

# A\_1

March 4, 2022

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
import warnings
warnings.filterwarnings('ignore')
```

1. Import a 311 NYC service request.

```
[2]: df = pd.read_csv('311_Service_Requests_from_2010_to_Present.csv')
df.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

```

```
[3]: df.shape
```

```
[3]: (300698, 53)
```

```
[4]: df.head()
```

```

[4]: Unique Key      Created Date      Closed Date Agency \
0      32310363  12/31/2015  11:59:45 PM  01-01-16 0:55  NYPD
1      32309934  12/31/2015  11:59:44 PM  01-01-16 1:26  NYPD
2      32309159  12/31/2015  11:59:29 PM  01-01-16 4:51  NYPD
3      32305098  12/31/2015  11:57:46 PM  01-01-16 7:43  NYPD

```

4 32306529 12/31/2015 11:56:58 PM 01-01-16 3:24 NYPD

	Agency Name	Complaint Type \
0	New York City Police Department	Noise - Street/Sidewalk
1	New York City Police Department	Blocked Driveway
2	New York City Police Department	Blocked Driveway
3	New York City Police Department	Illegal Parking
4	New York City Police Department	Illegal Parking

	Descriptor	Location Type	Incident Zip \
0	Loud Music/Party	Street/Sidewalk	10034.0
1	No Access	Street/Sidewalk	11105.0
2	No Access	Street/Sidewalk	10458.0
3	Commercial Overnight Parking	Street/Sidewalk	10461.0
4	Blocked Sidewalk	Street/Sidewalk	11373.0

	Incident Address	... Bridge Highway Name	Bridge Highway Direction \
0	71 VERMILYEA AVENUE	...	NaN NaN
1	27-07 23 AVENUE	...	NaN NaN
2	2897 VALENTINE AVENUE	...	NaN NaN
3	2940 BAISLEY AVENUE	...	NaN NaN
4	87-14 57 ROAD	...	NaN NaN

	Road Ramp Bridge Highway Segment	Garage Lot Name	Ferry Direction \
0	NaN	NaN	NaN NaN
1	NaN	NaN	NaN NaN
2	NaN	NaN	NaN NaN
3	NaN	NaN	NaN NaN
4	NaN	NaN	NaN NaN

	Ferry Terminal Name	Latitude	Longitude \
0	NaN	40.865682	-73.923501
1	NaN	40.775945	-73.915094
2	NaN	40.870325	-73.888525
3	NaN	40.835994	-73.828379
4	NaN	40.733060	-73.874170

	Location
0	(40.86568153633767, -73.92350095571744)
1	(40.775945312321085, -73.91509393898605)
2	(40.870324522111424, -73.88852464418646)
3	(40.83599404683083, -73.82837939584206)
4	(40.733059618956815, -73.87416975810375)

[5 rows x 53 columns]

```
[5]: df.isnull().sum()[df.isnull().sum()>0]
```

```
[5]: Closed Date                2164
      Descriptor                5914
      Location Type             131
      Incident Zip              2615
      Incident Address          44410
      Street Name               44410
      Cross Street 1            49279
      Cross Street 2            49779
      Intersection Street 1     256840
      Intersection Street 2     257336
      Address Type              2815
      City                      2614
      Landmark                  300349
      Facility Type             2171
      Due Date                   3
      Resolution Action Updated Date 2187
      X Coordinate (State Plane) 3540
      Y Coordinate (State Plane) 3540
      School Region             1
      School Code               1
      School Zip                1
      School or Citywide Complaint 300698
      Vehicle Type              300698
      Taxi Company Borough      300698
      Taxi Pick Up Location      300698
      Bridge Highway Name       300455
      Bridge Highway Direction  300455
      Road Ramp                 300485
      Bridge Highway Segment    300485
      Garage Lot Name           300698
      Ferry Direction           300697
      Ferry Terminal Name       300696
      Latitude                   3540
      Longitude                  3540
      Location                   3540
      dtype: int64
```

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)

```
[6]: import datetime
      from datetime import date
```

```
[7]: df['Created Date'] = df['Created Date'].astype('datetime64[ns]')
      df['Closed Date'] = df['Closed Date'].astype('datetime64[ns]')
```

```
[8]: df.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 datetime64[ns]
2   Closed Date                           298534 non-null datetime64[ns]
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: datetime64[ns](2), float64(10), int64(1), object(40)
memory usage: 121.6+ MB

```

```
[9]: df['Request_Closing_Time'] = df['Closed Date'] - df['Created Date']
df.head()
```

```
[9]: Unique Key      Created Date      Closed Date Agency \
0      32310363 2015-12-31 23:59:45 2016-01-01 00:55:00  NYPD
1      32309934 2015-12-31 23:59:44 2016-01-01 01:26:00  NYPD
2      32309159 2015-12-31 23:59:29 2016-01-01 04:51:00  NYPD
3      32305098 2015-12-31 23:57:46 2016-01-01 07:43:00  NYPD
4      32306529 2015-12-31 23:56:58 2016-01-01 03:24:00  NYPD
```

```

Agency Name      Complaint Type \
0 New York City Police Department Noise - Street/Sidewalk
1 New York City Police Department Blocked Driveway
2 New York City Police Department Blocked Driveway
3 New York City Police Department Illegal Parking
4 New York City Police Department Illegal Parking

```

```

Descriptor      Location Type Incident Zip \
0 Loud Music/Party Street/Sidewalk 10034.0
1 No Access Street/Sidewalk 11105.0
2 No Access Street/Sidewalk 10458.0
3 Commercial Overnight Parking Street/Sidewalk 10461.0
4 Blocked Sidewalk Street/Sidewalk 11373.0

```

```

Incident Address ... Bridge Highway Direction Road Ramp \
0 71 VERMILYEA AVENUE ... NaN NaN
1 27-07 23 AVENUE ... NaN NaN
2 2897 VALENTINE AVENUE ... NaN NaN
3 2940 BAISLEY AVENUE ... NaN NaN
4 87-14 57 ROAD ... NaN NaN

```

	Bridge	Highway	Segment	Garage	Lot	Name	Ferry	Direction	Ferry	Terminal	Name	\
0				NaN		NaN			NaN		NaN	
1				NaN		NaN			NaN		NaN	
2				NaN		NaN			NaN		NaN	
3				NaN		NaN			NaN		NaN	
4				NaN		NaN			NaN		NaN	

	Latitude	Longitude	Location	\
0	40.865682	-73.923501	(40.86568153633767, -73.92350095571744)	
1	40.775945	-73.915094	(40.775945312321085, -73.91509393898605)	
2	40.870325	-73.888525	(40.870324522111424, -73.88852464418646)	
3	40.835994	-73.828379	(40.83599404683083, -73.82837939584206)	
4	40.733060	-73.874170	(40.733059618956815, -73.87416975810375)	

	Request_Closing_Time
0	0 days 00:55:15
1	0 days 01:26:16
2	0 days 04:51:31
3	0 days 07:45:14
4	0 days 03:27:02

[5 rows x 54 columns]

```
[10]: df['Complaint Type'].value_counts()
```

```
[10]: 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
```

```
[11]: Complaint_Map = {'Noise - Street/Sidewalk' : 'Noise', 'Noise - Commercial' : 'Noise',
    ↪ 'Noise',
    'Noise - Park': 'Noise', 'Noise - House of Worship': 'Noise', 'Agency Issues':
    ↪ 'Other',
    'Squeegee': 'Other', 'Ferry Complaint': 'Other', 'Animal Abuse': 'Animal Abuse',
    'Illegal Fireworks': 'Noise', 'Graffiti': 'DnD', 'Drinking': 'DnD', 'Noise - Vehicle':
    ↪ 'Noise',
    'Urinating in Public': 'DnD', 'Bike/Roller/Skate Chronic': 'DnD', 'Blocked Driveway':
    ↪ 'Blocked Driveway',
    'Illegal Parking': 'Illegal Parking', 'Derelict Vehicle': 'Derelict Vehicle',
    ↪ 'Vending': 'Vending',
    'Traffic' : 'Traffic', 'Homeless Encampment': 'poor', 'Panhandling': 'poor',
    ↪ 'Disorderly Youth': 'DnD',
    'Posting Advertisement': 'Posting Advertisement'}
```

```
[12]: df['Complaint_short'] = df['Complaint Type'].map(Complaint_Map)
```

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.

