough	300698 non-null	object
25	X Coordinate (State Plane)	297158 non-null	float64
26	Y Coordinate (State Plane)	297158 non-null	float64
27	Park Facility Name	300698 non-null	object
28	Park Borough	300698 non-null	object
29	School Name	300698 non-null	object
30	School Number	300698 non-null	object
31	School Region	300697 non-null	object
32	School Code	300697 non-null	object
33	School Phone Number	300698 non-null	object
34	School Address	300698 non-null	object
35	School City	300698 non-null	object
36	School State	300698 non-null	object
37	School Zip	300697 non-null	object
38	School Not Found	300698 non-null	object
39	School or Citywide Complaint	0 non-null	float64
40	Vehicle Type	0 non-null	float64
41	Taxi Company Borough	0 non-null	float64
42	Taxi Pick Up Location	0 non-null	float64
43	Bridge Highway Name	243 non-null	object
44	Bridge Highway Direction	243 non-null	object
45	Road Ramp	213 non-null	object
46	Bridge Highway Segment	213 non-null	object
47	Garage Lot Name	0 non-null	float64
48	Ferry Direction	1 non-null	object
49	Ferry Terminal Name	2 non-null	object
50	Latitude	297158 non-null	float64
51	Longitude	297158 non-null	float64
52	Location	297158 non-null	object

```
dtypes: float64(10), int64(1), object(42)
```

```
memory usage: 121.6+ MB
```

```
In [5]: data.head(5)
```

```
Out[5]:
```

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Locatio
0	32310363	12/31/2015 11:59:45 PM	01-01-16 0:55	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/S
1	32309934	12/31/2015 11:59:44 PM	01-01-16 1:26	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/S
2	32309159	12/31/2015 11:59:29 PM	01-01-16 4:51	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/S
3	32305098	12/31/2015 11:57:46 PM	01-01-16 7:43	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/S
4	32306529	12/31/2015 11:56:58 PM	01-01-16 3:24	NYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street/S

5 rows × 53 columns

```
In [6]: data['Agency'].values
```

```
Out[6]: array(['NYPD', 'NYPD', 'NYPD', ..., 'NYPD', 'NYPD', 'NYPD'], dtype=object)
```

```
In [7]: pd.set_option('display.max_columns',None)
data.sample(5)
```

Out[7]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	L
243992	30690313	05/24/2015 01:33:08 PM	05/24/2015 03:19:48 PM	NYPD	New York City Police Department	Blocked Driveway	No Access	St
40550	32029961	11/20/2015 04:48:03 PM	11/21/2015 07:12:55 AM	NYPD	New York City Police Department	Illegal Parking	Posted Parking Sign Violation	St
226877	30792483	06-07-15 18:11	06-07-15 18:50	NYPD	New York City Police Department	Noise - Park	Loud Music/Party	Pa
85883	31721124	10-09-15 2:44	10-09-15 3:22	NYPD	New York City Police Department	Noise - Commercial	Loud Talking	
26086	32139424	12-05-15 16:16	12-06-15 2:19	NYPD	New York City Police Department	Noise - Vehicle	Car/Truck Music	St

```
In [8]: for column in data:  
        print(f'{column} : {data[column].unique()}')
```

Unique Key : [32310363 32309934 32309159 ... 30283424 30280004 30281825]  
Created Date : ['12/31/2015 11:59:45 PM' '12/31/2015 11:59:44 PM'  
'12/31/2015 11:59:29 PM' ... '03/29/2015 12:33:03 AM'  
'03/29/2015 12:33:02 AM' '03/29/2015 12:33:01 AM']  
Closed Date : ['01-01-16 0:55' '01-01-16 1:26' '01-01-16 4:51' ...  
'03/29/2015 02:33:59 AM' '03/29/2015 04:38:35 AM'  
'03/29/2015 04:41:50 AM']  
Agency : ['NYPD']  
Agency Name : ['New York City Police Department' 'NYPD' 'Internal Affairs Bureau']  
Complaint Type : ['Noise - Street/Sidewalk' 'Blocked Driveway' 'Illegal Parking'  
'Derelict Vehicle' 'Noise - Commercial' 'Noise - House of Worship'  
'Posting Advertisement' 'Noise - Vehicle' 'Animal Abuse' 'Vending'  
'Traffic' 'Drinking' 'Bike/Roller/Skate Chronic' 'Panhandling'  
'Noise - Park' 'Homeless Encampment' 'Urinating in Public' 'Graffiti'  
'Disorderly Youth' 'Illegal Fireworks' 'Ferry Complaint' 'Agency Issues'  
'Squeegee' 'Animal in a Park']  
Descriptor : ['Loud Music/Party' 'No Access' 'Commercial Overnight Parking'  
'Blocked Sidewalk' 'Posted Parking Sign Violation' 'Blocked Hydrant'  
'With License Plate' 'Partial Access' 'Unauthorized Bus Layover'  
'Double Parked Blocking Vehicle' 'Double Parked Blocking Traffic'  
'Vehicle' 'Loud Talking' 'Banging/Pounding' 'Car/Truck Music' 'Tortured'  
'In Prohibited Area' 'Congestion/Gridlock' 'Neglected' 'Car/Truck Horn'  
'In Public' 'Other (complaint details)' nan 'No Shelter'  
'Truck Route Violation' 'Unlicensed' 'Overnight Commercial Storage'  
'Engine Idling' 'After Hours - Licensed Est' 'Detached Trailer'  
'Underage - Licensed Est' 'Chronic Stoplight Violation' 'Loud Television'  
'Chained' 'Building' 'In Car' 'Police Report Requested'  
'Chronic Speeding' 'Playing in Unsuitable Place' 'Drag Racing'  
'Police Report Not Requested' 'Nuisance/Truant' 'Homeless Issue'  
'Language Access Complaint' 'Disruptive Passenger' 'Animal Waste']  
Location Type : ['Street/Sidewalk' 'Club/Bar/Restaurant' 'Store/Commercial'  
'House of Worship' 'Residential Building/House' 'Residential Building'  
'Park/Playground' 'Vacant Lot' 'House and Store' 'Highway' 'Commercial'  
'Roadway Tunnel' 'Subway Station' 'Parking Lot' 'Bridge' 'Terminal' nan  
'Ferry' 'Park']  
Incident Zip : [10034. 11105. 10458. 10461. 11373. 11215. 10032. 10457.  
11415. 11219.  
11372. 10453. 11208. 11379. 11374. 11412. 11217. 11234. 10026. 10456.  
10030. 10467. 11432. 10031. 11419. 10024. 11201. 11216. 10462. nan  
11385. 11414. 11213. 11375. 11211. 10312. 10017. 11417. 10002. 10027.  
11209. 10035. 11418. 11421. 11205. 10468. 11355. 11358. 11210. 11368.  
11427. 11436. 10308. 11364. 10011. 11423. 11230. 10003. 11221. 11416.  
11378. 11236. 11218. 10029. 10028. 11214. 11207. 11369. 11223. 11220.  
10302. 11420. 11354. 10473. 10301. 11103. 10465. 11377. 11212. 11365.  
10472. 10452. 11203. 10469. 11237. 11434. 11101. 10460. 11229. 11206.  
11102. 10466. 10009. 10033. 11694. 10022. 10470. 11433. 11428. 11413.  
10463. 10471. 10474. 11228. 10014. 10475. 11225. 11233. 11370. 11204.  
11435. 10459. 11238. 10304. 11367. 10306. 10305. 10001. 10314. 10019.  
11222. 10023. 11356. 11235. 10018. 10036. 11106. 10075. 10025. 10451.  
11366. 10005. 10303. 10455. 11361. 10016. 10309. 10013. 11226. 10012.  
11224. 11249. 10039. 10128. 10454. 10010. 11360. 11004. 11691. 10307.  
11232. 10038. 10310. 10040. 11426. 11362. 11411. 11429. 11422. 10007.  
10065. 10021. 10004. 11104. 11231. 11357. 11239. 11363. 10037. 11693.]

10280. 11430. 10464. 10006. 11692. 10044. 11001. 10282. 11371. 10281.  
 11109. 11040. 83. 10020. 10000. 11697. 11251. 10103. 10112. 10069.  
 11451. 10153. 10041. 11242. 10119. 10048. 10803. 11695. 10111. 10162.  
 10123. 11241.]

Incident Address : ['71 VERMILYEA AVENUE' '27-07 23 AVENUE' '2897 VALENT  
 INE AVENUE' ...  
 '77-03 79 PLACE' '81 HESTER STREET' '100-17 87 AVENUE']

Street Name : ['VERMILYEA AVENUE' '23 AVENUE' 'VALENTINE AVENUE' ... 'CR  
 AFTON AVENUE'  
 'LANDIS AVENUE' 'COOPER AVE']

Cross Street 1 : ['ACADEMY STREET' '27 STREET' 'EAST 198 STREET' ... '80  
 TH STREET' 'E 186'  
 'EAST 186']

Cross Street 2 : ['WEST 204 STREET' '28 STREET' 'EAST 199 STREET' ... 'B  
 ARNARD AVENUE'  
 'DEY STREET' 'GRAMERCY PARK']

Intersection Street 1 : [nan 'GRAND CENTRAL PARKWAY' 'DEVOE STREET' ...  
 'E 186' 'BAXTER AVENUE'  
 'east 186']

Intersection Street 2 : [nan '63 AVENUE' 'BUSHWICK AVENUE' ... 'MAYVILLE  
 STREET' 'STERLING AVENUE'  
 'LAWTON STREET']

Address Type : ['ADDRESS' nan 'INTERSECTION' 'LATLONG' 'BLOCKFACE' 'PLAC  
 ENAME']

City : ['NEW YORK' 'ASTORIA' 'BRONX' 'ELMHURST' 'BROOKLYN' 'KEW GARDENS'  
 'JACKSON HEIGHTS' 'MIDDLE VILLAGE' 'REGO PARK' 'SAINT ALBANS' 'JAMAICA'  
 'SOUTH RICHMOND HILL' nan 'RIDGEWOOD' 'HOWARD BEACH' 'FOREST HILLS'  
 'STATEN ISLAND' 'OZONE PARK' 'RICHMOND HILL' 'WOODHAVEN' 'FLUSHING'  
 'CORONA' 'QUEENS VILLAGE' 'OAKLAND GARDENS' 'HOLLIS' 'MASPETH'  
 'EAST ELMHURST' 'SOUTH OZONE PARK' 'WOODSIDE' 'FRESH MEADOWS'  
 'LONG ISLAND CITY' 'ROCKAWAY PARK' 'SPRINGFIELD GARDENS' 'COLLEGE POINT'  
 ,  
 'BAYSIDE' 'GLEN OAKS' 'FAR ROCKAWAY' 'BELLEROSE' 'LITTLE NECK'  
 'CAMBRIA HEIGHTS' 'ROSEDALE' 'SUNNYSIDE' 'WHITESTONE' 'ARVERNE'  
 'FLORAL PARK' 'NEW HYDE PARK' 'CENTRAL PARK' 'BREEZY POINT' 'QUEENS'  
 'Astoria' 'Long Island City' 'Woodside' 'East Elmhurst' 'Howard Beach']

Landmark : [nan 'MADISON SQUARE GARDEN' 'J F K AIRPORT' 'CLOVE LAKE' 'WI  
 SE TOWERS'  
 'FERRY TERMINAL' 'BRONX ZOO' 'UNION SQUARE PARK' 'SERRANO PLAYGROUND'  
 'WASHINGTON SQUARE PARK' 'CENTRAL PARK' 'BETH ISRAEL MED CENTER'  
 'BARCLAYS CENTER' 'LA GUARDIA AIRPORT' 'HIGHLAND PARK' 'WAGNER COLLEGE'  
 'WORLDS FAIR MARINA' 'RODMANS NECK' 'SUNSET PARK' 'UNITED NATIONS HQ'  
 'CONEY ISLAND HOSPITAL' 'WASHINGTON SQUARE' 'BRONX LEBANON HOSPITAL'  
 'MARCUS GARVEY MEMORIAL PARK' 'MANHATTAN COMMUNITY COLLEGE'  
 'BKLYN BATTERY TUNNEL' 'MARCUS GARVEY PARK'  
 'BKLYN PUBLIC LIBRARY CTR BRANCH' 'CITI FIELD' 'FLUSHING MEADOW PARK'  
 'LINCOLN TERRACE PARK' 'YANKEE STADIUM' 'MEADOW LAKE' 'MANHATTAN COLLEG  
 E'  
 'PROSPECT PARK' 'PS 174' 'MERCER PLAYGROUND' 'CITY HALL'  
 'BROOKLYN BRIDGE' 'PIER 94' 'LEIF ERICSON PARK' 'FORT GREENE PARK'  
 'GRAND ARMY PLAZA' 'METROPOLITAN MUSEUM' 'SI FERRY TERMINAL'  
 'MADISON SQUARE PARK' 'TRANSMITTER PARK' 'JOHN JAY PARK' 'SOUNDVIEW PAR  
 K'  
 'JOYCE KILMER PARK' 'QUEENSBRIDGE PARK' 'BROOKLYN MUSEUM' 'PIER 2'  
 'SOUTH STREET SEAPORT' 'COLUMBUS PARK' 'ST JOHN THE DIVINE'  
 'VAN CORTLAND PARK' 'WOODHULL MEDICAL CENTER' 'SEWARD PARK'  
 'EAST RIVER PARK' 'KOLBERT PARK' 'ST JOHNS UNIVERSITY' 'BARRY PARK'  
 'RAILROAD PARK' 'BRIGHTON BEACH' 'COOPER PARK' 'VALENTINO PARK'  
 'MORNINGSIDE PARK' 'ST MARYS PARK' 'VAN CORTLANDT PARK'  
 'CADMAN PLAZA PARK' 'JACKIE ROBINSON PARK' 'ASTORIA PARK' 'GAELIC PARK']



'PARKSIDE PLAYGROUND' '79 STREET BOAT BASIN' 'GRAND CENTRAL TERM'  
 'RIVERSIDE PARK' 'SOUTH BEACH' 'ROBERTO CLEMENTE STATE PARK BRDG'  
 'ROY WILKINS PARK' 'MARINE PARK' 'PELHAM BAY PARK' 'LINCOLN CENTER'  
 'MAURICE PARK' 'HAFFEN PARK' 'CONEY ISLAND BEACH' 'FORT TRYON PARK'  
 'PORT AUTH 42 STREET' 'COOPER UNION' 'TANAHEY PLAYGROUND'  
 'HAMILTON METZ FIELD' 'BATTERY PARK' 'CUYLER GORE' 'INWOOD HILL PARK'  
 'BETH ABRAHAM HOSPITAL' 'SEANS PLACE' 'BK BOTANIC GARDENS'  
 'JUMEL MANSION' 'POWELL COVE' 'CLAREMONT PARK' 'FORT INDEPENDENCE PARK'  
 'AQUEDUCT RACE TRACK' 'THE CLOISTERS' 'J HOOD WRIGHT PARK'  
 'LEHMAN COLLEGE' 'GREENWOOD PLAYGROUND' 'JUNIPER VALLEY PARK'  
 'MARIA HERNANDEZ PARK' 'JOHN JAY HS HISTORICAL' 'VICTORY FIELD'  
 'KINGS COUNTY HOSPITAL' 'MUSEUM NATURAL HIST' 'FLATIRON BUILDING'  
 'ROCKEFELLER CENTER' 'QUEENS HOSPITAL CENTER' 'INTREPID MUSEUM']  
 Facility Type : ['Precinct' nan]  
 Status : ['Closed' 'Open' 'Assigned' 'Draft']  
 Due Date : ['01-01-16 7:59' '01-01-16 7:57' '01-01-16 7:56' ...  
 '03/29/2015 08:33:03 AM' '03/29/2015 08:33:02 AM'  
 '03/29/2015 08:33:01 AM']  
 Resolution Description : ['The Police Department responded and upon arrival those responsible for the condition were gone.'  
 'The Police Department responded to the complaint and with the information available observed no evidence of the violation at that time.'  
 'The Police Department responded to the complaint and took action to fix the condition.'  
 'The Police Department issued a summons in response to the complaint.'  
 'The Police Department responded to the complaint and determined that police action was not necessary.'  
 'The Police Department reviewed your complaint and provided additional information below.'  
 'Your request can not be processed at this time because of insufficient contact information. Please create a new Service Request on NYC.gov and provide more detailed contact information.'  
 "This complaint does not fall under the Police Department's jurisdiction."  
 'Your complaint has been forwarded to the New York Police Department for a non-emergency response. 311 will have additional information in 8 hours. Please note your service request number for future reference.'  
 'The Police Department responded to the complaint and a report was prepared.'  
 'The Police Department responded to the complaint but officers were unable to gain entry into the premises.'  
 'The Police Department made an arrest in response to the complaint.'  
 "Your complaint has been forwarded to the New York Police Department for a non-emergency response. If the police determine the vehicle is illegally parked, they will ticket the vehicle and then you may either contact a private towing company to remove the vehicle or ask your local precinct to contact 'rotation tow'. Any fees charged for towing will have to be paid by the vehicle owner. 311 will have additional information in 8 hours. Please note your service request number for future reference."  
 'Your complaint has been received by the Police Department and it has been determined that a long-term investigation may be necessary. Additional information will be available at the conclusion of the investigation.'  
 'The Department of Transportation contacted the customer and resolved the Service Request or provided the information requested.'  
 'The New York City Police Department received your comments and forwarded them to the appropriate unit for resolution. You may follow up by calling (646) 610-6952 after 60 days from submitting your agency issue.'  
 'The Department of Transportation requires 30 days to respond to this type of complaint. Please note your Service Request number for future reference'

ference.'

'The condition was determined to be an issue appropriate for handling by an alternate entity. The Department of Parks and Recreation has notified the appropriate resource.']

Resolution Action Updated Date : ['01-01-16 0:55' '01-01-16 1:26' '01-01-16 4:51' ...

