Sam_Customerservice

May 10, 2020

[1]:
```
%matplotlib inline
### import libraries
import numpy as np
import pandas as pd


import matplotlib.pyplot as plt
from matplotlib import style
import seaborn as sns
```

[6]:
```
Customerservice = pd.read_csv ('311_Service_Requests_from_2010_to_Present.csv')
```

[8]:
```
Customerservice.head()
```

[8]:
```
   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]
```

[11]:
```
Customerservice.size
```

[11]:
```
15936994
```

[12]:
```
Customerservice.shape
```

[12]:
```
(300698, 53)
```

[17]:
```
Customerservice.columns
```

[17]:
```
Index(['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', 'Status', 'Due Date',
       'Resolution Description', 'Resolution Action Updated Date',
       'Community Board', 'Borough', 'X Coordinate (State Plane)',
       'Y Coordinate (State Plane)', 'Park Facility Name', 'Park Borough',
       'School Name', 'School Number', 'School Region', 'School Code',
```

```
       'School Phone Number', 'School Address', 'School City', 'School State',
       'School Zip', 'School Not Found', 'School or Citywide Complaint',
       'Vehicle Type', 'Taxi Company Borough', 'Taxi Pick Up Location',
       'Bridge Highway Name', 'Bridge Highway Direction', 'Road Ramp',
       'Bridge Highway Segment', 'Garage Lot Name', 'Ferry Direction',
       'Ferry Terminal Name', 'Latitude', 'Longitude', 'Location'],
      dtype='object')
```

[27]:
```
Customerservice['Complaint Type'].unique()
```

[27]:
```
array(['Noise - Street/Sidewalk', 'Blocked Driveway', 'Illegal Parking',
       'Derelict Vehicle', 'Noise - Commercial',
       'Noise - House of Worship', 'Posting Advertisement',
       'Noise - Vehicle', 'Animal Abuse', 'Vending', 'Traffic',
       'Drinking', 'Bike/Roller/Skate Chronic', 'Panhandling',
       'Noise - Park', 'Homeless Encampment', 'Urinating in Public',
       'Graffiti', 'Disorderly Youth', 'Illegal Fireworks',
       'Ferry Complaint', 'Agency Issues', 'Squeegee', 'Animal in a Park'],
      dtype=object)
```

[28]:
```
Customerservice['Descriptor'].unique()
```

[28]:
```
array(['Loud Music/Party', 'No Access', 'Commercial Overnight Parking',
       'Blocked Sidewalk', 'Posted Parking Sign Violation',
       'Blocked Hydrant', 'With License Plate', 'Partial Access',
       'Unauthorized Bus Layover', 'Double Parked Blocking Vehicle',
       'Double Parked Blocking Traffic', 'Vehicle', 'Loud Talking',
       'Banging/Pounding', 'Car/Truck Music', 'Tortured',
       'In Prohibited Area', 'Congestion/Gridlock', 'Neglected',
       'Car/Truck Horn', 'In Public', 'Other (complaint details)', nan,
       'No Shelter', 'Truck Route Violation', 'Unlicensed',
       'Overnight Commercial Storage', 'Engine Idling',
       'After Hours - Licensed Est', 'Detached Trailer',
       'Underage - Licensed Est', 'Chronic Stoplight Violation',
       'Loud Television', 'Chained', 'Building', 'In Car',
       'Police Report Requested', 'Chronic Speeding',
       'Playing in Unsuitable Place', 'Drag Racing',
       'Police Report Not Requested', 'Nuisance/Truant', 'Homeless Issue',
       'Language Access Complaint', 'Disruptive Passenger',
       'Animal Waste'], dtype=object)
```

[48]:
```
ComplaintCity = pd.DataFrame( {'count': Customerservice.groupby([ 'Complaint␣
↪Type', 'City']).size()}).reset_index()
```

[49]:
```
ComplaintCity
```

[49]:
```
     Complaint Type           City  count
0      Animal Abuse        ARVERNE     38
1      Animal Abuse        ASTORIA    125
2      Animal Abuse        BAYSIDE     37
3      Animal Abuse      BELLEROSE      7
4      Animal Abuse   BREEZY POINT      2
..              …            …    …
759         Vending  STATEN ISLAND     25
760         Vending      SUNNYSIDE     15
761         Vending     WHITESTONE      1
762         Vending      WOODHAVEN      6
763         Vending       WOODSIDE     15

[764 rows x 3 columns]
```