```
[13]: #Univariate Analysis
```

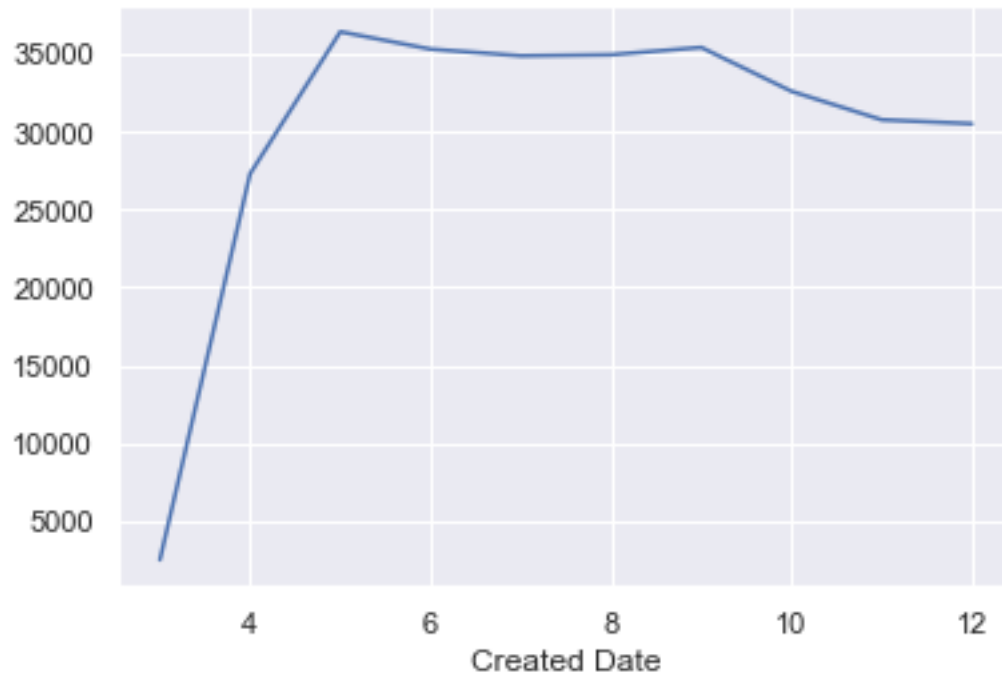
```
[14]: Report_months = df.groupby(pd.Grouper(key='Created Date', freq='M')).size()
Report_months.index = Report_months.index.month
Report_months
```

```
[14]: Created Date
3      2471
4      27305
5      36437
6      35315
7      34888
8      34956
9      35427
10     32605
11     30773
12     30521
dtype: int64
```

```
[15]: Report_months.plot()
```

```
[15]: <AxesSubplot:xlabel='Created Date'>
```





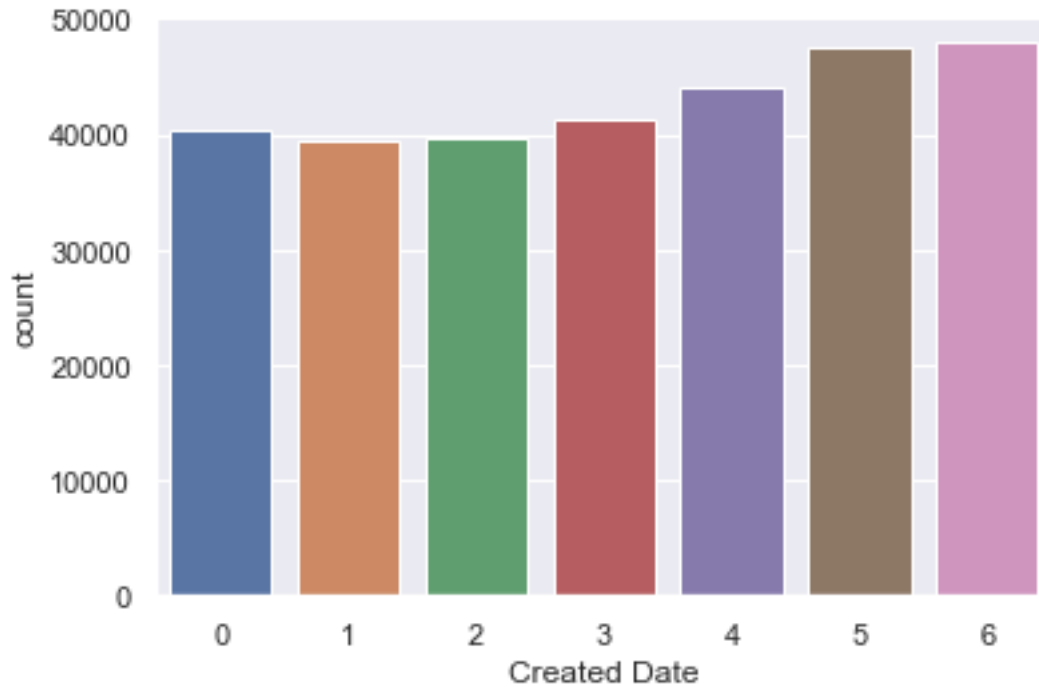
Report declines after 9th month

```
[16]: week = df['Created Date'].apply(lambda x: x.weekday())
      week
```

```
[16]: 0      3
      1      3
      2      3
      3      3
      4      3
      ..
      300693  6
      300694  6
      300695  6
      300696  6
      300697  6
      Name: Created Date, Length: 300698, dtype: int64
```

```
[17]: sns.countplot(x=week.index, data=week)
```

```
[17]: <AxesSubplot:xlabel='Created Date', ylabel='count'>
```



Number of Reports increase during weekends

```
[18]: #Bivariate Analysis
```

```
[19]: df['month'] = df['Created Date'].apply(lambda x: x.month)
df['hour'] = df['Created Date'].apply(lambda x: x.hour)
df['week'] = week
```

```
[20]: day_reports = df.groupby(['week' , 'Complaint_short']).size().unstack()
day_reports
```

```
[20]: Complaint_short  Animal Abuse  Blocked Driveway  Derelict Vehicle  DnD \
week
0                1196.0        11303.0        3132.0  363.0
1                1165.0        11287.0        2929.0  361.0
2                1250.0        10615.0        3032.0  357.0
3                1129.0        11120.0        2948.0  362.0
4                1102.0        10870.0        2464.0  403.0
5                 985.0        10706.0        1607.0  417.0
6                 951.0        11143.0        1606.0  435.0

Complaint_short  Illegal Parking  Noise  Other  Posting Advertisement \
week
0                11346.0  11281.0    2.0                93.0
```

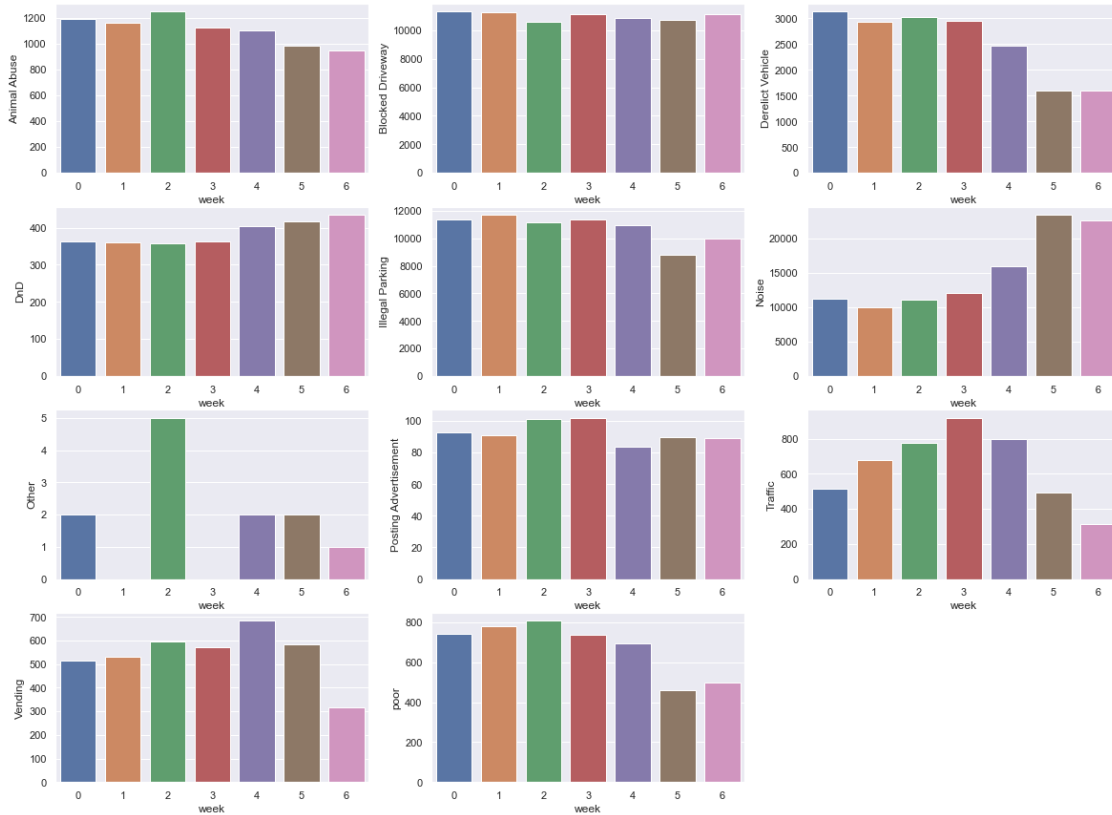
1	11711.0	10015.0	NaN	91.0
2	11159.0	11089.0	5.0	101.0
3	11385.0	12066.0	NaN	102.0
4	10946.0	15947.0	2.0	84.0
5	8812.0	23405.0	2.0	90.0
6	10002.0	22610.0	1.0	89.0

	Complaint_short	Traffic	Vending	poor
week				
0		517.0	515.0	741.0
1		680.0	531.0	781.0
2		776.0	597.0	807.0
3		920.0	573.0	737.0
4		798.0	683.0	696.0
5		492.0	585.0	462.0
6		315.0	318.0	499.0

```
[21]: Cols = day_reports.columns
      Cols
```

```
[21]: Index(['Animal Abuse', 'Blocked Driveway', 'Derelict Vehicle', 'DnD',
            'Illegal Parking', 'Noise', 'Other', 'Posting Advertisement', 'Traffic',
            'Vending', 'poor'],
            dtype='object', name='Complaint_short')
```

```
[22]: fig = plt.figure(figsize=(20,15))
      for index,col in enumerate(Cols):
          plt.subplot(4,3,index+1)
          sns.barplot(x=day_reports.index, y=col, data=day_reports)
```



1. Weekdays' highest reports are due to Animal Abuse, Blocked Driveway, Derelict Vehicle, Illegal Parking, Posting Advertisement, Vending, poor
2. Weekends' highest reports are from Noise
3. Traffic highest on Thursday and lowest on Sunday
4. DnD (such as Drinking, Bike/Roller/Skate Chronic) higher on weekends

```
[23]: month_report = df.groupby(['month' , 'Complaint_short']).size().unstack()
month_report
```

```
[23]: Complaint_short  Animal Abuse  Blocked Driveway  Derelict Vehicle  DnD \
month
3                    73.0          720.0          206.0    20.0
4                   747.0         6867.0         1672.0   200.0
5                   845.0         8103.0         1973.0   364.0
6                   953.0         8583.0         2074.0   352.0
7                  1043.0         7663.0         2109.0   382.0
8                   924.0         7765.0         2019.0   403.0
9                   881.0         8703.0         2104.0   386.0
10                  821.0         9435.0         1871.0   229.0
11                  702.0         9664.0         1795.0   194.0
12                  789.0         9541.0         1895.0   168.0
```

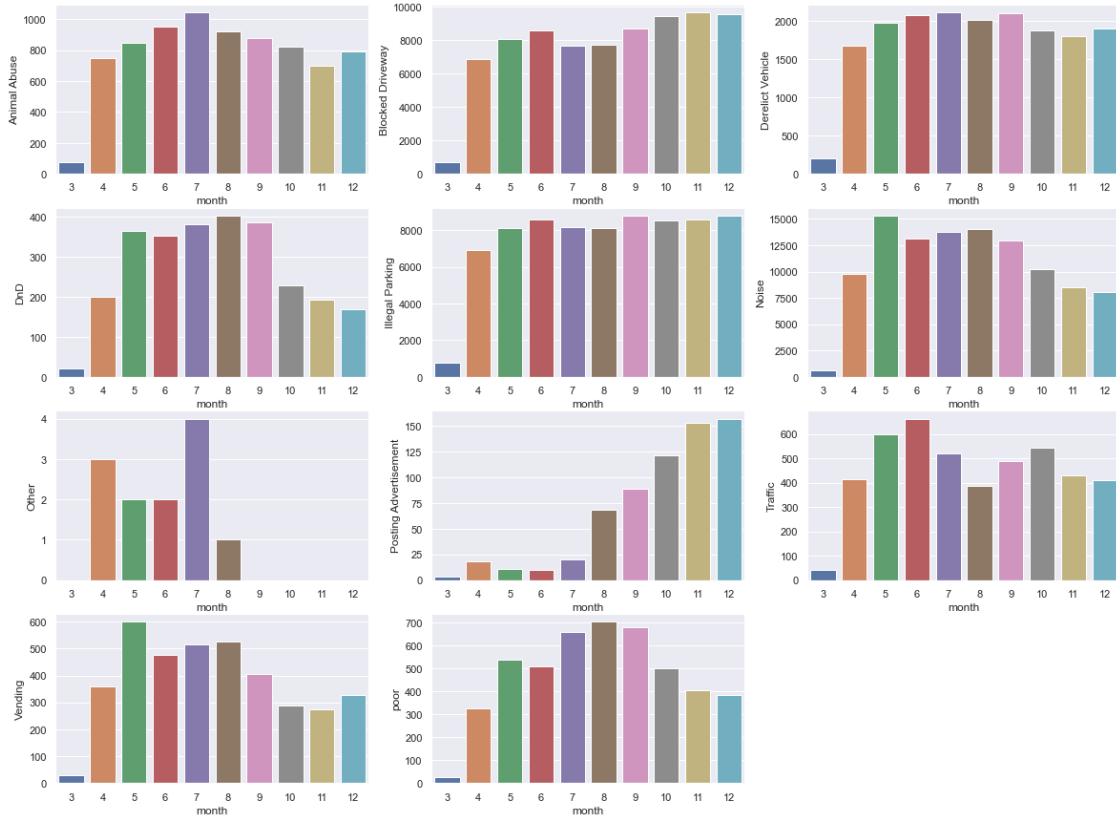
Complaint_short	Illegal Parking	Noise	Other	Posting Advertisement	\
month					
3	750.0	603.0	NaN		3.0
4	6918.0	9780.0	3.0		18.0
5	8105.0	15300.0	2.0		11.0
6	8604.0	13088.0	2.0		10.0
7	8185.0	13786.0	4.0		20.0
8	8102.0	14060.0	1.0		68.0
9	8770.0	12923.0	NaN		89.0
10	8524.0	10271.0	NaN		121.0
11	8611.0	8544.0	NaN		153.0
12	8792.0	8058.0	NaN		157.0

Complaint_short	Traffic	Vending	poor
month			
3	42.0	28.0	26.0
4	413.0	361.0	325.0
5	599.0	599.0	536.0
6	662.0	477.0	510.0
7	521.0	516.0	659.0
8	387.0	525.0	702.0
9	489.0	405.0	677.0
10	543.0	288.0	502.0
11	431.0	276.0	403.0
12	411.0	327.0	383.0

```
[24]: Cols2 = month_report.columns
      Cols2
```

```
[24]: Index(['Animal Abuse', 'Blocked Driveway', 'Derelict Vehicle', 'DnD',
            'Illegal Parking', 'Noise', 'Other', 'Posting Advertisement', 'Traffic',
            'Vending', 'poor'],
            dtype='object', name='Complaint_short')
```

```
[25]: fig = plt.figure(figsize=(20,15))
      for index,col in enumerate(Cols2):
          plt.subplot(4,3,index+1)
          sns.barplot(x=month_report.index, y=col, data=month_report)
```



1. Animal Abuse is highest in July
2. Blocked Driveway peaks from September to December - it is due to holiday season when people travel
3. Illegal Parking and Derelict Vehicle remains high through the months May to December
4. DnD peaks at August
5. Posting Advertisement is highest in November and December - it is due to advertising during Thanksgiving and Christmas Holiday season
6. Traffic is higher in Summer months of May and June when all travel

```
[26]: hour_report = df.groupby(['hour' , 'Complaint_short']).size().unstack()
hour_report
```

```
[26]: Complaint_short  Animal Abuse  Blocked Driveway  Derelict Vehicle  DnD \
hour
0                    137.0          2584.0          171.0  103.0
1                     92.0          1774.0          132.0   64.0
2                     78.0          1170.0           63.0   40.0
3                     44.0           881.0           35.0   30.0
4                     46.0           818.0           41.0   25.0
5                     41.0          1281.0           91.0   24.0
6                     93.0          2385.0          280.0   24.0
```

7	170.0	3856.0	639.0	39.0
8	277.0	4794.0	1133.0	77.0
9	444.0	4742.0	1447.0	88.0
10	465.0	4284.0	1480.0	112.0
11	474.0	3947.0	1451.0	125.0
12	518.0	3653.0	1336.0	121.0
13	560.0	3259.0	1252.0	131.0
14	533.0	3044.0	1208.0	151.0
15	530.0	2976.0	1158.0	167.0
16	529.0	3148.0	1160.0	176.0
17	504.0	3318.0	1068.0	183.0
18	482.0	3824.0	936.0	193.0
19	436.0	4027.0	853.0	192.0
20	407.0	4484.0	626.0	185.0
21	376.0	4640.0	466.0	151.0
22	310.0	4424.0	388.0	169.0
23	232.0	3731.0	304.0	128.0

Complaint_short hour	Illegal Parking	Noise	Other	Posting Advertisement \
0	2358.0	11505.0	NaN	10.0
1	1472.0	8386.0	NaN	3.0
2	984.0	5059.0	NaN	NaN
3	707.0	2981.0	NaN	NaN
4	657.0	2036.0	NaN	NaN
5	1012.0	1144.0	NaN	2.0
6	1889.0	1007.0	1.0	12.0
7	3381.0	989.0	NaN	10.0
8	4397.0	1106.0	2.0	5.0
9	4925.0	1257.0	2.0	13.0
10	4364.0	1314.0	NaN	11.0
11	4469.0	1585.0	NaN	9.0
12	4471.0	1838.0	NaN	13.0
13	3728.0	2031.0	1.0	11.0
14	3422.0	2211.0	1.0	12.0
15	3185.0	2418.0	NaN	8.0
16	3208.0	2926.0	2.0	18.0
17	3339.0	3210.0	1.0	15.0
18	3286.0	4063.0	1.0	9.0
19	3538.0	5165.0	1.0	24.0
20	3707.0	6871.0	NaN	23.0
21	5098.0	9319.0	NaN	63.0
22	4186.0	13072.0	NaN	176.0
23	3578.0	14920.0	NaN	203.0

Complaint_short hour	Traffic	Vending	poor
-------------------------	---------	---------	------

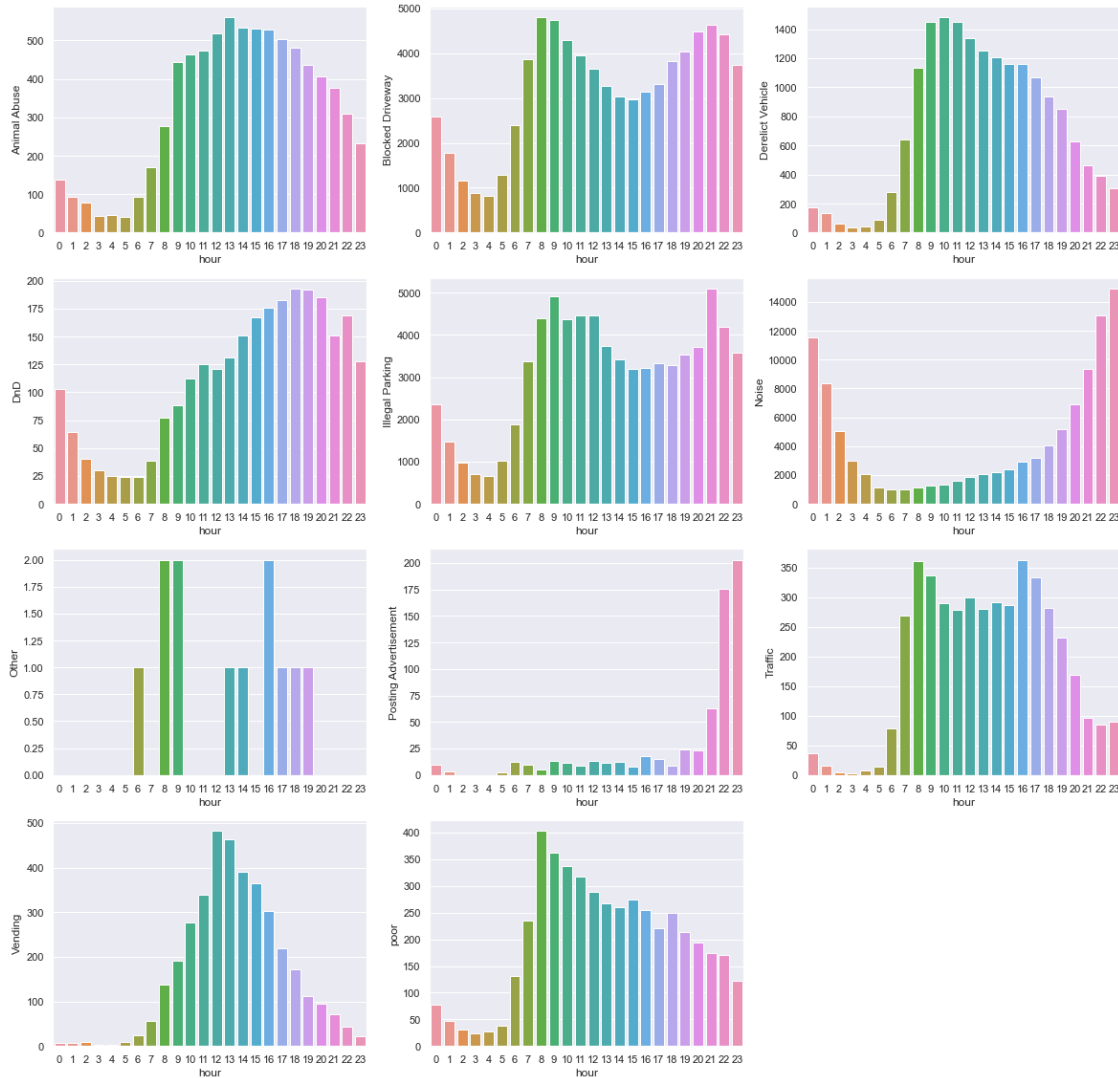
0	36.0	8.0	77.0
1	15.0	8.0	47.0
2	4.0	9.0	31.0
3	3.0	3.0	24.0
4	8.0	2.0	28.0
5	14.0	9.0	39.0
6	79.0	24.0	131.0
7	269.0	57.0	235.0
8	360.0	137.0	403.0
9	337.0	192.0	362.0
10	289.0	276.0	337.0
11	278.0	339.0	317.0
12	299.0	482.0	288.0
13	280.0	464.0	268.0
14	291.0	391.0	260.0
15	286.0	365.0	274.0
16	363.0	302.0	255.0
17	334.0	218.0	221.0
18	282.0	172.0	250.0
19	232.0	111.0	213.0
20	169.0	95.0	194.0
21	96.0	71.0	175.0
22	85.0	44.0	171.0
23	89.0	23.0	123.0

```
[27]: Cols3 = hour_report.columns
      Cols3
```

```
[27]: Index(['Animal Abuse', 'Blocked Driveway', 'Derelict Vehicle', 'DnD',
            'Illegal Parking', 'Noise', 'Other', 'Posting Advertisement', 'Traffic',
            'Vending', 'poor'],
          dtype='object', name='Complaint_short')
```

```
[28]: fig = plt.figure(figsize=(20,20))
      for index,col in enumerate(Cols3):
          plt.subplot(4,3,index+1)
          sns.barplot(x=hour_report.index, y=col, data=hour_report)
```





1. Animal Abuse is highest in the afternoon hours
2. DnD such as Drinking peaks in the evening hours 6-8PM
3. Noise and Posting Advertisement is reported highest around midnight
4. Blocked Driveway, Derelict Vehicle, Illegal Parking is highest during morning hours 8-10 AM (during Office commencing hours)
5. Traffic is spread through the day 9AM to 5PM due to office, school, people commuting
6. Vending has a normal distribution

```
[29]: import holidays
us_holidays = holidays.UnitedStates(state='NY')
```

```
[30]: no_of_holidays = 0
for ptr in holidays.UnitedStates(state='NY', years = 2015).items():
    print(ptr)
```

```
no_of_holidays +=1
```

```
(datetime.date(2015, 1, 1), "New Year's Day")
(datetime.date(2015, 1, 19), 'Martin Luther King Jr. Day')
(datetime.date(2015, 2, 12), "Lincoln's Birthday")
(datetime.date(2015, 2, 15), 'Susan B. Anthony Day')
(datetime.date(2015, 2, 16), "Washington's Birthday")
(datetime.date(2015, 5, 25), 'Memorial Day')
(datetime.date(2015, 7, 4), 'Independence Day')
(datetime.date(2015, 7, 3), 'Independence Day (Observed)')
(datetime.date(2015, 9, 7), 'Labor Day')
(datetime.date(2015, 10, 12), 'Columbus Day')
(datetime.date(2015, 11, 3), 'Election Day')
(datetime.date(2015, 11, 11), 'Veterans Day')
(datetime.date(2015, 11, 26), 'Thanksgiving')
(datetime.date(2015, 12, 25), 'Christmas Day')
```

```
[31]: reports_in_holidays = df['Created Date'].apply(lambda x: x in us_holidays).sum()
total_reports = len(df)
total_no_of_days = (df['Created Date'][0] - df['Created Date'].iloc[-1]).days
```

```
[32]: round(reports_in_holidays/no_of_holidays)
```

```
[32]: 693
```

```
[33]: round((total_reports - reports_in_holidays) / (total_no_of_days -
↳no_of_holidays))
```

```
[33]: 1106
```

Number of complaint reports in holidays are less compared to non holiday/regular days

```
[34]: location = df.groupby(['Complaint_short', 'Location Type']).size().unstack()
location
```

```
[34]: Location Type      Commercial  House and Store  Park/Playground  \
Complaint_short
Animal Abuse           62.0           93.0           123.0
Blocked Driveway       NaN           NaN           NaN
Derelict Vehicle       NaN           NaN           NaN
DnD                    NaN           NaN           136.0
Illegal Parking        NaN           NaN           NaN
Noise                  NaN           NaN           4049.0
Other                  NaN           NaN           NaN
Posting Advertisement  NaN           NaN           NaN
Traffic                NaN           NaN           NaN
Vending                NaN           NaN           106.0
```

poor	NaN	NaN	359.0
------	-----	-----	-------

Location Type	Parking Lot	Residential Building \
Complaint_short		
Animal Abuse	110.0	227.0
Blocked Driveway	NaN	NaN
Derelect Vehicle	NaN	NaN
DnD	NaN	NaN
Illegal Parking	NaN	NaN
Noise	NaN	NaN
Other	NaN	NaN
Posting Advertisement	7.0	NaN
Traffic	NaN	NaN
Vending	NaN	NaN
poor	NaN	NaN

Location Type	Residential Building/House	Store/Commercial \
Complaint_short		
Animal Abuse	5085.0	522.0
Blocked Driveway	NaN	NaN
Derelect Vehicle	NaN	NaN
DnD	588.0	249.0
Illegal Parking	NaN	NaN
Noise	33.0	18600.0
Other	NaN	NaN
Posting Advertisement	54.0	6.0
Traffic	NaN	NaN
Vending	201.0	432.0
poor	999.0	572.0

Location Type	Street/Sidewalk	Subway Station	Highway \
Complaint_short			
Animal Abuse	1531.0	22.0	NaN
Blocked Driveway	77007.0	NaN	NaN
Derelect Vehicle	17614.0	NaN	14.0
DnD	1324.0	12.0	NaN
Illegal Parking	75326.0	NaN	NaN
Noise	65806.0	NaN	NaN
Other	4.0	NaN	NaN
Posting Advertisement	582.0	NaN	NaN
Traffic	4278.0	NaN	186.0
Vending	3061.0	NaN	NaN
poor	2766.0	NaN	15.0

Location Type	Roadway Tunnel	Vacant Lot	Club/Bar/Restaurant \
Complaint_short			
Animal Abuse	NaN	NaN	NaN

Blocked Driveway	NaN	NaN	NaN
Derelict Vehicle	5.0	77.0	NaN
DnD	NaN	NaN	387.0
Illegal Parking	NaN	NaN	NaN
Noise	NaN	NaN	16973.0
Other	NaN	NaN	NaN
Posting Advertisement	NaN	NaN	NaN
Traffic	29.0	NaN	NaN
Vending	NaN	NaN	NaN
poor	1.0	NaN	NaN

Location Type	House of Worship	Ferry	Terminal	Bridge
Complaint_short				
Animal Abuse	NaN	NaN	NaN	NaN
Blocked Driveway	NaN	NaN	NaN	NaN
Derelict Vehicle	NaN	NaN	NaN	NaN
DnD	NaN	NaN	NaN	NaN
Illegal Parking	NaN	NaN	NaN	NaN
Noise	929.0	NaN	NaN	NaN
Other	NaN	1.0	1.0	NaN
Posting Advertisement	NaN	NaN	NaN	NaN
Traffic	NaN	NaN	NaN	NaN
Vending	NaN	NaN	NaN	NaN
poor	NaN	NaN	NaN	2.0

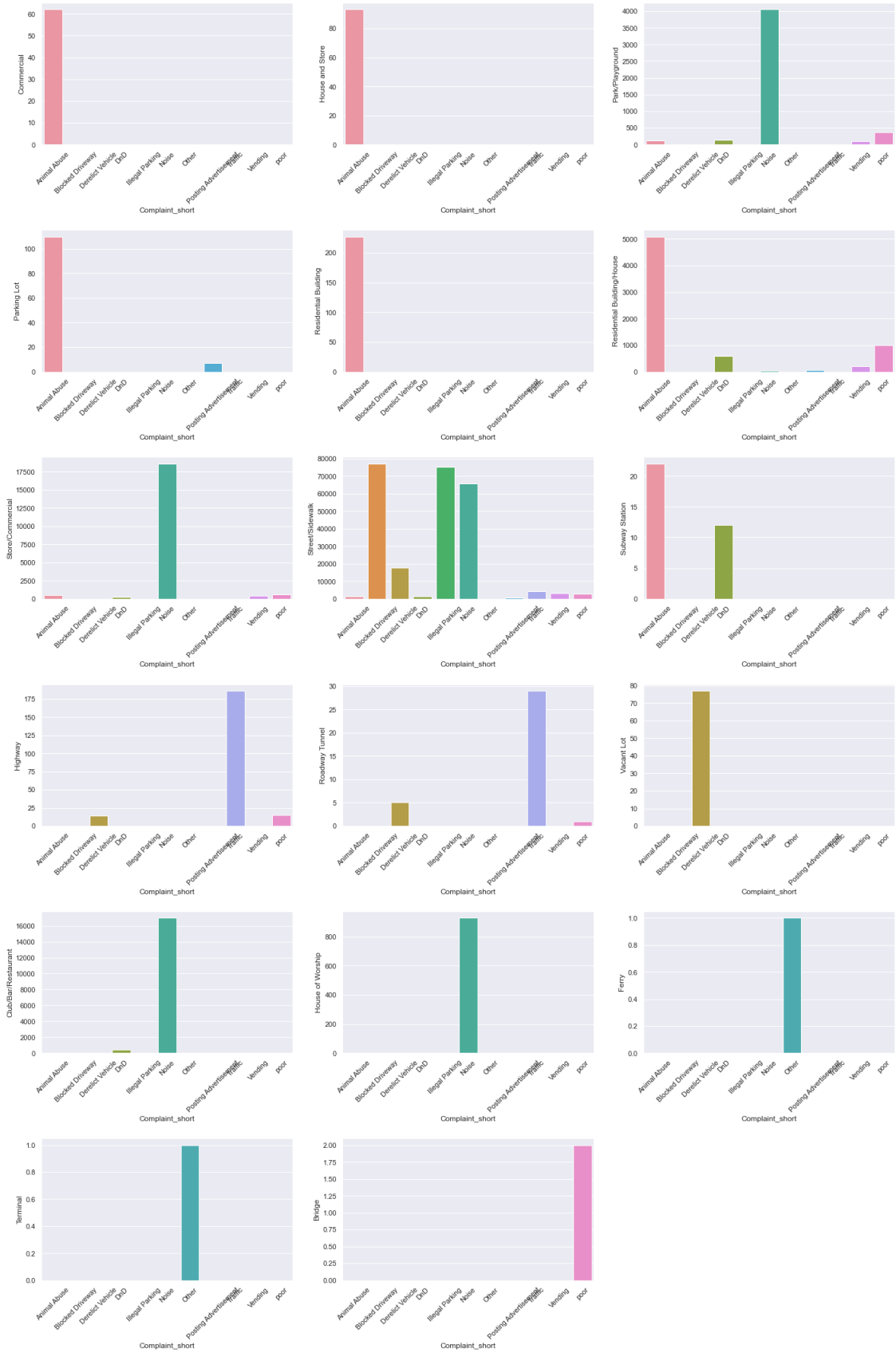
```
[35]: df['Location Type'].value_counts()
```

```
[35]: 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
```

```
[36]: Cols4 = location.columns  
Cols4
```

```
[36]: Index(['Commercial', 'House and Store', 'Park/Playground', 'Parking Lot',  
        'Residential Building', 'Residential Building/House',  
        'Store/Commercial', 'Street/Sidewalk', 'Subway Station', 'Highway',  
        'Roadway Tunnel', 'Vacant Lot', 'Club/Bar/Restaurant',  
        'House of Worship', 'Ferry', 'Terminal', 'Bridge'],  
        dtype='object', name='Location Type')
```

```
[37]: fig = plt.figure(figsize=(20,30))  
for index,col in enumerate(Cols4):  
    plt.subplot(6,3,index+1)  
    sns.barplot(x=location.index, y=col, data=location)  
    plt.xticks(rotation=45)  
fig.tight_layout(pad=2.0)
```



Complaints are very specific to the Location. Street and Sidewalk report various complaints

```
[38]: df.groupby(['Complaint_short', 'City']).size().unstack()
```

```
[38]: City          ARVERNE  ASTORIA  Astoria  BAYSIDE  BELLEROSE  \
Complaint_short
Animal Abuse          38.0    125.0      NaN    37.0         7.0
Blocked Driveway       35.0   2618.0   116.0   377.0        95.0
Derelict Vehicle       27.0    351.0    12.0   198.0        89.0
DnD                    5.0     66.0     NaN     5.0         5.0
Illegal Parking       58.0   1068.0   213.0   514.0       106.0
Noise                 51.0   1967.0   376.0    77.0        63.0
Other                 NaN      NaN     NaN     NaN         NaN
Posting Advertisement  NaN      1.0     NaN     NaN         1.0
Traffic               NaN     47.0     NaN     9.0         7.0
Vending               1.0     54.0     NaN     2.0         NaN
poor                  5.0     33.0     NaN     2.0         2.0

City          BREEZY POINT  BRONX  BROOKLYN  CAMBRIA HEIGHTS  \
Complaint_short
Animal Abuse           2.0   1415.0   2394.0             11.0
Blocked Driveway       3.0  12755.0  28148.0            147.0
Derelict Vehicle       3.0   1953.0   5181.0            115.0
DnD                    1.0    331.0    619.0             NaN
Illegal Parking       15.0   7859.0  27462.0             76.0
Noise                 6.0  15372.0  31952.0            117.0
Other                 NaN      NaN     NaN             NaN
Posting Advertisement  NaN     17.0     45.0             NaN
Traffic               NaN    355.0   1085.0             6.0
Vending               NaN    379.0    515.0             NaN
poor                  NaN    266.0    906.0             5.0

City          CENTRAL PARK  ...  SAINT ALBANS  SOUTH OZONE PARK  \
Complaint_short          ...
Animal Abuse           NaN  ...         30.0             55.0
Blocked Driveway       NaN  ...        244.0            942.0
Derelict Vehicle       NaN  ...        202.0            358.0
DnD                    NaN  ...          5.0             18.0
Illegal Parking        2.0  ...        181.0            494.0
Noise                 95.0  ...        151.0            268.0
Other                 NaN  ...         NaN             NaN
Posting Advertisement  NaN  ...         NaN             1.0
Traffic               NaN  ...        11.0            28.0
Vending               NaN  ...          2.0             5.0
poor                  NaN  ...          8.0             4.0
```

City	SOUTH RICHMOND HILL	SPRINGFIELD GARDENS	\
Complaint_short			
Animal Abuse	26.0	24.0	
Blocked Driveway	1548.0	262.0	
Derelect Vehicle	289.0	210.0	
DnD	26.0	9.0	
Illegal Parking	462.0	238.0	
Noise	377.0	119.0	
Other	NaN	NaN	
Posting Advertisement	NaN	2.0	
Traffic	11.0	11.0	
Vending	24.0	1.0	
poor	11.0	7.0	

City	STATEN ISLAND	SUNNYSIDE	WHITESTONE	WOODHAVEN	\
Complaint_short					
Animal Abuse	557.0	35.0	28.0	45.0	
Blocked Driveway	2142.0	206.0	208.0	1060.0	
Derelect Vehicle	1766.0	10.0	227.0	308.0	
DnD	221.0	17.0	8.0	7.0	
Illegal Parking	4886.0	122.0	525.0	682.0	
Noise	1947.0	289.0	84.0	341.0	
Other	NaN	NaN	NaN	NaN	
Posting Advertisement	516.0	2.0	NaN	NaN	
Traffic	200.0	16.0	17.0	6.0	
Vending	25.0	15.0	1.0	6.0	
poor	83.0	11.0	NaN	9.0	

City	WOODSIDE	Woodside
Complaint_short		
Animal Abuse	69.0	NaN
Blocked Driveway	1613.0	11.0
Derelect Vehicle	247.0	2.0
DnD	31.0	NaN
Illegal Parking	891.0	100.0
Noise	606.0	7.0
Other	NaN	NaN
Posting Advertisement	NaN	NaN
Traffic	39.0	NaN
Vending	15.0	NaN
poor	33.0	NaN

[11 rows x 53 columns]

```
[39]: City = df.groupby(['Complaint_short', 'City']).size().unstack()
City
```



[39]: City	ARVERNE	ASTORIA	Astoria	BAYSIDE	BELLEROSE	\
Complaint_short						
Animal Abuse	38.0	125.0	NaN	37.0	7.0	
Blocked Driveway	35.0	2618.0	116.0	377.0	95.0	
Derelect Vehicle	27.0	351.0	12.0	198.0	89.0	
DnD	5.0	66.0	NaN	5.0	5.0	
Illegal Parking	58.0	1068.0	213.0	514.0	106.0	
Noise	51.0	1967.0	376.0	77.0	63.0	
Other	NaN	NaN	NaN	NaN	NaN	
Posting Advertisement	NaN	1.0	NaN	NaN	1.0	
Traffic	NaN	47.0	NaN	9.0	7.0	
Vending	1.0	54.0	NaN	2.0	NaN	
poor	5.0	33.0	NaN	2.0	2.0	
City	BREEZY POINT	BRONX	BROOKLYN	CAMBRIA HEIGHTS	\	
Complaint_short						
Animal Abuse	2.0	1415.0	2394.0		11.0	
Blocked Driveway	3.0	12755.0	28148.0		147.0	
Derelect Vehicle	3.0	1953.0	5181.0		115.0	
DnD	1.0	331.0	619.0		NaN	
Illegal Parking	15.0	7859.0	27462.0		76.0	
Noise	6.0	15372.0	31952.0		117.0	
Other	NaN	NaN	NaN		NaN	
Posting Advertisement	NaN	17.0	45.0		NaN	
Traffic	NaN	355.0	1085.0		6.0	
Vending	NaN	379.0	515.0		NaN	
poor	NaN	266.0	906.0		5.0	
City	CENTRAL PARK	...	SAINT ALBANS	SOUTH OZONE PARK	\	
Complaint_short		...				
Animal Abuse	NaN	...	30.0		55.0	
Blocked Driveway	NaN	...	244.0		942.0	
Derelect Vehicle	NaN	...	202.0		358.0	
DnD	NaN	...	5.0		18.0	
Illegal Parking	2.0	...	181.0		494.0	
Noise	95.0	...	151.0		268.0	
Other	NaN	...	NaN		NaN	
Posting Advertisement	NaN	...	NaN		1.0	
Traffic	NaN	...	11.0		28.0	
Vending	NaN	...	2.0		5.0	
poor	NaN	...	8.0		4.0	
City	SOUTH RICHMOND HILL	SPRINGFIELD GARDENS	\			
Complaint_short						
Animal Abuse		26.0		24.0		
Blocked Driveway		1548.0		262.0		
Derelect Vehicle		289.0		210.0		

DnD	26.0	9.0
Illegal Parking	462.0	238.0
Noise	377.0	119.0
Other	NaN	NaN
Posting Advertisement	NaN	2.0
Traffic	11.0	11.0
Vending	24.0	1.0
poor	11.0	7.0

City	STATEN ISLAND	SUNNYSIDE	WHITESTONE	WOODHAVEN \
Complaint_short				
Animal Abuse	557.0	35.0	28.0	45.0
Blocked Driveway	2142.0	206.0	208.0	1060.0
Derelect Vehicle	1766.0	10.0	227.0	308.0
DnD	221.0	17.0	8.0	7.0
Illegal Parking	4886.0	122.0	525.0	682.0
Noise	1947.0	289.0	84.0	341.0
Other	NaN	NaN	NaN	NaN
Posting Advertisement	516.0	2.0	NaN	NaN
Traffic	200.0	16.0	17.0	6.0
Vending	25.0	15.0	1.0	6.0
poor	83.0	11.0	NaN	9.0

City	WOODSIDE	Woodside
Complaint_short		
Animal Abuse	69.0	NaN
Blocked Driveway	1613.0	11.0
Derelect Vehicle	247.0	2.0
DnD	31.0	NaN
Illegal Parking	891.0	100.0
Noise	606.0	7.0
Other	NaN	NaN
Posting Advertisement	NaN	NaN
Traffic	39.0	NaN
Vending	15.0	NaN
poor	33.0	NaN

[11 rows x 53 columns]

```
[40]: Total_report_city = pd.Series(name='Total', data=City.sum(axis=0))
Total_report_city
```

```
[40]: City
ARVERNE      220.0
ASTORIA     6330.0
Astoria      717.0
BAYSIDE     1221.0
```

BELLEROSE	375.0
BREEZY POINT	30.0
BRONX	40702.0
BROOKLYN	98307.0
CAMBRIA HEIGHTS	477.0
CENTRAL PARK	97.0
COLLEGE POINT	1220.0
CORONA	4295.0
EAST ELMHURST	2734.0
ELMHURST	2673.0
East Elmhurst	14.0
FAR ROCKAWAY	1179.0
FLORAL PARK	152.0
FLUSHING	5971.0
FOREST HILLS	1688.0
FRESH MEADOWS	1899.0
GLEN OAKS	306.0
HOLLIS	1012.0
HOWARD BEACH	931.0
Howard Beach	1.0
JACKSON HEIGHTS	1689.0
JAMAICA	7296.0
KEW GARDENS	771.0
LITTLE NECK	559.0
LONG ISLAND CITY	2437.0
Long Island City	134.0
MASPETH	2462.0
MIDDLE VILLAGE	1765.0
NEW HYDE PARK	98.0
NEW YORK	65994.0
OAKLAND GARDENS	551.0
OZONE PARK	2755.0
QUEENS	31.0
QUEENS VILLAGE	1814.0
REGO PARK	1486.0
RICHMOND HILL	1904.0
RIDGEWOOD	5163.0
ROCKAWAY PARK	745.0
ROSEDALE	922.0
SAINT ALBANS	834.0
SOUTH OZONE PARK	2173.0
SOUTH RICHMOND HILL	2774.0
SPRINGFIELD GARDENS	883.0
STATEN ISLAND	12343.0
SUNNYSIDE	723.0
WHITESTONE	1098.0
WOODHAVEN	2464.0