'03/29/2015 02:33:59 AM' '03/29/2015 04:38:35 AM'

'03/29/2015 04:41:50 AM']

Community Board : ['12 MANHATTAN' '01 QUEENS' '07 BRONX' '10 BRONX' '04 QUEENS'

'07 BROOKLYN' '03 BRONX' '09 QUEENS' '11 BROOKLYN' '03 QUEENS' '05 BRONX'

'05 BROOKLYN' '05 QUEENS' '06 QUEENS' '12 QUEENS' '06 BROOKLYN'

'18 BROOKLYN' '10 MANHATTAN' '04 BRONX' '12 BRONX' '09 MANHATTAN'

'10 QUEENS' '07 MANHATTAN' '02 BROOKLYN' '03 BROOKLYN' '11 BRONX'

'0 Unspecified' '08 BROOKLYN' '01 BROOKLYN' '03 STATEN ISLAND'

'06 MANHATTAN' '03 MANHATTAN' '10 BROOKLYN' '11 MANHATTAN' '07 QUEENS'

'11 QUEENS' '17 BROOKLYN' '13 QUEENS' '06 BRONX' '04 MANHATTAN'

'14 BROOKLYN' '12 BROOKLYN' '08 MANHATTAN' '13 BROOKLYN' '15 BROOKLYN'

'01 STATEN ISLAND' '09 BRONX' '02 STATEN ISLAND' '02 QUEENS'

'16 BROOKLYN' '08 QUEENS' '09 BROOKLYN' '04 BROOKLYN' '14 QUEENS'

'05 MANHATTAN' '08 BRONX' '02 BRONX' '02 MANHATTAN' '01 BRONX'

'01 MANHATTAN' '83 QUEENS' '95 STATEN ISLAND' '81 QUEENS' '27 BRONX'

'Unspecified STATEN ISLAND' '64 MANHATTAN' '80 QUEENS' '55 BROOKLYN'

'26 BRONX' '82 QUEENS' '56 BROOKLYN' '28 BRONX' 'Unspecified QUEENS'

'Unspecified MANHATTAN' '84 QUEENS']

Borough : ['MANHATTAN' 'QUEENS' 'BRONX' 'BROOKLYN' 'Unspecified' 'STATEN ISLAND']

X Coordinate (State Plane) : [1005409. 1007766. 1015081. ... 1003221. 1016065. 1016436.]

Y Coordinate (State Plane) : [254678. 221986. 256380. ... 178766. 222214. 222234.]

Park Facility Name : ['Unspecified' 'Alley Pond Park - Nature Center']

Park Borough : ['MANHATTAN' 'QUEENS' 'BRONX' 'BROOKLYN' 'Unspecified' 'STATEN ISLAND']

School Name : ['Unspecified' 'Alley Pond Park - Nature Center']

School Number : ['Unspecified' 'Q001']

School Region : ['Unspecified' nan]

School Code : ['Unspecified' nan]

School Phone Number : ['Unspecified' '7182176034']

School Address : ['Unspecified' 'Grand Central Parkway, near the soccer field']

School City : ['Unspecified' 'QUEENS']

School State : ['Unspecified' 'NY']

School Zip : ['Unspecified' nan]

School Not Found : ['N']

School or Citywide Complaint : [nan]

Vehicle Type : [nan]

Taxi Company Borough : [nan]

Taxi Pick Up Location : [nan]

Bridge Highway Name : [nan 'BQE/Gowanus Expwy' 'Grand Central Pkwy' 'Cross Bronx Expwy'

'Park Ave Tunnel - E 34th St./Grand Central' 'Long Island Expwy' 'FDR Dr'

'Prospect Expwy' 'Jackie Robinson/Interboro Pkwy' 'Staten Island Expwy'

'Hutchinson River Pkwy' 'Belt Pkwy' 'First Ave Tunnel - UN Plaza'

'Van Wyck Expwy' 'West Street' 'FDR Southbound' 'Battery Park Underpass'

,

'Harlem River Dr' 'Bronx River Pkwy' 'Richmond Pkwy/Korean War Vets'

'Nassau Expwy' 'Henry Hudson Pkwy/Rt 9A' 'Third Ave Br - Fifth St Basin'

```

'
'Major Deegan Expwy' 'Sheridan Expwy' 'Whitestone Expwy'
'Cross Island Pkwy' 'Bruckner Expwy' 'Third Ave Br - E 129th St'
'Clearview Expwy']
Bridge Highway Direction : [nan 'East/Queens Bound' 'West/Toward Triboro
ugh Br'
'North/Westbound (To GW Br)' 'West/Staten Island Bound' 'North Bound'
'West/Manhattan Bound' 'Northbound/Uptown' 'Westbound/To BQE' 'Westbound'
d'
'Westbound/To Goethals Br' 'North/Westchester County Bound'
'East/Brooklyn Bound' 'West/Brooklyn Bound' 'Southbound/Downtown'
'East/Long Island Bound' 'East/Bronx Bound' 'South/JFK Airport Bound'
'South Bound' 'Eastbound/To Ocean Pkwy' 'North/Bronx Bound'
'To West St/West Side' 'South/Downtown' 'To FDR/East Side' 'Eastbound'
'Northbound' 'South/Toward Triborough Br' 'Southbound'
'Southbound/To Triborough Br' 'South/East (To Throgs Neck Br)'
'Manhattan Bound' 'North/Eastbound' 'South/New Jersey Bound'
'South/Queens Bound' 'South/Long Island Bound']
Road Ramp : [nan 'Roadway' 'Ramp']
Bridge Highway Segment : [nan
'Brooklyn-Queens Expwy(I-278) (Exit 25) - Brooklyn-Battery Tunnel (I-47
8)'
'La Guardia Airport-Main Terminal (Exit 7) - La Guardia Airport/Steinwa
y St (Exit 5)'
'Whitestone Br (I-678) (Exit 6A) - Cross Bronx Expwy Extension (I-295)
(Exit 6B)'
'Meeker Ave/Morgan Ave (Exit 34)' 'Grand Central Pkwy (Exit 22 A-B)'
'38th St / 39th St (Exit 23) - Prospect Expwy (NY 27) (Exit 24)'
'East 96th St (Exit 14) - Triborough Br (Exit 17)'
'4th Ave (Exit 2B) - Hamilton Ave (Exit 2A)'
'Cypress Ave Vermont Place (Exit 2) - Bushwick Ave (Exit 1)'
'Clove Rd/Richmond Rd (Exit 13) - Todt Hill Road/Slosson Ave (Exit 12)'
'East Tremont Ave / Westchester Ave (Exit 2) - Pelham Pkwy (Exit 3E)'
'Victory Blvd (Exit 8) - Dr Martin Luther King Jr Expwy (NY 440) (Exit
9)'
'Van Wyck Expwy (I-678) (Exit 19) - Lefferts Blvd. (Exit 18)'
'East 42nd St (Exit 9) - East 61st St / East 63rd St (Exit 12)'
'Triborough Br (Exit 17) - East 116th St (Exit 16)'
'Cross Bay Blvd (Exit 17 S-N) - Nassau Expwy (NY 878)/Van Wyck Expwy (I
-678) (Exit 19)'
'Roosevelt Ave / Broadway (Exit 37) - Northern Blvd (NY 25A) (Exit 38)'
'McGuinness Blvd / Humboldt St (Exit 33) - Kosciuszko Br (to Queens)'
'Hillside Ave/Jamaica Ave (Exit 6)'
'Jackie Robinson Pkwy (Exit 7) - Hillside Ave/Jamaica Ave (Exit 6)'
'East 96th St (Exit 14) - East 71st St (Exit 13)'
'BEGIN Staten Island Expwy (Exit 15N) - Lily Pond Ave/Bay St (Exit 15
S)'
'31st St (Exit 3)' 'Clove Rd/Richmond Rd (Exit 13)'
'END FDR Dr/Battery Park Underpass'
'10Ave/11Ave (Exit 4) - Park Circle (Exit 5)'
'Flushing Meadows-Corona Park (Exit 9P) - Whitestone Expwy (I-678) (Exi
t 9E)'
'Nassau Expwy (NY 878) (Exit 1A) - Belt Pkwy (1 E-W)'
'E 125th St 2nd Ave (Exit 19)'
'Flushing Ave (Exit 30) - Williamsburg Br / Metropolitan Ave (Exit 32)'
'East 71st St (Exit 13) - East 61st St / East 63rd St (Exit 12)'
'BEGIN Staten Island Expwy - Forest Ave (Exit 4)'
'Ocean Pkwy (Exit 7 S-N) - Coney Island Ave (Exit 8)'
'East 61st St / East 63rd St (Exit 12) - East 96th St (Exit 14)'
'Forest Ave (Exit 4) - West Shore Expwy (NY 440) (Exit 5)'

```

'East 71st St (Exit 13)' 'East 20th St/East 23rd St (Exit 7)'  
 'Rockaway Pkwy (Exit 13) - Flatbush Ave (Exit 11N)'  
 'Entrance Bronx River Pkwy'  
 'Cypress Hills St (Exit 3) - Forest Park Dr (Exit 4)'  
 'Arden Ave - Richmond Ave'  
 'Houston St (Exit 5) - East 20th St / East 23rd St (Exit 7)'  
 'Castle Hill Ave (Exit 5B) - Bronx-Whitestone Br (I-678) (Exit 6A)'  
 'JFK Expwy (Exit 2S)- JFK Expwy (Exit 2N)'  
 'Flatbush Ave (Exit 11N) - Flatbush Ave (Exit 11S)'  
 'W 57 St - W 72 St (Exit 9)'  
 'Gateway National Recreation Area (Exit 12) - Rockaway Pkwy (Exit 13)'  
 'East 106th St (Exit 15) - East 96th St (Exit 14)'  
 'Grand Central Pkwy (Exit 22 A-B) - Junction Blvd (Exit 20)' 'Roadway'  
 'Kissena Blvd (Exit 24)'  
 'Jerome Ave (Exit 2A) - Webster Ave (US 1) (Exit 2B)'  
 'Metropolitan Ave (Exit 6) -Van Wyck Expwy (I-678) (Exit 7)'  
 'Bronx River Pkwy (Exit 4B) - Westchester Ave / White Plains Road (Exit 5A)'  
 'Cross Bronx / GWB (Exit 14) - Riverside Dr (Exit 15)'  
 'Grand Central Pkwy (Split) (Exit 39)'  
 'Brooklyn-Queens Expwy (I-278) (Exit 4) - La Guardia Airport/Astoria Blvd. (Exit 5)'  
 'East 96th St (Exit 14)'  
 'Westchester Ave / White Plains Road (Exit 5A) - Castle Hill Ave (Exit 5B)'  
 'E 161st (Exit 6) - E 155th St/W 153rd St (Exit 5)'  
 'Bradley Ave (Exit 11) - Todt Hill Road/Slosson Ave (Exit 12)'  
 'Richmond Ave (Exit 7) - Victory Blvd (Exit 8)'  
 'Entrance ramp from George Washington Bridge'  
 'BEGIN Sheridan Expwy - Westchester Ave'  
 'Bronx River Pkwy/Westchester County'  
 '92nd St (From Lower Level) (Exit 17)' 'Entrance Staten Island Expwy'  
 'Triborough Bridge Connector (I-278) (Exit 17)'  
 '78th Ave Queens Blvd (NY 25) (Exit 12) - Van Wyck Expwy (I-678) (Exit 13)'  
 'Triborough Bridge (Exit 17)'  
 'Amsterdam Ave. W. 179th St. / George Washington Br (Exit 24) - Dyckman St 10th Ave'  
 'Van Cortlandt Park South (Exit 11) - W 230th St (Exit 10)'  
 '20th Ave (Exit 15) - Cross Island Pkwy (Exit 16)'  
 'South St (Exit 1) - Battery Park Underpass'  
 'Throgs Neck Br/Pennyfield Ave (Exit 8) - Throgs Neck Expwy (I-695) (Exit 10)'  
 'South St (Exit 3) - Pearl St / Brooklyn Br (Exit 2)'  
 'Crospey Ave Stillwell Ave (Exit 6N) - Crospey Ave Stillwell Ave (Exit 6S)'  
 'Belt Pkwy (1 E-W) - Rockaway Blvd (Exit 2)'  
 'East 61st St/East 63rd St (Exit 12)'  
 'Northern Blvd (NY 25A) (Exit 31E) - Northern Blvd (NY 25A) (Exit 31W)'  
 '3rd Ave (Exit 21)'  
 'Sheridan Expwy (I-895) (Exit 4A) - Bronx River Pkwy (Exit 4B)'  
 'Westchester Ave (Exit 8A) - East Tremont Ave (Exit 7B)'  
 'BEGIN Cross Bronx Expwy (Extension) - Randall Ave (Exit 11)'  
 'Springfield Blvd. (Exit 22) - Sunrise Highway(NY 27) (Exit 23)'  
 '8Ave/7Ave (Exit 3) - 4th Ave (Exit 2B)' 'Entrance Belt Pkwy'  
 '7th Ave/65th St (Exit 20) - 3rd Ave (Exit 21)' 'Entrance FDR Dr'  
 'Webster Ave (US 1) (Exit 2B) - Sheridan Expwy (I-895) (Exit 4A)' 'Ramp'  
 'Linden Place (Exit 14) - 20th Ave (Exit 15)'  
 'Wythe Ave / Kent Ave (Exit 31) - Tillary St /Manhattan Br (Exit 29)'

'Begin Brooklyn-Queens Expwy'  
 'Boston Post Road (US 1N) (Exit 6) - East 177th St / Sheridan Expwy (I-895) (Exit 5)'  
 'Flatbush Ave (Exit 11S) - Flatbush Ave (Exit 11N)'  
 'Cross Bay Blvd (Exit 17N) - Cross Bay Blvd (Exit 17S)'  
 'Clove Rd/Richmond Rd (Exit 13) - Lily Pond Ave/Bay St (Exit 15 S-N)'  
 'Westchester Ave - END Sheridan Expwy (Bruckner Expwy) (I-278)'  
 'Todt Hill Road/Slosson Ave (Exit 12) - Bradley Ave (Exit 11)'  
 'W. Fordham Rd (Exit 9) - I95 / US 1 (Exit 7 N-S)'  
 'Van Wyck Service Roads - Federal Circle'  
 'Dr Martin Luther King Jr Expwy (NY 440) (Exit 9) - Bradley Ave (Exit 11)'  
 'Queens Blvd (NY 25) / Woodhaven Blvd (Exit 19) - 108th St (Exit 21)'  
 'Hoyt Ave (Exit 3) - Triboro Br' 'Maurice Ave (Exit 18)'  
 '30th Ave (Exit 40) West Leg - Astoria Blvd (Exit 41) West Leg'  
 'Francis Lewis Blvd. (Exit 24A) - 130th Ave. (Exit 24B)'  
 'JFK Expwy (Exit 20) - Van Wyck Expwy (I-678) (Exit 19)'  
 'Clearview Expwy (I-295) (Exit 21 S-N) - Union Turnpike (Exit 22)'  
 'South St (Exit 1)' 'Erskine St (Exit 15) - Pennsylvania Ave (Exit 14)'  
 'Crospey Ave Stillwell Ave (Exit 6 S-N) - Ocean Pkwy (Exit 7 S-N)'  
 'W 79 St (Exit 10) - W 57 St'  
 'Knapp St (Exits 9 A-B) - Coney Island Ave (Exit 8)'  
 'Bay Pkwy (Exit 5) - Bay 8th St 14th Ave (Exit 4)'  
 'Northern Blvd (Exit 5) - 35th Ave (Exit 6A)' 'Rockaway Pkwy (Exit 13)'  
 'N. Conduit Ave. (NY27) (Exit 17W) - Cross Bay Blvd (Exit 17N)'  
 'Castle Hill Ave (Exit 5B)' 'W 79 St (Exit 10) - W 96 St (Exit 11)'  
 'Bronx River Pkwy (Exit 48 S-N) - White Plains Road / Westchester Ave (Exit 49)'  
 'Cross Island Pkwy (Exit 30S) - Little Neck Pkwy (Exit 32)'  
 '150th St (Exit 20) - Farmers Blvd. Guy R. Brewer Blvd. (Exit 21B)'  
 'Begin Korean War Vets Pkwy/Arthur Kill Rd - Arden Ave'  
 '130th Ave. (Exit 24B) - Southern State Pkwy (Exit 25A)'  
 'East 177th St/Sheridan Expwy (I-895) (Exit 5)'  
 'End Major Deegan Expwy/Westchester'  
 'Lafayette Ave - END Hutchinson River Pkwy' 'Bushwick Ave (Exit 1)'  
 'I95/US 1 (Exit 7 N-S)' '6th Ave/65th St (Exit 20)'  
 '6th Ave (Exit 20) - 86th St (Exit 19)'  
 'Southern State Pkwy (Exit 25A) - END Cross Island Pkwy'  
 'Clearview Expwy (I-295) (Exit 27 S-N) - Utopia Pkwy (Exit 25)'  
 'Van Dam St (Exit 15) - Queens Midtown Tunnel'  
 'Flatbush Ave (Exit 11N) - Gateway National Recreation Area (Exit 12)'  
 'South Ave (Exit 6) - West Shore Expwy (NY 440) (Exit 5)'  
 'Brooklyn-Queens Expwy (I-278) (Exit 17 E-W) - Queens Blvd (NY 25) / Woodhaven Blvd (Exit 19)'  
 'Brooklyn-Queens Expwy (I-278) (Exit 17W) - Greenpoint Ave (Exit 16)'  
 'Hamilton Ave (Exit 26) - Atlantic Ave (Exit 27)'  
 'East 61st St / East 63rd St (Exit 12) - East 53rd St (Exit 11)'  
 '92nd St (Exit 17) - Fort Hamilton Pkwy (Exit 18)'  
 'Entrance Major Deegan Expwy' 'W 125 St (Exit 12) - W 95 St (Exit 11)'  
 'W 237th St/W 239th St (Exit 20) - W 246th St/W 250th St (Exit 21)'  
 'Belt Pkwy - 65th St - 67th St (Exit 1)'  
 'Westchester Ave/White Plains Road (Exit 5A)'  
 '30th Ave (From East Leg) (Exit 40)'  
 'Brooklyn Br (Exit 28B) - Manhattan Br (Exit 29A)'  
 'Van Wyck Expwy (I-678) (Exit 7)'  
 'Hamilton Ave (Exit 2A) - Gowanus Expwy (I-278) (Exit 1)'  
 'East 233rd St (Exit 10) - Westchester County'  
 'W 230th (Exit 10) - Van Cortlandt Park South (Exit 11)'  
 'END Brooklyn-Queens Expwy' '38th St/39th St (Exit 23)'  
 'W 125 St (Exit 12) - W 158 St (Exit 13)'

```

'Queens Blvd (NY 25)/Woodhaven Blvd (Exit 19)'
'Wythe Ave/Kent Ave (Exit 31)'
'20th Ave (Exit 15) - Linden Place (Exit 14)']
Garage Lot Name : [nan]
Ferry Direction : [nan 'Manhattan Bound']
Ferry Terminal Name : [nan 'St. George Terminal (Staten Island)' 'Barber
i']
Latitude : [40.86568154 40.77594531 40.87032452 ... 40.77664592 40.70635
259
40.71605291]
Longitude : [-73.92350096 -73.91509394 -73.88852464 ... -73.94880526 -7
3.87124456
-73.9913785 ]
Location : ['(40.86568153633767, -73.92350095571744)'
'(40.775945312321085, -73.91509393898605)'
'(40.870324522111424, -73.88852464418646)' ...
'(40.77664591586459, -73.94880525662063)']

```

In [9]: data.columns

```

Out[9]: Index(['Unique Key', 'Created Date', 'Closed Date', 'Agency', 'Agency Na
me',
      'Complaint Type', 'Descriptor', 'Location Type', 'Incident Zip',
      'Incident Address', 'Street Name', 'Cross Street 1', 'Cross Stree
t 2',
      'Intersection Street 1', 'Intersection Street 2', 'Address Type',
      'City', 'Landmark', 'Facility Type', 'Status', 'Due Date',
      'Resolution Description', 'Resolution Action Updated Date',
      'Community Board', 'Borough', 'X Coordinate (State Plane)',
      'Y Coordinate (State Plane)', 'Park Facility Name', 'Park Borough
',
      'School Name', 'School Number', 'School Region', 'School Code',
      'School Phone Number', 'School Address', 'School City', 'School S
tate',
      'School Zip', 'School Not Found', 'School or Citywide Complaint',
      'Vehicle Type', 'Taxi Company Borough', 'Taxi Pick Up Location',
      'Bridge Highway Name', 'Bridge Highway Direction', 'Road Ramp',
      'Bridge Highway Segment', 'Garage Lot Name', 'Ferry Direction',
      'Ferry Terminal Name', 'Latitude', 'Longitude', 'Location'],
      dtype='object')

```

```

In [10]: data = data.drop(['Unique Key', 'School Region', 'School Code', 'School Zip
', 'School Not Found',
      'School or Citywide Complaint', 'Vehicle Type', 'Taxi Com
pany Borough', 'Taxi Pick Up Location', 'Garage Lot Name'], axis=1)

```

In [11]: data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300698 entries, 0 to 300697
Data columns (total 43 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Created Date                             300698 non-null object
1   Closed Date                              298534 non-null object
2   Agency                                   300698 non-null object
3   Agency Name                             300698 non-null object
4   Complaint Type                           300698 non-null object
5   Descriptor                               294784 non-null object
6   Location Type                             300567 non-null object
7   Incident Zip                             298083 non-null float64
8   Incident Address                         256288 non-null object
9   Street Name                              256288 non-null object
10  Cross Street 1                           251419 non-null object
11  Cross Street 2                           250919 non-null object
12  Intersection Street 1                     43858 non-null object
13  Intersection Street 2                     43362 non-null object
14  Address Type                             297883 non-null object
15  City                                     298084 non-null object
16  Landmark                                 349 non-null object
17  Facility Type                             298527 non-null object
18  Status                                   300698 non-null object
19  Due Date                                 300695 non-null object
20  Resolution Description                    300698 non-null object
21  Resolution Action Updated Date            298511 non-null object
22  Community Board                          300698 non-null object
23  Borough                                  300698 non-null object
24  X Coordinate (State Plane)                297158 non-null float64
25  Y Coordinate (State Plane)                297158 non-null float64
26  Park Facility Name                        300698 non-null object
27  Park Borough                             300698 non-null object
28  School Name                              300698 non-null object
29  School Number                            300698 non-null object
30  School Phone Number                      300698 non-null object
31  School Address                           300698 non-null object
32  School City                              300698 non-null object
33  School State                             300698 non-null object
34  Bridge Highway Name                       243 non-null object
35  Bridge Highway Direction                  243 non-null object
36  Road Ramp                                213 non-null object
37  Bridge Highway Segment                    213 non-null object
38  Ferry Direction                           1 non-null object
39  Ferry Terminal Name                       2 non-null object
40  Latitude                                  297158 non-null float64
41  Longitude                                 297158 non-null float64
42  Location                                  297158 non-null object
dtypes: float64(5), object(38)
memory usage: 98.6+ MB
```

2. Read or convert the columns 'Created Date' and 'Closed Date' to datetime datatype and create a new column 'Request\_Closing\_Time' as the time elapsed between request creation and request closing. (Hint: Explore the package/module datetime)

```
In [12]: data['Created Date'] = pd.to_datetime(data['Created Date'])
data['Closed Date'] = pd.to_datetime(data['Closed Date'])
```

```
In [13]: data['Created Date']
```

```
Out[13]: 0          2015-12-31 23:59:45
1          2015-12-31 23:59:44
2          2015-12-31 23:59:29
3          2015-12-31 23:57:46
4          2015-12-31 23:56:58
...
300693     2015-03-29 00:33:41
300694     2015-03-29 00:33:28
300695     2015-03-29 00:33:03
300696     2015-03-29 00:33:02
300697     2015-03-29 00:33:01
Name: Created Date, Length: 300698, dtype: datetime64[ns]
```

```
In [14]: data['Closed Date']
```

```
Out[14]: 0          2016-01-01 00:55:00
1          2016-01-01 01:26:00
2          2016-01-01 04:51:00
3          2016-01-01 07:43:00
4          2016-01-01 03:24:00
...
300693          NaT
300694     2015-03-29 02:33:59
300695     2015-03-29 03:40:20
300696     2015-03-29 04:38:35
300697     2015-03-29 04:41:50
Name: Closed Date, Length: 300698, dtype: datetime64[ns]
```

```
In [15]: data['Request_Closing_Time'] = data['Closed Date'] - data['Created Date']
```



In [16]: `data.sample(5)`

Out[16]:

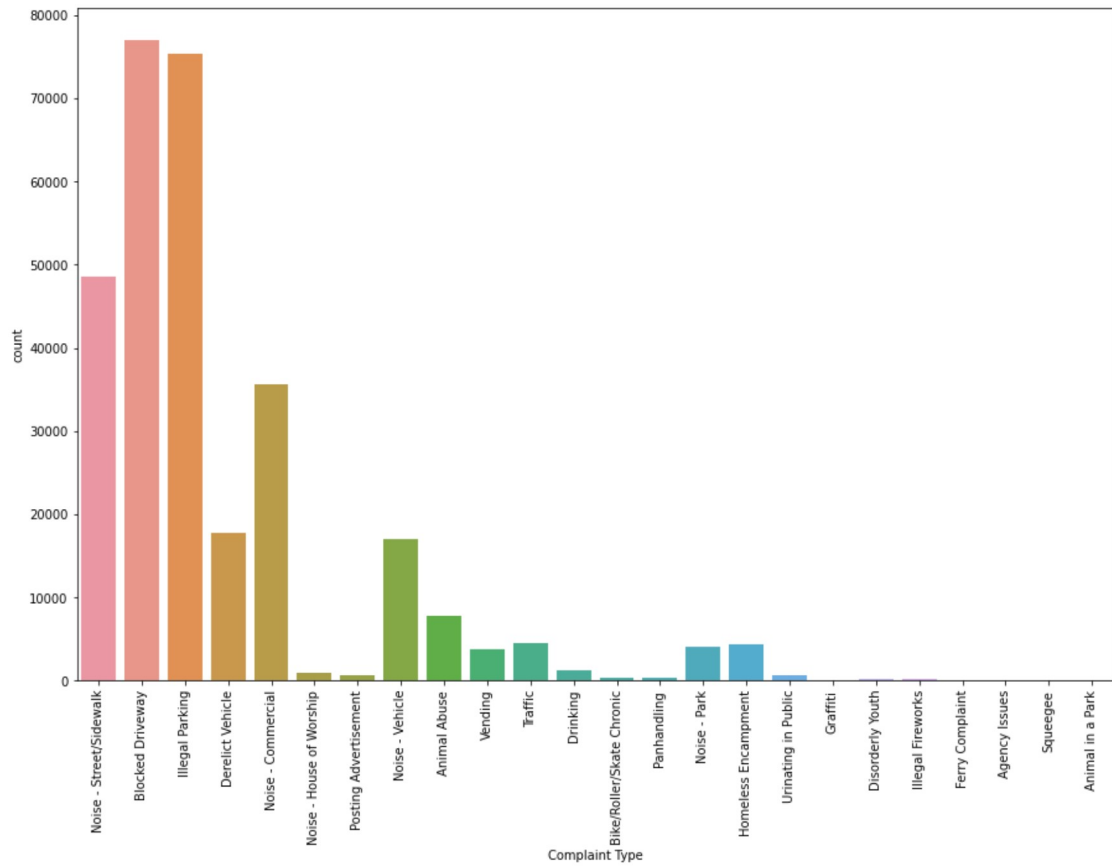
	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Ty
20113	2015-12-11 09:00:00	2015-12-11 20:41:00	NYPD	New York City Police Department	Blocked Driveway	Partial Access	Street/Sidew
174548	2015-07-22 22:51:25	2015-07-23 02:21:53	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Store/Commer
111348	2015-09-16 20:58:35	2015-09-16 22:56:45	NYPD	New York City Police Department	Noise - Park	Loud Music/Party	Park/Playgrou
267299	2015-05-04 07:56:00	2015-05-04 23:26:00	NYPD	New York City Police Department	Illegal Parking	Posted Parking Sign Violation	Street/Sidew
65127	2015-10-28 11:03:47	2015-10-28 13:03:56	NYPD	New York City Police Department	Blocked Driveway	Partial Access	Street/Sidew

In [17]: `data.columns`

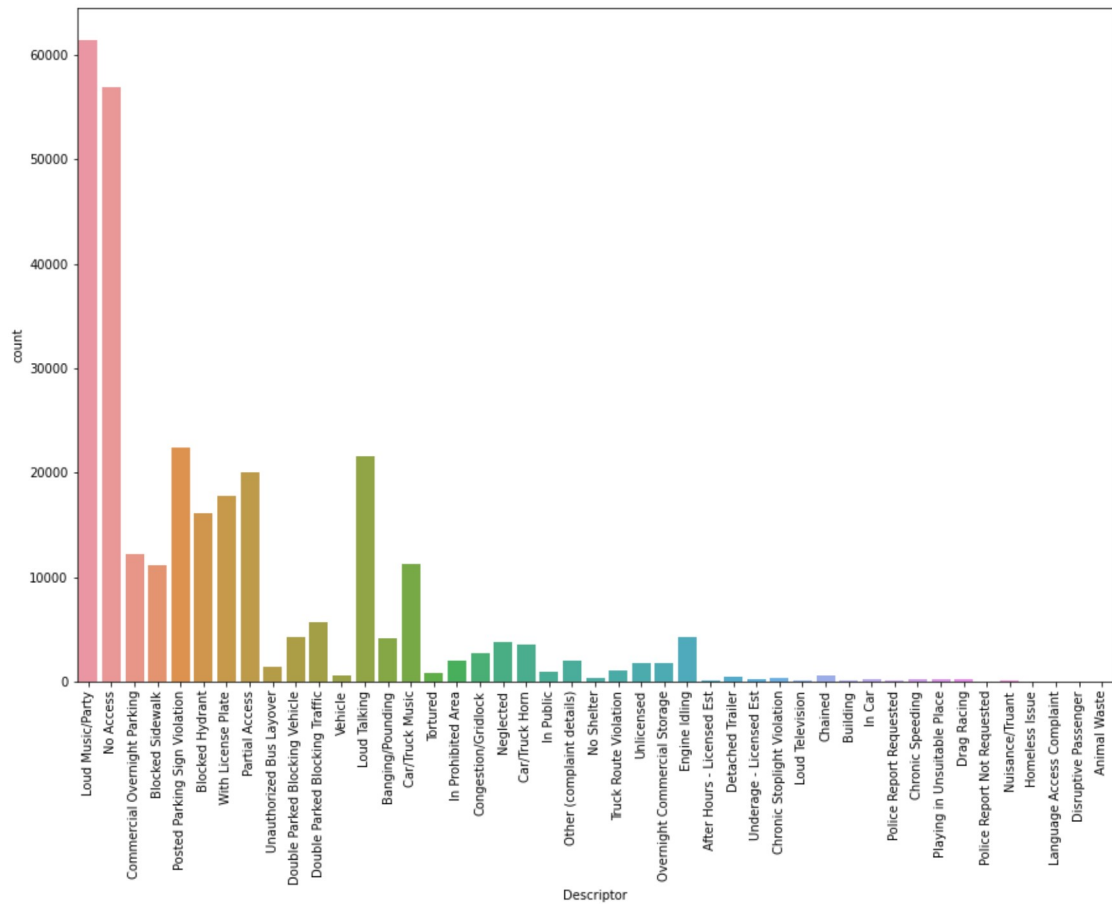
Out[17]: Index(['Created Date', 'Closed Date', 'Agency', 'Agency Name', 'Complaint Type', 'Descriptor', 'Location Type', 'Incident Zip', 'Incident Address', 'Street Name', 'Cross Street 1', 'Cross Street 2', 'Intersection Street 1', 'Intersection Street 2', 'Address Type', 'City', 'Landmark', 'Facility Type', 'Status', 'Due Date', 'Resolution Description', 'Resolution Action Updated Date', 'Community Board', 'Borough', 'X Coordinate (State Plane)', 'Y Coordinate (State Plane)', 'Park Facility Name', 'Park Borough', 'School Name', 'School Number', 'School Phone Number', 'School Address', 'School City', 'School State', 'Bridge Highway Name', 'Bridge Highway Direction', 'Road Ramp', 'Bridge Highway Segment', 'Ferry Direction', 'Ferry Terminal Name', 'Latitude', 'Longitude', 'Location', 'Request\_Closing\_Time'], dtype='object')

3. Provide major insights/patterns that you can offer in a visual format (graphs or tables); at least 4 major conclusions that you can come up with after generic data mining.

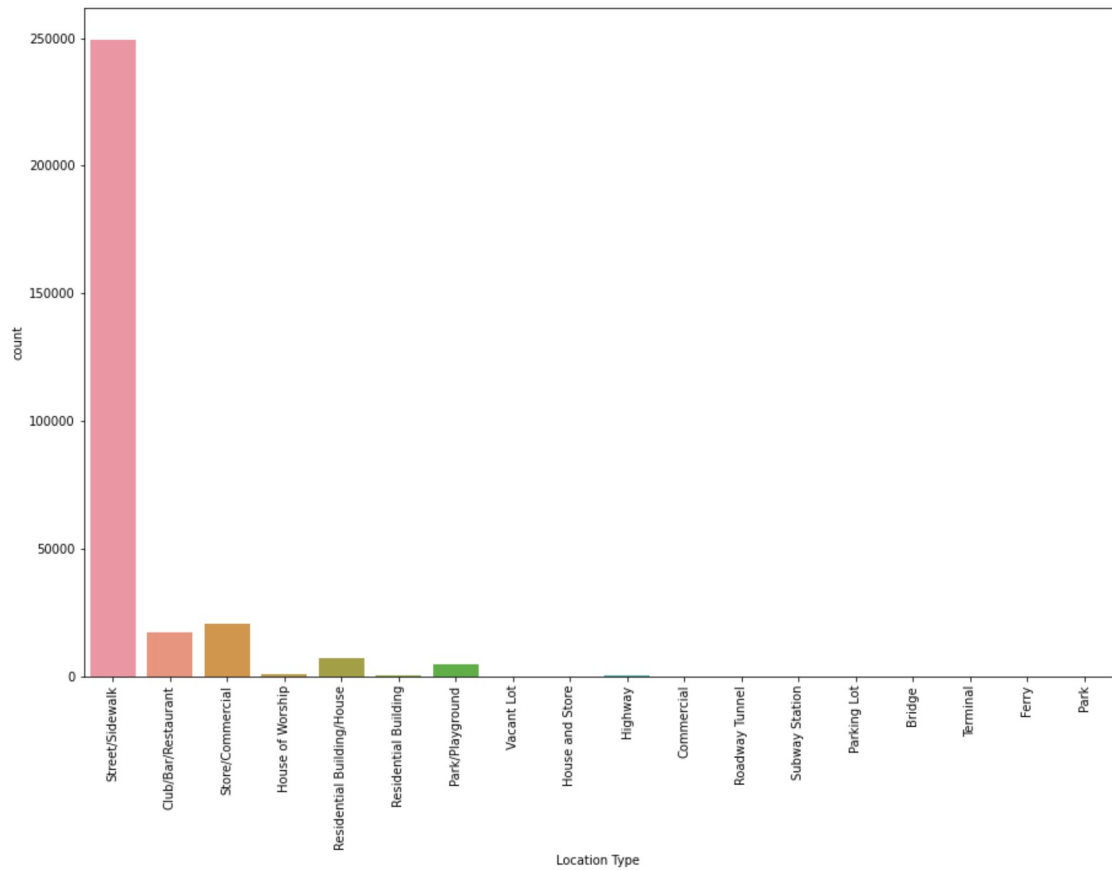
```
In [18]: plt.figure(figsize=(15,10))
sns.countplot(x='Complaint Type',data=data)
plt.xticks(x=data['Complaint Type'],rotation='vertical')
plt.show()
```



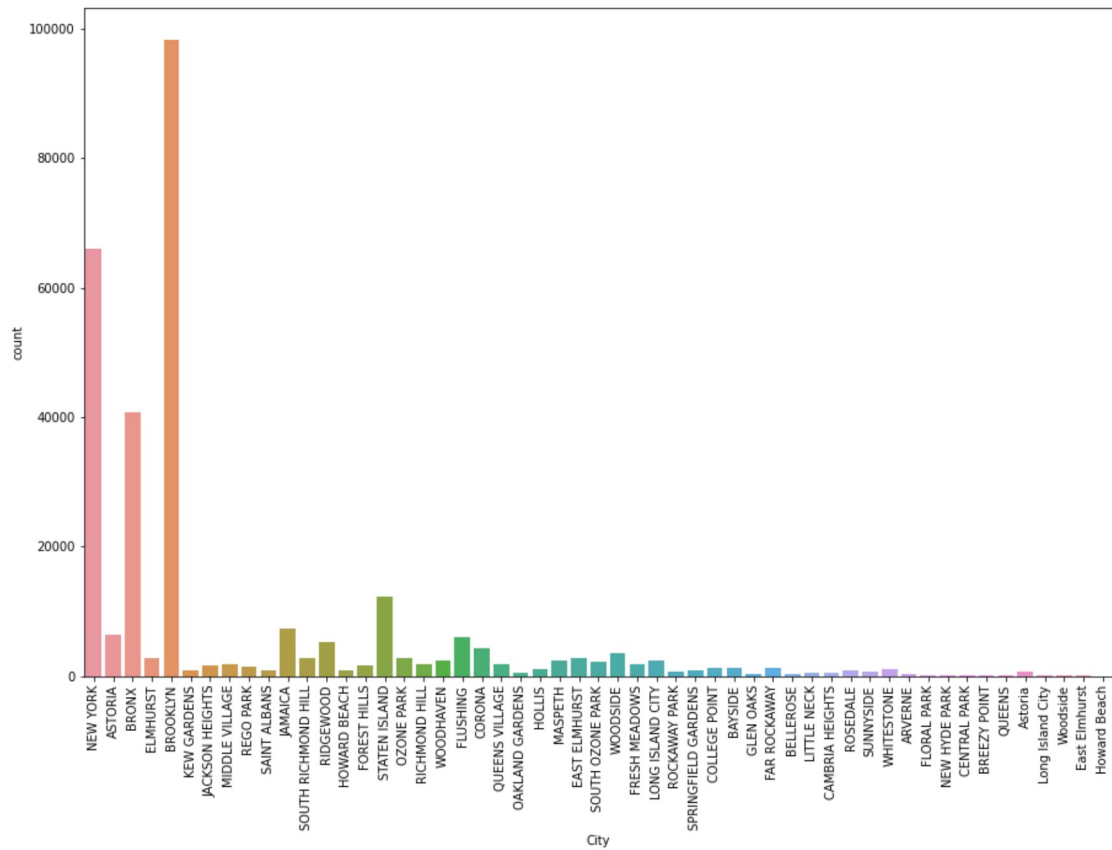
```
In [19]: plt.figure(figsize=(15,10))
sns.countplot(x='Descriptor',data=data)
plt.xticks(x=data['Descriptor'],rotation='vertical')
plt.show()
```



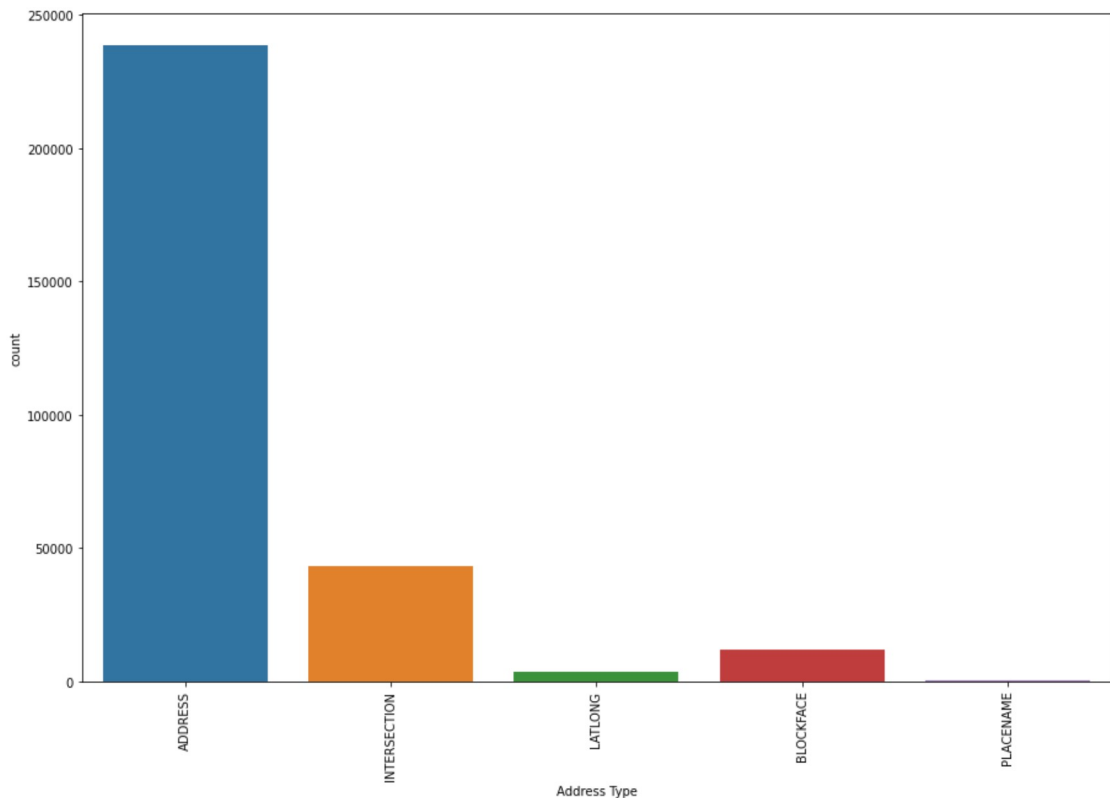
```
In [20]: plt.figure(figsize=(15,10))
sns.countplot(x='Location Type',data=data)
plt.xticks(x=data['Location Type'],rotation='vertical')
plt.show()
```



```
In [21]: plt.figure(figsize=(15,10))
sns.countplot(x='City',data=data)
plt.xticks(x=data['City'],rotation='vertical')
plt.show()
```



```
In [22]: plt.figure(figsize=(15,10))
sns.countplot(x='Address Type',data=data)
plt.xticks(x=data['Address Type'],rotation='vertical')
plt.show()
```



```
In [23]: complaint = data['Complaint Type'].value_counts()
complaint
```

```
Out[23]: Blocked Driveway          77044
Illegal Parking          75361
Noise - Street/Sidewalk  48612
Noise - Commercial      35577
Derelict Vehicle         17718
Noise - Vehicle          17083
Animal Abuse             7778
Traffic                  4498
Homeless Encampment      4416
Noise - Park             4042
Vending                  3802
Drinking                 1280
Noise - House of Worship  931
Posting Advertisement     650
Urinating in Public       592
Bike/Roller/Skate Chronic  427
Panhandling               307
Disorderly Youth          286
Illegal Fireworks         168
Graffiti                 113
Agency Issues             6
Squeegee                  4
Ferry Complaint           2
Animal in a Park          1
Name: Complaint Type, dtype: int64
```

```
In [24]: complaint = complaint.to_frame()  
complaint = complaint.rename(columns={'Complaint Type': 'Counts'})  
complaint
```

Out[24]:

	Counts
<b>Blocked Driveway</b>	77044
<b>Illegal Parking</b>	75361
<b>Noise - Street/Sidewalk</b>	48612
<b>Noise - Commercial</b>	35577
<b>Derelict Vehicle</b>	17718
<b>Noise - Vehicle</b>	17083
<b>Animal Abuse</b>	7778
<b>Traffic</b>	4498
<b>Homeless Encampment</b>	4416
<b>Noise - Park</b>	4042
<b>Vending</b>	3802
<b>Drinking</b>	1280
<b>Noise - House of Worship</b>	931
<b>Posting Advertisement</b>	650
<b>Urinating in Public</b>	592
<b>Bike/Roller/Skate Chronic</b>	427
<b>Panhandling</b>	307
<b>Disorderly Youth</b>	286
<b>Illegal Fireworks</b>	168
<b>Graffiti</b>	113
<b>Agency Issues</b>	6
<b>Squeegee</b>	4
<b>Ferry Complaint</b>	2
<b>Animal in a Park</b>	1

```
In [25]: complaint = complaint.reset_index()  
complaint = complaint.rename(columns={'index':'Complaint Type'})  
complaint
```

Out[25]:

	Complaint Type	Counts
0	Blocked Driveway	77044
1	Illegal Parking	75361
2	Noise - Street/Sidewalk	48612
3	Noise - Commercial	35577
4	Derelict Vehicle	17718
5	Noise - Vehicle	17083
6	Animal Abuse	7778
7	Traffic	4498
8	Homeless Encampment	4416
9	Noise - Park	4042
10	Vending	3802
11	Drinking	1280
12	Noise - House of Worship	931
13	Posting Advertisement	650
14	Urinating in Public	592
15	Bike/Roller/Skate Chronic	427
16	Panhandling	307
17	Disorderly Youth	286
18	Illegal Fireworks	168
19	Graffiti	113
20	Agency Issues	6
21	Squeegee	4
22	Ferry Complaint	2
23	Animal in a Park	1



```
In [26]: complaint['percentage'] = np.around((complaint.Counts/complaint.Counts.sum())*100,decimals=1)
complaint
```

Out[26]:

	Complaint Type	Counts	percentage
0	Blocked Driveway	77044	25.6
1	Illegal Parking	75361	25.1
2	Noise - Street/Sidewalk	48612	16.2
3	Noise - Commercial	35577	11.8
4	Derelict Vehicle	17718	5.9
5	Noise - Vehicle	17083	5.7
6	Animal Abuse	7778	2.6
7	Traffic	4498	1.5
8	Homeless Encampment	4416	1.5
9	Noise - Park	4042	1.3
10	Vending	3802	1.3
11	Drinking	1280	0.4
12	Noise - House of Worship	931	0.3
13	Posting Advertisement	650	0.2
14	Urinating in Public	592	0.2
15	Bike/Roller/Skate Chronic	427	0.1
16	Panhandling	307	0.1
17	Disorderly Youth	286	0.1
18	Illegal Fireworks	168	0.1
19	Graffiti	113	0.0
20	Agency Issues	6	0.0
21	Squeegee	4	0.0
22	Ferry Complaint	2	0.0
23	Animal in a Park	1	0.0

```
In [27]: complaint = complaint[complaint.percentage > 1]
complaint
```

Out[27]:

	Complaint Type	Counts	percentage
0	Blocked Driveway	77044	25.6
1	Illegal Parking	75361	25.1
2	Noise - Street/Sidewalk	48612	16.2
3	Noise - Commercial	35577	11.8
4	Derelict Vehicle	17718	5.9
5	Noise - Vehicle	17083	5.7
6	Animal Abuse	7778	2.6
7	Traffic	4498	1.5
8	Homeless Encampment	4416	1.5
9	Noise - Park	4042	1.3
10	Vending	3802	1.3

```
In [28]: descriptor = data['Descriptor'].value_counts()  
descriptor
```

```
Out[28]: Loud Music/Party          61430  
No Access          56976  
Posted Parking Sign Violation  22440  
Loud Talking       21584  
Partial Access     20068  
With License Plate 17718  
Blocked Hydrant    16081  
Commercial Overnight Parking  12189  
Car/Truck Music    11273  
Blocked Sidewalk   11121  
Double Parked Blocking Traffic  5731  
Double Parked Blocking Vehicle  4211  
Engine Idling      4189  
Banging/Pounding   4165  
Neglected         3787  
Car/Truck Horn     3511  
Congestion/Gridlock 2761  
In Prohibited Area 2025  
Other (complaint details) 1969  
Unlicensed         1777  
Overnight Commercial Storage  1757  
Unauthorized Bus Layover  1367  
Truck Route Violation 1014  
In Public          932  
Tortured           854  
Vehicle            590  
Chained            535  
Detached Trailer   464  
No Shelter         382  
Chronic Stoplight Violation  280  
Underage - Licensed Est  271  
Chronic Speeding   268  
In Car             251  
Playing in Unsuitable Place  245  
Drag Racing        175  
Loud Television     93  
Police Report Requested  90  
After Hours - Licensed Est  77  
Building           60  
Nuisance/Truant    41  
Police Report Not Requested  23  
Language Access Complaint  6  
Homeless Issue     1  
Disruptive Passenger 1  
Animal Waste       1  
Name: Descriptor, dtype: int64
```

```
In [29]: descriptor = descriptor.to_frame()  
descriptor = descriptor.rename(columns={'Descriptor': 'Counts'})  
descriptor
```

Out[29]:

	Counts
<b>Loud Music/Party</b>	61430
<b>No Access</b>	56976
<b>Posted Parking Sign Violation</b>	22440
<b>Loud Talking</b>	21584
<b>Partial Access</b>	20068
<b>With License Plate</b>	17718
<b>Blocked Hydrant</b>	16081
<b>Commercial Overnight Parking</b>	12189
<b>Car/Truck Music</b>	11273
<b>Blocked Sidewalk</b>	11121
<b>Double Parked Blocking Traffic</b>	5731
<b>Double Parked Blocking Vehicle</b>	4211
<b>Engine Idling</b>	4189
<b>Banging/Pounding</b>	4165
<b>Neglected</b>	3787
<b>Car/Truck Horn</b>	3511
<b>Congestion/Gridlock</b>	2761
<b>In Prohibited Area</b>	2025
<b>Other (complaint details)</b>	1969
<b>Unlicensed</b>	1777
<b>Overnight Commercial Storage</b>	1757
<b>Unauthorized Bus Layover</b>	1367
<b>Truck Route Violation</b>	1014
<b>In Public</b>	932
<b>Tortured</b>	854
<b>Vehicle</b>	590
<b>Chained</b>	535
<b>Detached Trailer</b>	464
<b>No Shelter</b>	382
<b>Chronic Stoplight Violation</b>	280
<b>Underage - Licensed Est</b>	271
<b>Chronic Speeding</b>	268
<b>In Car</b>	251
<b>Playing in Unsuitable Place</b>	245
<b>Drag Racing</b>	175

	Counts
Loud Television	93
Police Report Requested	90
After Hours - Licensed Est	77
Building	60
Nuisance/Truant	41
Police Report Not Requested	23
Liquor Access Complaint	6

```
In [30]: descriptor = descriptor.reset_index()  
descriptor = descriptor.rename(columns={'index': 'Descriptor'})  
descriptor
```

Out[30]:

	Descriptor	Counts
0	Loud Music/Party	61430
1	No Access	56976
2	Posted Parking Sign Violation	22440
3	Loud Talking	21584
4	Partial Access	20068
5	With License Plate	17718
6	Blocked Hydrant	16081
7	Commercial Overnight Parking	12189
8	Car/Truck Music	11273
9	Blocked Sidewalk	11121
10	Double Parked Blocking Traffic	5731
11	Double Parked Blocking Vehicle	4211
12	Engine Idling	4189
13	Banging/Pounding	4165
14	Neglected	3787
15	Car/Truck Horn	3511
16	Congestion/Gridlock	2761
17	In Prohibited Area	2025
18	Other (complaint details)	1969
19	Unlicensed	1777
20	Overnight Commercial Storage	1757
21	Unauthorized Bus Layover	1367
22	Truck Route Violation	1014
23	In Public	932
24	Tortured	854
25	Vehicle	590
26	Chained	535
27	Detached Trailer	464
28	No Shelter	382
29	Chronic Stoplight Violation	280
30	Underage - Licensed Est	271
31	Chronic Speeding	268
32	In Car	251
33	Playing in Unsuitable Place	245
34	Drag Racing	175



	Descriptor	Counts
35	Loud Television	93
36	Police Report Requested	90
37	After Hours - Licensed Est	77
38	Building	60
39	Nuisance/Tenant	41
40	Police Report Not Requested	23
41	Licensee Access Complaint	6

```
In [31]: descriptor['percentage'] = np.around((descriptor.Counts/descriptor.Counts.sum())*100,decimals=1)
descriptor
```

Out[31]:

	Descriptor	Counts	percentage
0	Loud Music/Party	61430	20.8
1	No Access	56976	19.3
2	Posted Parking Sign Violation	22440	7.6
3	Loud Talking	21584	7.3
4	Partial Access	20068	6.8
5	With License Plate	17718	6.0
6	Blocked Hydrant	16081	5.5
7	Commercial Overnight Parking	12189	4.1
8	Car/Truck Music	11273	3.8
9	Blocked Sidewalk	11121	3.8
10	Double Parked Blocking Traffic	5731	1.9
11	Double Parked Blocking Vehicle	4211	1.4
12	Engine Idling	4189	1.4
13	Banging/Pounding	4165	1.4
14	Neglected	3787	1.3
15	Car/Truck Horn	3511	1.2
16	Congestion/Gridlock	2761	0.9
17	In Prohibited Area	2025	0.7
18	Other (complaint details)	1969	0.7
19	Unlicensed	1777	0.6
20	Overnight Commercial Storage	1757	0.6
21	Unauthorized Bus Layover	1367	0.5
22	Truck Route Violation	1014	0.3
23	In Public	932	0.3
24	Tortured	854	0.3
25	Vehicle	590	0.2
26	Chained	535	0.2
27	Detached Trailer	464	0.2
28	No Shelter	382	0.1
29	Chronic Stoplight Violation	280	0.1
30	Underage - Licensed Est	271	0.1
31	Chronic Speeding	268	0.1
32	In Car	251	0.1
33	Playing in Unsuitable Place	245	0.1
34	Drag Racing	175	0.1

	Descriptor	Counts	percentage
35	Loud Television	93	0.0
36	Police Report Requested	90	0.0
37	After Hours - Licensed Est	77	0.0
38	Building	60	0.0
39	Nuisance/Truant	41	0.0
40	Police Report Not Requested	23	0.0
41	License Access Complaint	6	0.0

```
In [32]: descriptor = descriptor[descriptor.percentage>1]
descriptor
```

Out[32]:

	Descriptor	Counts	percentage
0	Loud Music/Party	61430	20.8
1	No Access	56976	19.3
2	Posted Parking Sign Violation	22440	7.6
3	Loud Talking	21584	7.3
4	Partial Access	20068	6.8
5	With License Plate	17718	6.0
6	Blocked Hydrant	16081	5.5
7	Commercial Overnight Parking	12189	4.1
8	Car/Truck Music	11273	3.8
9	Blocked Sidewalk	11121	3.8
10	Double Parked Blocking Traffic	5731	1.9
11	Double Parked Blocking Vehicle	4211	1.4
12	Engine Idling	4189	1.4
13	Banging/Pounding	4165	1.4
14	Neglected	3787	1.3
15	Car/Truck Horn	3511	1.2

```
In [33]: city = data['City'].value_counts()  
city
```

```
Out[33]: BROOKLYN          98307  
NEW YORK          65994  
BRONX             40702  
STATEN ISLAND     12343  
JAMAICA           7296  
ASTORIA           6330  
FLUSHING          5971  
RIDGEWOOD         5163  
CORONA            4295  
WOODSIDE          3544  
SOUTH RICHMOND HILL 2774  
OZONE PARK        2755  
EAST ELMHURST     2734  
ELMHURST          2673  
WOODHAVEN         2464  
MASPETH           2462  
LONG ISLAND CITY  2437  
SOUTH OZONE PARK  2173  
RICHMOND HILL     1904  
FRESH MEADOWS     1899  
QUEENS VILLAGE    1814  
MIDDLE VILLAGE    1765  
JACKSON HEIGHTS   1689  
FOREST HILLS      1688  
REGO PARK         1486  
BAYSIDE           1221  
COLLEGE POINT     1220  
FAR ROCKAWAY      1179  
WHITESTONE        1098  
HOLLIS            1012  
HOWARD BEACH      931  
ROSEDALE          922  
SPRINGFIELD GARDENS 883  
SAINT ALBANS      834  
KEW GARDENS       771  
ROCKAWAY PARK     745  
SUNNYSIDE         723  
Astoria           717  
LITTLE NECK       559  
OAKLAND GARDENS   551  
CAMBRIA HEIGHTS   477  
BELLEROSE         375  
GLEN OAKS         306  
ARVERNE           220  
FLORAL PARK       152  
Long Island City  134  
Woodside          120  
NEW HYDE PARK     98  
CENTRAL PARK      97  
QUEENS            32  
BREEZY POINT      30  
East Elmhurst     14  
Howard Beach      1  
Name: City, dtype: int64
```

```
In [34]: city = city.to_frame()  
city = city.rename(columns={'City':'Counts'})  
city
```

Out[34]:

	Counts
BROOKLYN	98307
NEW YORK	65994
BRONX	40702
STATEN ISLAND	12343
JAMAICA	7296
ASTORIA	6330
FLUSHING	5971
RIDGEWOOD	5163
CORONA	4295
WOODSIDE	3544
SOUTH RICHMOND HILL	2774
OZONE PARK	2755
EAST ELMHURST	2734
ELMHURST	2673
WOODHAVEN	2464
MASPETH	2462
LONG ISLAND CITY	2437
SOUTH OZONE PARK	2173
RICHMOND HILL	1904
FRESH MEADOWS	1899
QUEENS VILLAGE	1814
MIDDLE VILLAGE	1765
JACKSON HEIGHTS	1689
FOREST HILLS	1688
REGO PARK	1486
BAYSIDE	1221
COLLEGE POINT	1220
FAR ROCKAWAY	1179
WHITESTONE	1098
HOLLIS	1012
HOWARD BEACH	931
ROSEDALE	922
SPRINGFIELD GARDENS	883
SAINT ALBANS	834
KEW GARDENS	771
ROCKAWAY PARK	745
SUNNYSIDE	723

	Counts
Astoria	717
LITTLE NECK	559
OAKLAND GARDENS	551
CAMBRIA HEIGHTS	477
BELLEROSE	375
GLEN OAKS	306
ARVERNE	220
FLORAL PARK	152
Long Island City	134
Woodside	120
NEW HYDE PARK	98
CENTRAL PARK	97
QUEENS	32
BREEZY POINT	30
East Elmhurst	11



```
In [35]: city = city.reset_index()  
city = city.rename(columns={'index':'City'})  
city
```

Out[35]:

	City	Counts
0	BROOKLYN	98307
1	NEW YORK	65994
2	BRONX	40702
3	STATEN ISLAND	12343
4	JAMAICA	7296
5	ASTORIA	6330
6	FLUSHING	5971
7	RIDGEWOOD	5163
8	CORONA	4295
9	WOODSIDE	3544
10	SOUTH RICHMOND HILL	2774
11	OZONE PARK	2755
12	EAST ELMHURST	2734
13	ELMHURST	2673
14	WOODHAVEN	2464
15	MASPETH	2462
16	LONG ISLAND CITY	2437
17	SOUTH OZONE PARK	2173
18	RICHMOND HILL	1904
19	FRESH MEADOWS	1899
20	QUEENS VILLAGE	1814
21	MIDDLE VILLAGE	1765
22	JACKSON HEIGHTS	1689
23	FOREST HILLS	1688
24	REGO PARK	1486
25	BAYSIDE	1221
26	COLLEGE POINT	1220
27	FAR ROCKAWAY	1179
28	WHITESTONE	1098
29	HOLLIS	1012
30	HOWARD BEACH	931
31	ROSEDALE	922
32	SPRINGFIELD GARDENS	883
33	SAINT ALBANS	834
34	KEW GARDENS	771

	City	Counts
35	ROCKAWAY PARK	745
36	SUNNYSIDE	723
37	Astoria	717
38	LITTLE NECK	559
39	OAKLAND GARDENS	551
40	CAMBRIA HEIGHTS	477
41	BELLEROSE	375
42	GLEN OAKS	306
43	ARVERNE	220
44	FLORAL PARK	152
45	Long Island City	134
46	Woodside	120
47	NEW HYDE PARK	98
48	CENTRAL PARK	97
49	QUEENS	33

```
In [36]: city['percentage'] = np.around((city.Counts/city.Counts.sum())*100,decimals=1)
city = city[city.percentage > 1]
city
```

Out[36]:

	City	Counts	percentage
0	BROOKLYN	98307	33.0
1	NEW YORK	65994	22.1
2	BRONX	40702	13.7
3	STATEN ISLAND	12343	4.1
4	JAMAICA	7296	2.4
5	ASTORIA	6330	2.1
6	FLUSHING	5971	2.0
7	RIDGEWOOD	5163	1.7
8	CORONA	4295	1.4
9	WOODSIDE	3544	1.2

```
In [37]: location = data['Location Type'].value_counts()
location
```

```
Out[37]: Street/Sidewalk      249299
Store/Commercial      20381
Club/Bar/Restaurant   17360
Residential Building/House  6960
Park/Playground       4773
House of Worship      929
Residential Building   227
Highway               215
Parking Lot           117
House and Store        93
Vacant Lot             77
Commercial             62
Roadway Tunnel         35
Subway Station         34
Bridge                 2
Terminal               1
Ferry                  1
Park                   1
Name: Location Type, dtype: int64
```

```
In [38]: location = location.to_frame()
location = location.rename(columns={'Location Type':'Counts'})
location
```

```
Out[38]:
```

	Counts
Street/Sidewalk	249299
Store/Commercial	20381
Club/Bar/Restaurant	17360
Residential Building/House	6960
Park/Playground	4773
House of Worship	929
Residential Building	227
Highway	215
Parking Lot	117
House and Store	93
Vacant Lot	77
Commercial	62
Roadway Tunnel	35
Subway Station	34
Bridge	2
Terminal	1
Ferry	1
Park	1

```
In [39]: location = location.reset_index()
location = location.rename(columns={'index':'Location Type'})
location['percentage'] = np.around((location.Counts/location.Counts.sum
())/100,decimals=1)
location
```

Out[39]:

	Location Type	Counts	percentage
0	Street/Sidewalk	249299	82.9
1	Store/Commercial	20381	6.8
2	Club/Bar/Restaurant	17360	5.8
3	Residential Building/House	6960	2.3
4	Park/Playground	4773	1.6
5	House of Worship	929	0.3
6	Residential Building	227	0.1
7	Highway	215	0.1
8	Parking Lot	117	0.0
9	House and Store	93	0.0
10	Vacant Lot	77	0.0
11	Commercial	62	0.0
12	Roadway Tunnel	35	0.0
13	Subway Station	34	0.0
14	Bridge	2	0.0
15	Terminal	1	0.0
16	Ferry	1	0.0
17	Park	1	0.0

```
In [40]: location = location[location.percentage > 0.1]
location
```

Out[40]:

	Location Type	Counts	percentage
0	Street/Sidewalk	249299	82.9
1	Store/Commercial	20381	6.8
2	Club/Bar/Restaurant	17360	5.8
3	Residential Building/House	6960	2.3
4	Park/Playground	4773	1.6
5	House of Worship	929	0.3

```
In [41]: address = data['Address Type'].value_counts()
address
```

```
Out[41]: ADDRESS          238644
INTERSECTION      43366
BLOCKFACE         12014
LATLONG           3509
PLACENAME          350
Name: Address Type, dtype: int64
```

```
In [42]: address = address.to_frame()
address = address.rename(columns={'Address Type': 'Counts'})
address
```

```
Out[42]:
```

	Counts
ADDRESS	238644
INTERSECTION	43366
BLOCKFACE	12014
LATLONG	3509
PLACENAME	350

```
In [43]: address = address.reset_index()
address = address.rename(columns={'index': 'Address Type'})
address
```

```
Out[43]:
```

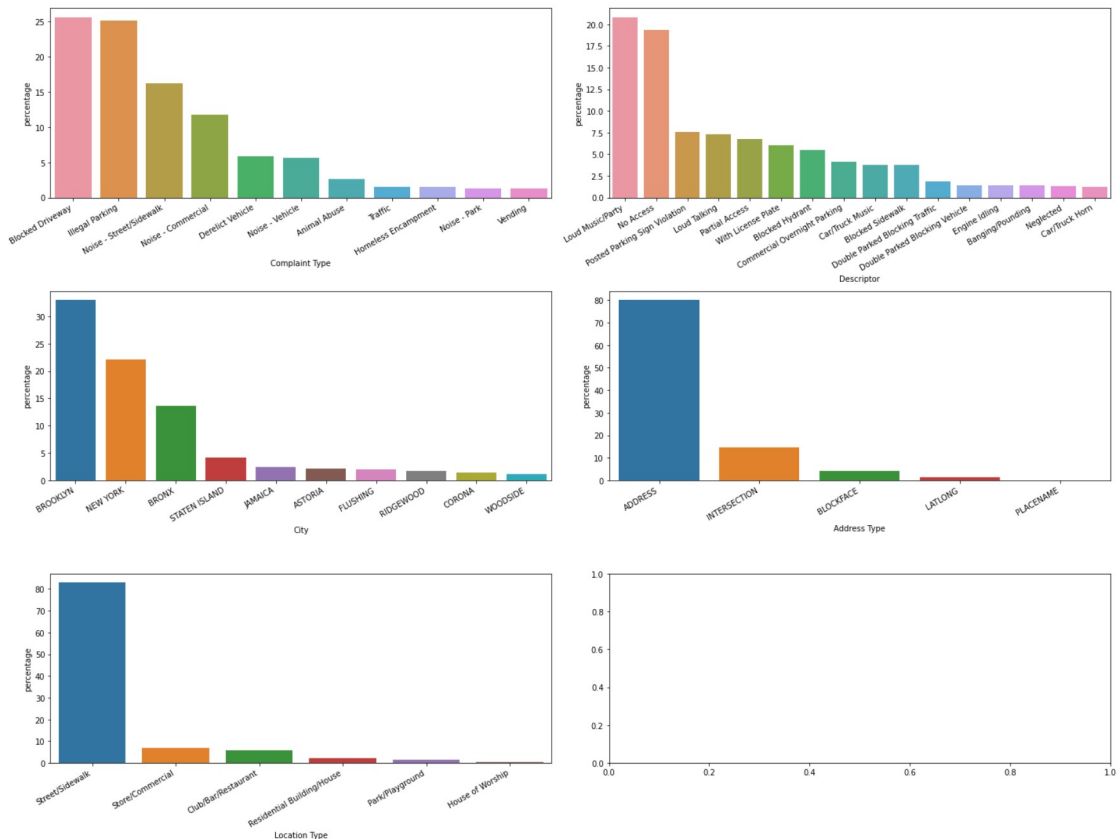
	Address Type	Counts
0	ADDRESS	238644
1	INTERSECTION	43366
2	BLOCKFACE	12014
3	LATLONG	3509
4	PLACENAME	350

```
In [44]: address['percentage'] = np.around((address.Counts/address.Counts.sum())*100, decimals=1)
address
```

```
Out[44]:
```

	Address Type	Counts	percentage
0	ADDRESS	238644	80.1
1	INTERSECTION	43366	14.6
2	BLOCKFACE	12014	4.0
3	LATLONG	3509	1.2
4	PLACENAME	350	0.1

```
In [45]: fig, ax = plt.subplots(3, 2, figsize=(20, 15))
comp = sns.barplot(ax = ax[0,0],x=complaint['Complaint Type'], y=complaint.percentage)
comp.set_xticklabels(comp.get_xticklabels(), rotation=30, ha="right")
des = sns.barplot(ax = ax[0,1],x=descriptor['Descriptor'],y=descriptor.percentage)
des.set_xticklabels(des.get_xticklabels(), rotation=30, ha="right")
ci = sns.barplot(ax = ax[1,0],x=city['City'], y=city.percentage)
ci.set_xticklabels(ci.get_xticklabels(), rotation=30, ha="right")
addr = sns.barplot(ax = ax[1,1],x=address['Address Type'], y=address.percentage)
addr.set_xticklabels(addr.get_xticklabels(), rotation=30, ha="right")
loc = sns.barplot(ax = ax[2,0],x=location['Location Type'], y=location.percentage)
loc.set_xticklabels(loc.get_xticklabels(), rotation=30, ha="right")
plt.tight_layout()
```



```
In [46]: C_CT_RCT = data[['City', 'Complaint Type', 'Request_Closing_Time']]
C_CT_RCT.dropna(subset = ['City', 'Complaint Type', 'Request_Closing_Time'], inplace = True)
C_CT_RCT['DTime'] = np.around( (C_CT_RCT['Request_Closing_Time'].astype(np.int64)/
                                                                    (pow(10,9)*3600) ),
                                decimals=2)
C_CT_RCT.head(6)
```

/usr/local/lib/python3.7/dist-packages/pandas/util/\_decorators.py:311: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

return func(\*args, \*\*kwargs)  
/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:3: FutureWarning: casting timedelta64[ns] values to int64 with .astype(...) is deprecated and will raise in a future version. Use .view(...) instead.

This is separate from the ipykernel package so we can avoid doing imports until

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:4: 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: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
after removing the cwd from sys.path.

Out[46]:

	City	Complaint Type	Request_Closing_Time	DTime
0	NEW YORK	Noise - Street/Sidewalk	0 days 00:55:15	0.92
1	ASTORIA	Blocked Driveway	0 days 01:26:16	1.44
2	BRONX	Blocked Driveway	0 days 04:51:31	4.86
3	BRONX	Illegal Parking	0 days 07:45:14	7.75
4	ELMHURST	Illegal Parking	0 days 03:27:02	3.45
5	BROOKLYN	Illegal Parking	0 days 01:53:30	1.89



```
In [47]: Avarage_time = np.around((C_CT_RCT['DTime'].mean()),decimals=2)
print('Avarage or mean time gap between logging the complaint and problem
solved = ',Avarage_time, 'hour')
Central_val = np.around((C_CT_RCT['DTime'].median()),decimals=2)
print('Central or median value of the distribution = ',Central_val, 'hour
')
Most_occoor = np.around((C_CT_RCT['DTime'].mode()),decimals=2)
print('Freuqently occered value = ',Most_occoor, 'hour')
stand_dev = np.around((C_CT_RCT['DTime'].std()),decimals=2)
print('Standard Deviation is = ',stand_dev)
```

Avarage or mean time gap between logging the complaint and problem solve  
d = 4.31 hour  
Central or median value of the distribution = 2.71 hour  
Freuqently occered value = 0 0.88  
dtype: float64 hour  
Standard Deviation is = 6.08

```
In [48]: data['Created Date'].head(5)
```

```
Out[48]: 0    2015-12-31 23:59:45
1    2015-12-31 23:59:44
2    2015-12-31 23:59:29
3    2015-12-31 23:57:46
4    2015-12-31 23:56:58
Name: Created Date, dtype: datetime64[ns]
```

In [49]: *# Creating a data frame Contain Days and Months of Complaint date*

```
-----
Year_Month_Day = pd.to_datetime(data['Created Date'].dt.date)
Month_Day = pd.DataFrame()
Month_Day['Date'] = pd.to_datetime(Year_Month_Day.dt.date)
Month_Day['Month'] = Year_Month_Day.dt.month
Month_Day['Day'] = Year_Month_Day.dt.day
Month_Day['Month Name'] = Month_Day['Month'].apply(lambda x: calendar.mon
th_abbrev[x])
Month_Day['Day No'] = Month_Day['Date'].dt.weekday
Month_Day['Day Name'] = Month_Day['Day No'].map({0: 'Monday', 1: 'Tuesday',
2: 'Wednesday', 3: 'Thursday', 4: 'Friday',
5: 'Saturday', 6: 'Sunday'
'})
Month_Day.sample(20)
```

Out[49]:

	Date	Month	Day	Month Name	Day No	Day Name
<b>104844</b>	2015-09-21	9	21	Sep	0	Monday
<b>129848</b>	2015-08-31	8	31	Aug	0	Monday
<b>169059</b>	2015-07-27	7	27	Jul	0	Monday
<b>74388</b>	2015-10-19	10	19	Oct	0	Monday
<b>177356</b>	2015-07-20	7	20	Jul	0	Monday
<b>13041</b>	2015-12-17	12	17	Dec	3	Thursday
<b>265579</b>	2015-05-05	5	5	May	1	Tuesday
<b>106248</b>	2015-09-20	9	20	Sep	6	Sunday
<b>214567</b>	2015-06-18	6	18	Jun	3	Thursday
<b>90973</b>	2015-10-04	10	4	Oct	6	Sunday
<b>77091</b>	2015-10-17	10	17	Oct	5	Saturday
<b>10492</b>	2015-12-20	12	20	Dec	6	Sunday
<b>112853</b>	2015-09-15	9	15	Sep	1	Tuesday
<b>261877</b>	2015-05-08	5	8	May	4	Friday
<b>17570</b>	2015-12-13	12	13	Dec	6	Sunday
<b>296123</b>	2015-04-03	4	3	Apr	4	Friday
<b>60362</b>	2015-11-01	11	1	Nov	6	Sunday
<b>8336</b>	2015-12-22	12	22	Dec	1	Tuesday
<b>119759</b>	2015-09-09	9	9	Sep	2	Wednesday
<b>157194</b>	2015-08-07	8	7	Aug	4	Friday

```
In [50]: Month_plot = Month_Day['Month Name'].value_counts()  
Month_plot = Month_plot.to_frame()  
Month_plot = Month_plot.rename(columns={'Month Name': 'Counts'})  
Month_plot
```

Out[50]:

	Counts
<b>May</b>	36437
<b>Sep</b>	35427
<b>Jun</b>	35315
<b>Aug</b>	34956
<b>Jul</b>	34888
<b>Oct</b>	32605
<b>Nov</b>	30773
<b>Dec</b>	30521
<b>Apr</b>	27305
<b>Mar</b>	2471

```
In [51]: Day_plot = Month_Day['Day Name'].value_counts()  
Day_plot = Day_plot.to_frame()  
Day_plot = Day_plot.rename(columns={'Day Name': 'Counts'})  
Day_plot
```

Out[51]:

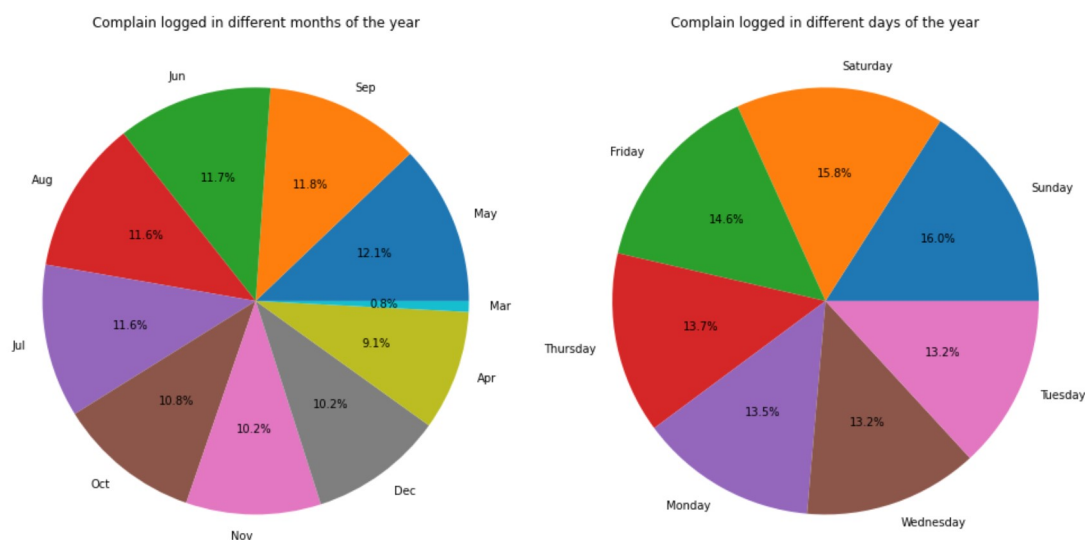
	Counts
<b>Sunday</b>	47969
<b>Saturday</b>	47564
<b>Friday</b>	43995
<b>Thursday</b>	41342
<b>Monday</b>	40489
<b>Wednesday</b>	39788
<b>Tuesday</b>	39551

```
In [52]: fig, axes = plt.subplots(1,2, figsize=(14,8))

axes[0].pie(Month_plot['Counts'], labels = Month_plot.index,autopct='%1.1f%%')
axes[0].set_title('Complain logged in different months of the year')

axes[1].pie(Day_plot['Counts'], labels = Day_plot.index,autopct='%1.1f%%')
axes[1].set_title('Complain logged in different days of the year')

plt.tight_layout()
```



```
In [53]: Month_Day_grouped = Month_Day.groupby(['Month Name', 'Day Name'], as_index=
False)['Day No'].count()
Month_Day_grouped_final = Month_Day_grouped.rename(columns={'Day No': 'Counts'})
Month_Day_grouped_final.head(10)
```

Out[53]:

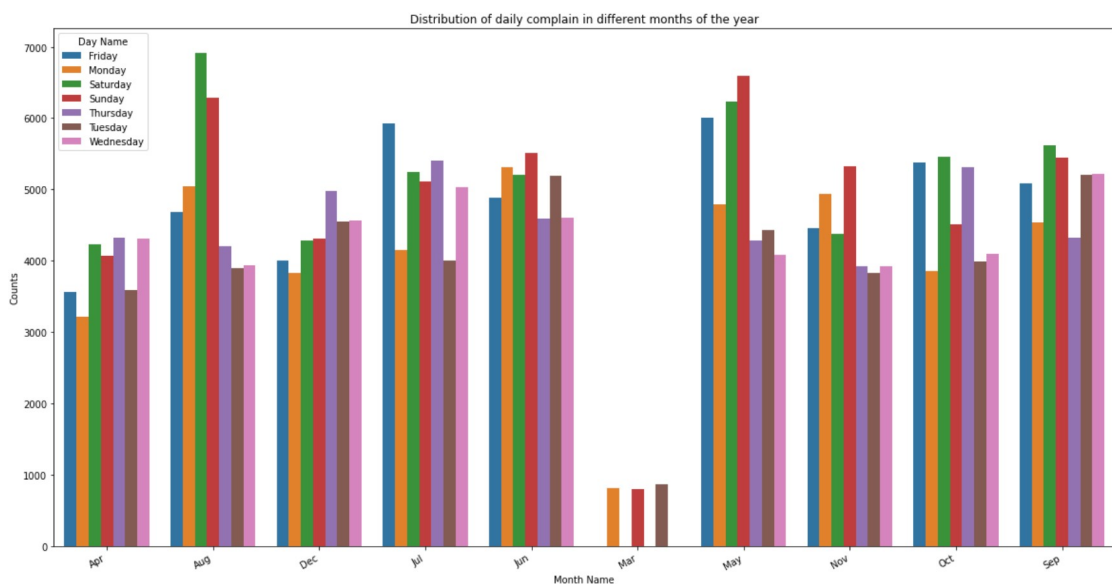
	Month Name	Day Name	Counts
0	Apr	Friday	3565
1	Apr	Monday	3222
2	Apr	Saturday	4227
3	Apr	Sunday	4069
4	Apr	Thursday	4323
5	Apr	Tuesday	3586
6	Apr	Wednesday	4313
7	Aug	Friday	4684
8	Aug	Monday	5042
9	Aug	Saturday	6913

```
In [54]: Month_Day[( (Month_Day['Month Name'] == 'Mar') & (Month_Day['Day Name'] =
= 'Monday') )].count()
```

```
Out[54]: Date          807
Month          807
Day           807
Month Name     807
Day No        807
Day Name       807
dtype: int64
```

```
In [55]: plt.figure(figsize=(20,10))

month_day_plot = sns.barplot(x=Month_Day_grouped_final['Month Name'], y=Month_Day_grouped_final['Counts'],
                             hue=Month_Day_grouped_final['Day Name'], data=Month_Day_grouped_final)
month_day_plot.set_xticklabels(month_day_plot.get_xticklabels(), rotation=30, ha="right")
plt.title('Distribution of daily complain in different months of the year')
plt.show()
plt.tight_layout()
```



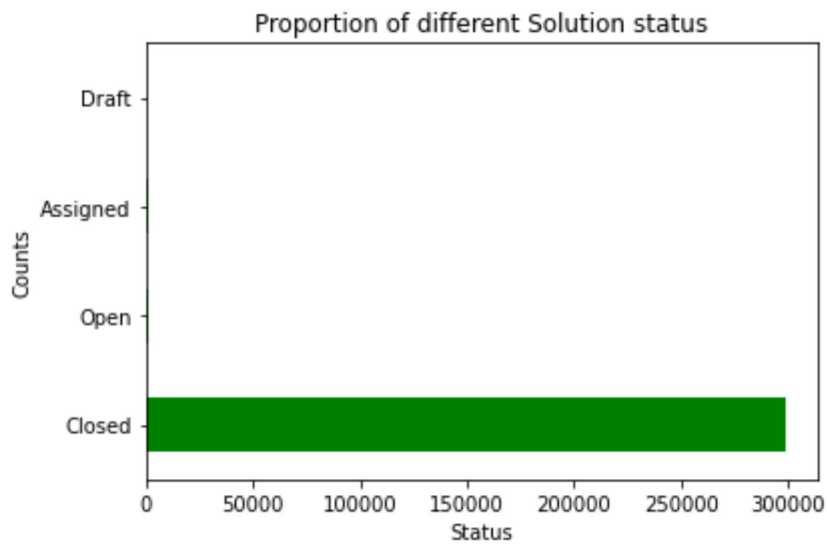
<Figure size 432x288 with 0 Axes>

```
In [56]: Month_Day_grouped[Month_Day_grouped['Month Name'] == 'Mar']
```

```
Out[56]:
```

	Month Name	Day Name	Day No
35	Mar	Monday	807
36	Mar	Sunday	802
37	Mar	Tuesday	862

```
In [57]: data['Status'].value_counts().plot(kind='barh',color='g')
plt.xlabel('Status')
plt.ylabel('Counts')
plt.title('Proportion of different Solution status')
plt.show()
plt.tight_layout()
```



<Figure size 432x288 with 0 Axes>

4. Order the complaint types based on the average 'Request\_Closing\_Time', grouping them for different locations.

```
In [67]: Complaint_City_AvgTime_grouped = C_CT_RCT.groupby(['City', 'Complaint Type'])
        .agg({'DTime': 'mean'})
        Complaint_City_AvgTime_grouped = Complaint_City_AvgTime_grouped.rename(
        columns={'DTime': 'Avg. Time(Given City, Complaint Type)'})
        Complaint_City_AvgTime_grouped = Complaint_City_AvgTime_grouped.sort_valu
        es(['City', 'Avg. Time(Given City, Complaint Type)'])
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', None)
        Complaint_City_AvgTime_grouped
```

Out[67]:

		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
ARVERNE	Drinking	0.240000
	Vending	0.480000
	Urinating in Public	0.690000
	Panhandling	1.030000
	Noise - Park	1.285000
	Graffiti	1.530000
	Noise - House of Worship	1.562727
	Homeless Encampment	1.812500
	Noise - Vehicle	1.860000
	Noise - Street/Sidewalk	1.992759
	Animal Abuse	2.153158
	Noise - Commercial	2.285000
	Illegal Parking	2.316207
	Blocked Driveway	2.526286
	Derelict Vehicle	2.968519
ASTORIA	Disorderly Youth	3.595000
	Panhandling	1.150000
	Bike/Roller/Skate Chronic	1.740667
	Noise - House of Worship	2.022632
	Illegal Fireworks	2.772500
	Disorderly Youth	2.903333
	Noise - Park	2.994754
	Noise - Commercial	3.133039
	Noise - Street/Sidewalk	3.450881
	Noise - Vehicle	3.509020
	Urinating in Public	4.626667
	Drinking	4.722571
	Blocked Driveway	4.816108
	Illegal Parking	4.833371
	Homeless Encampment	4.918750



		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
	Vending	4.935556
	Animal Abuse	5.000640
	Traffic	5.410851
	Posting Advertisement	5.870000
	Derelict Vehicle	9.689145
	Graffiti	14.097500
Astoria	Noise - Commercial	3.542069
	Noise - Street/Sidewalk	3.713333
	Illegal Parking	4.711362
	Blocked Driveway	4.915172
	Derelict Vehicle	6.234167
BAYSIDE	Traffic	1.526667
	Noise - Street/Sidewalk	1.530667
	Noise - Vehicle	1.709375
	Vending	1.880000
	Drinking	1.900000
	Noise - Commercial	2.234500
	Illegal Parking	2.562763
	Blocked Driveway	2.562997
	Homeless Encampment	2.875000
	Disorderly Youth	2.970000
	Animal Abuse	3.274865
	Noise - Park	3.275000
	Derelict Vehicle	3.360000
	Noise - House of Worship	3.535000
	Graffiti	4.553333
BELLEROSE	Noise - Park	1.410000
	Disorderly Youth	1.850000
	Noise - House of Worship	2.200000
	Posting Advertisement	2.260000
	Noise - Vehicle	2.584000
	Drinking	3.920000
	Bike/Roller/Skate Chronic	4.900000
	Traffic	5.760000
	Illegal Fireworks	6.670000
	Noise - Commercial	6.740811
	Panhandling	7.480000

		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
BREEZY POINT	Urinating in Public	7.540000
	Illegal Parking	8.203019
	Noise - Street/Sidewalk	9.069231
	Blocked Driveway	10.099474
	Animal Abuse	12.725714
	Derelict Vehicle	17.167978
	Homeless Encampment	39.130000
	Noise - Street/Sidewalk	1.000000
	Noise - Vehicle	1.320000
	Blocked Driveway	1.326667
BRONX	Noise - Commercial	2.540000
	Animal Abuse	2.615000
	Drinking	2.630000
	Illegal Parking	3.942667
	Derelict Vehicle	7.143333
	Bike/Roller/Skate Chronic	3.459500
	Posting Advertisement	3.461176
	Disorderly Youth	4.238571
	Noise - House of Worship	4.559494
	Noise - Commercial	4.697402
BROOKLYN	Noise - Park	4.698556
	Traffic	4.923606
	Noise - Street/Sidewalk	5.226292
	Urinating in Public	5.389804
	Noise - Vehicle	5.560851
	Illegal Fireworks	5.607083
	Drinking	5.793404
	Blocked Driveway	6.261773
	Illegal Parking	6.580888
	Vending	6.825673
BROOKLYN	Animal Abuse	7.335534
	Homeless Encampment	7.442186
	Graffiti	8.898889
	Derelict Vehicle	9.227623
	Panhandling	14.214211
BROOKLYN	Illegal Fireworks	2.340000
	Noise - Commercial	2.986138

		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
	Noise - House of Worship	3.069765
	Traffic	3.112959
	Noise - Park	3.150495
	Noise - Vehicle	3.283404
	Noise - Street/Sidewalk	3.296003
	Posting Advertisement	3.361778
	Drinking	3.540467
	Urinating in Public	3.899191
	Disorderly Youth	4.150833
	Illegal Parking	4.272241
	Panhandling	4.316735
	Blocked Driveway	4.410757
	Vending	4.528097
	Homeless Encampment	4.694912
	Animal Abuse	4.832427
	Bike/Roller/Skate Chronic	5.004865
	Derelict Vehicle	5.947511
	Graffiti	8.242791
	Illegal Fireworks	1.530000
	Noise - House of Worship	2.640000
CAMBRIA HEIGHTS	Noise - Commercial	3.809167
	Noise - Street/Sidewalk	4.607600
	Noise - Vehicle	6.917273
	Blocked Driveway	7.696054
	Traffic	8.546667
	Illegal Parking	11.243947
	Animal Abuse	11.355455
	Derelict Vehicle	16.110870
	Homeless Encampment	22.788000
	Noise - Street/Sidewalk	3.266526
CENTRAL PARK	Illegal Parking	4.595000
	Disorderly Youth	0.600000
COLLEGE POINT	Homeless Encampment	1.443333
	Traffic	2.293571
	Noise - Vehicle	2.650687
	Noise - Street/Sidewalk	3.016667
	Illegal Parking	3.125227

		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
CORONA	Noise - Park	3.180000
	Blocked Driveway	3.380667
	Derelict Vehicle	3.518913
	Noise - Commercial	3.972571
	Animal Abuse	4.643929
	Vending	4.660000
	Graffiti	11.860000
	Graffiti	0.730000
	Panhandling	1.170000
	Posting Advertisement	1.540000
	Urinating in Public	1.951429
	Traffic	2.285833
	Noise - Park	2.320417
	Noise - Street/Sidewalk	2.364160
	Noise - Commercial	2.622177
	Noise - Vehicle	2.660500
	Disorderly Youth	2.876667
EAST ELMHURST	Vending	3.101613
	Blocked Driveway	3.320837
	Illegal Parking	3.361045
	Drinking	3.648485
	Animal Abuse	3.668033
	Homeless Encampment	3.669474
	Noise - House of Worship	3.753333
	Derelict Vehicle	4.585088
	Bike/Roller/Skate Chronic	0.250000
	Noise - House of Worship	1.439444
	Posting Advertisement	1.860000
	Drinking	1.861111
	Urinating in Public	2.100000
	Noise - Vehicle	2.168033
	Noise - Commercial	2.354000
	Noise - Street/Sidewalk	2.474673
	Noise - Park	2.654000
	Traffic	2.687500
	Illegal Parking	3.378139
	Blocked Driveway	3.733274

		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
ELMHURST	Vending	4.021111
	Animal Abuse	4.055254
	Derelict Vehicle	5.687080
	Homeless Encampment	6.300000
	Disorderly Youth	6.900000
	Graffiti	7.650000
	Posting Advertisement	0.730000
	Disorderly Youth	0.860000
	Illegal Fireworks	0.980000
	Noise - House of Worship	1.886000
	Noise - Park	2.314412
	Noise - Street/Sidewalk	2.509554
	Urinating in Public	2.539000
	Noise - Vehicle	2.626170
	Traffic	2.627857
	Noise - Commercial	2.790123
	Drinking	3.062308
	Illegal Parking	3.278196
	Panhandling	3.300000
	Blocked Driveway	3.433396
East Elmhurst	Homeless Encampment	3.674688
	Animal Abuse	3.872105
FAR ROCKAWAY	Vending	3.994762
	Bike/Roller/Skate Chronic	4.625000
	Derelict Vehicle	4.818333
	Illegal Parking	5.784615
	Derelict Vehicle	9.490000
	Noise - House of Worship	1.130000
	Noise - Park	1.504348
	Urinating in Public	1.510000
	Noise - Commercial	1.927708
	Noise - Vehicle	2.180909
	Traffic	2.385000
	Drinking	2.417500
	Blocked Driveway	2.634648
	Illegal Parking	2.706441
	Animal Abuse	2.717865

		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
FLORAL PARK	Vending	2.805556
	Homeless Encampment	3.019286
	Noise - Street/Sidewalk	3.038382
	Disorderly Youth	3.370000
	Derelict Vehicle	3.665615
	Noise - Vehicle	1.950000
	Disorderly Youth	3.260000
	Noise - Commercial	4.366667
	Noise - Street/Sidewalk	6.913333
	Blocked Driveway	7.701500
	Drinking	8.230000
	Illegal Parking	9.254062
	Derelict Vehicle	16.654464
	Animal Abuse	26.580000
FLUSHING	Illegal Fireworks	0.695000
	Panhandling	1.150000
	Urinating in Public	1.226667
	Graffiti	1.722500
	Traffic	2.063617
	Disorderly Youth	2.145000
	Vending	2.450000
	Noise - Street/Sidewalk	2.830533
	Noise - Commercial	2.845029
	Noise - Park	2.889310
	Blocked Driveway	2.951002
	Illegal Parking	2.982001
	Homeless Encampment	3.006538
	Drinking	3.052250
	Noise - Vehicle	3.359690
FOREST HILLS	Animal Abuse	3.557762
	Noise - House of Worship	3.640000
	Derelict Vehicle	3.692932
	Bike/Roller/Skate Chronic	5.733333
	Posting Advertisement	6.130000
	Illegal Fireworks	0.670000
	Urinating in Public	1.370000
	Noise - Park	1.539000

		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
	Noise - Commercial	1.880993
	Traffic	2.007000
	Noise - Street/Sidewalk	2.320842
	Noise - Vehicle	2.383333
	Vending	2.745000
	Animal Abuse	3.262889
	Graffiti	3.273333
	Illegal Parking	3.324554
	Homeless Encampment	3.510556
	Blocked Driveway	3.715339
	Derelict Vehicle	3.725962
	Noise - House of Worship	3.900000
	Drinking	4.100000
	Disorderly Youth	4.150000
	Bike/Roller/Skate Chronic	4.824000
	Panhandling	5.816000
	Posting Advertisement	5.866667
	FRESH MEADOWS Panhandling	1.570000
	Urinating in Public	1.600000
	Traffic	1.631538
	Noise - Vehicle	2.145455
	Noise - Commercial	2.407857
	Noise - Street/Sidewalk	2.467381
	Illegal Parking	2.623379
	Vending	2.630000
	Animal Abuse	3.279778
	Noise - Park	3.416250
	Drinking	3.735000
	Blocked Driveway	3.973936
	Derelict Vehicle	4.511512
	Homeless Encampment	5.900000
GLEN OAKS	Noise - Park	4.402432
	Vending	4.843889
	Traffic	5.550000
	Noise - Commercial	6.117692
	Illegal Parking	8.967162
	Urinating in Public	11.010000

		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
HOLLIS	Blocked Driveway	11.299667
	Noise - Vehicle	11.422500
	Noise - Street/Sidewalk	11.915000
	Animal Abuse	12.990000
	Derelict Vehicle	15.245102
	Noise - Street/Sidewalk	2.830732
	Noise - Vehicle	3.190851
	Noise - Park	3.512353
	Traffic	3.819091
	Disorderly Youth	3.870000
	Urinating in Public	3.950000
	Noise - House of Worship	4.052193
	Homeless Encampment	4.283333
	Animal Abuse	4.370909
	Blocked Driveway	4.795117
HOWARD BEACH	Noise - Commercial	6.715200
	Illegal Parking	6.792318
	Drinking	7.296667
	Derelict Vehicle	11.565035
	Vending	1.810000
	Illegal Fireworks	1.846667
	Drinking	2.007500
	Bike/Roller/Skate Chronic	2.630000
	Homeless Encampment	2.646667
	Noise - Vehicle	3.084000
	Traffic	3.096667
	Panhandling	3.300000
	Noise - Park	3.875000
	Noise - Street/Sidewalk	3.943333
	Disorderly Youth	4.190000
Howard Beach	Blocked Driveway	4.254970
	Animal Abuse	4.411613
	Illegal Parking	5.757438
	Noise - Commercial	5.834922
	Noise - House of Worship	6.800000
	Derelict Vehicle	11.476377
	Blocked Driveway	4.030000



		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
JACKSON HEIGHTS	Illegal Fireworks	0.700000
	Noise - House of Worship	1.105000
	Posting Advertisement	1.380000
	Panhandling	1.680000
	Noise - Vehicle	2.211207
	Noise - Street/Sidewalk	2.321198
	Noise - Park	2.418750
	Homeless Encampment	2.612727
	Noise - Commercial	2.956350
	Traffic	3.246923
	Illegal Parking	3.649781
	Vending	3.714615
	Blocked Driveway	3.739313
	Derelict Vehicle	3.768966
	Urinating in Public	3.820000
	Animal Abuse	4.059762
	Bike/Roller/Skate Chronic	4.115000
	Drinking	5.727778
JAMAICA	Panhandling	1.740000
	Illegal Fireworks	2.607500
	Bike/Roller/Skate Chronic	2.915000
	Noise - Park	3.181316
	Noise - Vehicle	3.463775
	Noise - Street/Sidewalk	3.664749
	Noise - Commercial	3.923823
	Drinking	4.412059
	Traffic	4.706464
	Blocked Driveway	4.981122
	Animal Abuse	5.059127
	Disorderly Youth	5.077500
	Illegal Parking	5.079275
	Urinating in Public	5.108182
	Noise - House of Worship	5.351538
	Graffiti	5.410000
	Posting Advertisement	5.487143
	Vending	7.512000
	Derelict Vehicle	7.946527

		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
KEW GARDENS	Homeless Encampment	8.077468
	Drinking	1.020000
	Noise - House of Worship	1.160000
	Homeless Encampment	1.900000
	Noise - Street/Sidewalk	2.584000
	Animal Abuse	3.164211
	Noise - Vehicle	3.625556
	Traffic	3.793000
	Noise - Commercial	3.857073
	Illegal Parking	4.753396
	Blocked Driveway	6.129553
	Vending	6.220000
	Derelict Vehicle	7.191429
	Urinating in Public	7.196667
LITTLE NECK	Noise - Park	1.070000
	Noise - Commercial	1.352763
	Drinking	1.580000
	Noise - Street/Sidewalk	1.982500
	Noise - Vehicle	2.152000
	Traffic	2.200588
	Posting Advertisement	2.230000
	Animal Abuse	2.303333
	Blocked Driveway	2.417355
	Illegal Parking	2.850763
	Urinating in Public	3.080000
	Derelict Vehicle	3.597049
	Disorderly Youth	4.355000
LONG ISLAND CITY	Posting Advertisement	0.820000
	Drinking	3.032857
	Traffic	3.384583
	Graffiti	3.505000
	Panhandling	3.900000
	Urinating in Public	4.193333
	Noise - Street/Sidewalk	4.392358
	Noise - Commercial	4.542130
	Noise - Vehicle	5.062056
	Noise - Park	5.333704

		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
Long Island City	Disorderly Youth	5.500000
	Animal Abuse	6.058667
	Blocked Driveway	6.076969
	Bike/Roller/Skate Chronic	6.763333
	Homeless Encampment	7.015000
	Illegal Parking	7.467154
	Vending	9.279333
	Derelict Vehicle	10.488462
	Noise - Street/Sidewalk	2.970385
	Noise - Commercial	3.357222
MASPETH	Blocked Driveway	3.527059
	Illegal Parking	4.945577
	Derelict Vehicle	8.680000
	Illegal Fireworks	1.620000
	Drinking	2.324444
	Urinating in Public	3.695000
	Traffic	3.754545
	Disorderly Youth	4.015000
	Noise - Street/Sidewalk	4.208347
	Noise - Vehicle	4.601579
MIDDLE VILLAGE	Vending	4.635000
	Blocked Driveway	4.815724
	Noise - Commercial	4.925577
	Illegal Parking	5.495492
	Homeless Encampment	7.183000
	Animal Abuse	7.278333
	Derelict Vehicle	7.777995
	Noise - House of Worship	7.900000
	Bike/Roller/Skate Chronic	8.840000
	Noise - Park	11.033333
	Drinking	1.235000
	Traffic	3.531667
	Homeless Encampment	4.352000
	Blocked Driveway	4.368993
	Noise - Park	4.545000
	Noise - Vehicle	4.743095
	Noise - Street/Sidewalk	4.895676

		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
	Illegal Parking	4.998780
	Noise - Commercial	5.113000
	Animal Abuse	6.881364
	Derelict Vehicle	8.245000
	Bike/Roller/Skate Chronic	15.680000
NEW HYDE PARK	Animal Abuse	1.920000
	Noise - Vehicle	3.345000
	Illegal Parking	7.589286
	Blocked Driveway	7.738491
	Derelict Vehicle	7.802143
NEW YORK	Illegal Fireworks	1.720278
	Noise - House of Worship	2.305206
	Disorderly Youth	2.408261
	Noise - Vehicle	2.621935
	Traffic	2.649457
	Noise - Street/Sidewalk	2.733194
	Noise - Commercial	2.735005
	Bike/Roller/Skate Chronic	2.892178
	Urinating in Public	2.893984
	Noise - Park	2.945041
	Posting Advertisement	2.955854
	Drinking	3.057831
	Vending	3.306084
	Illegal Parking	3.390036
	Panhandling	3.474404
	Blocked Driveway	3.558807
	Animal Abuse	3.685252
	Homeless Encampment	3.703297
	Squeegee	4.047500
	Derelict Vehicle	4.266071
	Graffiti	5.063636
OAKLAND GARDENS	Bike/Roller/Skate Chronic	1.115000
	Disorderly Youth	1.430000
	Noise - Street/Sidewalk	1.643684
	Illegal Parking	2.292928
	Noise - Vehicle	2.314000
	Traffic	2.411667

		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
OZONE PARK	Drinking	2.440000
	Blocked Driveway	2.546667
	Noise - Park	2.749286
	Animal Abuse	2.764737
	Derelict Vehicle	3.719419
	Vending	3.785000
	Homeless Encampment	28.650000
	Illegal Fireworks	0.320000
	Noise - House of Worship	0.840000
	Homeless Encampment	1.963333
	Disorderly Youth	2.207500
	Noise - Park	2.905556
	Noise - Vehicle	3.340704
	Noise - Street/Sidewalk	3.822847
	Noise - Commercial	3.907391
	Drinking	3.944211
	Vending	4.280000
	Traffic	4.343684
	Urinating in Public	4.397500
	Panhandling	4.582857
QUEENS	Blocked Driveway	4.956267
	Animal Abuse	4.963125
	Posting Advertisement	4.973333
	Illegal Parking	5.119742
	Bike/Roller/Skate Chronic	7.190000
	Derelict Vehicle	10.677619
	Urinating in Public	0.350000
	Noise - Commercial	1.325000
	Noise - Vehicle	1.325000
	Traffic	2.115000
	Noise - House of Worship	2.600000
	Illegal Parking	3.417500
	Noise - Street/Sidewalk	3.655000
	Blocked Driveway	3.995000
	Homeless Encampment	7.200000
	Derelict Vehicle	8.690000
	Animal in a Park	336.830000

		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
QUEENS VILLAGE	Noise - House of Worship	2.515000
	Posting Advertisement	3.060000
	Illegal Fireworks	3.284000
	Noise - Park	3.525000
	Drinking	4.680000
	Noise - Street/Sidewalk	5.581667
	Urinating in Public	6.334000
	Traffic	7.098846
	Noise - Vehicle	8.879268
	Panhandling	9.060000
	Homeless Encampment	9.383333
	Blocked Driveway	9.538376
	Illegal Parking	9.935484
	Noise - Commercial	10.036512
	Animal Abuse	12.821970
	Vending	14.600000
	Derelict Vehicle	16.077216
	Graffiti	53.330000
REGO PARK	Graffiti	0.950000
	Homeless Encampment	1.528333
	Noise - Commercial	2.392785
	Drinking	2.562500
	Noise - Street/Sidewalk	2.604912
	Noise - Vehicle	2.946744
	Illegal Parking	3.367356
	Blocked Driveway	3.602733
	Noise - Park	3.749091
	Traffic	3.764286
	Animal Abuse	4.424615
	Derelict Vehicle	4.549753
	Vending	5.113333
	Urinating in Public	5.600000
RICHMOND HILL	Noise - House of Worship	9.300000
	Noise - Park	2.242500
	Graffiti	2.430000
	Drinking	2.956667
	Posting Advertisement	3.000000

		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
RIDGEWOOD	Traffic	3.261429
	Illegal Fireworks	3.477500
	Noise - Vehicle	4.076562
	Noise - Commercial	4.089444
	Blocked Driveway	5.052434
	Noise - Street/Sidewalk	5.149195
	Animal Abuse	5.539688
	Illegal Parking	5.778579
	Vending	5.806154
	Urinating in Public	6.292000
	Homeless Encampment	8.084643
	Derelict Vehicle	9.591867
	Posting Advertisement	0.250000
	Graffiti	0.990000
	Illegal Fireworks	1.080000
	Vending	3.052500
	Noise - House of Worship	3.075000
	Bike/Roller/Skate Chronic	3.480000
	Noise - Park	3.592857
	Traffic	3.633571
ROCKAWAY PARK	Noise - Commercial	3.683417
	Disorderly Youth	3.996667
	Blocked Driveway	4.009716
	Noise - Street/Sidewalk	4.128868
	Urinating in Public	4.150000
	Drinking	4.347000
	Noise - Vehicle	4.349862
	Illegal Parking	4.435233
	Homeless Encampment	5.779565
	Animal Abuse	6.428974
ROCKAWAY PARK	Derelict Vehicle	7.576242
	Urinating in Public	0.720000
	Noise - Park	1.025000
	Homeless Encampment	1.717500
	Noise - Street/Sidewalk	1.833704
ROCKAWAY PARK	Noise - Commercial	2.013492
	Noise - Vehicle	2.260370

		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
ROSEDALE	Animal Abuse	2.283000
	Drinking	2.303500
	Blocked Driveway	2.417429
	Illegal Parking	2.620662
	Derelict Vehicle	2.727778
	Vending	2.945000
	Disorderly Youth	3.427500
	Traffic	3.562857
	Graffiti	0.160000
	Noise - House of Worship	4.505000
	Drinking	5.505000
	Noise - Park	5.778406
	Noise - Vehicle	5.845200
	Noise - Street/Sidewalk	5.873750
	Vending	6.798125
	Blocked Driveway	7.457773
	Noise - Commercial	7.758800
	Bike/Roller/Skate Chronic	8.525000
	Illegal Parking	10.225487
	Traffic	10.535217
SAINT ALBANS	Derelict Vehicle	14.507548
	Animal Abuse	14.593939
	Homeless Encampment	18.862500
	Noise - Park	0.820000
	Drinking	0.963333
	Disorderly Youth	1.760000
	Vending	2.390000
	Noise - Commercial	3.100345
	Noise - Street/Sidewalk	3.366582
	Noise - Vehicle	3.458537
	Traffic	3.965455
	Illegal Parking	4.184309
	Noise - House of Worship	4.650000
	Blocked Driveway	4.686844
	Derelict Vehicle	5.900545
	Urinating in Public	6.510000
	Animal Abuse	7.372000



		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
SOUTH OZONE PARK	Homeless Encampment	7.570000
	Illegal Fireworks	0.510000
	Posting Advertisement	1.290000
	Urinating in Public	1.695000
	Homeless Encampment	1.835000
	Noise - House of Worship	2.160000
	Disorderly Youth	2.685000
	Noise - Commercial	3.339571
	Animal Abuse	3.492000
	Vending	3.520000
	Traffic	3.556786
	Noise - Street/Sidewalk	3.660476
	Noise - Vehicle	3.793765
	Bike/Roller/Skate Chronic	4.060000
	Blocked Driveway	4.427643
	Illegal Parking	4.695729
	Drinking	4.880769
	Noise - Park	5.707500
	Derelict Vehicle	10.398687
SOUTH RICHMOND HILL	Illegal Fireworks	0.475000
	Noise - House of Worship	1.983333
	Disorderly Youth	3.065000
	Noise - Park	3.605000
	Noise - Vehicle	3.607407
	Noise - Commercial	3.873586
	Noise - Street/Sidewalk	4.121868
	Animal Abuse	4.366923
	Blocked Driveway	4.866428
	Vending	5.267917
	Homeless Encampment	5.335455
	Illegal Parking	5.708247
	Drinking	5.766522
	Traffic	5.776364
	Derelict Vehicle	11.877024
SPRINGFIELD GARDENS	Bike/Roller/Skate Chronic	30.910000
	Noise - Park	1.670000
	Noise - House of Worship	3.710000

		Avg. Time(Given City, Complaint Type)
City	Complaint Type	
STATEN ISLAND	Noise - Commercial	3.724722
	Homeless Encampment	3.778000
	Noise - Street/Sidewalk	4.217368
	Urinating in Public	4.790000
	Drinking	4.846667
	Noise - Vehicle	5.045476
	Illegal Fireworks	5.560000
	Panhandling	6.350000
	Vending	6.850000
	Blocked Driveway	9.035725
	Illegal Parking	9.178067
	Traffic	10.343636
	Derelict Vehicle	11.669619
	Animal Abuse	14.861250
	Posting Advertisement	19.505000
	Posting Advertisement	1.542117
	Urinating in Public	2.478571
	Noise - House of Worship	2.502941
	Noise - Park	2.933134
	Noise - Street/Sidewalk	2.969044
SUNNYSIDE	Noise - Commercial	3.000517
	Noise - Vehicle	3.237753
	Drinking	3.493200
	Traffic	3.567500
	Illegal Fireworks	3.735000
	Illegal Parking	3.846005
	Disorderly Youth	3.893913
	Blocked Driveway	4.071195
	Bike/Roller/Skate Chronic	4.078571
	Vending	4.272000
	Panhandling	4.670833
	Animal Abuse	4.969587
	Homeless Encampment	4.980141
	Derelict Vehicle	5.039490
	Graffiti	9.560000
	Graffiti	0.590000
	Bike/Roller/Skate Chronic	1.205000

City	Avg. Time(Given City, Complaint Type)	
	Complaint Type	
	Urinating in Public	3.405000
	Disorderly Youth	3.740000
	Drinking	4.606000
	Noise - Vehicle	4.923125
	Traffic	5.951875
	Noise - Street/Sidewalk	6.232000
	Noise - Park	6.464667
	Noise - Commercial	6.622174
	Illegal Parking	6.665492
	Blocked Driveway	6.968398
	Homeless Encampment	7.190000
	Posting Advertisement	7.270000
	Vending	9.724000
	Derelict Vehicle	9.829000
	Animal Abuse	11.553143

5.Perform a statistical test for the following

In [69]: `import scipy.stats as stat`

```
In [70]: Complaint_AvgTime = C_CT_RCT.groupby(['Complaint Type']).agg({'DTime':'mean'})
Complaint_AvgTime = pd.DataFrame(Complaint_AvgTime)
Complaint_AvgTime = Complaint_AvgTime.sort_values(['DTime']).reset_index()
Complaint_AvgTime
```

Out[70]:

	Complaint Type	DTime
0	Posting Advertisement	1.975926
1	Illegal Fireworks	2.761190
2	Noise - Commercial	3.136907
3	Noise - House of Worship	3.193240
4	Noise - Park	3.401706
5	Noise - Street/Sidewalk	3.438573
6	Traffic	3.446291
7	Disorderly Youth	3.558916
8	Noise - Vehicle	3.588570
9	Urinating in Public	3.626486
10	Bike/Roller/Skate Chronic	3.756611
11	Drinking	3.855354
12	Vending	4.013619
13	Squeegee	4.047500
14	Homeless Encampment	4.366029
15	Panhandling	4.372852
16	Illegal Parking	4.486005
17	Blocked Driveway	4.738187
18	Animal Abuse	5.213471
19	Graffiti	7.151062
20	Derelict Vehicle	7.346105
21	Animal in a Park	336.830000

```
In [72]: Tmean_without = float(Complaint_AvgTime[Complaint_AvgTime['Complaint Type']!= 'Animal in a Park'].mean())
print("Without complaint type 'Animal in a Park' ----- ",Tmean_without)
Tmean_with = float(Complaint_AvgTime['DTime'].mean())
print("With complaint type 'Animal in a Park' ----- ",Tmean_with)
```

```
Without complaint type 'Animal in a Park' -----  4.070219157949681
With complaint type 'Animal in a Park' -----  19.19566374167924
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.
    """Entry point for launching an IPython kernel.
```

```
In [73]: ttest_with, pval_with = stat.ttest_1samp(Complaint_AvgTime['DTime'], Tmean_with)
print('T-statistic is ', ttest_with)
print('p value is ', np.around(pval_with))
```

```
T-statistic is = 0.0
p value is = 1.0
```

```
In [74]: if (pval_with < 0.05):
print('Null hypothesis is rejected since p value ({} ) is less than 0.05'.format(np.around(pval_with, decimals=2)))
else:
print('Null hypothesis is accepted since p value ({} ) is greater than 0.05'.format(np.around(pval_with, decimals=2)))
```

```
Null hypothesis is accepted since p value (1.0) is greater than 0.05
```

```
In [75]: Complaint_AvgTime_without = Complaint_AvgTime.drop([len(Complaint_AvgTime)-1], axis=0)
Complaint_AvgTime_without
```

Out[75]:

	Complaint Type	DTime
0	Posting Advertisement	1.975926
1	Illegal Fireworks	2.761190
2	Noise - Commercial	3.136907
3	Noise - House of Worship	3.193240
4	Noise - Park	3.401706
5	Noise - Street/Sidewalk	3.438573
6	Traffic	3.446291
7	Disorderly Youth	3.558916
8	Noise - Vehicle	3.588570
9	Urinating in Public	3.626486
10	Bike/Roller/Skate Chronic	3.756611
11	Drinking	3.855354
12	Vending	4.013619
13	Squeegee	4.047500
14	Homeless Encampment	4.366029
15	Panhandling	4.372852
16	Illegal Parking	4.486005
17	Blocked Driveway	4.738187
18	Animal Abuse	5.213471
19	Graffiti	7.151062
20	Derelict Vehicle	7.346105

```
In [76]: ttest_without, pval_without = stat.ttest_1samp(Complaint_AvgTime_without
['DTime'], Tmean_without)
print('T-statistic is ', ttest_without)
print('p value is ', np.around(pval_without, decimals=8))
```

```
T-statistic is = 0.0
p value is = 1.0
```

```
In [77]: if (pval_without < 0.05):
        print('Null hypothesis is rejected since p value ({} ) is less than 0.
05'.format(np.around(pval_without, decimals=2)))
    else:
        print('Null hypothesis is accepted since p value ({} ) is greater than
0.05'.format(np.around(pval_without, decimals=2)))
```

Null hypothesis is accepted since p value (1.0) is greater than 0.05

```
In [78]: sample1 = Complaint_AvgTime.sample(frac=.5)
sample1
```

Out[78]:

	Complaint Type	DTime
8	Noise - Vehicle	3.588570
19	Graffiti	7.151062
5	Noise - Street/Sidewalk	3.438573
2	Noise - Commercial	3.136907
20	Derelict Vehicle	7.346105
3	Noise - House of Worship	3.193240
10	Bike/Roller/Skate Chronic	3.756611
16	Illegal Parking	4.486005
4	Noise - Park	3.401706
6	Traffic	3.446291
17	Blocked Driveway	4.738187

```
In [79]: sample2 = Complaint_AvgTime.drop(sample1.index)
sample2
```

Out[79]:

	Complaint Type	DTime
0	Posting Advertisement	1.975926
1	Illegal Fireworks	2.761190
7	Disorderly Youth	3.558916
9	Urinating in Public	3.626486
11	Drinking	3.855354
12	Vending	4.013619
13	Squeegee	4.047500
14	Homeless Encampment	4.366029
15	Panhandling	4.372852
18	Animal Abuse	5.213471
21	Animal in a Park	336.830000

```
In [80]: print('Mean of 1st sample =',np.around(float(sample1['DTime'].mean()),dec
imals=2))
print('Standard dev. of 1st sample =',np.around(float(sample1['DTime'].st
d()),decimals=2))
print('Mean of 2nd sample =',np.around(float(sample2['DTime'].mean()),dec
imals=2))
print('Standard dev. of 2nd sample =',np.around(float(sample2['DTime'].st
d()),decimals=2))
```

```
Mean of 1st sample = 4.33
Standard dev. of 1st sample = 1.53
Mean of 2nd sample = 34.06
Standard dev. of 2nd sample = 100.42
```

```
In [81]: ttest_2sp, p_val = stat.ttest_ind(sample1['DTime'],sample2['DTime'])
print('T-statistic is ',ttest_2sp)
print('p value is ',np.around(p_val,decimals=2))
```

```
T-statistic is = -0.9814975717515023
p value is = 0.34
```

```
In [82]: if (p_val<0.05):
print('Null hypothesis is rejected since p value ({} ) is less than 0.
05'.format(np.around(p_val,decimals=2)))
else:
print('Null hypothesis is accepted since p value ({} ) is greater than
0.05'.format(np.around(p_val,decimals=2)))
```

```
Null hypothesis is accepted since p value (0.34) is greater than 0.05
```

```
In [83]: sample1_anova = Complaint_AvgTime.sample(frac=1/3)
sample1_anova
```

Out[83]:

	Complaint Type	DTime
21	Animal in a Park	336.830000
9	Urinating in Public	3.626486
7	Disorderly Youth	3.558916
1	Illegal Fireworks	2.761190
11	Drinking	3.855354
12	Vending	4.013619
14	Homeless Encampment	4.366029

```
In [84]: rest_data = Complaint_AvgTime.drop(sample1_anova.index)
rest_data
```

Out[84]:

	Complaint Type	DTime
0	Posting Advertisement	1.975926
2	Noise - Commercial	3.136907
3	Noise - House of Worship	3.193240
4	Noise - Park	3.401706
5	Noise - Street/Sidewalk	3.438573
6	Traffic	3.446291
8	Noise - Vehicle	3.588570
10	Bike/Roller/Skate Chronic	3.756611
13	Squeegee	4.047500
15	Panhandling	4.372852
16	Illegal Parking	4.486005
17	Blocked Driveway	4.738187
18	Animal Abuse	5.213471
19	Graffiti	7.151062
20	Derelict Vehicle	7.346105



```
In [85]: sample2_anova = rest_data.sample(frac=1/2)
sample2_anova
```

Out[85]:

	Complaint Type	DTime
20	Derelict Vehicle	7.346105
3	Noise - House of Worship	3.193240
19	Graffiti	7.151062
13	Squeegee	4.047500
10	Bike/Roller/Skate Chronic	3.756611
2	Noise - Commercial	3.136907
16	Illegal Parking	4.486005
0	Posting Advertisement	1.975926

```
In [86]: sample3_anova = rest_data.drop(sample2_anova.index)
sample3_anova
```

Out[86]:

	Complaint Type	DTime
4	Noise - Park	3.401706
5	Noise - Street/Sidewalk	3.438573
6	Traffic	3.446291
8	Noise - Vehicle	3.588570
15	Panhandling	4.372852
17	Blocked Driveway	4.738187
18	Animal Abuse	5.213471

```
In [87]: print('Mean of 1st sample = ',np.around(float(sample1_anova['DTime'].mean
()),decimals=2))
print('Standard dev. of 1st sample = ',np.around(float(sample1_anova['DTime'].std()),decimals=2))
print('Mean of 2nd sample = ',np.around(float(sample2_anova['DTime'].mean
()),decimals=2))
print('Standard dev. of 2nd sample = ',np.around(float(sample2_anova['DTime'].std()),decimals=2))
print('Mean of 3rd sample = ',np.around(float(sample3_anova['DTime'].mean
()),decimals=2))
print('Standard dev. of 3rd sample = ',np.around(float(sample3_anova['DTime'].std()),decimals=2))
```

```
Mean of 1st sample = 51.29
Standard dev. of 1st sample = 125.91
Mean of 2nd sample = 4.39
Standard dev. of 2nd sample = 1.92
Mean of 3rd sample = 4.03
Standard dev. of 3rd sample = 0.74
```

```
In [88]: f_val,p_val = stat.shapiro(sample1_anova['DTime'])
print('F-statistic is ',f_val)
print('p value is ',np.around(p_val,decimals=2))

F-statistic is = 0.45682263374328613
p value is = 0.0
```

```
In [89]: f_val,p_val = stat.shapiro(sample2_anova['DTime'])
print('F-statistic is ',f_val)
print('p value is ',np.around(p_val,decimals=2))

F-statistic is = 0.8815967440605164
p value is = 0.2
```

```
In [90]: f_val,p_val = stat.shapiro(sample3_anova['DTime'])
print('F-statistic is ',f_val)
print('p value is ',np.around(p_val,decimals=2))

F-statistic is = 0.8285542726516724
p value is = 0.08
```

```
In [91]: f_val,p_val = stat.shapiro(sample3_anova['DTime'])
print('F-statistic is ',f_val)
print('p value is ',np.around(p_val,decimals=2))

F-statistic is = 0.8285542726516724
p value is = 0.08
```

```
In [92]: f_val,p_val = stat.f_oneway(sample1_anova['DTime'],sample2_anova['DTime'],
sample3_anova['DTime'])
print('F-statistic is ',f_val)
print('p value is ',np.around(p_val,decimals=2))

F-statistic is = 1.0556755952554207
p value is = 0.37
```

```
In [93]: if (p_val<0.05):
print('Null hypothesis is rejected since p value ({} ) is less than 0.05'.format(np.around(p_val,decimals=2)))
else:
print('Null hypothesis is accepted since p value ({} ) is greater than 0.05'.format(np.around(p_val,decimals=2)))

Null hypothesis is accepted since p value (0.37) is greater than 0.05
```

```
In [94]: t_val,p_val = stat.ttest_ind(sample1_anova['DTime'],sample2_anova['DTime'])
print('T-statistic for sample 1 and 2 is =',t_val)
print('p value is =',np.around(p_val,decimals=2))
t_val,p_val = stat.ttest_ind(sample1_anova['DTime'],sample3_anova['DTime'])
print('T-statistic for sample 1 and 3 is =',t_val)
print('p value is =',np.around(p_val,decimals=2))
t_val,p_val = stat.ttest_ind(sample2_anova['DTime'],sample3_anova['DTime'])
print('T-statistic for sample 2 and 3 is =',t_val)
print('p value is =',np.around(p_val,decimals=2))
```

T-statistic for sample 1 and 2 is = 1.0592366047297563  
p value is = 0.31  
T-statistic for sample 1 and 3 is = 0.9930075351702778  
p value is = 0.34  
T-statistic for sample 2 and 3 is = 0.46316590585416006  
p value is = 0.65

```
In [95]: print('Null data in Complaint Type =',data['Complaint Type'].isnull().sum())
print('Null data in City =',data['City'].isnull().sum())
```

Null data in Complaint Type = 0  
Null data in City = 2614

```
In [96]: df_cc = data[['Complaint Type','City']]
df_cc = df_cc.dropna()
```

```
In [97]: City_Complaint = pd.crosstab(data['Complaint Type'],data['City'],margins=True, margins_name='Total')
City_Complaint.head(6)
```

Out[97]:

	City	ARVERNE	ASTORIA	Astoria	BAYSIDE	BELLEROSE	BREEZY POINT	BRONX	BROOKLYN
Complaint Type									
Animal Abuse		38	125	0	37	7	2	1415	
Animal in a Park		0	0	0	0	0	0	0	
Bike/Roller /Skate Chronic		0	15	0	0	1	0	20	
Blocked Driveway		35	2618	116	377	95	3	12755	
Derelict Vehicle		27	351	12	198	89	3	1953	
Disorderly Youth		2	3	0	1	2	0	63	

```
In [98]: print("For 'ARVERNE' and 'ASTORIA' pair -----")
f_val,p_val = stat.f_oneway(City_Complaint['ARVERNE'],City_Complaint['ASTORIA'])
print('F-statistic is ',f_val)
print('p value is ',np.around(p_val,decimals=2))
print("For 'ARVERNE' and 'BROOKLYN' pair -----")
f_val,p_val = stat.f_oneway(City_Complaint['ARVERNE'],City_Complaint['BROOKLYN'])
print('F-statistic is ',f_val)
print('p value is ',np.around(p_val,decimals=2))
print("For 'HOLLIS' and 'JAMAICA' pair -----")
f_val,p_val = stat.f_oneway(City_Complaint['HOLLIS'],City_Complaint['JAMAICA'])
print('F-statistic is ',f_val)
print('p value is ',np.around(p_val,decimals=2))
print("For 'MASPETH' and 'QUEENS' pair -----")
f_val,p_val = stat.f_oneway(City_Complaint['MASPETH'],City_Complaint['QUEENS'])
print('F-statistic is ',f_val)
print('p value is ',np.around(p_val,decimals=2))
```

```
For 'ARVERNE' and 'ASTORIA' pair -----
F-statistic is = 3.3097701947747975
p value is = 0.08
For 'ARVERNE' and 'BROOKLYN' pair -----
F-statistic is = 3.716772993046823
p value is = 0.06
For 'HOLLIS' and 'JAMAICA' pair -----
F-statistic is = 2.666621070410633
p value is = 0.11
For 'MASPETH' and 'QUEENS' pair -----
F-statistic is = 3.368313812374042
p value is = 0.07
```

```
In [99]: chai2, p_val, df, exp_frq = stat.chi2_contingency(City_Complaint)
print('Chai square value =',chai2)
print('p-value is ',p_val)
```

```
Chai square value = 119769.34666374495
p-value is = 0.0
```

```
In [100]: if (p_val<0.05):
            print('Null hypothesis is rejected since p value ({} ) is less than 0.05'.format(np.around(p_val,decimals=2)))
        else:
            print('Null hypothesis is accepted since p value ({} ) is greater than 0.05'.format(np.around(p_val,decimals=2)))
```

```
Null hypothesis is rejected since p value (0.0) is less than 0.05
```