[51]:
```
Customerservice.groupby(['Borough','Complaint Type','Descriptor']).size()
```

[51]:
```
Borough      Complaint Type         Descriptor
BRONX        Animal Abuse           Chained                      132
                                    In Car                        36
                                    Neglected                    673
                                    No Shelter                    71
                                    Other (complaint details)    311
                                                                 …
Unspecified  Noise - Vehicle        Engine Idling                 11
             Posting Advertisement  Vehicle                        1
             Traffic                Truck Route Violation          1
             Vending                In Prohibited Area             2
                                    Unlicensed                     5
Length: 288, dtype: int64
```

[52]:
```
Customerservice.groupby(['Community Board','Complaint Type','Descriptor']).
↪size()
```

[52]:
```
Community Board           Complaint Type          Descriptor
0 Unspecified             Agency Issues           Language Access Complaint
6
                          Animal Abuse            In Car
1
                                                  Neglected
5
                                                  Other (complaint details)
2
                                                  Tortured
3
..
Unspecified MANHATTAN     Noise - Street/Sidewalk  Loud Talking
```

```
1
Unspecified QUEENS         Illegal Parking          Blocked Sidewalk
1
                           Noise - Street/Sidewalk  Loud Music/Party
1
Unspecified STATEN ISLAND  Noise - Commercial       Banging/Pounding
1
                           Noise - Street/Sidewalk  Loud Music/Party
1
Length: 2793, dtype: int64
```

[53]:
```
Customerservice.groupby(['School Name','Complaint Type','Descriptor']).size()
```

[53]:
```
School Name                       Complaint Type           Descriptor
Alley Pond Park - Nature Center   Animal in a Park         Animal Waste
1
Unspecified                       Agency Issues            Language Access
Complaint             6
                                  Animal Abuse             Chained
535
                                                           In Car
251
                                                           Neglected
3787
                                                           No Shelter
382
                                                           Other (complaint
details)          1969
                                                           Tortured
854
                                  Blocked Driveway         No Access
56976
                                                           Partial Access
20068
                                  Derelict Vehicle         With License Plate
17718
                                  Disorderly Youth         Nuisance/Truant
41
                                                           Playing in Unsuitable
Place         245
                                  Drinking                 After Hours -
Licensed Est           77
                                                           In Public
932
                                                           Underage - Licensed
Est             271
                                  Ferry Complaint          Disruptive Passenger
```

1		
1		Homeless Issue
	Graffiti	Police Report Not
Requested 23		
		Police Report
Requested 90		
	Illegal Parking	Blocked Hydrant
16081		
		Blocked Sidewalk
11121		
		Commercial Overnight
Parking 12189		
		Detached Trailer
464		
		Double Parked
Blocking Traffic 5731		
		Double Parked
Blocking Vehicle 4211		
		Overnight Commercial
Storage 1757		
		Posted Parking Sign
Violation 22440		
		Unauthorized Bus
Layover 1367		
	Noise - Commercial	Banging/Pounding
3917		
		Car/Truck Horn
991		
		Car/Truck Music
899		
		Loud Music/Party
25787		
		Loud Talking
3891		
		Loud Television
92		
	Noise - House of Worship	Banging/Pounding
248		
		Loud Music/Party
504		
		Loud Talking
178		
		Loud Television
1		
	Noise - Park	Loud Music/Party
3109		

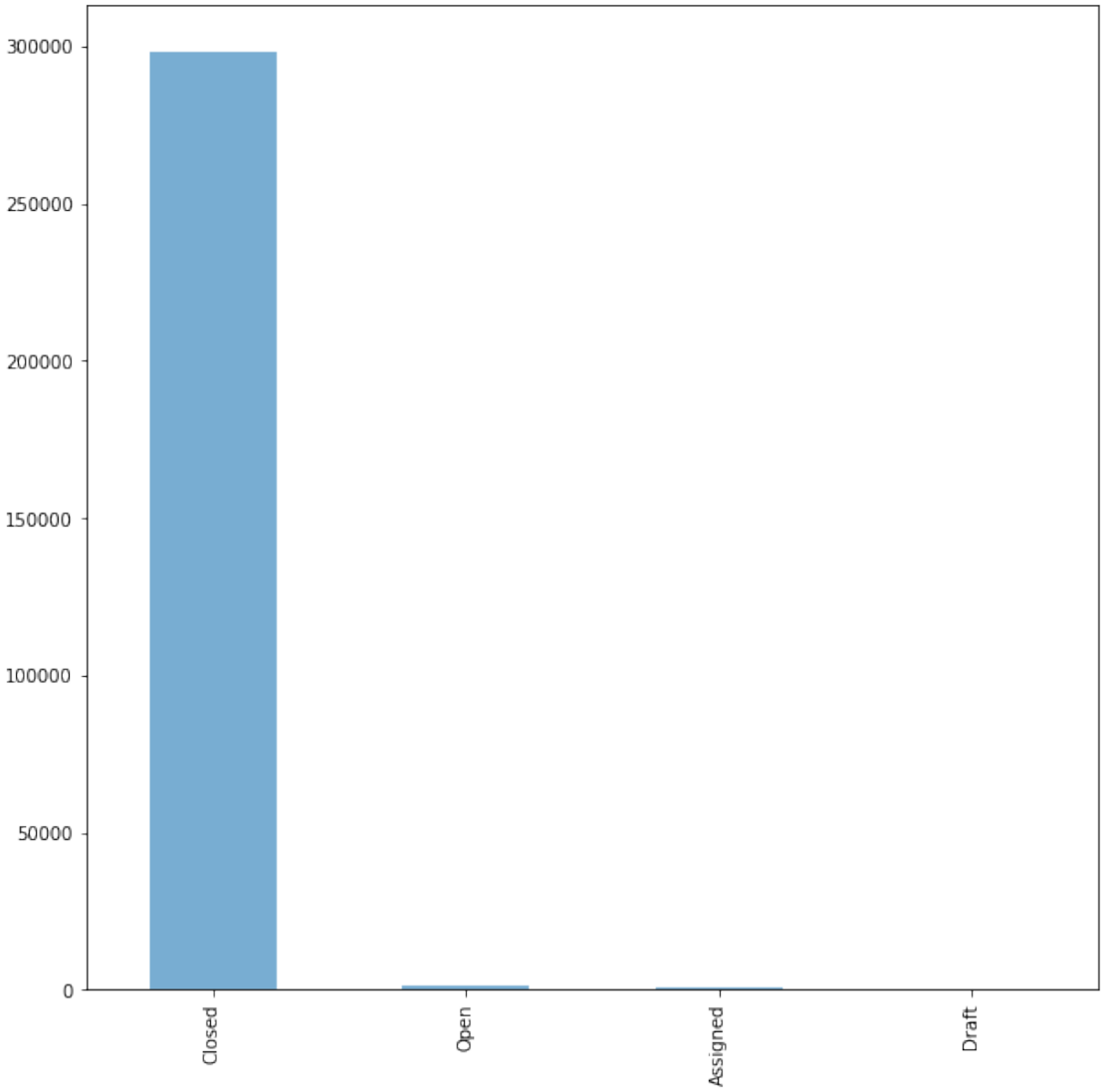
```
                                                  Loud Talking
933
                              Noise - Street/Sidewalk   Loud Music/Party
32030
                                                  Loud Talking
16582
                              Noise - Vehicle           Car/Truck Horn
2520
                                                  Car/Truck Music
10374
                                                  Engine Idling
4189
                              Posting Advertisement     Building
60
                                                  Vehicle
590
                              Traffic                   Chronic Speeding
268
                                                  Chronic Stoplight
Violation          280
                                                  Congestion/Gridlock
2761
                                                  Drag Racing
175
                                                  Truck Route Violation
1014
                              Vending                   In Prohibited Area
2025
                                                  Unlicensed
1777
dtype: int64
```