```
WOODSIDE          3544.0
Woodside          120.0
Name: Total, dtype: float64
```

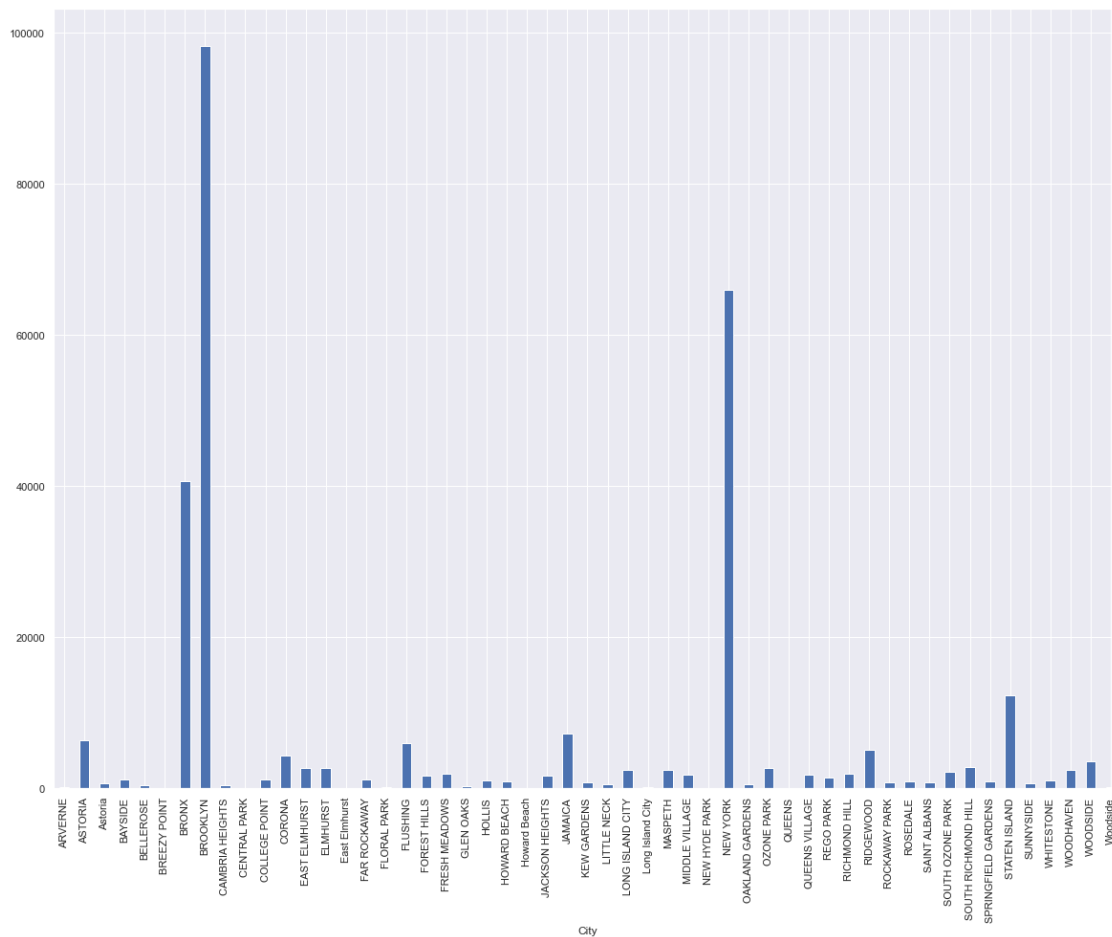
```
[41]: plt.figure(figsize=(20,15))
Total_report_city.plot(kind='bar')
plt.xticks(rotation=90)
```

```
[41]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
              17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
              34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
              51, 52]),
      [Text(0, 0, 'ARVERNE'),
       Text(1, 0, 'ASTORIA'),
       Text(2, 0, 'Astoria'),
       Text(3, 0, 'BAYSIDE'),
       Text(4, 0, 'BELLEROSE'),
       Text(5, 0, 'BREEZY POINT'),
       Text(6, 0, 'BRONX'),
       Text(7, 0, 'BROOKLYN'),
       Text(8, 0, 'CAMBRIA HEIGHTS'),
       Text(9, 0, 'CENTRAL PARK'),
       Text(10, 0, 'COLLEGE POINT'),
       Text(11, 0, 'CORONA'),
       Text(12, 0, 'EAST ELMHURST'),
       Text(13, 0, 'ELMHURST'),
       Text(14, 0, 'East Elmhurst'),
       Text(15, 0, 'FAR ROCKAWAY'),
       Text(16, 0, 'FLORAL PARK'),
       Text(17, 0, 'FLUSHING'),
       Text(18, 0, 'FOREST HILLS'),
       Text(19, 0, 'FRESH MEADOWS'),
       Text(20, 0, 'GLEN OAKS'),
       Text(21, 0, 'HOLLIS'),
       Text(22, 0, 'HOWARD BEACH'),
       Text(23, 0, 'Howard Beach'),
       Text(24, 0, 'JACKSON HEIGHTS'),
       Text(25, 0, 'JAMAICA'),
       Text(26, 0, 'KEW GARDENS'),
       Text(27, 0, 'LITTLE NECK'),
       Text(28, 0, 'LONG ISLAND CITY'),
       Text(29, 0, 'Long Island City'),
       Text(30, 0, 'MASPETH'),
       Text(31, 0, 'MIDDLE VILLAGE'),
       Text(32, 0, 'NEW HYDE PARK'),
       Text(33, 0, 'NEW YORK'),
       Text(34, 0, 'OAKLAND GARDENS'),
```

```

Text(35, 0, 'OZONE PARK'),
Text(36, 0, 'QUEENS'),
Text(37, 0, 'QUEENS VILLAGE'),
Text(38, 0, 'REGO PARK'),
Text(39, 0, 'RICHMOND HILL'),
Text(40, 0, 'RIDGEWOOD'),
Text(41, 0, 'ROCKAWAY PARK'),
Text(42, 0, 'ROSEDALE'),
Text(43, 0, 'SAINT ALBANS'),
Text(44, 0, 'SOUTH OZONE PARK'),
Text(45, 0, 'SOUTH RICHMOND HILL'),
Text(46, 0, 'SPRINGFIELD GARDENS'),
Text(47, 0, 'STATEN ISLAND'),
Text(48, 0, 'SUNNYSIDE'),
Text(49, 0, 'WHITESTONE'),
Text(50, 0, 'WOODHAVEN'),
Text(51, 0, 'WOODSIDE'),
Text(52, 0, 'Woodside')]

```



Highest number of Complaints reported in Brooklyn. Followed by New York and Bronx.

Univariate : Month and Day wise insights. Bivariate : Location type wise insights City wise insights  
Hour, Month, Day wise insights Holiday insights were provided for Question 3.

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

```
[42]: df2 = df[['Created Date', 'Closed Date', 'City', 'Complaint Type']].copy()
df2.head()
```

```
[42]:
```

	Created Date	Closed Date	City	Complaint Type
0	2015-12-31 23:59:45	2016-01-01 00:55:00	NEW YORK	Noise - Street/Sidewalk
1	2015-12-31 23:59:44	2016-01-01 01:26:00	ASTORIA	Blocked Driveway
2	2015-12-31 23:59:29	2016-01-01 04:51:00	BRONX	Blocked Driveway
3	2015-12-31 23:57:46	2016-01-01 07:43:00	BRONX	Illegal Parking
4	2015-12-31 23:56:58	2016-01-01 03:24:00	ELMHURST	Illegal Parking

```
[43]: df2.isnull().sum()
```

```
[43]: Created Date      0
Closed Date      2164
City             2614
Complaint Type    0
dtype: int64
```

```
[44]: df2['City'] = df2['City'].fillna('Unknown City')
df2.dropna(inplace=True)
df2.isnull().sum()
```

```
[44]: Created Date      0
Closed Date      0
City             0
Complaint Type    0
dtype: int64
```

```
[45]: df2['Request_Closing_Time'] = df2['Closed Date'] - df2['Created Date']
df2.head()
```

```
[45]:
```

	Created Date	Closed Date	City	Complaint Type \
0	2015-12-31 23:59:45	2016-01-01 00:55:00	NEW YORK	Noise - Street/Sidewalk
1	2015-12-31 23:59:44	2016-01-01 01:26:00	ASTORIA	Blocked Driveway
2	2015-12-31 23:59:29	2016-01-01 04:51:00	BRONX	Blocked Driveway
3	2015-12-31 23:57:46	2016-01-01 07:43:00	BRONX	Illegal Parking
4	2015-12-31 23:56:58	2016-01-01 03:24:00	ELMHURST	Illegal Parking

	Request_Closing_Time
0	0 days 00:55:15
1	0 days 01:26:16

```

2      0 days 04:51:31
3      0 days 07:45:14
4      0 days 03:27:02

```

```
[46]: df2.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 298534 entries, 0 to 300697
Data columns (total 5 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Created Date          298534 non-null  datetime64[ns]
1   Closed Date           298534 non-null  datetime64[ns]
2   City                  298534 non-null  object
3   Complaint Type        298534 non-null  object
4   Request_Closing_Time  298534 non-null  timedelta64[ns]
dtypes: datetime64[ns](2), object(2), timedelta64[ns](1)
memory usage: 13.7+ MB

```

```
[47]: df2['Closing_Time_Seconds']=df2['Request_Closing_Time'].map(lambda x: pd.
      ↪to_timedelta(x).seconds)
```

```
[48]: df2.head()
```

```
[48]:
```

	Created Date	Closed Date	City	Complaint Type \
0	2015-12-31 23:59:45	2016-01-01 00:55:00	NEW YORK	Noise - Street/Sidewalk
1	2015-12-31 23:59:44	2016-01-01 01:26:00	ASTORIA	Blocked Driveway
2	2015-12-31 23:59:29	2016-01-01 04:51:00	BRONX	Blocked Driveway
3	2015-12-31 23:57:46	2016-01-01 07:43:00	BRONX	Illegal Parking
4	2015-12-31 23:56:58	2016-01-01 03:24:00	ELMHURST	Illegal Parking

	Request_Closing_Time	Closing_Time_Seconds
0	0 days 00:55:15	3315
1	0 days 01:26:16	5176
2	0 days 04:51:31	17491
3	0 days 07:45:14	27914
4	0 days 03:27:02	12422

```
[49]: final_set = (df2.groupby(['Complaint Type', 'City']).mean()).
      ↪sort_values(by=['Closing_Time_Seconds'])
final_set.head()
```

```
[49]:
```

Complaint Type	City	Closing_Time_Seconds
Graffiti	ROSEDALE	563.0
Drinking	ARVERNE	859.0
Posting Advertisement	RIDGEWOOD	895.0

Bike/Roller/Skate Chronic	EAST ELMHURST	908.0
Illegal Fireworks	OZONE PARK	1140.0

The above final\_set dataframe shows grouping of Complaint type and City ordered by Closing time

5. Perform a statistical test for the following: Please note: For the below statements you need to state the Null and Alternate and then provide a statistical test to accept or reject the Null Hypothesis along with the corresponding 'p-value'. Whether the average response time across complaint types is similar or not (overall) Are the type of complaint or service requested and location related?

```
[50]: df3 = df2.groupby(['City', 'Complaint Type'])[['Closing_Time_Seconds']].mean()
```

```
[51]: df3.head()
```

```
[51]:
```

		Closing_Time_Seconds
City	Complaint Type	
ARVERNE	Animal Abuse	7753.052632
	Blocked Driveway	9093.485714
	Derelict Vehicle	10685.592593
	Disorderly Youth	12928.500000
	Drinking	859.000000

```
[52]: df3
```

```
[52]:
```

		Closing_Time_Seconds
City	Complaint Type	
ARVERNE	Animal Abuse	7753.052632
	Blocked Driveway	9093.485714
	Derelict Vehicle	10685.592593
	Disorderly Youth	12928.500000
	Drinking	859.000000
...		...
Woodside	Blocked Driveway	23062.363636
	Derelict Vehicle	17880.000000
	Illegal Parking	17925.130000
	Noise - Commercial	8619.000000
	Noise - Street/Sidewalk	12280.600000

[778 rows x 1 columns]

```
[53]: df4=pd.DataFrame(df['Complaint Type'].value_counts().
    ↳sort_values(ascending=False).head(10))
df4
```

```
[53]:
```

	Complaint Type
Blocked Driveway	77044



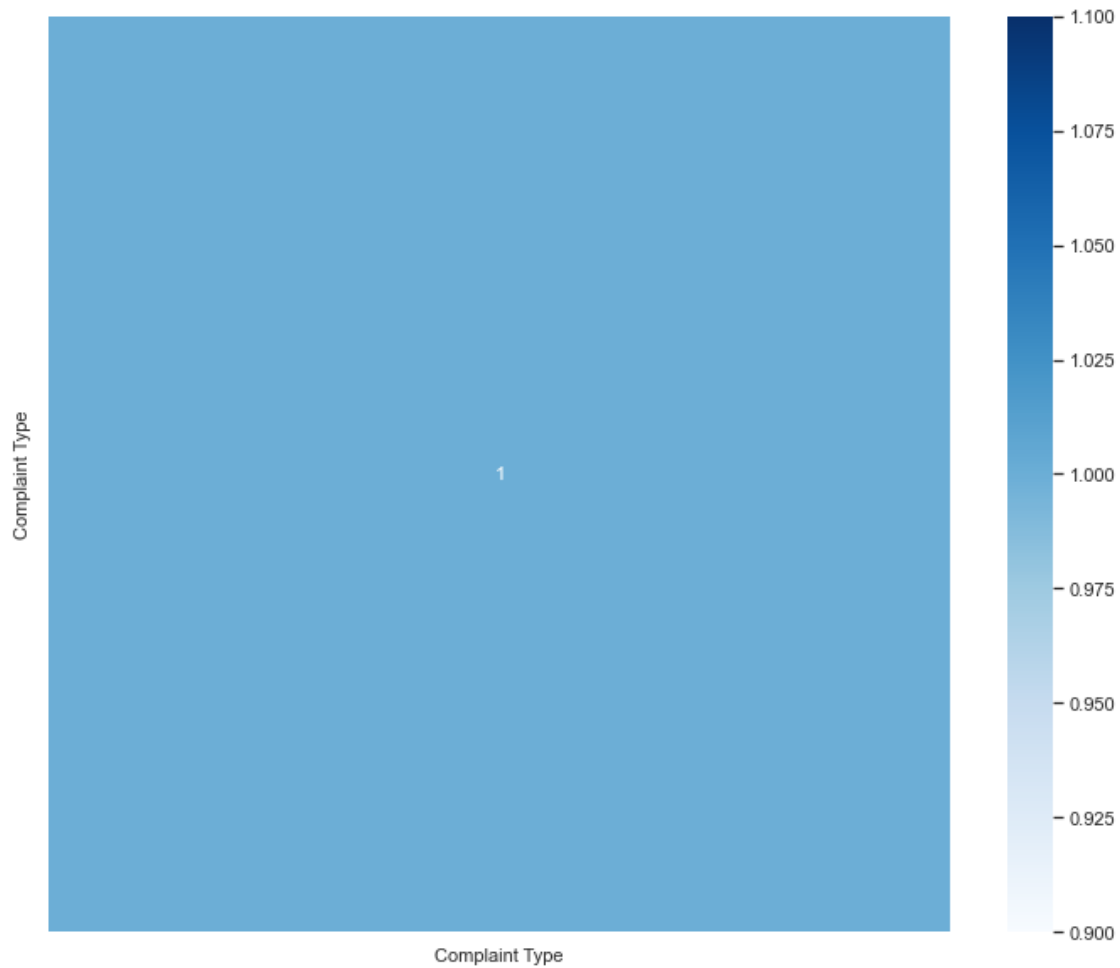
Illegal Parking	75361
Noise - Street/Sidewalk	48612
Noise - Commercial	35577
Derelict Vehicle	17718
Noise - Vehicle	17083
Animal Abuse	7778
Traffic	4498
Homeless Encampment	4416
Noise - Park	4042

```
[54]: # calculate the correlation matrix
corr1 = df4.corr(method='pearson')
corr1
print(corr1.columns)
```

```
Index(['Complaint Type'], dtype='object')
```

```
[55]: plt.figure(figsize=(12,10))
# plot the heatmap
sns.heatmap(corr1, xticklabels=corr1.columns, yticklabels=corr1.
↪columns,cmap="Blues",annot=True)
```

```
[55]: <AxesSubplot:>
```



```
[56]: df4.skew()
```

```
[56]: Complaint Type    0.911963
      dtype: float64
```

Null hypothesis: Complaint type is not Related to Location Alternative hypothesis: Complaint type is related to location we will use Spearmanr correlation for the test, correlation of more than 0.8 will overturn our null hypothesis.

```
[57]: df5=df[['Complaint Type','City', 'Longitude', 'Latitude']].copy()
```

```
[58]: df5.dropna(inplace=True)
```

```
[59]: from scipy.stats import f_oneway
      from sklearn.preprocessing import OrdinalEncoder
      from scipy.stats import spearmanr
```

```
[60]: ord_enc = OrdinalEncoder()
df5['Complaint Type'] = ord_enc.fit_transform(df5[['Complaint Type']])
df5['City'] = ord_enc.fit_transform(df5[['City']])
df5.head()
```

```
[60]:
```

	Complaint Type	City	Longitude	Latitude
0	13.0	33.0	-73.923501	40.865682
1	2.0	1.0	-73.915094	40.775945
2	2.0	6.0	-73.888525	40.870325
3	9.0	6.0	-73.828379	40.835994
4	9.0	13.0	-73.874170	40.733060

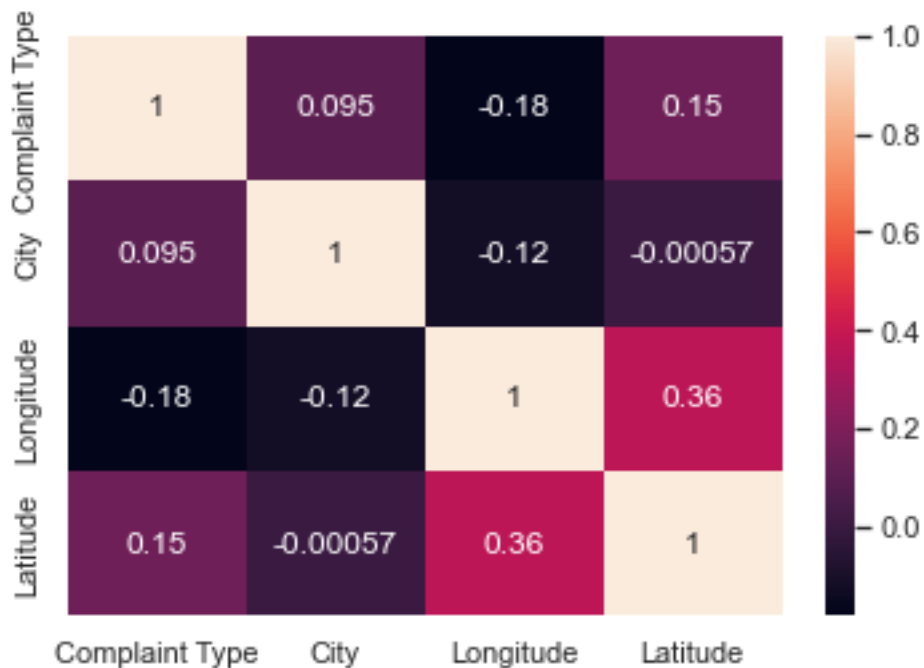
```
[61]: stat, p = spearmanr(df5)
print(p)
```

```
[[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
```

As we can see all the p values are 0, that means there is no significant correlation between the features.

```
[62]: sns.heatmap(df5.corr(), annot=True)
```

```
[62]: <AxesSubplot:>
```



[ ]: