Customer Service Requests Analysis

#libraries imported

```
import numpy as np
import pandas as pd
import datetime as dt
from datetime import timedelta
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from scipy.integrate import quad
from scipy.stats import norm
```

#read path of dataset file

```
df_311 = pd.read_csv('C:\\Users\\SandipG\\Desktop\\Python Programs\\311_Service_Requests_from_2010_to_Present.csv')
```

```
E:\Programs\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3058: DtypeWarning: Columns (48,49) have mixed types. Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

#check if the dataset file is accessed correctly

```
df_311.shape
```

```
(300698, 53)
```

Data Ingestion to display all 53 columns

```
df = pd.read_csv("C:\\Users\\SandipG\\Desktop\\Python Programs\\311_Service_Requests_from_2010_to_Present.csv")
```

```
pd.set_option('display.max_columns',None)
```

```
E:\Programs\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3058: DtypeWarning: Columns (48,49) have mixed types. Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

Converting dates

```
df['Created Date'] = df['Created Date'].astype('datetime64[ns]')
df['Closed Date'] = df['Closed Date'].astype('datetime64[ns]')
df['Resolution Action Updated Date'] = df['Resolution Action Updated Date'].astype('datetime64[ns]')

#New feature created Request_Closing_Time

df['Request_Closing_Time']=df['Closed Date']-df['Created Date']


#New feature created Month &

count = 0

#Create an empty list called "month"

month=[]

# for all the rows in 'Created Date' (calculated by len(df)) extract month and append to "month" list

while count < len(df):
    month.append(df['Created Date'][count].month)
    count = count+1

# create a new month column

df['Month'] = month

#new feature created Minute
minutes=[]

# calculate duration by finding diffrenece bteween 'Closed Date' and Created Date
# Create new column called "Duration_Minutes"

df['Duration_Minutes'] = df['Closed Date'] - df['Created Date']

for x in df['Duration_Minutes']:
    minutes.append(x.seconds / 60)

df['Duration_Minutes'] = minutes

# Preparing Data

df['Agency'].value_counts()
```

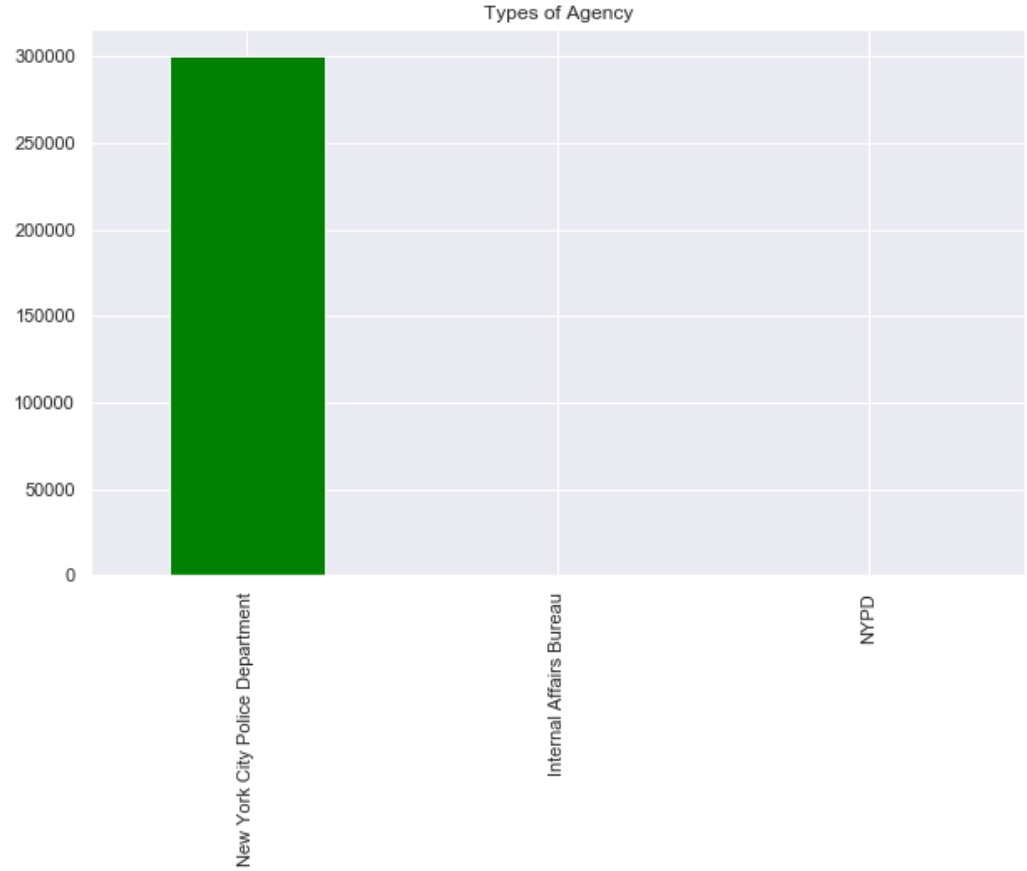
```
NYPD    300698
Name: Agency, dtype: int64
```

Data Profiling of 'Agency Name' feature.

```
(df['Agency Name'].value_counts()).head().plot(kind = 'bar', figsize= (10,6), title='Types of Agency',color='green')
```

Please Note:
There are only 2 types of agency's viz New York Police Dept and Internal Affairs Bureau and
Majority of complaints come from New York Police Dept.

```
<matplotlib.axes._subplots.AxesSubplot at 0x270f52a8b08>
```



Data Profiling of 'Complaint Type' feature.

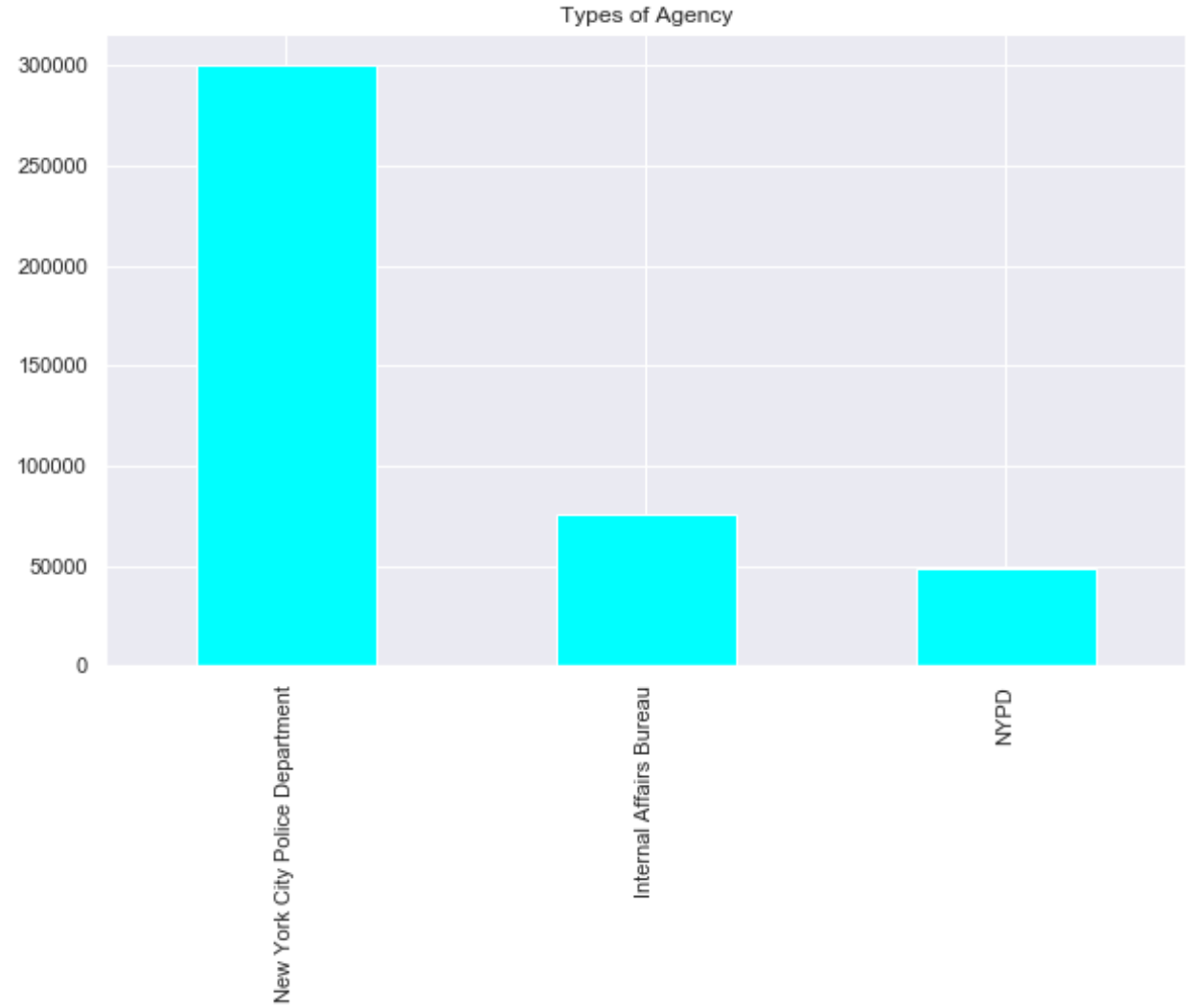
```
df['Complaint Type'].value_counts()
(df['Complaint Type'].value_counts()).head().plot(kind = 'bar', figsize= (10,6), title='Maximum Occuring Complaints(top 5)',color='cyan')
```

Data Profiling of 'Agency Name' feature.

```
(df['Agency Name'].value_counts()).head().plot(kind = 'bar', figsize= (10,6), title='Types of Agency',color='cyan')
```

Please Note:
There are only 2 types of agency's viz New York Police Dept and Internal Affairs Bureau and
Majority of complaints come from New York Police Dept.
The major complaint type are Blocked Driveway , Illegal Parking,Noise - Street/Sidewalk , Noise - Commercial and Derelict Vehicle

```
<matplotlib.axes._subplots.AxesSubplot at 0x270e0dd5e48>
```




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

```
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
```

df.head()

	Unique Key	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
0	32310363	2015-12-31 23:59:45	2016-01-01 00:55:00	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk	10034.0	71 VERMILYEA AVENUE	VERMILYEA AVENUE	ACADEMY STREET	WEST 204 STREET	NaN	NaN	ADDRESS	NEW YORK	NaN	Precinct
1	32309934	2015-12-31	2016-01-01	NYPD	New York City	Blocked Driveway	No Access	Street/Sidewalk	11105.0	27-07 23 AVENUE	23 AVENUE	27 STREET	28 STREET	NaN	NaN	ADDRESS	ASTORIA	NaN	Precinct

	Unique Key	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
		23:59:44	01:26:00		Police Department														
2	32309159	2015-12-31 23:59:29	2016-01-01 04:51:00	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	10458.0	2897 VALENTINE AVENUE	VALENTINE AVENUE	EAST 198 STREET	EAST 199 STREET	NaN	NaN	ADDRESS	BRONX	NaN	Precinct
3	32305098	2015-12-31 23:57:46	2016-01-01 07:43:00	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Sidewalk	10461.0	2940 BAISLEY AVENUE	BAISLEY AVENUE	EDISON AVENUE	BSTREET	NaN	NaN	ADDRESS	BRONX	NaN	Precinct
4	32306529	2015-12-31 23:56:58	2016-01-01 03:24:00	NYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street/Sidewalk	11373.0	87-1457 ROAD	57 ROAD	SEABURY STREET	HOFFMAN DRIVE	NaN	NaN	ADDRESS	ELMHURST	NaN	Precinct

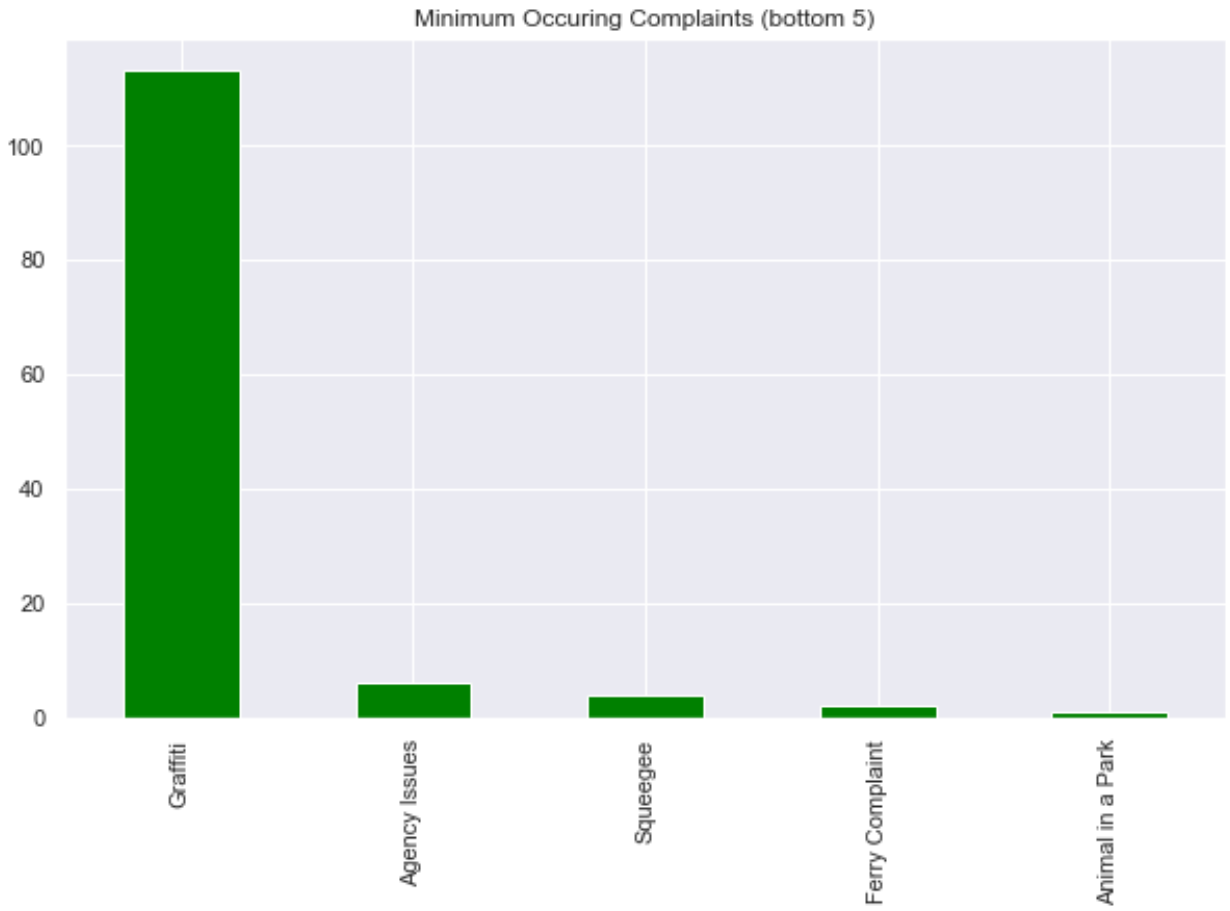
df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300698 entries, 0 to 300697
Data columns (total 56 columns):
Unique Key                          300698 non-null int64
Created Date                        300698 non-null datetime64[ns]
Closed Date                         298534 non-null datetime64[ns]
Agency                              300698 non-null object
Agency Name                         300698 non-null object
Complaint Type                      300698 non-null object
Descriptor                          294784 non-null object
Location Type                       300567 non-null object
Incident Zip                        298083 non-null float64
Incident Address                    256288 non-null object
Street Name                         256288 non-null object
Cross Street 1                      251419 non-null object
Cross Street 2                      250919 non-null object
Intersection Street 1               43858 non-null object
Intersection Street 2               43362 non-null object
Address Type                        297883 non-null object
City                                298084 non-null object
Landmark                            349 non-null object
Facility Type                       298527 non-null object
Status                              300698 non-null object
Due Date                            300695 non-null object
Resolution Description              300698 non-null object
Resolution Action Updated Date      298511 non-null datetime64[ns]
Community Board                     300698 non-null object
Borough                             300698 non-null object
X Coordinate (State Plane)          297158 non-null float64
Y Coordinate (State Plane)          297158 non-null float64
Park Facility Name                  300698 non-null object
Park Borough                        300698 non-null object
School Name                         300698 non-null object
School Number                       300698 non-null object
School Region                       300697 non-null object
School Code                         300697 non-null object
School Phone Number                 300698 non-null object
School Address                      300698 non-null object
School City                         300698 non-null object
School State                        300698 non-null object
School Zip                          300697 non-null object
School Not Found                    300698 non-null object
School or Citywide Complaint        0 non-null float64
```

```
Vehicle Type                      0 non-null float64
Taxi Company Borough              0 non-null float64
Taxi Pick Up Location             0 non-null float64
Bridge Highway Name               243 non-null object
Bridge Highway Direction          243 non-null object
Road Ramp                         213 non-null object
Bridge Highway Segment            213 non-null object
Garage Lot Name                   0 non-null float64
Ferry Direction                   1 non-null object
Ferry Terminal Name               2 non-null object
Latitude                          297158 non-null float64
Longitude                         297158 non-null float64
Location                          297158 non-null object
Request_Closing_Time              298534 non-null timedelta64[ns]
Month                             300698 non-null int64
Duration_Minutes                  298534 non-null float64
dtypes: datetime64[ns](3), float64(11), int64(2), object(39),
timedelta64[ns](1)
memory usage: 128.5+ MB
```

```
(df['Complaint Type'].value_counts()).tail().plot(kind = 'bar', figsize= (10,6), title='Minimum Occuring Complaints (bottom 5)',color='green')
```

```
# Please Note:
# The occurrence of the following  complaint type are less:
# Illegal Fireworks          168
# Graffiti                   113
# Agency Issues                6
# Squeegee                     4
# Ferry Complaint              2
# Animal in a Park             1
<matplotlib.axes._subplots.AxesSubplot at 0x270f4717a48>
```

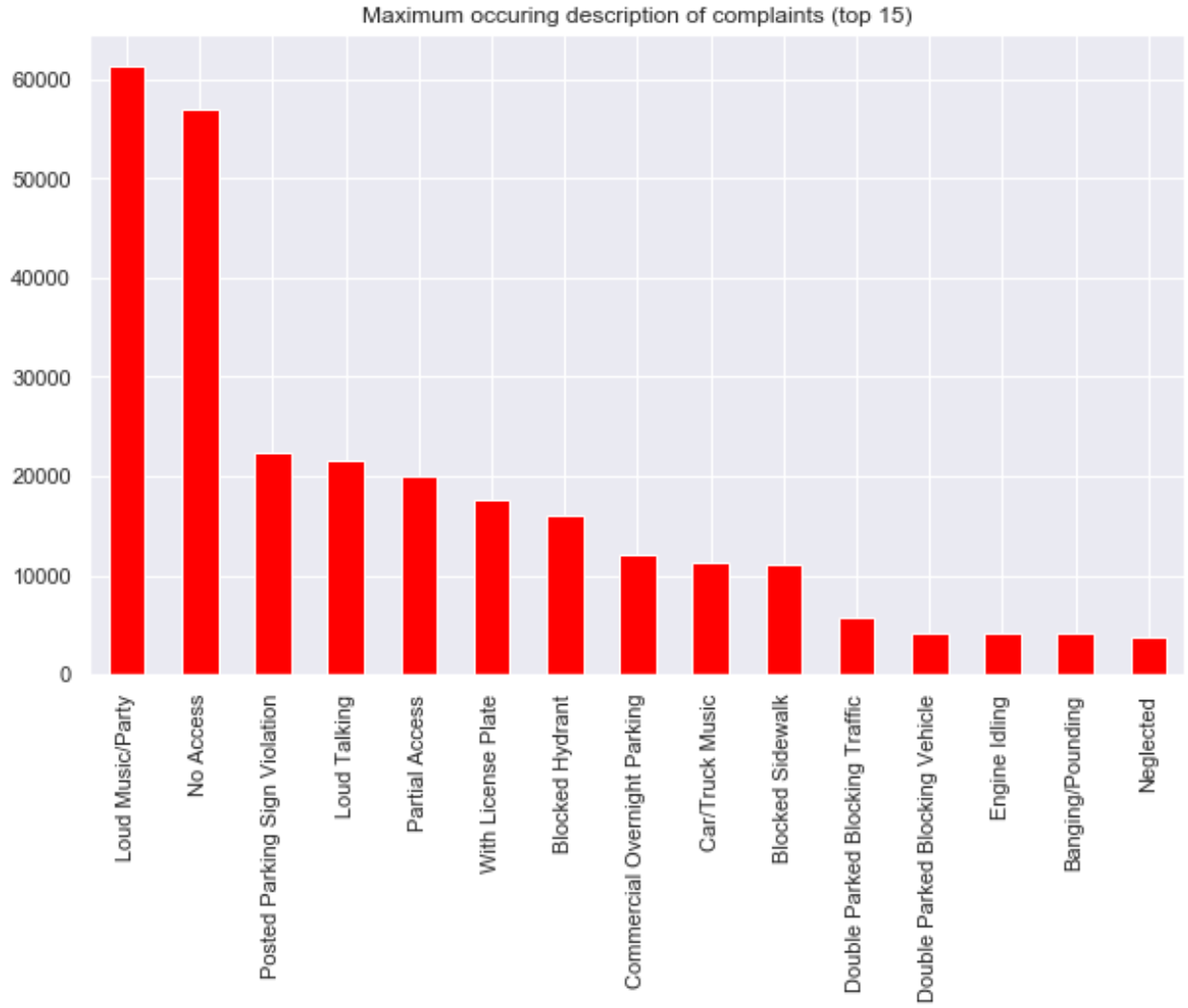


Data Profiling of 'Descriptor' feature.

```
df['Descriptor'].value_counts()
(df['Descriptor'].value_counts()).head(15).plot(kind = 'bar', figsize= (10,6), title='Maximum occuring description of complaints (top 15)',color='red')
```

Please Note:
Major complaint types are for the following description:
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

```
<matplotlib.axes._subplots.AxesSubplot at 0x270f46ff788>
```



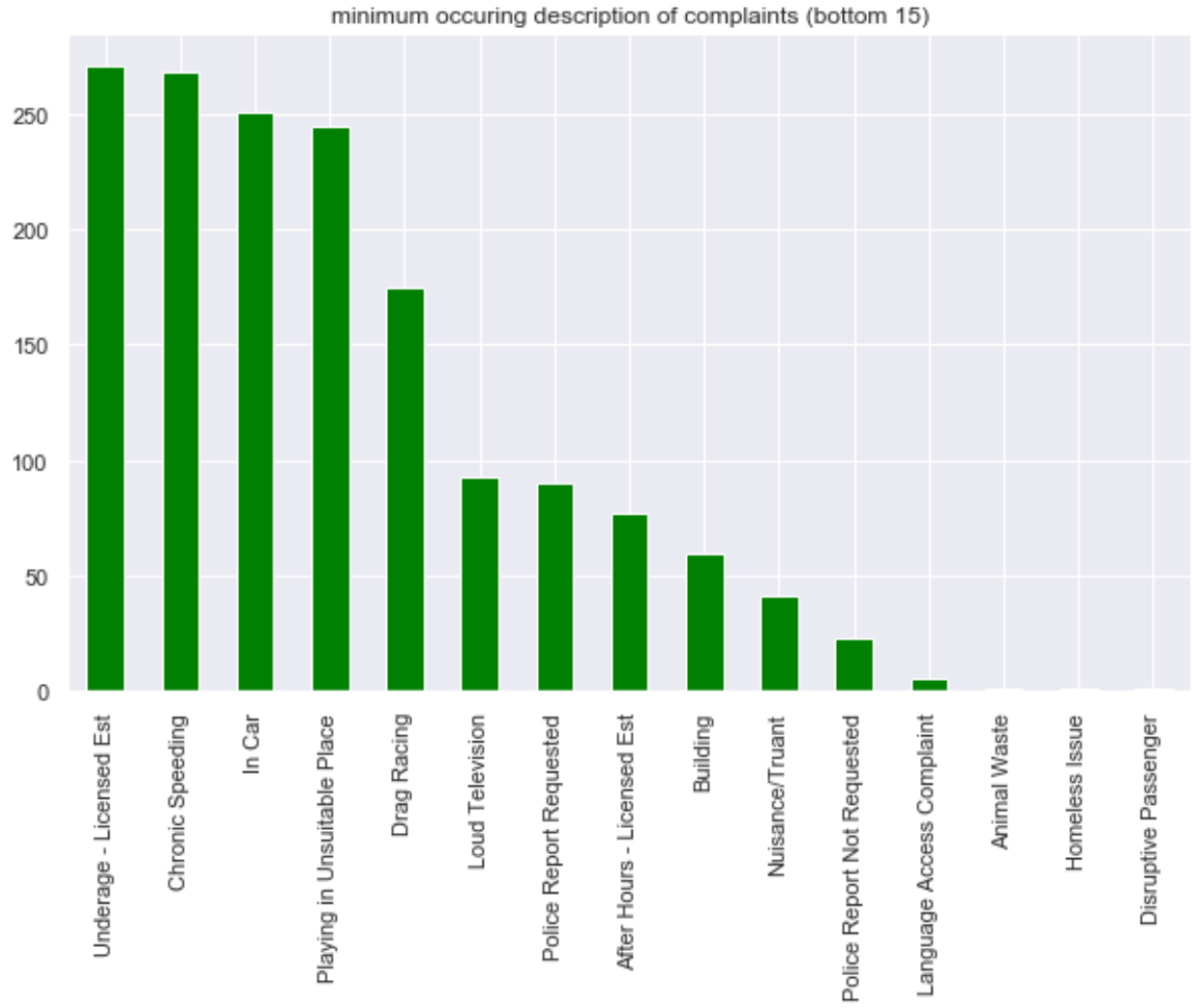
df['Descriptor'].value_counts()

```
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
```

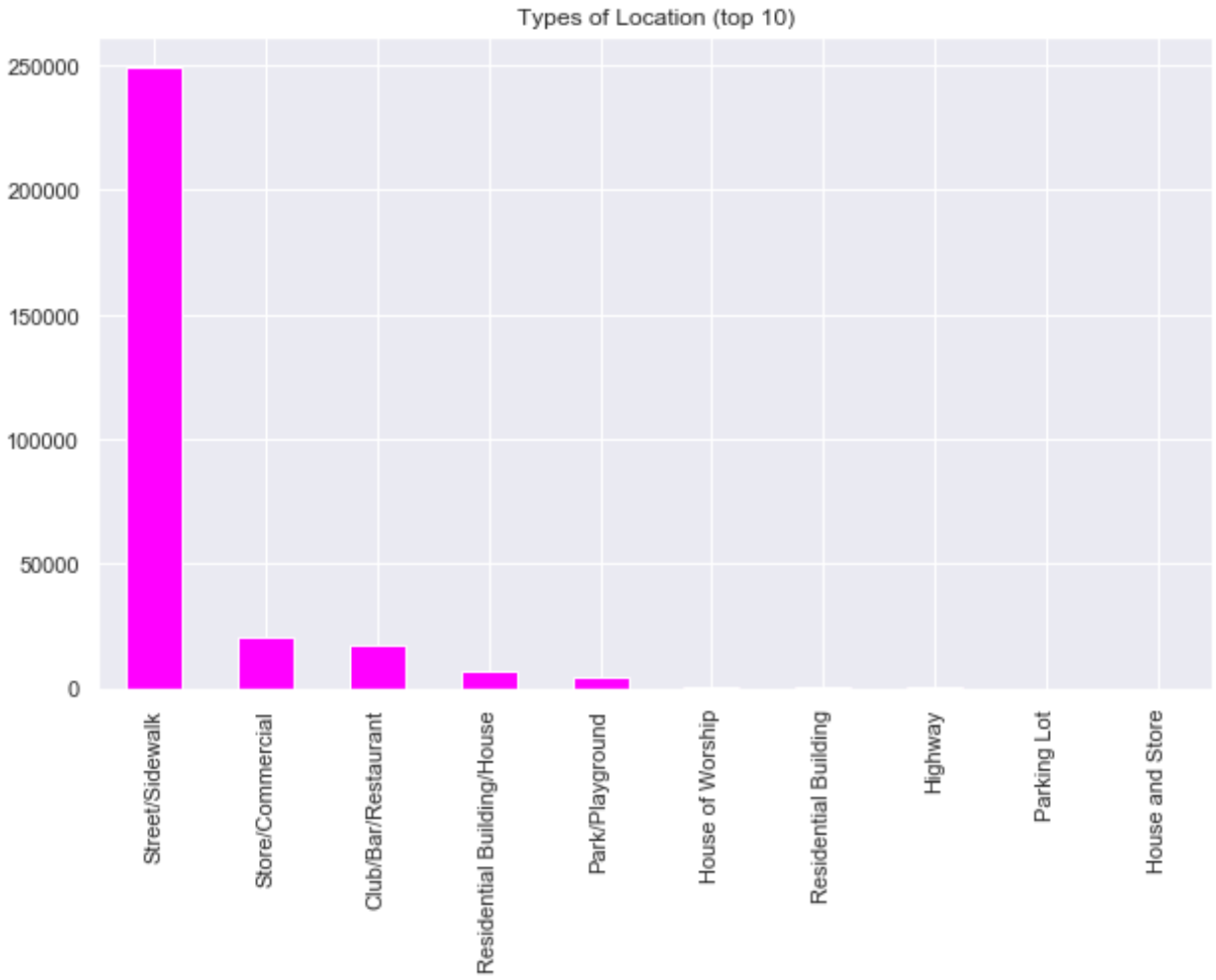
```
Banging/Pounding                  4165
Neglected                         3787
Car/Truck Horn                    3511
Congestion/Gridlock               2761
In Prohibited Area                2025
Other (complaint details)         1969
Unlicensed                        1777
Overnight Commercial Storage      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
Animal Waste                         1
Homeless Issue                       1
Disruptive Passenger                 1
Name: Descriptor, dtype: int64
```

```
(df['Descriptor'].value_counts()).tail(15).plot(kind = 'bar', figsize= (10,6), title='minimum occuring description of complaints (bottom 15)',color='green')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x270b5b87f88>
```




```
# Location Type
#  Data Profiling of 'Location Type' feature.

df['Location Type'].value_counts()
(df['Location Type'].value_counts()).head(10).plot(kind = 'bar', figsize= (10,6), title='Types of Location (top 10)',color='magenta')

# Please Note:
# Major complaints have come from the following location type:
# Street/Sidewalk              249299
# Store/Commercial              20381
# Club/Bar/Restaurant           17360
# Residential Building/House     6960
# Park/Playground               4773#
<matplotlib.axes._subplots.AxesSubplot at 0x270e3e71b08>
```



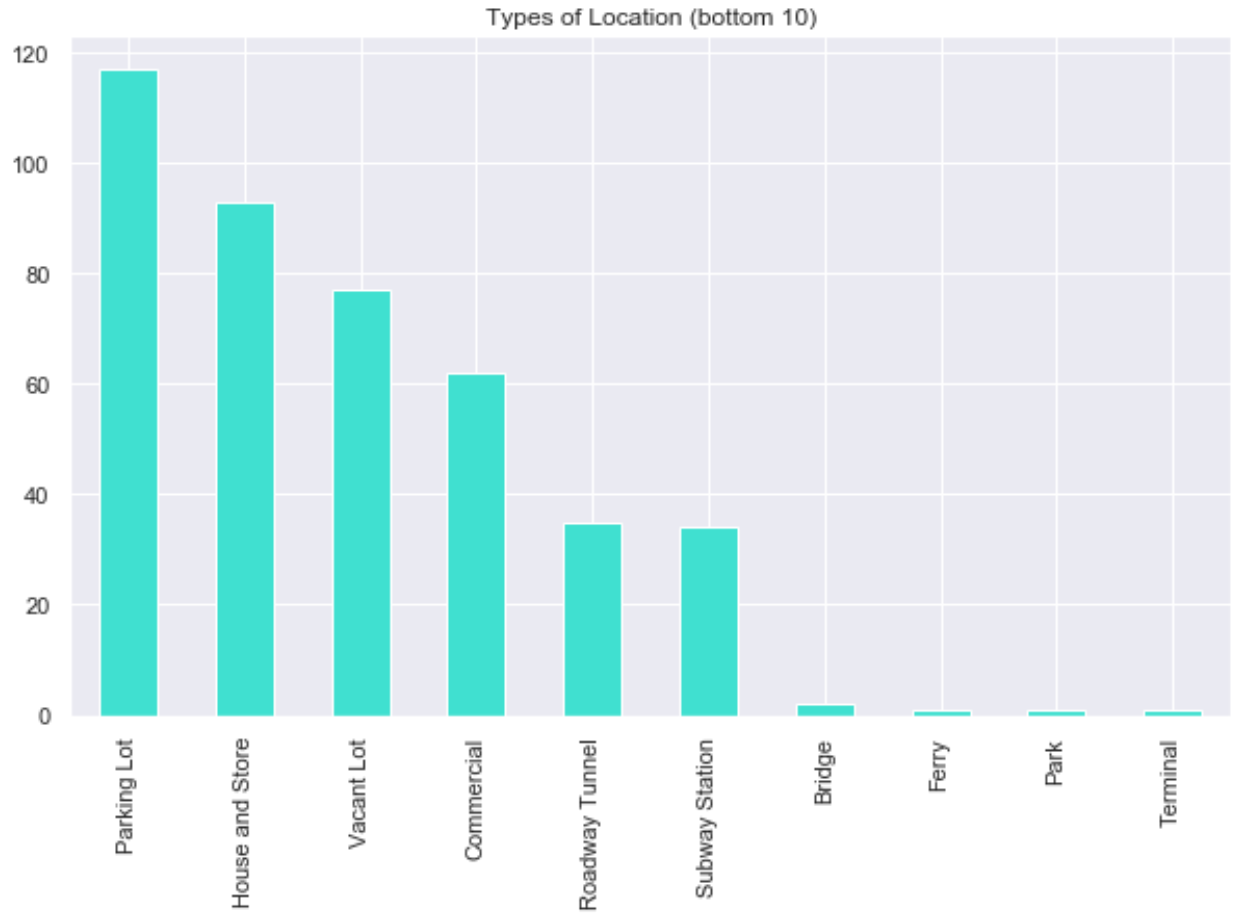
df['Location Type'].value_counts()

```
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
Ferry                               1
```

```
Park                                1
Terminal                            1
Name: Location Type, dtype: int64
```

```
(df['Location Type'].value_counts()).tail(10).plot(kind = 'bar', figsize= (10,6), title='Types of Location (bottom 10)',color='turquoise')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x270e2819488>
```

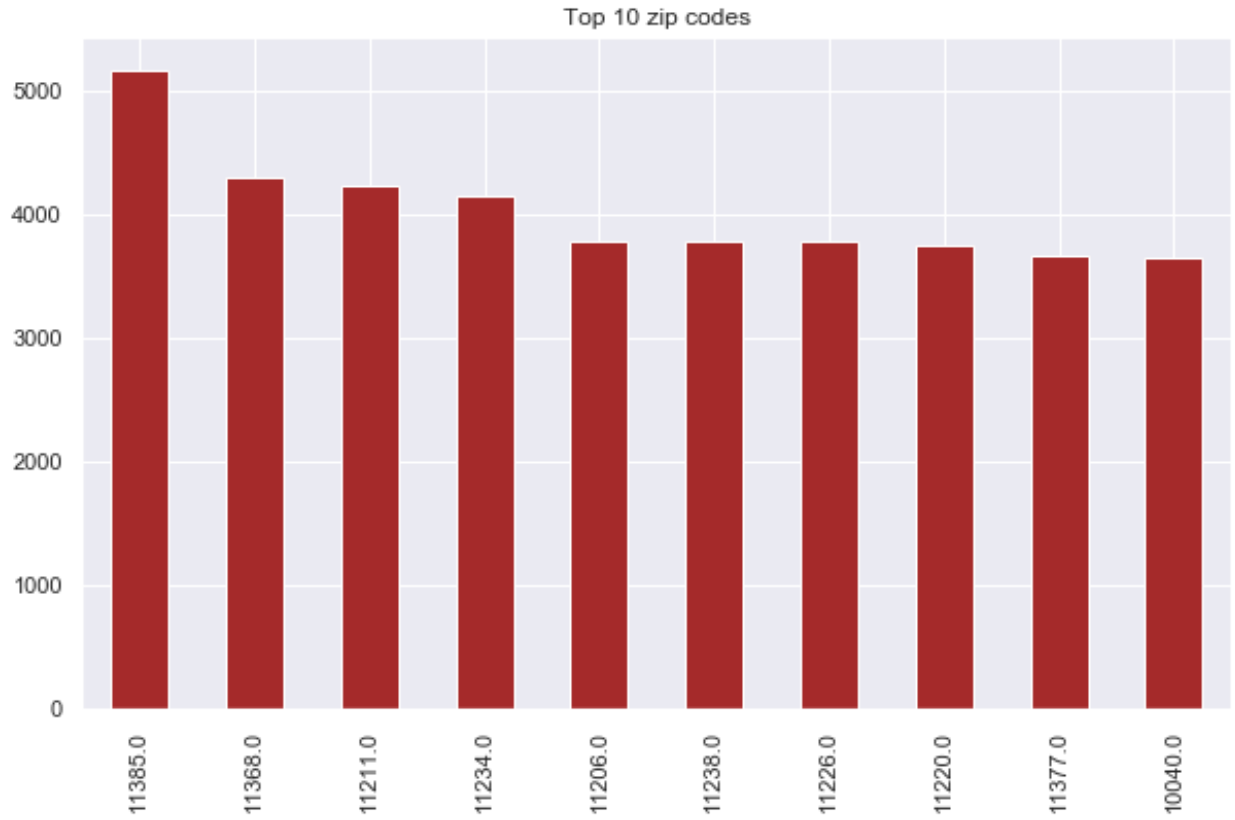


Incident Zip
Data Profiling of 'Incident Zip ' feature.

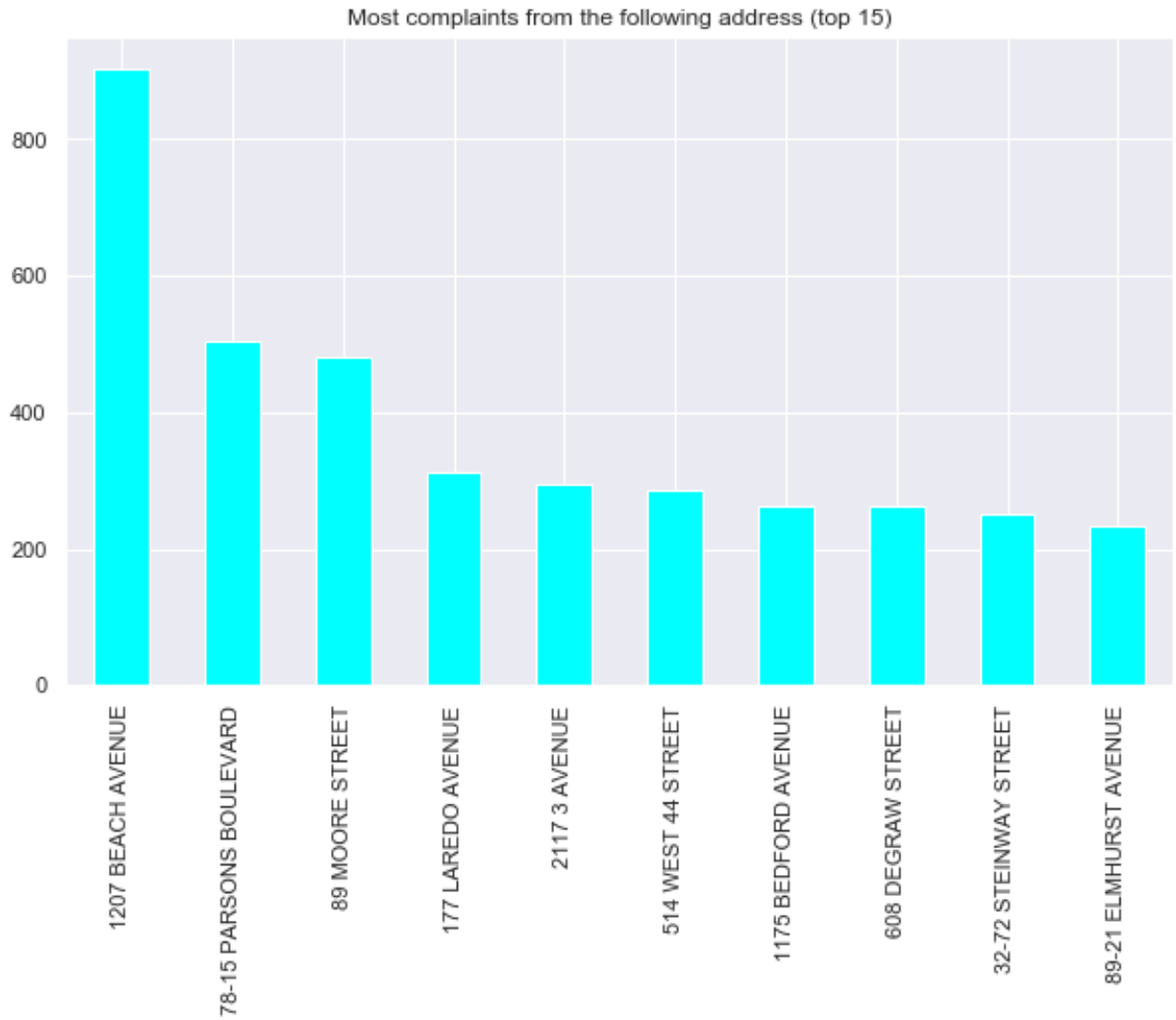
```
df['Incident Zip'].value_counts().head(20)
(df['Incident Zip'].value_counts()).head(10).plot(kind = 'bar', figsize= (10,6), title='Top 10 zip codes',color='brown')
```

Please Note:
a lot of complaints have come from the following zip code : 11385
followed by zip code : 11368tur

```
<matplotlib.axes._subplots.AxesSubplot at 0x270bb993088>
```




```
# Incident Address
# Data Profiling of 'Descriptor' feature.

(df['Incident Address'].value_counts()) .head (10)
(df['Incident Address'].value_counts()).head(10).plot(kind = 'bar', figsize= (10,6), title='Most complaints from the following address (top 15)',color='cyan')

# Please Note:
# Repeated complaints have come from the following address.
# 1207 BEACH AVENUE          904
# 78-15 PARSONS BOULEVARD    505
# 89 MOORE STREET            480
# 177 LAREDO AVENUE          311
# 2117 3 AVENUE              295
# 514 WEST 44 STREET         287
# 1175 BEDFORD AVENUE        264
# 608 DEGRAW STREET          263
# 32-72 STEINWAY STREET      251
# 89-21 ELMHURST AVENUE      234

<matplotlib.axes._subplots.AxesSubplot at 0x270c7a34ac8>
```

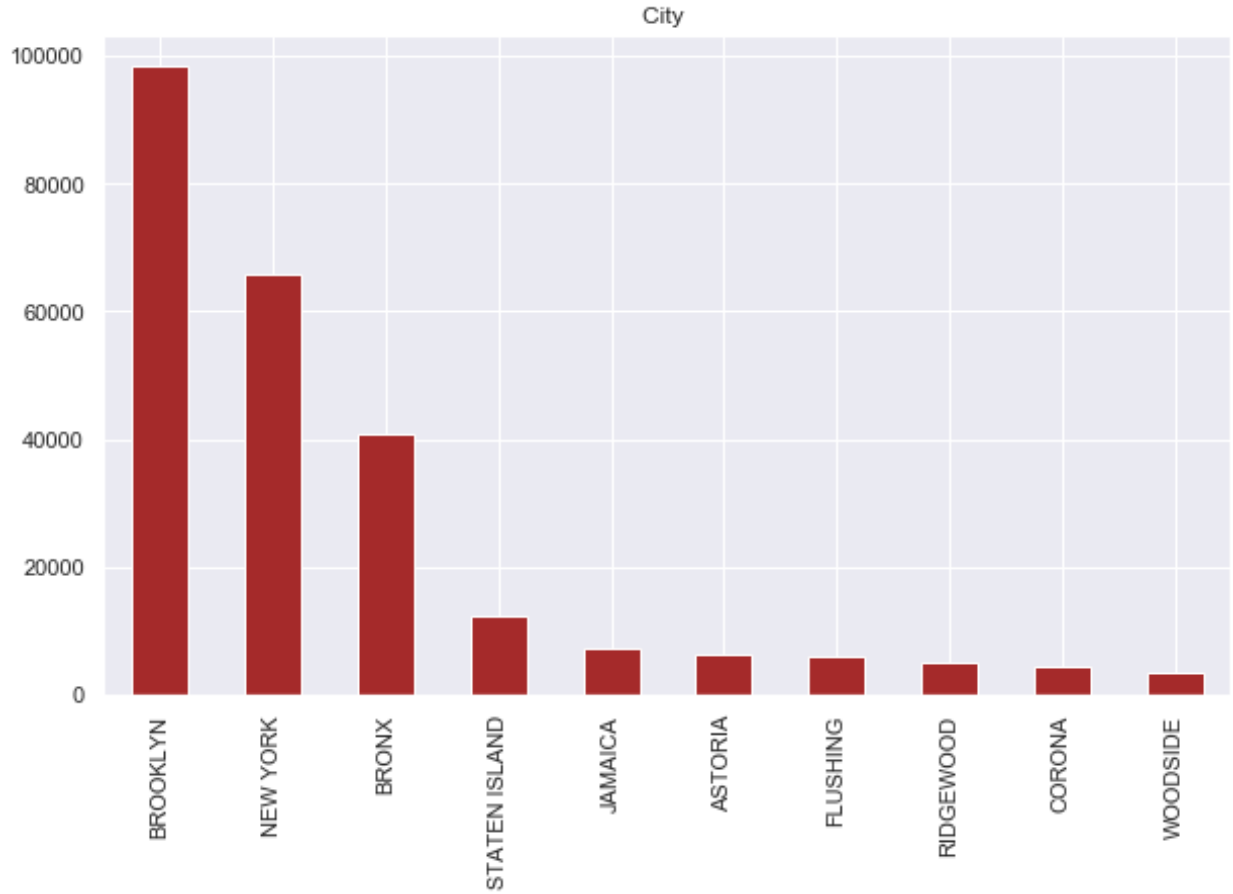



```
# Data Profiling of 'City' feature.
df['City'].value_counts()
(df['City'].value_counts()).head(10).plot(kind = 'bar', figsize= (10,6), title='City ',color='brown')
```

```
# Please Note:
# Major complaints have come from Brooklyn,New York and Bronx.
```

```
# BROOKLYN          98307
# NEW YORK          65994
# BRONX             40702
# STATEN ISLAND     12343
# JAMAICA            7296
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x270c79f1208>
```




```
# Status
df['Status'].value_counts()

# Please Note:
# 298471 complaints have been closed.
# 1439 complaints are still open.
# 786 complaints are assigned to other department.
# 2 complaints are getting drafted.
```

```
Closed      298471
Open          1439
Assigned       786
Draft            2
Name: Status, dtype: int64
```

```
# Data profiling for 'Request_Closing_Time' column.
df['Request_Closing_Time'].describe()

# please Note:
#1) the minimum time taken to close a complaint or to (attend and close) is 1 hour.
```

#2) the maximum time taken to close a complaint or to (attend and close) is 24 days 16 hours and 52 minutes.

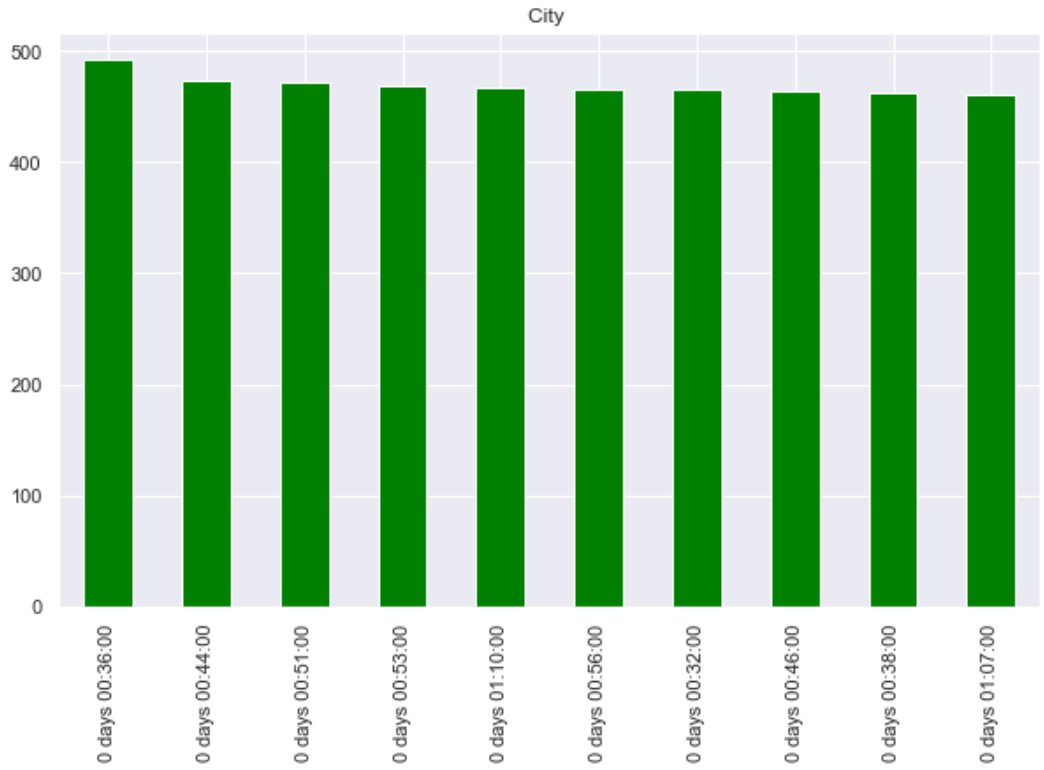
```
mean      0 days 04:18:51.832782
std       0 days 06:05:22.141833
min              0 days 00:01:00
25%              0 days 01:16:33
50%       0 days 02:42:55.500000
75%              0 days 05:21:00
max             24 days 16:52:22
Name: Request_Closing_Time, dtype: object
```

df['Request_Closing_Time']

```
df['Request_Closing_Time'].value_counts().head()
(df['Request_Closing_Time'].value_counts()).head(10).plot(kind = 'bar', figsize= (10,6), title='City ',color='green')
```

it basically gives top 5 complaints which were closed in minimum time.
Please Note:
492 complaints were closed in 36 minutes.
473 complaints were closed in 44 minutes.
472 complaints were closed in 51 minutes.
468 complaints were closed in 53 minutes.
467 complaints were closed in 1hr 10 minutes.

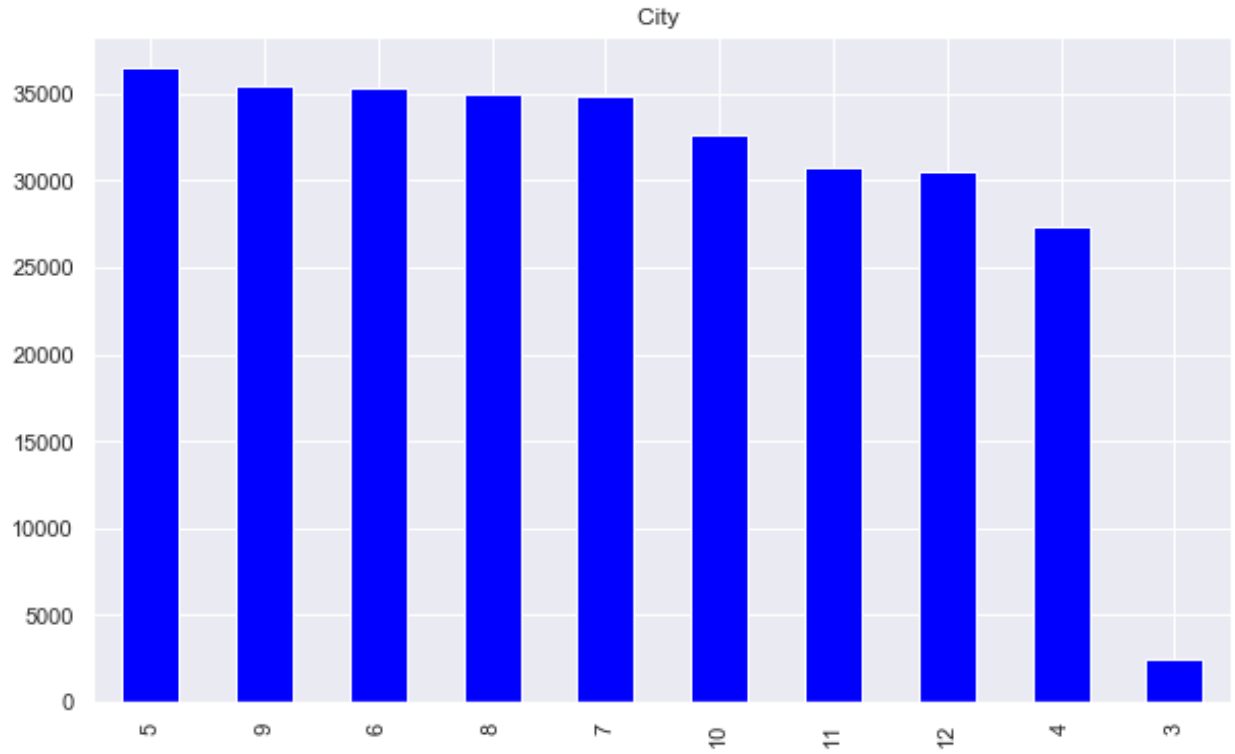
```
<matplotlib.axes._subplots.AxesSubplot at 0x270f3d49948>
```



```
df['Month'].value_counts()
(df['Month'].value_counts()).head(10).plot(kind = 'bar', figsize= (10,6), title='City ',color='blue')
```

Please Note:
Major complaints are in the months of May 2015. followed by sept 2015
In March the complaints are less because the data collected is from 29 March 2015 , we do not have date for the entire month.
5 (May) 36437
9 (Sept) 35427
6 (June) 35315
8 (Aug) 34956
7 (July) 34888

```
<matplotlib.axes._subplots.AxesSubplot at 0x270e0a5db08>
```




```
df['Duration_Minutes'].value_counts().head(20)
```

```
36.0    492
44.0    476
51.0    474
53.0    471
70.0    469
32.0    467
46.0    467
56.0    466
```

```
67.0    463
38.0    463
42.0    462
35.0    461
47.0    459
39.0    459
43.0    458
55.0    456
52.0    456
40.0    455
60.0    454
54.0    451
Name: Duration_Minutes, dtype: int64
```

df[['Duration_Minutes','Request_Closing_Time']].describe()

	Duration_Minutes	Request_Closing_Time
count	298534.000000	298534
mean	235.763797	0 days 04:18:51.832782
std	231.619407	0 days 06:05:22.141833
min	0.000000	0 days 00:01:00
25%	76.000000	0 days 01:16:33
50%	161.300000	0 days 02:42:55.500000
75%	316.000000	0 days 05:21:00
max	1439.916667	24 days 16:52:22

these are the least occuring complaint types in the below mentioned months.

df.groupby("Month")['Complaint Type'].agg("min")

```
Month
3       Animal Abuse
4     Agency Issues
5     Agency Issues
6     Agency Issues
```

```
7      Agency Issues
8       Animal Abuse
9       Animal Abuse
10      Animal Abuse
11      Animal Abuse
12      Animal Abuse
Name: Complaint Type, dtype: object
```

these are the maximum occuring complaint types in the below mentioned months

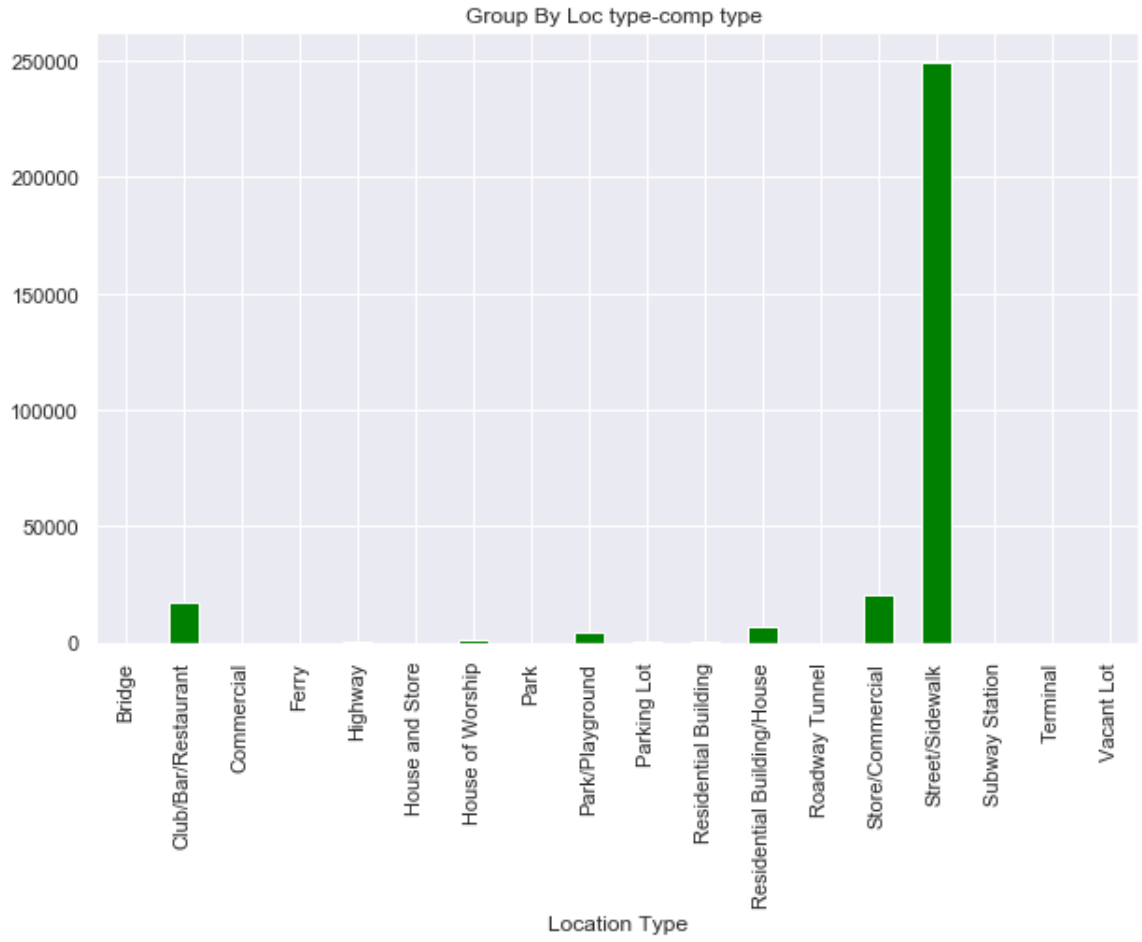
```
df.groupby("Month")['Complaint Type'].agg("max")
```

```
Month
3      Vending
4      Vending
5      Vending
6      Vending
7      Vending
8      Vending
9      Vending
10     Vending
11     Vending
12     Vending
Name: Complaint Type, dtype: object
```

```
df.groupby('Location Type')['Complaint Type'].agg("count")
(df.groupby('Location Type')['Complaint Type'].agg("count")).head (20).plot(kind = 'bar', figsize= (10,6),
title='Group By Loc type-comp type ',color='green')
```

Please Note
A majority of complaints have come from location type 'Street/Sidewalk' (249299 complaints),
followed by Store/commercial (20381 complaints)

```
<matplotlib.axes._subplots.AxesSubplot at 0x270f33a2908>
```




```
df.groupby('Location Type')['Complaint Type'].agg("max")
```

```
# Please Note:
# According to the location type these are the maximum occuring complaints received.
```

```
# Bridge                        Homeless Encampment
# Club/Bar/Restaurant                Urinating in Public
# Commercial                          Animal Abuse
# Ferry                          Ferry Complaint
# Highway                             Traffic
# House and Store                     Animal Abuse
# House of Worship             Noise - House of Worship
# Park                           Animal in a Park
# Park/Playground                        Vending
# Parking Lot                  Posting Advertisement
# Residential Building                Animal Abuse
# Residential Building/House                Vending
# Roadway Tunnel                      Traffic
# Store/Commercial                       Vending
```

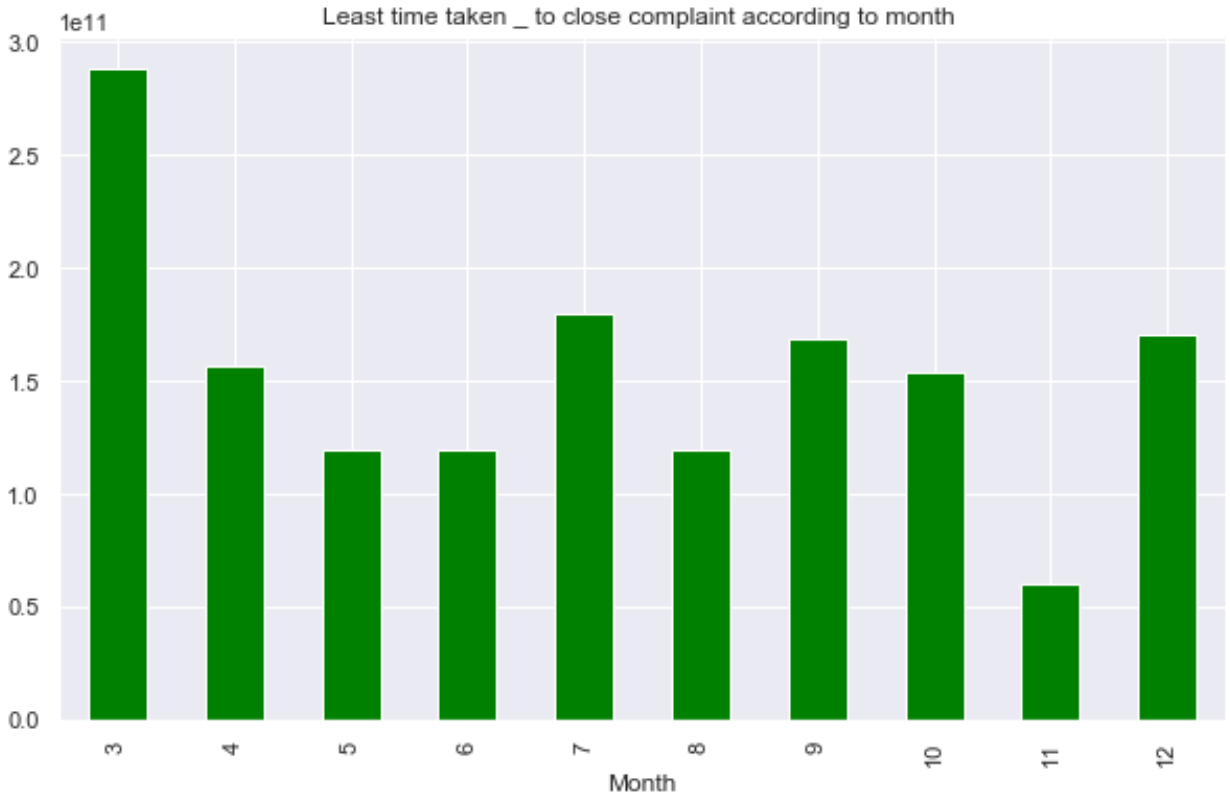
```
# Street/Sidewalk                    Vending
Location Type
Bridge                                Homeless Encampment
Club/Bar/Restaurant                   Urinating in Public
Commercial                                   Animal Abuse
Ferry                                    Ferry Complaint
Highway                                           Traffic
House and Store                              Animal Abuse
House of Worship                  Noise - House of Worship
Park                                      Animal in a Park
Park/Playground                                   Vending
Parking Lot                          Posting Advertisement
Residential Building                         Animal Abuse
Residential Building/House                        Vending
Roadway Tunnel                                    Traffic
Store/Commercial                                  Vending
Street/Sidewalk                                   Vending
Subway Station                        Urinating in Public
Terminal                                 Ferry Complaint
Vacant Lot                               Derelict Vehicle
Name: Complaint Type, dtype: object
```

To Find out according to particular Month what is the least time taken to close the complaint.

```
df.groupby("Month")['Request_Closing_Time'].agg("min")
(df.groupby("Month")['Request_Closing_Time'].agg("min")).head(10).plot(kind = 'bar', figsize= (10,6), title='Least time taken _ to close complaint according to month ',color='green')
```

Please Note:
In March 2015 the least time taken the close a given complaint is 4 hrs 48 min.
In April 2015 the least time taken the close a given complaint is 2hrs 37 min. and so on for other

```
<matplotlib.axes._subplots.AxesSubplot at 0x270ee8db488>
```



df.groupby("Month")['Request_Closing_Time'].agg("max")

```
Month
3      2 days 08:23:14
4     14 days 00:50:05
5     24 days 16:52:22
6      4 days 01:10:41
7      4 days 09:50:00
8      4 days 12:09:00
9      3 days 23:37:21
10     5 days 21:09:56
11     7 days 01:06:48
12    24 days 01:21:36
Name: Request_Closing_Time, dtype: timedelta64[ns]
```

#complaint types on the average “Request_closing_time in minutes” .

df.groupby("Complaint Type")['Duration_Minutes'].agg("max")

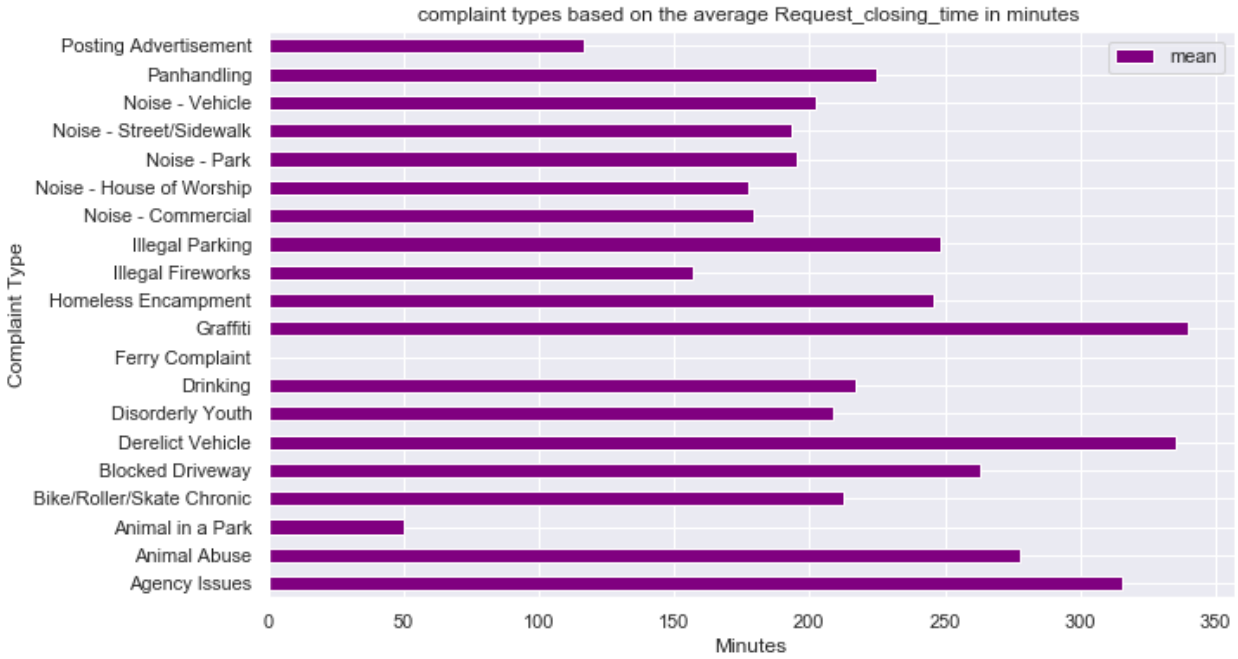
```
Complaint Type
Agency Issues                  623.000000
Animal Abuse                  1439.133333
Animal in a Park                50.083333
```

```
Bike/Roller/Skate Chronic    1079.916667
Blocked Driveway             1439.883333
Derelict Vehicle             1439.800000
Disorderly Youth             1236.116667
Drinking                     1378.000000
Ferry Complaint                      NaN
Graffiti                     1344.000000
Homeless Encampment          1411.000000
Illegal Fireworks             915.000000
Illegal Parking              1439.000000
Noise - Commercial           1435.266667
Noise - House of Worship     1294.000000
Noise - Park                 1398.000000
Noise - Street/Sidewalk      1439.916667
Noise - Vehicle              1437.000000
Panhandling                  1223.683333
Posting Advertisement         934.000000
Squeegee                      407.200000
Traffic                      1386.033333
Urinating in Public          1348.000000
Vending                      1430.000000
Name: Duration_Minutes, dtype: float64
```

```
df.groupby("Complaint Type")['Duration_Minutes'].agg("mean")
(df.groupby("Complaint Type")['Duration_Minutes'].agg(["mean"])).head(20).plot(kind = 'barh',figsize= (10,6),
title='complaint types based on the average Request_closing_time in minutes ',color='purple')
plt.ylabel('Complaint Type')
plt.xlabel('Minutes')
```

```
# Please Note:
# This tells us the "avg" time taken in minutes to close a particular complaint type.
# Agency Issues              315.619444
# Animal Abuse               277.573208
# Animal in a Park            50.083333
# Bike/Roller/Skate Chronic   212.402830
# Blocked Driveway            263.082093
# Derelict Vehicle            335.330298
# Disorderly Youth            208.480070
# Drinking                    217.027621
```

```
Text(0.5, 0, 'Minutes')
```



df.groupby(["Location Type",'Complaint Type'])['Duration_Minutes'].agg(["mean",'max'])

		mean	max
Location Type	**Complaint Type**		
Bridge	Homeless Encampment	229.158333	281.000000
	Drinking	241.187078	1378.000000
Club/Bar/Restaurant	Noise - Commercial	173.489104	1435.266667
	Urinating in Public	269.485714	1348.000000
Commercial	Animal Abuse	274.114516	1139.000000
...	...	...	...
Street/Sidewalk	Vending	227.460780	1430.000000
Subway Station	Animal Abuse	182.136364	472.016667
	Urinating in Public	69.127778	278.800000

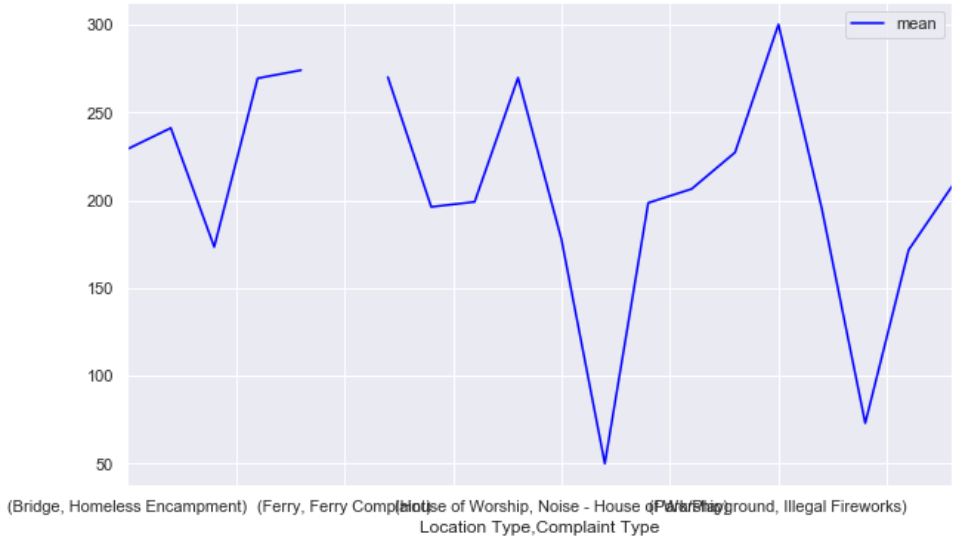
		mean	max
Location Type	**Complaint Type**		
Terminal	Ferry Complaint	NaN	NaN
Vacant Lot	Derelict Vehicle	242.721212	907.900000

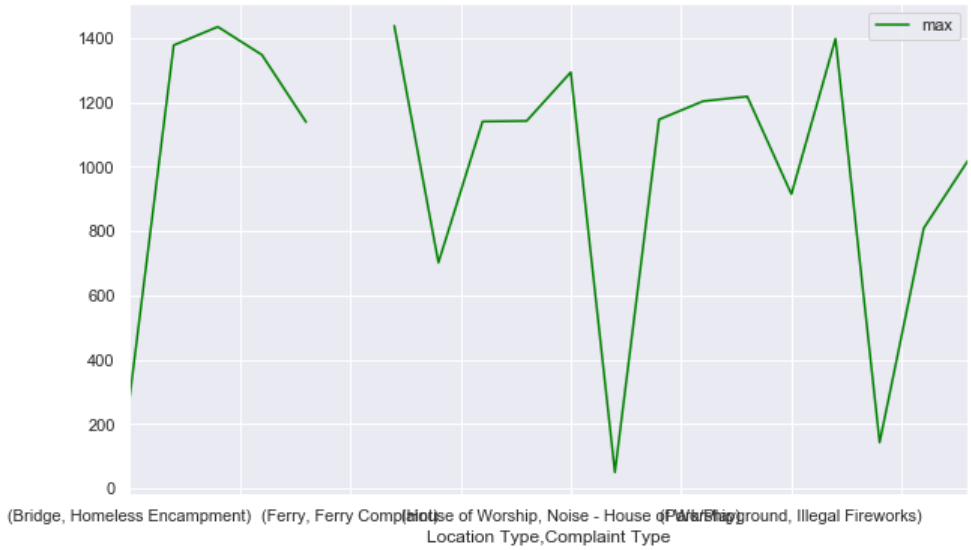
71 rows × 2 columns

```
df.groupby(["Location Type",'Complaint Type'])['Duration_Minutes'].agg(["mean",'max'])

(df.groupby(["Location Type",'Complaint Type'])['Duration_Minutes'].agg(["mean"])).head
(20).plot(figsize=(10,6),color=['blue'])
(df.groupby(["Location Type",'Complaint Type'])['Duration_Minutes'].agg(["max"])).head
(20).plot(figsize=(10,6),color=['green'])
```

Please Note:
This table tells us the "AVG time and maximum time " taken to close a particular complaint type at particular location.
For e.g:
Under location type "Bridge" and complaint type " Homeless Encampment" the averge time taken to close the complaint is 229 min
maximum time taken is 281 minutes.
Complaint Type 'Graffiti' has highest average of 434 minutes to close the complaint for the location store/commercial.

```
<matplotlib.axes._subplots.AxesSubplot at 0x270e2732548>
```





df.groupby(['Complaint Type'])['Duration_Minutes'].agg(["mean"])

Please Note:

The average time taken to close a particular Complaint type is as follows:

 # agency Issues
 # 315.619444

 # Animal Abuse
 # 277.573208

 # Animal in a Park
 # 50.083333

 # Bike/Roller/Skate Chronic
 # 212.402830

 # Blocked Driveway
 # 263.082093

 # Derelict Vehicle
 # 335.330298

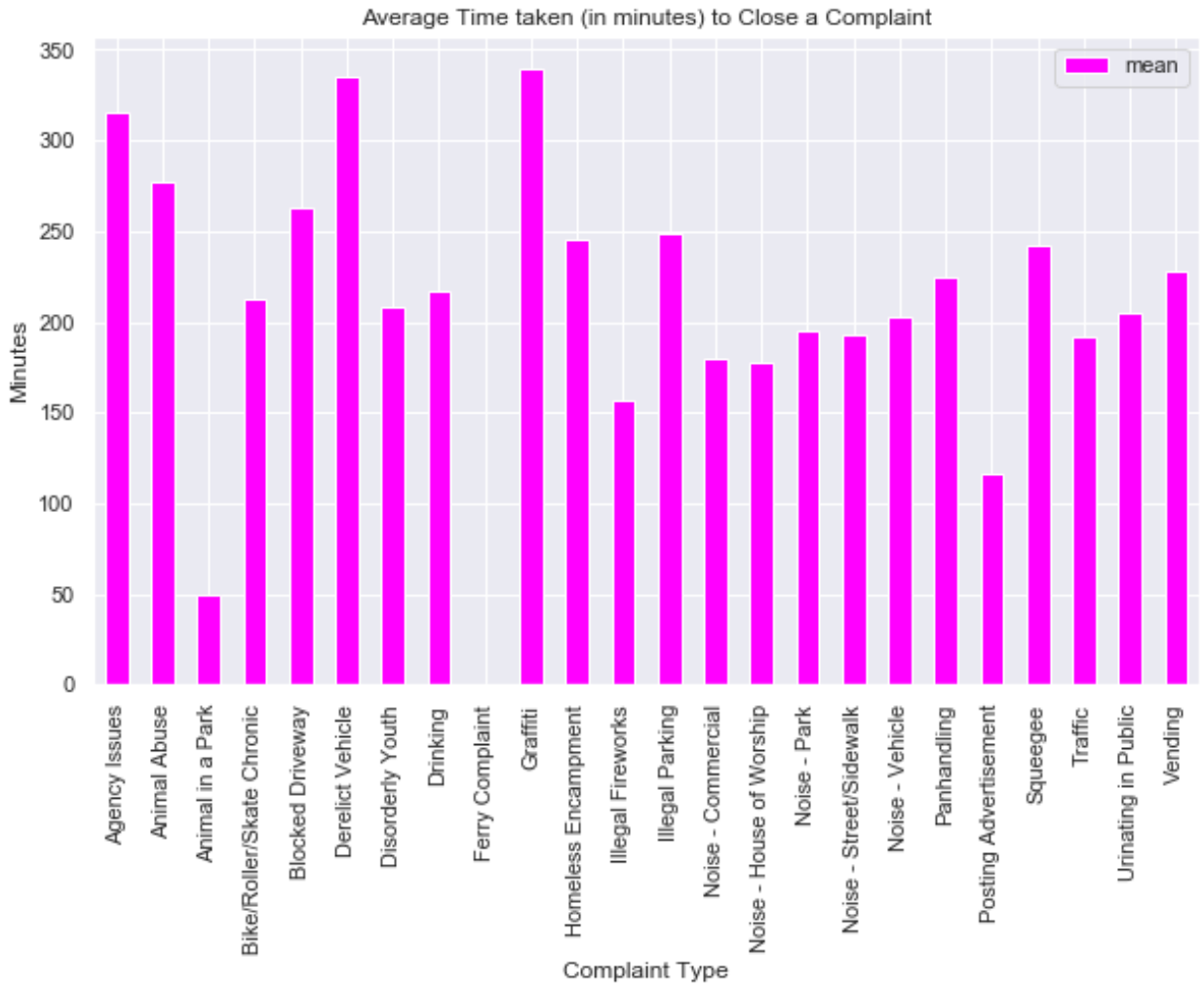
 # Disorderly Youth
 # 208.480070

Complaint Type	mean
Agency Issues	315.619444
Animal Abuse	277.573208
Animal in a Park	50.083333
Bike/Roller/Skate Chronic	212.402830
Blocked Driveway	263.082093
Derelict Vehicle	335.330298
Disorderly Youth	208.480070
Drinking	217.027621
Ferry Complaint	NaN
Graffiti	339.871534
Homeless Encampment	245.955948
Illegal Fireworks	157.096925
Illegal Parking	248.584628
Noise - Commercial	179.473576
Noise - House of Worship	177.647417
Noise - Park	195.335148
Noise - Street/Sidewalk	193.294691
Noise - Vehicle	202.573797

Complaint Type	mean
Agency Issues	315.619444
Animal Abuse	277.573208
Panhandling	224.595574
Posting Advertisement	116.325977
Squeegee	242.737500
Traffic	191.867486
Urinating in Public	205.437697
Vending	228.313382

```
df.groupby(['Complaint Type'])['Duration_Minutes'].agg(["mean"]).plot (kind = 'bar',figsize = (10,6) ,color= 'magenta')

plt.xlabel('Complaint Type')
plt.ylabel('Minutes')
plt.title('Average Time taken (in minutes) to Close a Complaint')
```

```
Text(0.5, 1.0, 'Average Time taken (in minutes) to Close a Complaint')
```



df.groupby(['Month'])['Duration_Minutes'].agg(["mean"])

Please Note:

This tells us the average time taken (in minutes) to close a particular complaint in the particular months.

 # 3
 # 214.345828

 # 4
 # 207.810550

 # 5
 # 218.354513

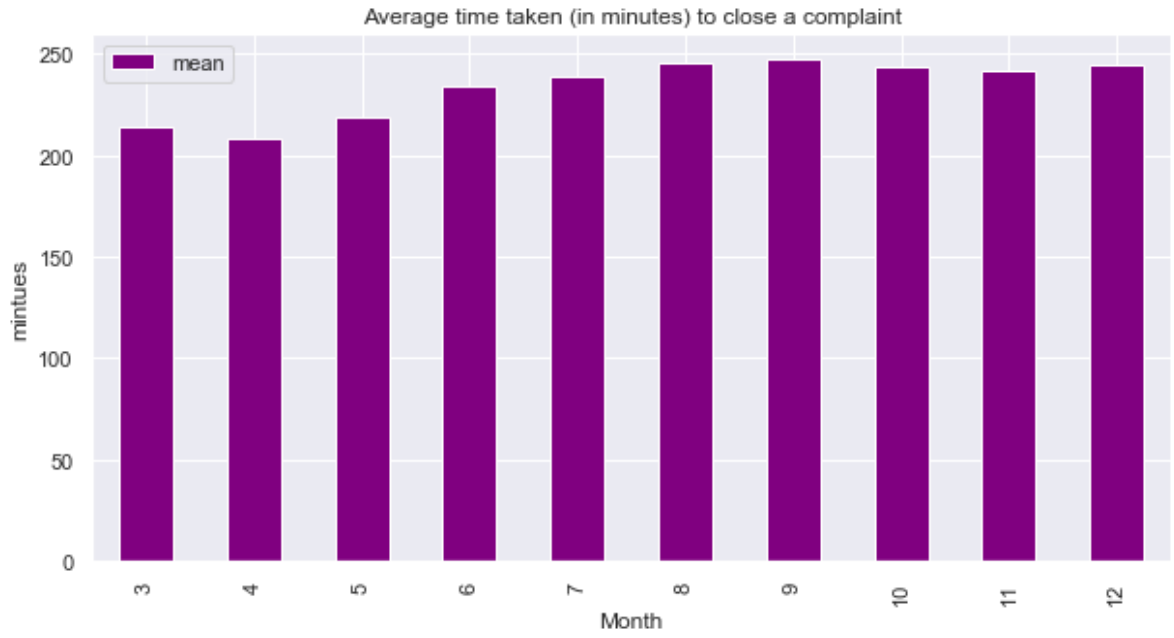
 # 6

```
# 233.716719

# 7
# 238.769073

# 8
# 245.543672

# 9
# 247.157852

# 10
# 243.121011

# 11
# 241.428750

# 12
# 244.20099
```

Month	mean
3	214.345828
4	207.810550
5	218.354513
6	233.716719
7	238.769073
8	245.543672
9	247.157852
10	243.121011
11	241.428750

Month	mean
12	244.200998

```
(df.groupby(['Month'])['Duration_Minutes'].agg(["mean"])).plot(kind = 'bar',figsize = (10,5),color= 'purple')

plt.ylabel('mintues')
plt.title('Average time taken (in minutes) to close a complaint')
Text(0.5, 1.0, 'Average time taken (in minutes) to close a complaint')
```




```
df.groupby(['Month'])['Duration_Minutes'].agg(["max"])
```

Please Note:
The maximum time taken (in minutes) to complete a given complaint in the Month of March is 1409 minutes.
The maximum time taken (in minutes) to complete a given complaint in the Month of April is 1424 minutes and so on.

Month	max
3	1409.083333
4	1424.150000
5	1438.000000

Month	max
6	1439.883333
7	1439.300000
8	1439.916667
9	1439.133333
10	1438.850000
11	1439.800000
12	1439.683333

#Question 1: whether the average response time across complaint types is similar or not (overall)

#Choosing the features for the above Problem.
#Complaint Type, Request Closing Time and Duration Minutes are the features choosen for solving the above problem

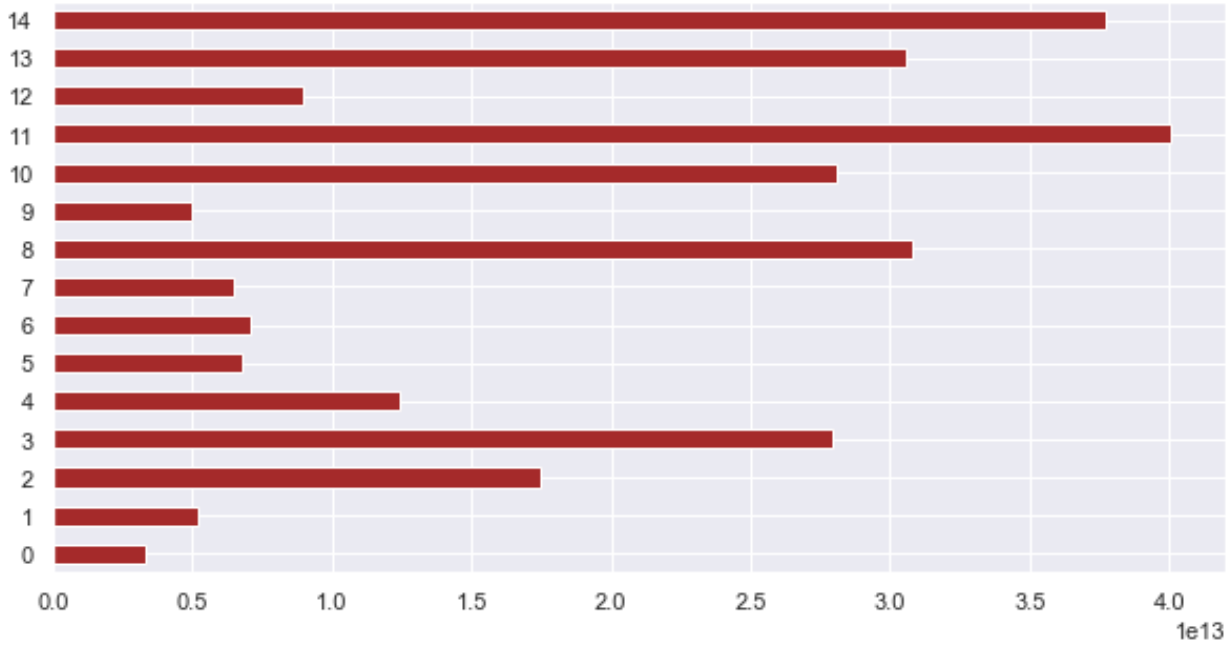
assigning the selected features to a new variable.
dropping all the NaN values in the rows.

```
dfnew= df[['Complaint Type','Request_Closing_Time','Duration_Minutes']]

dfnew_after_drop=dfnew.dropna(axis=0, inplace=False)

dfnew_after_drop.info()
<class 'pandas.core.frame.DataFrame'>
Int64Index: 298534 entries, 0 to 300697
Data columns (total 3 columns):
Complaint Type          298534 non-null object
Request_Closing_Time    298534 non-null timedelta64[ns]
Duration_Minutes        298534 non-null float64
dtypes: float64(1), object(1), timedelta64[ns](1)
memory usage: 9.1+ MB

dfnew_after_drop.isna().sum()
Complaint Type          0
```

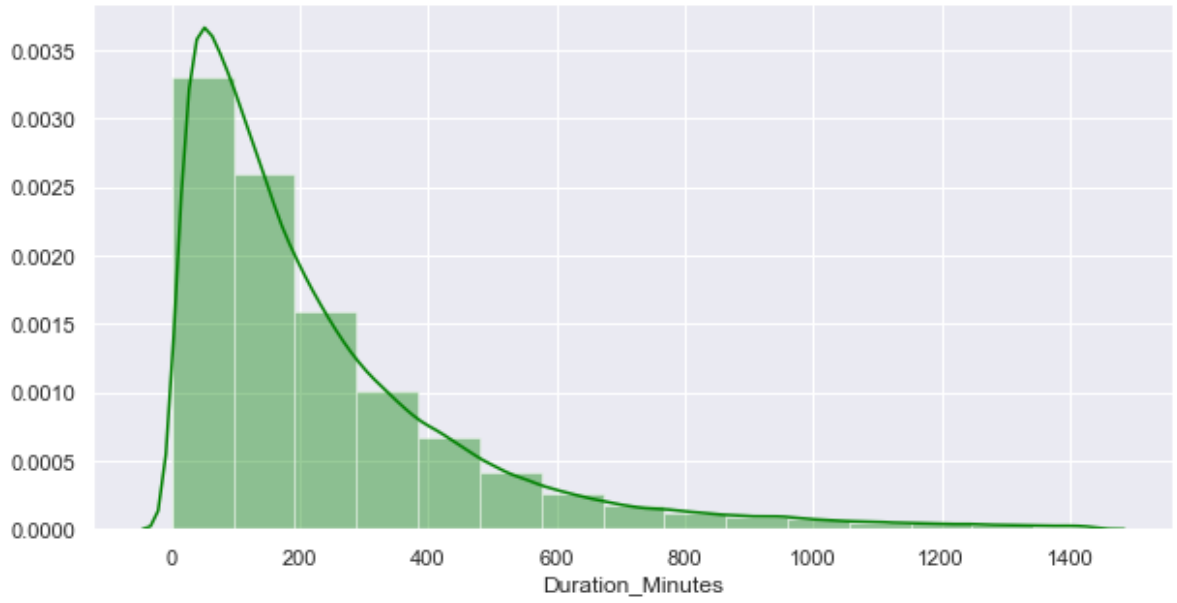
```
Request_Closing_Time     0
Duration_Minutes         0
dtype: int64
```

```
dfnew_after_drop['Request_Closing_Time'].head(15).plot(kind = 'barh',figsize = (10,5),color= 'brown')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x270f3935cc8>
```




```
sns.set(rc={'figure.figsize':(10,5)})
sns.distplot(dfnew_after_drop['Duration_Minutes'],rug = False,hist=True,kde = True,color='green',bins = 15)
```

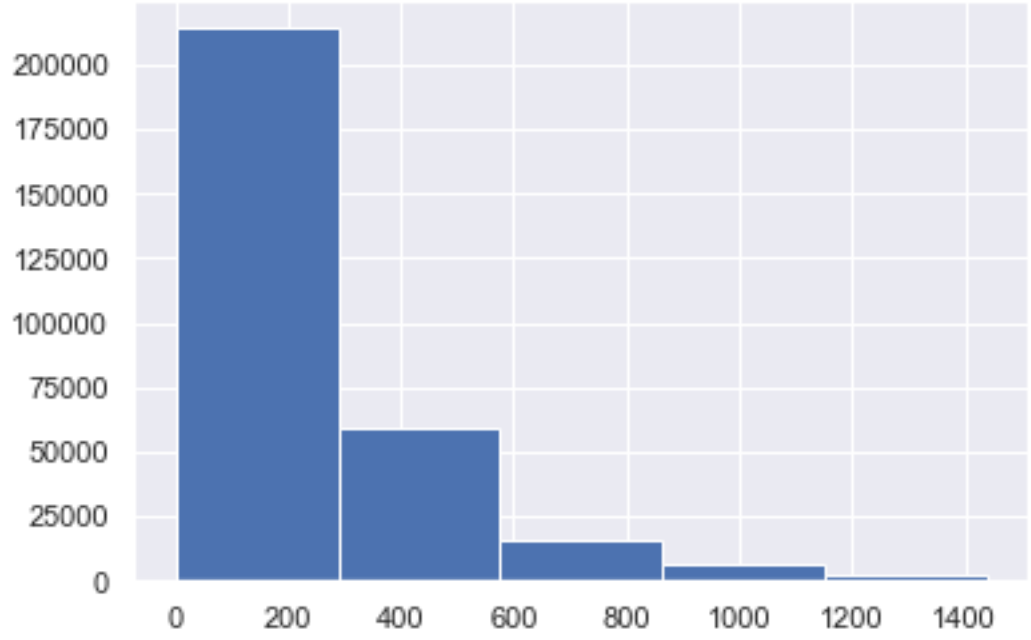


Please Note:

As this distribution is right skewed, there are lot of outliers.

majority of the data falls between 0 to 400 minutes.

```
<matplotlib.axes._subplots.AxesSubplot at 0x270854ac448>
```




```
dfnew_after_drop['Duration_Minutes'].hist(bins=5)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x270852e5508>
```

Please Note:

When we have factors with more than 2 levels we use '2 'T' TEST
The features chosen for the '2t TEST' are 'Location Type' and 'Duration_Minutes' and "Request_Closing_Time'

-Null Hypothesis (H0)-- the average response time across the complaint types is similar --H0 is accepted (Fail to reject Null Hypothesis)
-Alternate Hypothesis (H1) ---the average response time across the complaint types is NOT similar (different) --- H0(Null Hypothesis) is rejected

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

```
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
```

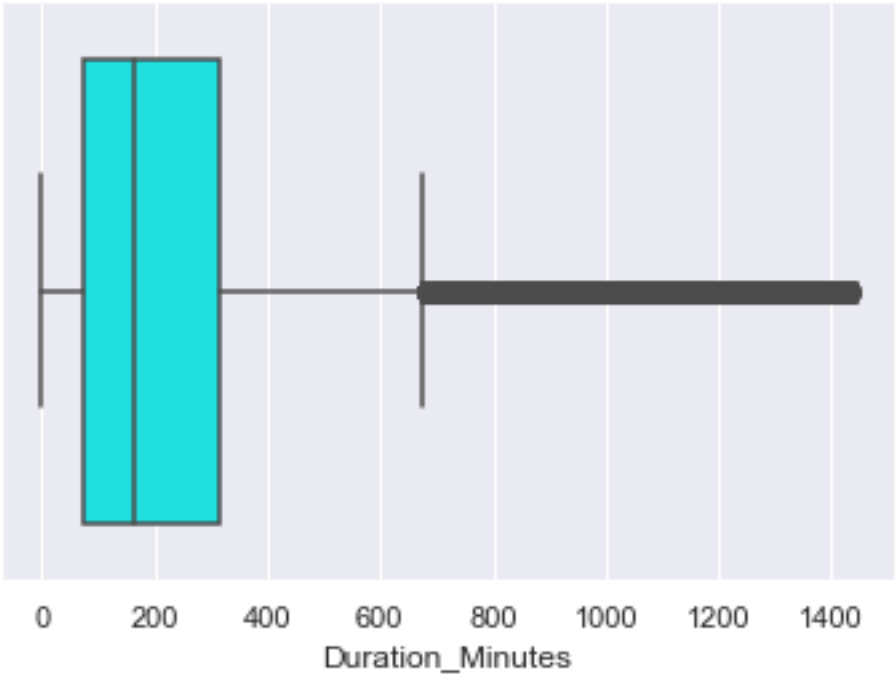
```
df[[ 'Duration_Minutes' , 'Request_Closing_Time']].describe()
```

	Duration_Minutes	Request_Closing_Time
count	298534.000000	298534
mean	235.763797	0 days 04:18:51.832782
std	231.619407	0 days 06:05:22.141833
min	0.000000	0 days 00:01:00
25%	76.000000	0 days 01:16:33
50%	161.300000	0 days 02:42:55.500000
75%	316.000000	0 days 05:21:00
max	1439.916667	24 days 16:52:22

```
import statsmodels.api as sm
from statsmodels.formula.api import ols
sns.boxplot('Duration_Minutes',data= df,color='cyan' )
```

Please Note:
There are lot of outliers
the mean of duration minutes is around 180 minutes

```
<matplotlib.axes._subplots.AxesSubplot at 0x270ee9939c8>
```



#dfnew_after_drop Request_Closing_Time Duration_Minutes Location Type

```
dfnew_after_drop [dfnew_after_drop['Complaint Type']=='Blocked Driveway'] [['Request_Closing_Time']].mean()
```

```
Request_Closing_Time    04:44:27.258638
dtype: timedelta64[ns]
```

#dfnew_after_drop Request_Closing_Time Duration_Minutes Location Type

```
dfnew_after_drop [dfnew_after_drop['Complaint Type']=='Illegal Parking'] [['Request_Closing_Time']].mean()
```

```
Request_Closing_Time    04:30:04.145454
dtype: timedelta64[ns]
```

```
from scipy import stats
```

```
dfnew_after_drop['Complaint Type'].value_counts()
Blocked Driveway             76810
Illegal Parking              74532
Noise - Street/Sidewalk      48076
Noise - Commercial           35247
```

```
Derelict Vehicle             17588
Noise - Vehicle              17033
Animal Abuse                  7768
Traffic                       4496
Homeless Encampment           4416
Noise - Park                  4022
Vending                       3795
Drinking                      1275
Noise - House of Worship       929
Posting Advertisement          648
Urinating in Public            592
Bike/Roller/Skate Chronic      424
Panhandling                    305
Disorderly Youth               286
Illegal Fireworks              168
Graffiti                       113
Agency Issues                    6
Squeegee                         4
Animal in a Park                 1
Name: Complaint Type, dtype: int64
```

CREATE blocked_driveway_duration_minutes, illegal_parking_duration_minutes
'Blocked Driveway ',Illegal Parking and Noise - Street/Sidewalk are taken as these are the most repeated complaint types.

```
blocked_driveway_duration_minutes = dfnew_after_drop[dfnew_after_drop['Complaint Type']=='Blocked Driveway']['Duration_Minutes']
illegal_parking_duration_minutes = dfnew_after_drop[dfnew_after_drop['Complaint Type']=='Illegal Parking']['Duration_Minutes']
noise_street_sidewalk_duration_minutes = dfnew_after_drop[dfnew_after_drop['Complaint Type']=='Noise - Street/Sidewalk']['Duration_Minutes']
```

```
stats.ttest_ind(blocked_driveway_duration_minutes, illegal_parking_duration_minutes)
Ttest_indResult(statistic=11.878368607446648, pvalue=1.5852726168784882e-32)
```

```
stats.ttest_ind(illegal_parking_duration_minutes,noise_street_sidewalk_duration_minutes)
Ttest_indResult(statistic=42.5796940247014, pvalue=0.0)
```

Please Note:
-The P value = 1.585272616878488e-32 i.e (10^-32). [blocked_driveway_duration_minutes, illegal_parking_duration_minutes] as the p-value is less than 0.05 , the null hypothesis is rejected.-The average response time across the complaint types is NOT similar (it is different).it is distributed -blocked_driveway_duration_minutes, illegal_parking_duration_minutes

-The P value = 0.0 i.e (10^-308).
[illegal_parking_duration_minutes,noise_street_sidewalk_duration_minutes]
The p-value is less than 0.05 , the null hypothesis is rejected. the average response time across the complaint types is NOT similar (different).it is distributed - illegal_parking_duration_minutes,noise_street_sidewalk_duration_minutes

Question 2:
Are the type of complaint or service requested and location related ?

-Null Hypothesis (H0)-- The type of complaint **or** service requested **and** locations are related to each other.
-H0 **is** accepted (Fail to reject Null Hypothesis)

-Alternate Hypothesis (H1) ---The type of complaint **or** service requested **and** locations are NOT related to each other. -H0(Null Hypothesis) **is** rejected

#Preparing Data

```
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300698 entries, 0 to 300697
Data columns (total 56 columns):
Unique Key                      300698 non-null int64
Created Date                    300698 non-null datetime64[ns]
Closed Date                     298534 non-null datetime64[ns]
Agency                          300698 non-null object
Agency Name                     300698 non-null object
Complaint Type                  300698 non-null object
Descriptor                      294784 non-null object
Location Type                   300567 non-null object
Incident Zip                    298083 non-null float64
Incident Address                256288 non-null object
Street Name                     256288 non-null object
Cross Street 1                  251419 non-null object
Cross Street 2                  250919 non-null object
Intersection Street 1           43858 non-null object
Intersection Street 2           43362 non-null object
Address Type                    297883 non-null object
City                            298084 non-null object
Landmark                        349 non-null object
Facility Type                   298527 non-null object
Status                          300698 non-null object
Due Date                        300695 non-null object
Resolution Description          300698 non-null object
Resolution Action Updated Date  298511 non-null datetime64[ns]
```

```
Community Board                   300698 non-null object
Borough                           300698 non-null object
X Coordinate (State Plane)        297158 non-null float64
Y Coordinate (State Plane)        297158 non-null float64
Park Facility Name                300698 non-null object
Park Borough                      300698 non-null object
School Name                       300698 non-null object
School Number                     300698 non-null object
School Region                     300697 non-null object
School Code                       300697 non-null object
School Phone Number               300698 non-null object
School Address                    300698 non-null object
School City                       300698 non-null object
School State                      300698 non-null object
School Zip                        300697 non-null object
School Not Found                  300698 non-null object
School or Citywide Complaint      0 non-null float64
Vehicle Type                      0 non-null float64
Taxi Company Borough              0 non-null float64
Taxi Pick Up Location             0 non-null float64
Bridge Highway Name               243 non-null object
Bridge Highway Direction          243 non-null object
Road Ramp                         213 non-null object
Bridge Highway Segment            213 non-null object
Garage Lot Name                   0 non-null float64
Ferry Direction                   1 non-null object
Ferry Terminal Name               2 non-null object
Latitude                          297158 non-null float64
Longitude                         297158 non-null float64
Location                          297158 non-null object
Request_Closing_Time              298534 non-null timedelta64[ns]
Month                             300698 non-null int64
Duration_Minutes                  298534 non-null float64
dtypes: datetime64[ns](3), float64(11), int64(2), object(39),
timedelta64[ns](1)
memory usage: 128.5+ MB
```

```
dfnew_1= df[['Complaint Type','Latitude','Longitude']]
dfnew_1.isna().sum()
```

```
Complaint Type       0
Latitude          3540
Longitude         3540
dtype: int64
```

```
dfnew_after_drop_1=dfnew_1.dropna(axis=0, inplace=False)
```

```
# A new feature is created 'Location' as its from a dataframe object type and cannot be converted to int or float
```

```
dfnew_after_drop_1['Location'] = list(zip(dfnew_after_drop_1.Latitude,dfnew_after_drop_1.Longitude))
```

```
E:\Programs\Anaconda3\lib\site-packages\ipykernel_launcher.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

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

```
dfnew_after_drop_1.head()
```

	Complaint Type	Latitude	Longitude	Location
0	Noise - Street/Sidewalk	40.865682	-73.923501	(40.86568154, -73.92350096)
1	Blocked Driveway	40.775945	-73.915094	(40.77594531, -73.91509394)
2	Blocked Driveway	40.870325	-73.888525	(40.87032452, -73.88852464)
3	Illegal Parking	40.835994	-73.828379	(40.83599405, -73.82837940000002)
4	Illegal Parking	40.733060	-73.874170	(40.73305962, -73.87416976)

```
dfnew_after_drop_1.isna().sum()
```

```
Complaint Type    0
Latitude          0
Longitude         0
Location          0
dtype: int64
```

```
dfnew_after_drop_1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 297158 entries, 0 to 300697
Data columns (total 4 columns):
Complaint Type    297158 non-null object
Latitude          297158 non-null float64
Longitude         297158 non-null float64
```

```
Location          297158 non-null object
dtypes: float64(2), object(2)
memory usage: 11.3+ MB
```

dfnew_after_drop_1['Complaint Type'].value_counts()

```
Blocked Driveway             76723
Illegal Parking              74066
Noise - Street/Sidewalk      47793
Noise - Commercial           35176
Derelict Vehicle             17519
Noise - Vehicle              16873
Animal Abuse                  7747
Traffic                       4475
Homeless Encampment           4367
Noise - Park                  3929
Vending                       3776
Drinking                      1271
Noise - House of Worship       922
Posting Advertisement          649
Urinating in Public            592
Bike/Roller/Skate Chronic      414
Panhandling                    301
Disorderly Youth               285
Illegal Fireworks              163
Graffiti                       113
Squeegee                         4
Name: Complaint Type, dtype: int64
```

dfnew_after_drop_1.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 297158 entries, 0 to 300697
Data columns (total 4 columns):
Complaint Type    297158 non-null object
Latitude          297158 non-null float64
Longitude         297158 non-null float64
Location          297158 non-null object
dtypes: float64(2), object(2)
memory usage: 11.3+ MB
```

On combining latitude and longitude results into object type & can't be converted to float hence 'location' variable is ignored

dfnew_after_drop_1['Complaint Type'].value_counts()

```
Blocked Driveway             76723
Illegal Parking              74066
Noise - Street/Sidewalk      47793
Noise - Commercial           35176
Derelict Vehicle             17519
Noise - Vehicle              16873
Animal Abuse                  7747
Traffic                       4475
Homeless Encampment           4367
Noise - Park                  3929
Vending                       3776
Drinking                      1271
Noise - House of Worship       922
Posting Advertisement          649
Urinating in Public            592
Bike/Roller/Skate Chronic      414
Panhandling                    301
Disorderly Youth               285
Illegal Fireworks              163
Graffiti                       113
Squeegee                         4
Name: Complaint Type, dtype: int64
```

dfnew_after_drop_1[dfnew_after_drop_1['Complaint Type']=='Blocked Driveway'] [['Location']].mode()

	Location
0	(40.76150067, -73.81339940000001)

dfnew_after_drop_1[dfnew_after_drop_1['Complaint Type']=='Blocked Driveway'] [['Location']].head()

	Location
1	(40.77594531, -73.91509394)
2	(40.87032452, -73.88852464)
7	(40.83750263, -73.90290517)
9	(40.62379307, -73.9995389)
10	(40.75259967, -73.89336321)

creating blocked_driveway_location, illegal_parking_location
'Blocked Driveway ',Illegal Parking and Noise - Street/Sidewalk. as taken as these are the most repeated complaint types.

```
blocked_driveway = dfnew_after_drop_1[dfnew_after_drop_1['Complaint Type']=='Blocked Driveway'][['Latitude','Longitude']]
illegal_parking = dfnew_after_drop_1[dfnew_after_drop_1['Complaint Type']=='Illegal Parking'][['Latitude','Longitude']]
noise_street_sidewalk = dfnew_after_drop_1[dfnew_after_drop_1['Complaint Type']=='Noise - Street/Sidewalk'][['Latitude','Longitude']]

stats.ttest_ind(blocked_driveway,illegal_parking)

Ttest_indResult(statistic=array([20.72505512, 71.36519054]),
pvalue=array([2.79800334e-95, 0.00000000e+00]))

stats.ttest_ind(illegal_parking,noise_street_sidewalk)
Ttest_indResult(statistic=array([-117.76704045,    8.70740178]),
pvalue=array([0.00000000e+00, 3.14703712e-18]))
```

Please Note:

-The P value =([2.79800334e-95, 0.00000000e+00]) , i.e (10^-95, 0).
[blocked_driveway_location, illegal_parking_location]
the p-value is less than 0.05 , the null hypothesis is rejected. The type of complaint or service requested and locations are NOT related to each other. It is distributed - blocked_driveway_location,illegal_parking_location.

-The P value =([0.00000000e+00, 3.14703712e-18]) , i.e (0, 10^-18).
[illegal_parking_location,noise_street_sidewalk_location]
-The p-value is less than 0.05 , the null hypothesis is rejected. -The type of complaint or service requested and locations are NOT related to each other. It is distributed - illegal_parking_location,noise_street_sidewalk_location.

<End of document>