[54]:
```
import datetime
```

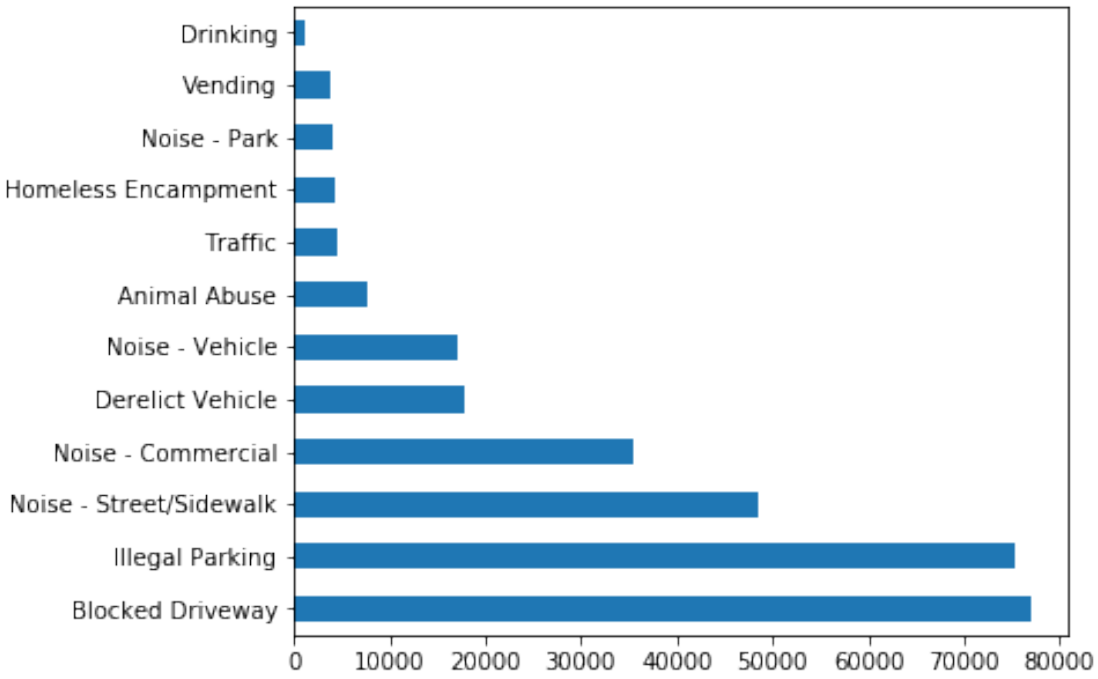
[61]:
```
df = pd.read_csv ('311_Service_Requests_from_2010_to_Present.csv' ,␣
↪parse_dates= ['Closed Date' , 'Created Date'])
```

[62]:
```
df["Request_Closing_Time"] = df["Closed Date"] - df["Created Date"]
```

[91]:
```
df['Status'].value_counts().plot(kind='bar',alpha=0.6,figsize=(10,10))
plt.show()
```



[76]:
```
#Bar graph of 12 Complaints
Customerservice['Complaint Type'].value_counts().head(12).
 ↪plot(kind='barh',figsize=(6,5));
```

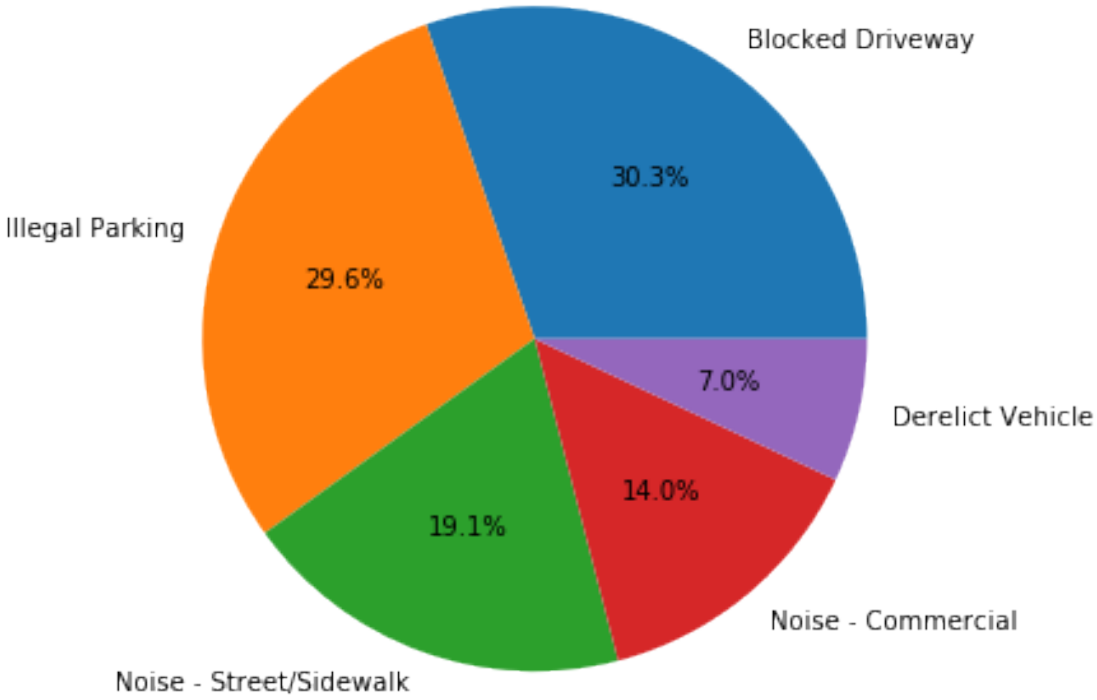


[79]:
```
majorcomplints=Customerservice.dropna(subset=["Complaint Type"])
majorcomplints=Customerservice.groupby("Complaint Type")

sortedComplaintType = majorcomplints.size().sort_values(ascending = False)
sortedComplaintType = sortedComplaintType.to_frame('count').reset_index()

sortedComplaintType
sortedComplaintType.head(12)
```

[79]:

	Complaint Type	count
0	Blocked Driveway	77044
1	Illegal Parking	75361
2	Noise - Street/Sidewalk	48612
3	Noise - Commercial	35577
4	Derelict Vehicle	17718
5	Noise - Vehicle	17083
6	Animal Abuse	7778
7	Traffic	4498
8	Homeless Encampment	4416
9	Noise - Park	4042
10	Vending	3802
11	Drinking	1280

[83]:
```
sortedComplaintType = sortedComplaintType.head()
plt.figure(figsize=(6,6))
plt.pie(sortedComplaintType['count'],labels=sortedComplaintType["Complaint␣
↪Type"], autopct="%1.1f%%")
plt.show()
```



[84]:
```
#Group dataset by complaint type to display plot against city
groupedby_complainttype = df.groupby('Complaint Type')
```

[85]:
```
group_data = groupedby_complainttype.get_group('Blocked Driveway')
group_data.shape
```

[85]: (77044, 54)

[86]:
```
group_data = groupedby_complainttype.get_group('Illegal Parking')
group_data.shape
```

[86]: (75361, 54)

[87]:
```
group_data = groupedby_complainttype.get_group('Illegal Parking')
group_data.size
```

[87]:
```
4069494
```

[88]:
```
group_data = groupedby_complainttype.get_group('Blocked Driveway')
group_data.size
```

[88]:
```
4160376
```

[90]:
```
#NaN values in entire dataset
df.isnull().sum()
```

[90]:
```
Unique Key                          0
Created Date                        0
Closed Date                      2164
Agency                              0
Agency Name                         0
Complaint Type                      0
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
Status                              0
Due Date                            3
Resolution Description              0
Resolution Action Updated Date   2187
Community Board                     0
Borough                             0
X Coordinate (State Plane)       3540
Y Coordinate (State Plane)       3540
Park Facility Name                  0
Park Borough                        0
School Name                         0
School Number                       0
School Region                       1
School Code                         1
School Phone Number                 0
```

```
School Address                          0
School City                             0
School State                            0
School Zip                              1
School Not Found                        0
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
Request_Closing_Time                 2164
dtype: int64
```

[92]:
```
#fix blank values in City column
df['City'].dropna(inplace=True)
```

[96]:
```
#Shape after dropping nan values
df['City'].size
```

[96]:
```
298084
```

[97]:
```
#Shape after dropping nan values
df['City'].shape
```

[97]:
```
(298084,)
```

[99]:
```
#count of null values in grouped city column data
group_data['City'].isnull().sum()
```

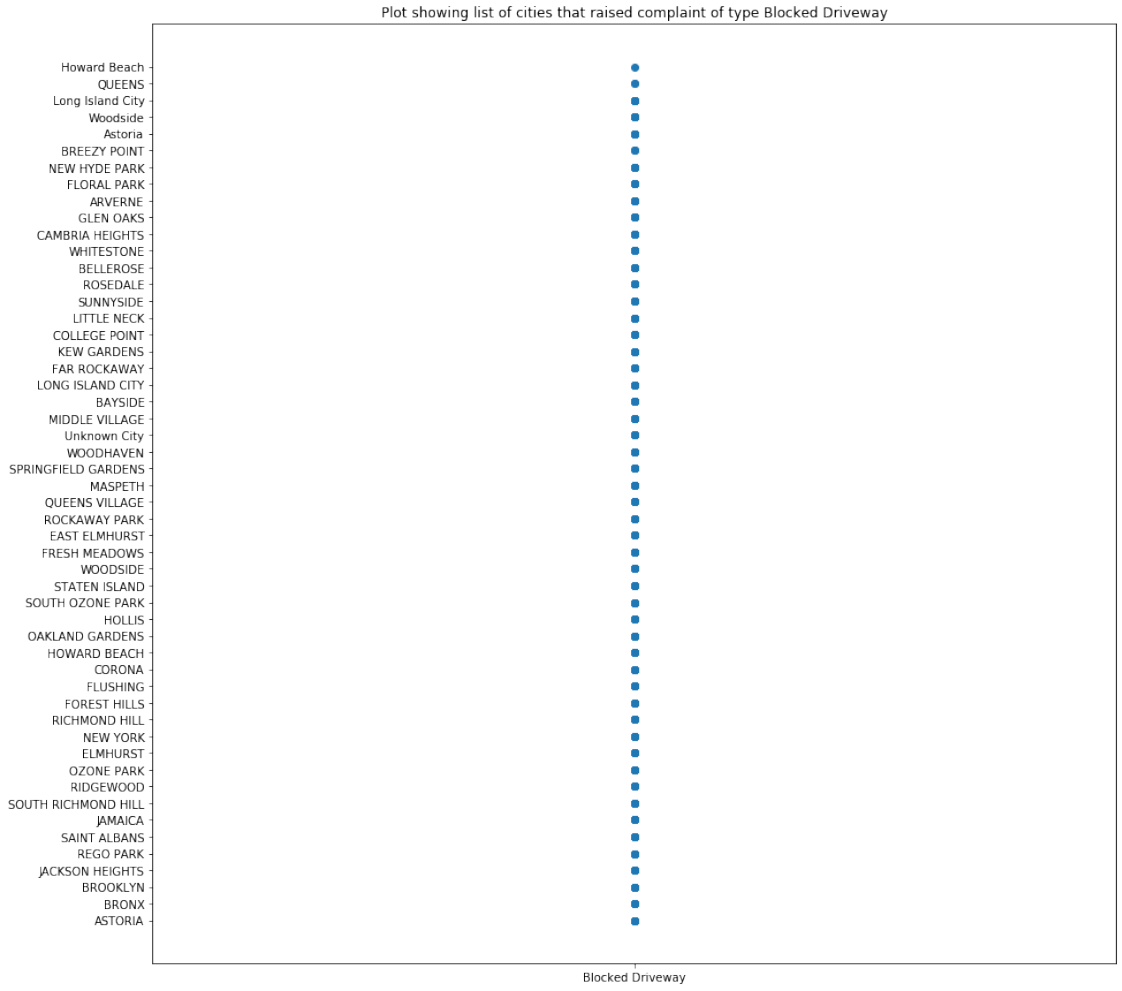
[99]:
```
283
```

[105]:
```
#fix those NAN with "unknown city" value instead
group_data['City'].fillna('Unknown City', inplace =False)
```

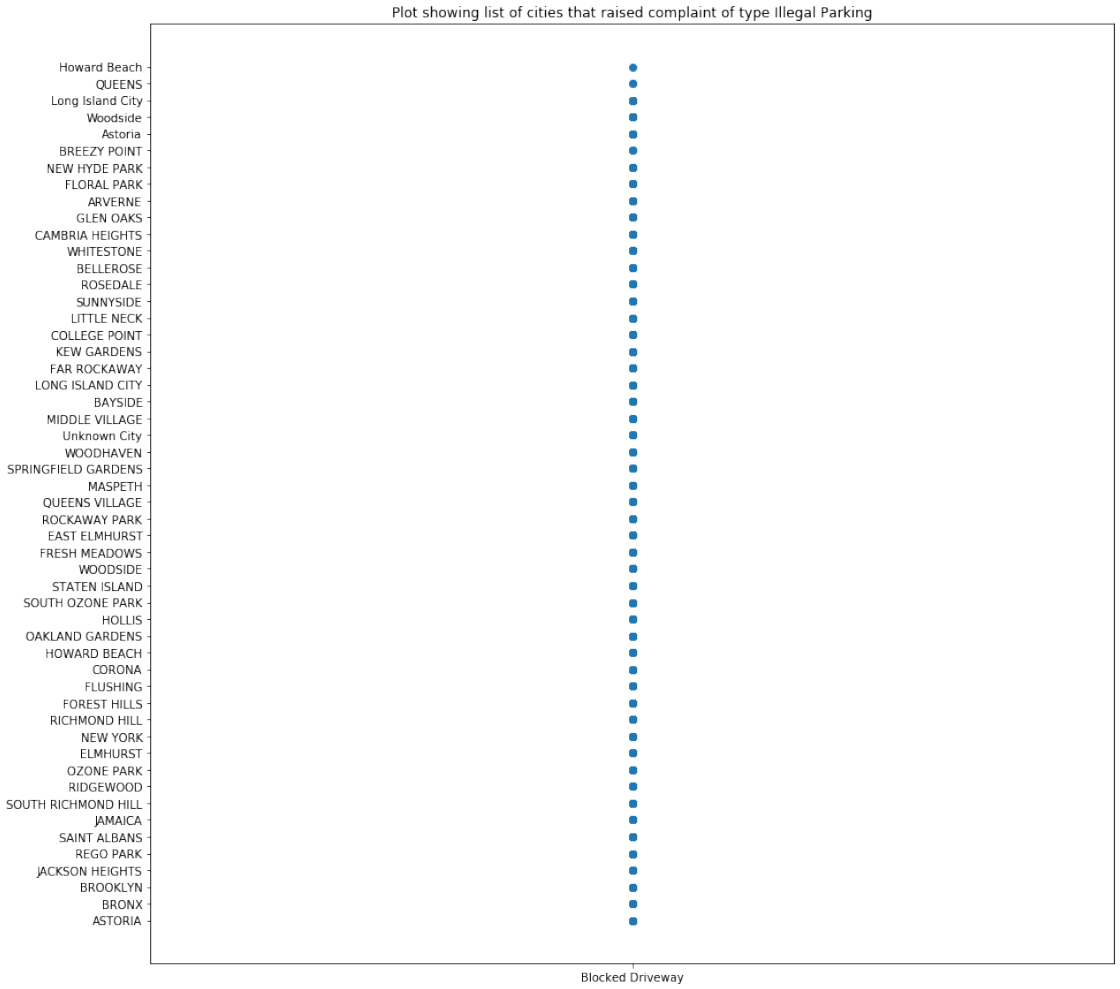
[105]:
```
1                 ASTORIA
2                   BRONX
7                   BRONX
9                BROOKLYN
```

```
10        JACKSON HEIGHTS
                ...
300665            BROOKLYN
300672            BROOKLYN
300679           RIDGEWOOD
300687           RIDGEWOOD
300694       RICHMOND HILL
Name: City, Length: 77044, dtype: object
```

[109]:
```
#Scatter plot displaying all the cities that raised complaint of type 'Blocked
 ↪Driveway'
plt.figure(figsize=(15, 15))
plt.scatter(group_data['Complaint Type'],group_data['City'])
plt.title('Plot showing list of cities that raised complaint of type Blocked
 ↪Driveway')
plt.show()
```



[110]:
```
#Scatter plot displaying all the cities that raised complaint of type 'illegal␣
 ↪Parking'
plt.figure(figsize=(15, 15))
plt.scatter(group_data['Complaint Type'],group_data['City'])
plt.title('Plot showing list of cities that raised complaint of type Illegal␣
 ↪Parking')
plt.show()
```



[ ]: