customer-service-requests-analysis

Oct 11, 2023

1 Customer Service Requests Analysis

[1]:
```
#import library
import pandas as pd
import numpy as np
import matplotlib as mpl
from matplotlib import pyplot as plt
%matplotlib inline
plt.style.use(['fivethirtyeight'])
mpl.rcParams['lines.linewidth'] = 4
import warnings
warnings.filterwarnings("ignore")
import scipy.stats as stats
import statsmodels.api as sm
from statsmodels.formula.api import ols
```

1.0.1 Importing Dataset : NY 311 Service-Requests

[2]:
```
#Import the data
df = pd.read_csv('311_Service_Requests_from_2010_to_Present.
 ↪csv',low_memory=False, header=0,
    sep=',', parse_dates=['Created Date', 'Closed Date', 'Resolution Action␣
 ↪Updated Date'],index_col='Unique Key')
```

Converting 'Created Date' and Closed Date' to datetime datatype Removing null and junk data

[3]:
```
def prepareData(df):
    df['Resolution_Time'] = (df['Closed Date'] - df['Created Date']).dt.
 ↪total_seconds() ####This Days divide by 3600
    df_clean=df[df['Resolution_Time'].notnull()]
    df_perfect = df_clean[df_clean['Closed Date'] >= df_clean['Created Date']]
    df_perfect['Day of Week'] = df_perfect['Created Date'].dt.dayofweek
    df_perfect['Day of Month'] = df_perfect['Created Date'].dt.day
    df_perfect['Month'] = df_perfect['Created Date'].dt.month
    df_perfect['Year'] = df_perfect['Created Date'].dt.year
```

```
    df_perfect=df_perfect[df_perfect.Borough!='Unspecified']
    return df_perfect
```

[4]:
```
#Looking for shape
df_perfect = prepareData(df)
df_perfect.shape
```

[4]:
```
(361542, 57)
```

[5]:
```
df_perfect.head()
```

[5]:

Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	Street Name	…	Ferry Direction	Ferry Terminal Name	Latitude	Longitude
32310363	2015-12-31 23:59:45	2016-01-01 00:55:15	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk	10034.0	71 VERMILYEA AVENUE	VERMILYEA AVENUE	…	NaN	NaN	40.865682	-73.923501
32309934	2015-12-31 23:59:44	2016-01-01 01:26:57	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	11105.0	27-07 23 AVENUE	23 AVENUE	…	NaN	NaN	40.775945	-73.915094
32309159	2015-12-31 23:59:29	2016-01-01 04:51:03	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	10458.0	2897 VALENTINE AVENUE	VALENTINE AVENUE	…	NaN			
32305098	2015-12-31 23:57:46	2016-01-01 07:43:13	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Sidewalk	10461.0	2940 BAISLEY AVENUE	BAISLEY AVENUE	…	NaN			
32306529	2015-12-31 23:56:58	2016-01-01 03:24:42	NYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street/Sidewalk	11373.0	87-14 57 ROAD	57 ROAD	…	NaN			

```
32309159                    NaN  40.870325 -73.888525
32305098                    NaN  40.835994 -73.828379
32306529                    NaN  40.733060 -73.874170

                                                Location Resolution_Time  \
Unique Key
32310363     (40.86568153633767, -73.92350095571744)          3330.0
32309934    (40.775945312321085, -73.91509393898605)          5233.0
32309159    (40.870324522111424, -73.88852464418646)         17494.0
32305098     (40.83599404683083, -73.82837939584206)         27927.0
32306529    (40.733059618956815, -73.87416975810375)         12464.0

           Day of Week Day of Month Month  Year
Unique Key
32310363             3           31    12  2015
32309934             3           31    12  2015
32309159             3           31    12  2015
32305098             3           31    12  2015
32306529             3           31    12  2015

[5 rows x 57 columns]
```

[6]:
```
df_perfect.tail()
```

[6]:
```
                 Created Date         Closed Date Agency  \
Unique Key
29609918   2015-01-01 00:04:44 2015-01-01 10:22:31   NYPD
29608392   2015-01-01 00:04:28 2015-01-01 02:25:02   NYPD
29607589   2015-01-01 00:01:30 2015-01-01 00:20:33   NYPD
29610889   2015-01-01 00:01:29 2015-01-01 02:42:22   NYPD
29611816   2015-01-01 00:00:50 2015-01-01 02:47:50   NYPD

                              Agency Name           Complaint Type  \
Unique Key
29609918    New York City Police Department          Illegal Parking
29608392    New York City Police Department          Noise - Vehicle
29607589    New York City Police Department  Noise - Street/Sidewalk
29610889    New York City Police Department         Blocked Driveway
29611816    New York City Police Department         Blocked Driveway

                  Descriptor    Location Type  Incident Zip  \
Unique Key
29609918     Blocked Hydrant  Street/Sidewalk       11421.0
29608392      Car/Truck Horn  Street/Sidewalk       10468.0
29607589    Loud Music/Party  Street/Sidewalk       10031.0
29610889           No Access  Street/Sidewalk       10466.0
29611816           No Access  Street/Sidewalk       11420.0
```

```
                Incident Address      Street Name  … Ferry Direction  \
Unique Key                                         …
29609918           84-25 85 ROAD          85 ROAD  …              NaN
29608392    2555 SEDGWICK AVENUE  SEDGWICK AVENUE  …              NaN
29607589     508 WEST 139 STREET  WEST 139 STREET  …              NaN
29610889     931 EAST 226 STREET  EAST 226 STREET  …              NaN
29611816       123-19 135 STREET       135 STREET  …              NaN

           Ferry Terminal Name   Latitude  Longitude  \
Unique Key
29609918                   NaN  40.695145 -73.860949
29608392                   NaN  40.867830 -73.907178
29607589                   NaN  40.821647 -73.950873
29610889                   NaN  40.886361 -73.853290
29611816                   NaN  40.674212 -73.803585

                                          Location Resolution_Time  \
Unique Key
29609918     (40.69514470265117, -73.86094888534394)         37067.0
29608392     (40.86782963689454, -73.90717786644662)          8434.0
29607589    (40.821646626438095, -73.95087342885292)          1143.0
29610889     (40.88636077906953, -73.85329048666742)          9653.0
29611816    (40.674211762243935, -73.80358548685278)         10020.0

           Day of Week Day of Month Month  Year
Unique Key
29609918             3            1     1  2015
29608392             3            1     1  2015
29607589             3            1     1  2015
29610889             3            1     1  2015
29611816             3            1     1  2015

[5 rows x 57 columns]
```

[7]:
```
df.isna().sum()
```

[7]:
```
Created Date                    0
Closed Date                  2381
Agency                          0
Agency Name                     0
Complaint Type                  0
Descriptor                   6501
Location Type                 133
Incident Zip                 2998
Incident Address            51699
Street Name                 51699
```

```
Cross Street 1                    57188
Cross Street 2                    57805
Intersection Street 1            313438
Intersection Street 2            314046
Address Type                       3252
City                               2997
Landmark                         364183
Facility Type                      2389
Status                                0
Due Date                              3
Resolution Description                0
Resolution Action Updated Date     2402
Community Board                       0
Borough                               0
X Coordinate (State Plane)         4030
Y Coordinate (State Plane)         4030
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     364558
Vehicle Type                     364558
Taxi Company Borough             364558
Taxi Pick Up Location            364558
Bridge Highway Name              364261
Bridge Highway Direction         364261
Road Ramp                        364296
Bridge Highway Segment           364296
Garage Lot Name                  364558
Ferry Direction                  364557
Ferry Terminal Name              364556
Latitude                           4030
Longitude                          4030
Location                           4030
Resolution_Time                    2381
dtype: int64
```

[8]:
```
df_perfect.info()
```

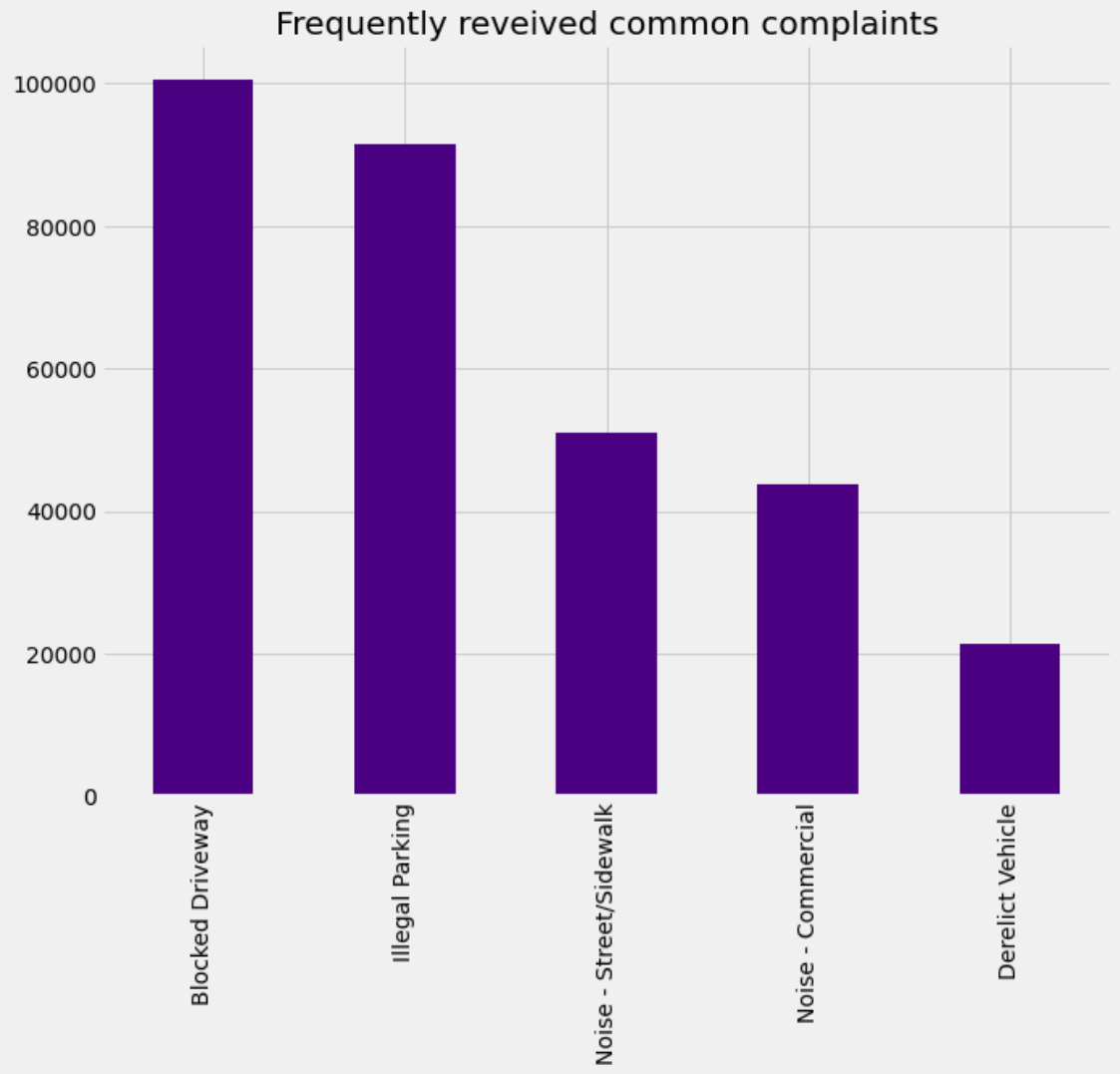
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 361542 entries, 32310363 to 29611816
Data columns (total 57 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   Created Date                    361542 non-null  datetime64[ns]
 1   Closed Date                     361542 non-null  datetime64[ns]
 2   Agency                          361542 non-null  object
 3   Agency Name                     361542 non-null  object
 4   Complaint Type                  361542 non-null  object
 5   Descriptor                      355049 non-null  object
 6   Location Type                   361420 non-null  object
 7   Incident Zip                    361501 non-null  float64
 8   Incident Address                309865 non-null  object
 9   Street Name                     309865 non-null  object
 10  Cross Street 1                  306722 non-null  object
 11  Cross Street 2                  306713 non-null  object
 12  Intersection Street 1           50504 non-null   object
 13  Intersection Street 2           50504 non-null   object
 14  Address Type                    361247 non-null  object
 15  City                            361501 non-null  object
 16  Landmark                        375 non-null     object
 17  Facility Type                   361533 non-null  object
 18  Status                          361542 non-null  object
 19  Due Date                        361541 non-null  object
 20  Resolution Description          361542 non-null  object
 21  Resolution Action Updated Date  361503 non-null  datetime64[ns]
 22  Community Board                 361542 non-null  object
 23  Borough                         361542 non-null  object
 24  X Coordinate (State Plane)      360469 non-null  float64
 25  Y Coordinate (State Plane)      360469 non-null  float64
 26  Park Facility Name              361542 non-null  object
 27  Park Borough                    361542 non-null  object
 28  School Name                     361542 non-null  object
 29  School Number                   361542 non-null  object
 30  School Region                   361542 non-null  object
 31  School Code                     361542 non-null  object
 32  School Phone Number             361542 non-null  object
 33  School Address                  361542 non-null  object
 34  School City                     361542 non-null  object
 35  School State                    361542 non-null  object
 36  School Zip                      361542 non-null  object
 37  School Not Found                361542 non-null  object
 38  School or Citywide Complaint    0 non-null       float64
 39  Vehicle Type                    0 non-null       float64
 40  Taxi Company Borough            0 non-null       float64
 41  Taxi Pick Up Location           0 non-null       float64
 42  Bridge Highway Name             297 non-null     object
```

```
 43  Bridge Highway Direction        297 non-null     object
 44  Road Ramp                       262 non-null     object
 45  Bridge Highway Segment          262 non-null     object
 46  Garage Lot Name                 0 non-null       float64
 47  Ferry Direction                 0 non-null       object
 48  Ferry Terminal Name             0 non-null       object
 49  Latitude                        360469 non-null  float64
 50  Longitude                       360469 non-null  float64
 51  Location                        360469 non-null  object
 52  Resolution_Time                 361542 non-null  float64
 53  Day of Week                     361542 non-null  int64
 54  Day of Month                    361542 non-null  int64
 55  Month                           361542 non-null  int64
 56  Year                            361542 non-null  int64
dtypes: datetime64[ns](3), float64(11), int64(4), object(39)
memory usage: 160.0+ MB
```

Investigating on Insights and Patterns

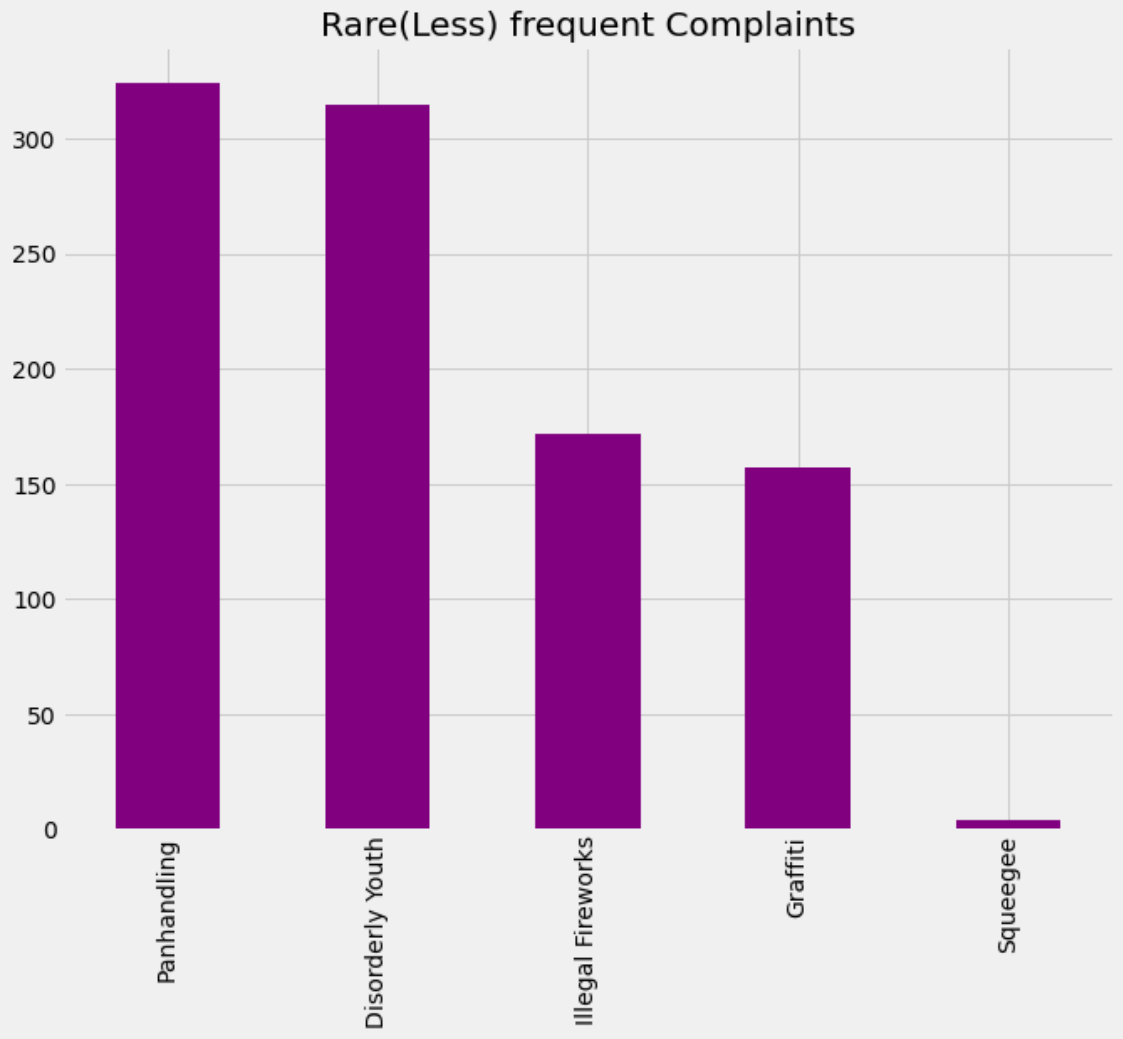
[9]:
```
(df_perfect['Complaint Type'].value_counts()).head().plot(kind='bar',
                    figsize=(10,8),color='indigo', title = 'Frequently reveived␣
 ↪common complaints')
```

[9]:
```
<AxesSubplot:title={'center':'Frequently reveived common complaints'}>
```

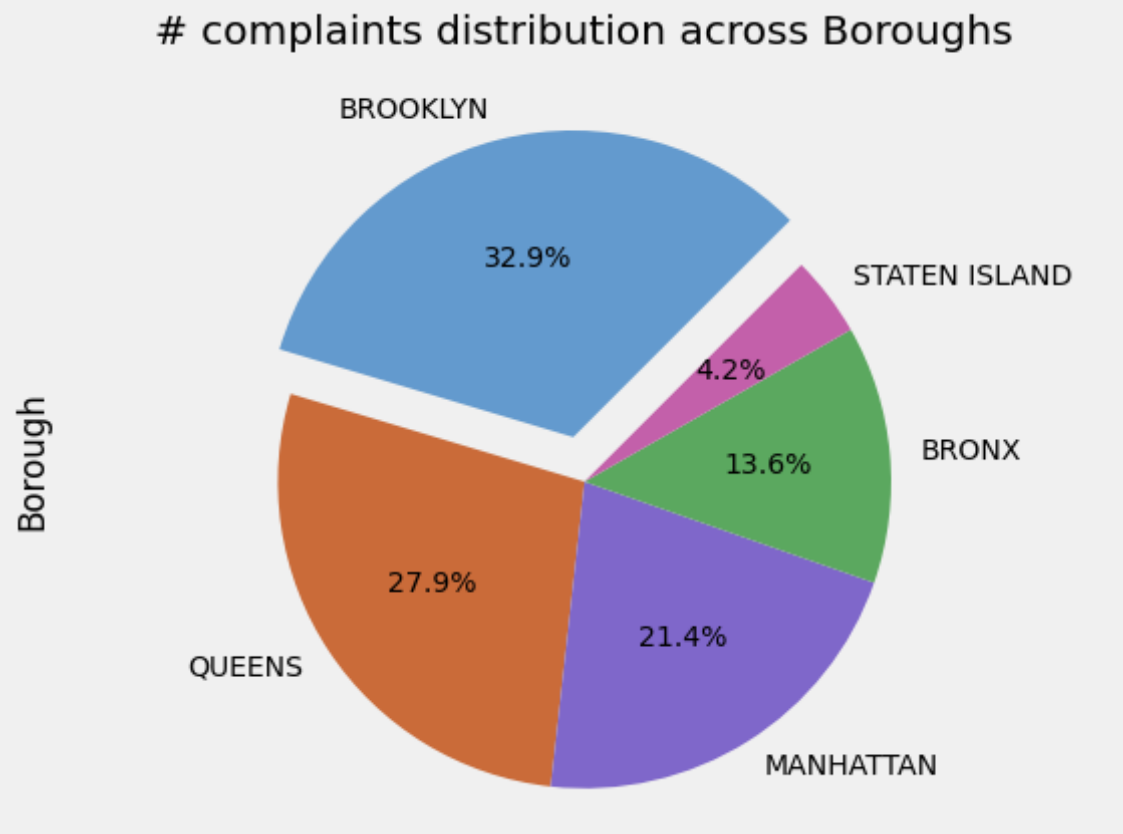


[10]:
```
#Rare(Less) frequent Complaints
(df_perfect['Complaint Type'].value_counts()).tail().plot(kind='bar',
                    figsize=(10,8),color='purple', title = 'Rare(Less) frequent␣
 ↪Complaints')
```

[10]: <AxesSubplot:title={'center':'Rare(Less) frequent Complaints'}>



[11]:
```
# Distribution of complaints across Boroughs
colors = ['#639ace','#ca6b39','#7f67ca','#5ba85f','#c360aa','#a7993f','#cc566a']
df_perfect['Borough'].value_counts().plot(kind='pie',autopct='%1.1f%%',
                        explode = (0.15, 0, 0, 0, 0), startangle=45,␣
 ↪shadow=False, colors = colors,
                        figsize = (8,6))
#plt.legend(title='BOROUGH', loc='upper right', bbox_to_anchor=(1.5,1))
plt.axis('equal')
plt.title('# complaints distribution across Boroughs\n')
plt.tight_layout()
plt.show()
```



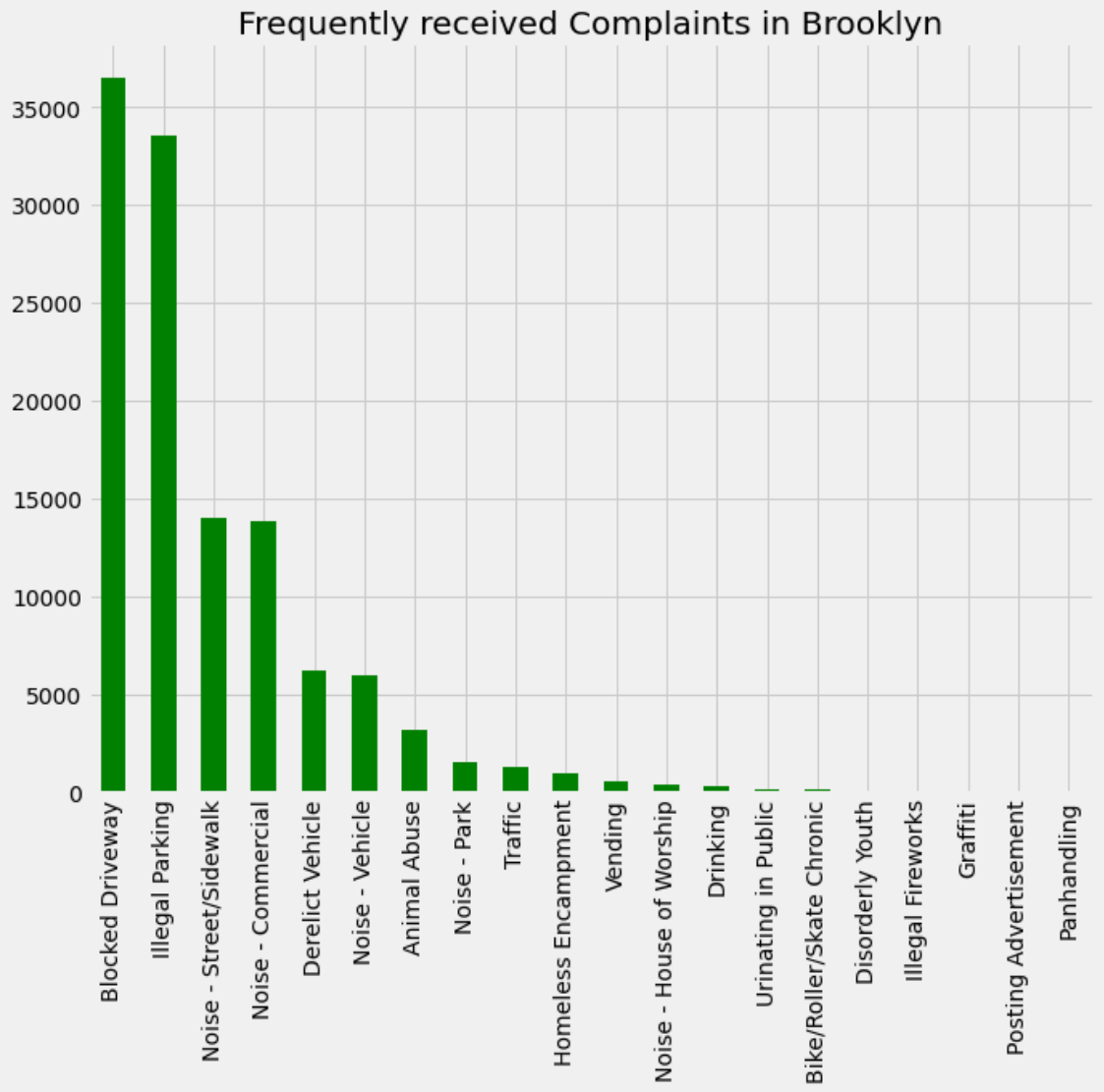
[12]:
```
#Analysis for Brooklyn borough which has highest number of complains
df_Brooklyn = df_perfect[df_perfect['Borough']=='BROOKLYN']
```

[13]:
```
df_Brooklyn.shape
```

[13]:
```
(118851, 57)
```

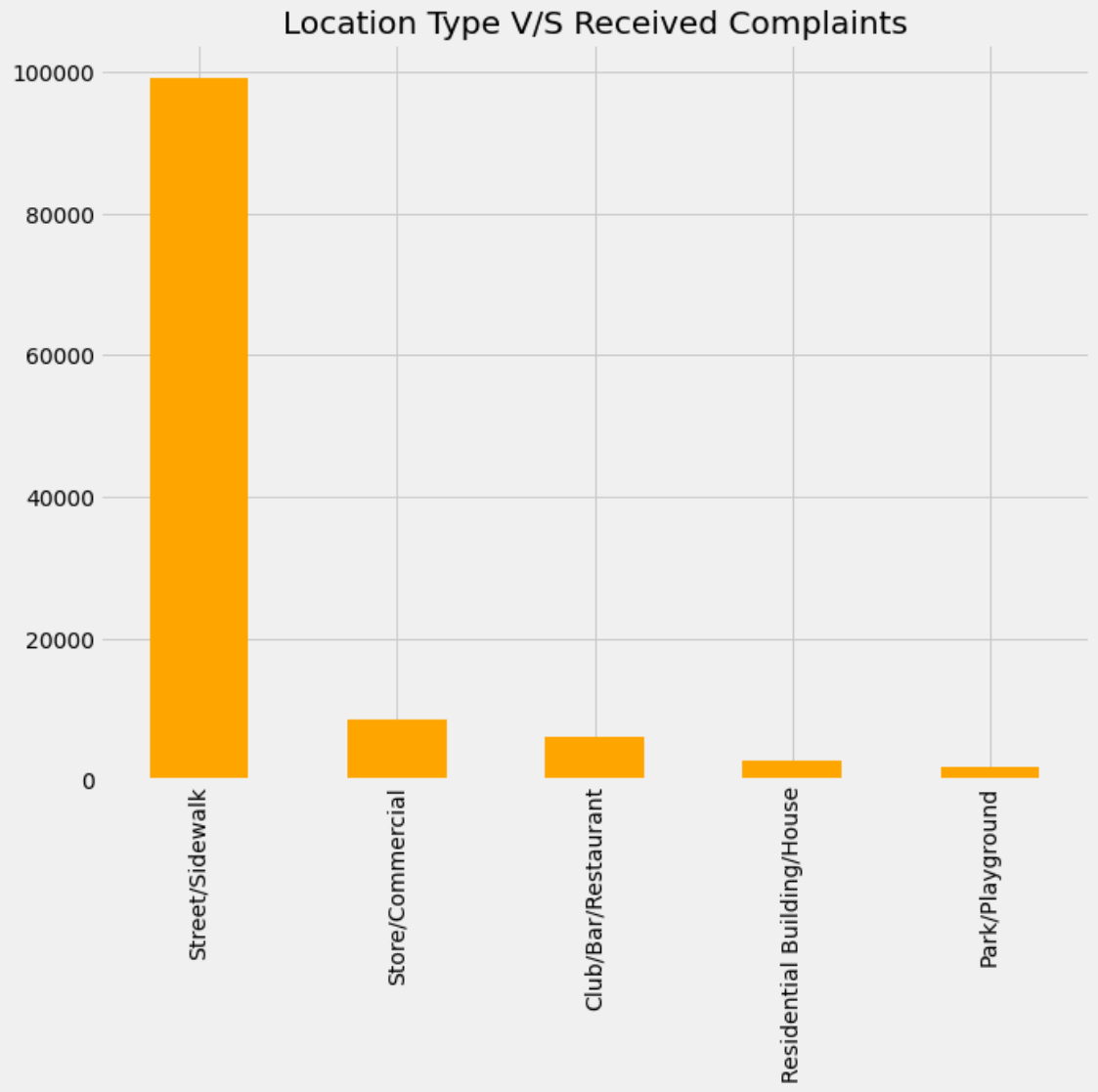
[14]:
```
(df_Brooklyn['Complaint Type'].value_counts()).head(25).
 ↪plot(kind='bar',color='green',
                                                    figsize=(10,8),title =␣
 ↪'Frequently received Complaints in Brooklyn')
```

[14]:
```
<AxesSubplot:title={'center':'Frequently received Complaints in Brooklyn'}>
```



[15]:
```
#Plotting graph : location type vs complaints received
(df_Brooklyn['Location Type'].value_counts()).head().plot(kind='bar',␣
 ↪figsize=(10,8),color='Orange',title = 'Location Type V/S Received␣
 ↪Complaints')
```

[15]: <AxesSubplot:title={'center':'Location Type V/S Received Complaints'}>



[16]:
```
#Anlyzing the Most Frequent complaint in Brooklyn
df_perfect[df_perfect['Complaint Type'] == 'Blocked Driveway']['Descriptor'].
 ↪value_counts()
```

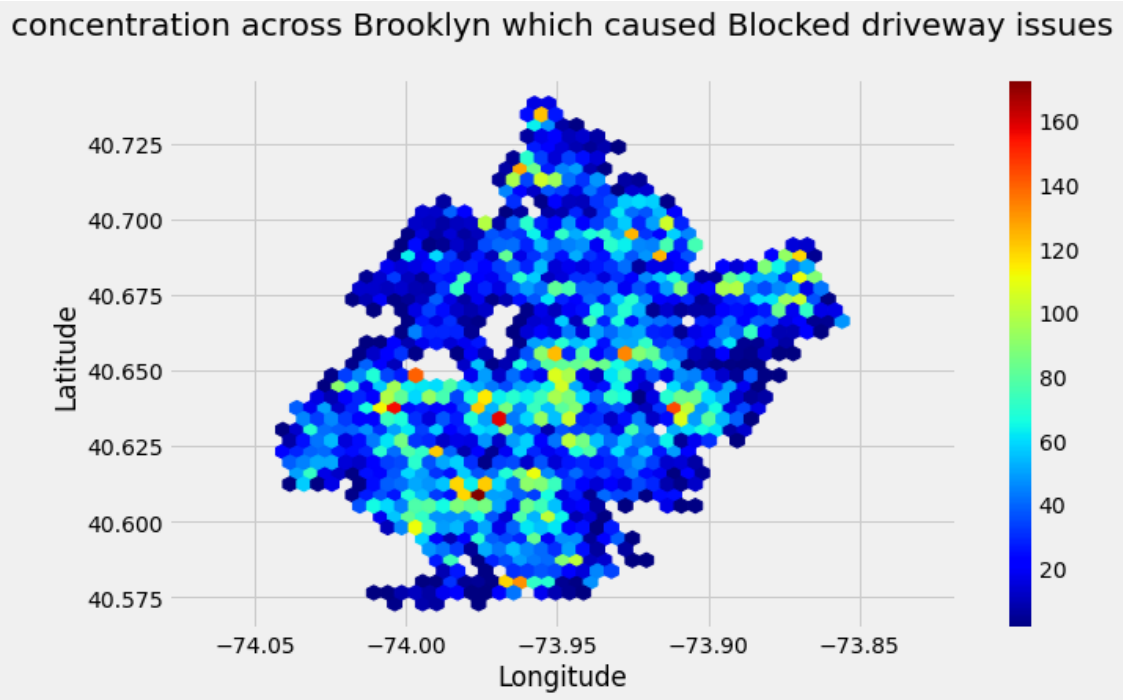
[16]:
```
No Access         75657
Partial Access    24882
Name: Descriptor, dtype: int64
```

[17]:
```
df_Brook_blocked = df_Brooklyn[df_Brooklyn['Complaint Type'] == 'Blocked␣
 ↪Driveway']
```

[18]:
```
df_Brook_blocked.plot(
```

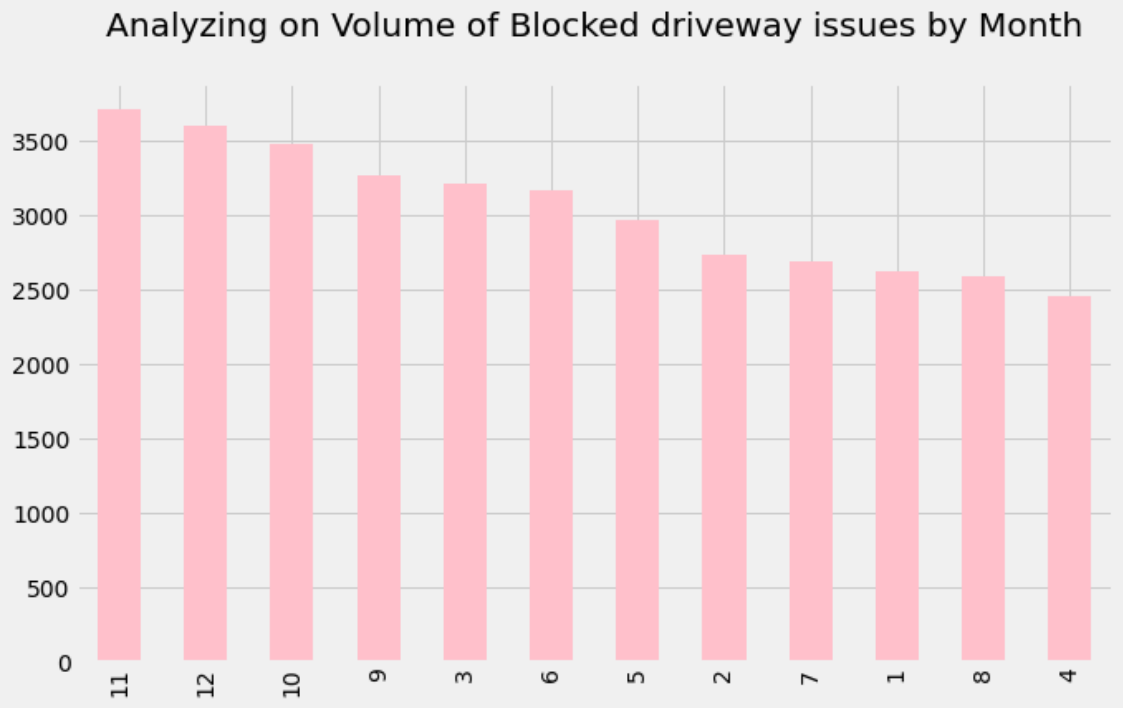
```
    kind='hexbin', x='Longitude', y='Latitude', gridsize=40,title =␣
↪'concentration across Brooklyn which caused Blocked driveway issues\n',
    colormap='jet', mincnt=1, figsize=(10,6)).axis('equal')
```

[18]: (-74.04994269716303, -73.84653003063751, 40.56459183795005, 40.746496092182724)



[19]:
```
df_Brook_blocked['Month'].value_counts().plot(kind =␣
↪'bar',color='pink',figsize=(10,6), title = 'Analyzing on Volume of Blocked␣
↪driveway issues by Month\n')
```

[19]: <AxesSubplot:title={'center':'Analyzing on Volume of Blocked driveway issues by Month\n'}>



1.0.2 'Request_Closing_Time' in Seconds by grouping them for different location and order by complaint type

[20]:
```
df_avg_res_time_city = df_perfect.groupby(['City','Complaint Type']).
  ↪Resolution_Time.mean()
#df_perfect.sort_values('Complaint Type').groupby('City')
#
df_avg_res_time_city.head(25)
```

[20]:
```
City      Complaint Type
ARVERNE   Animal Abuse                 8399.195652
          Blocked Driveway             8318.840000
          Derelict Vehicle            11394.000000
          Disorderly Youth            12928.500000
          Drinking                      859.000000
          Graffiti                     5508.000000
          Homeless Encampment          6541.250000
          Illegal Parking              8406.080645
          Noise - Commercial           8234.000000
          Noise - House of Worship     6653.428571
          Noise - Park                 4638.000000
          Noise - Street/Sidewalk      7173.206897
          Noise - Vehicle              5673.600000
          Panhandling                  3673.000000
```

```
          Traffic                        4014.000000
          Urinating in Public            2491.000000
          Vending                        1735.000000
ASTORIA   Animal Abuse                   17206.882353
          Bike/Roller/Skate Chronic       6718.750000
          Blocked Driveway               16515.934517
          Derelict Vehicle               32446.598592
          Disorderly Youth                9713.800000
          Drinking                       15518.279070
          Graffiti                       50742.250000
          Homeless Encampment            17704.312500
Name: Resolution_Time, dtype: float64
```

1.0.3 Average response time(in seconds) across complaint types

[21]:
```
df_avg_res_time = df_perfect.groupby('Complaint Type').Resolution_Time.mean().
 ↪sort_values(ascending=True)
df_avg_res_time.head(21)
```

[21]:
```
Complaint Type
Posting Advertisement          7286.256259
Illegal Fireworks             10113.482558
Noise - Commercial            11040.936795
Noise - House of Worship      11391.087079
Noise - Park                  12194.009799
Noise - Street/Sidewalk       12206.102697
Traffic                       12309.120092
Disorderly Youth              12363.749206
Noise - Vehicle               12557.269801
Urinating in Public           12959.293292
Bike/Roller/Skate Chronic     13089.997886
Drinking                      13800.002855
Vending                       14365.309034
Squeegee                      14560.250000
Homeless Encampment           15451.384505
Illegal Parking               15590.340164
Panhandling                   15765.632716
Blocked Driveway              16218.697660
Animal Abuse                  18033.102859
Graffiti                      23276.343949
Derelict Vehicle              25279.866465
Name: Resolution_Time, dtype: float64
```

From the above data rejecting null hypothesis, since the average response time across complaint type are not equal. Null Hypothesis (Average response time across complaint type are equal). Alternate Hypothesis (Average response time across complaint

type are equal)

Following complains have resolution times which are very close. Disorderly Youth 12810.902098 Noise - Vehicle 12918.914430 One group can be formed for these complaints and one way Anova for this can be performed

[22]:
```
df_dis_youth = df_perfect[df_perfect['Complaint Type']=='Disorderly Youth']
df_dis_youth = df_dis_youth.loc[:,['Resolution_Time']]
df_dis_youth.head()
```

[22]:
```
            Resolution_Time
Unique Key
32274507              713.0
32244468             4605.0
32225263             2345.0
32227341            19415.0
32191432             6849.0
```

[23]:
```
df_noise_veh = df_perfect[df_perfect['Complaint Type']=='Noise - Vehicle']
df_noise_veh = df_noise_veh.loc[:,['Resolution_Time']]
df_noise_veh.head()
```

[23]:
```
            Resolution_Time
Unique Key
32307159            22994.0
32308722             7254.0
32308107            11319.0
32308108            10937.0
32306622             2615.0
```

[24]:
```
df_type_res = df_perfect.loc[:, ['Complaint Type','Resolution_Time']]
df_type_res.head()
df_type_res.columns
```

[24]:
```
Index(['Complaint Type', 'Resolution_Time'], dtype='object')
```

[25]:
```
fvalue, pvalue = stats.f_oneway(df_dis_youth, df_noise_veh)
pvalue
```

[25]:
```
array([0.83131389])
```

1.0.4 One Way Anova for Posting Advertisement and Derelict Vehicle

[26]:
```
df_post_ad = df_perfect[df_perfect['Complaint Type']=='Posting Advertisement']
df_post_ad = df_post_ad.loc[:,['Resolution_Time']]
df_post_ad.head()
```

[26]:
```
             Resolution_Time
Unique Key
32306752              7643.0
32307464              7798.0
32308949              7893.0
32307323              8047.0
32306034              8144.0
```

[27]:
```
df_der_veh = df_perfect[df_perfect['Complaint Type']=='Derelict Vehicle']
df_der_veh = df_der_veh.loc[:,['Resolution_Time']]
df_der_veh.head()
```

[27]:
```
             Resolution_Time
Unique Key
32309424             37785.0
32306497             14224.0
32305124              4913.0
32308002             14879.0
32305798              2712.0
```

[28]:
```
# stats f_oneway functions takes the groups as input and returns F and P-value
fvalue, pvalue = stats.f_oneway(df_post_ad, df_der_veh)
pvalue
```

[28]:
```
array([1.52053985e-35])
```

Anova table for complain type and resolution time

[30]:
```
df_perfect['Complaint_Type']=df_perfect['Complaint Type']
df_type_res = df_perfect.loc[:, ['Complaint_Type','Resolution_Time']] ␣
 ↪#Complaint Type
# Ordinary Least Squares (OLS) model
model = ols('Resolution_Time ~ Complaint_Type', data=df_type_res).fit()
anova_table = sm.stats.anova_lm(model, typ=2)
anova_table
```

[30]:
```
                      sum_sq        df           F  PR(>F)
Complaint_Type  3.892068e+12      20.0  454.111878     0.0
Residual        1.549249e+14  361521.0         NaN     NaN
```

Crosstab and Chi Square test for Location and Complaint type

[31]:
```
df_city_type = pd.crosstab(df_perfect.City , df_perfect.Complaint_Type)
```

[32]:
```
# chi-squared test with similar proportions
from scipy.stats import chi2_contingency
from scipy.stats import chi2
# contingency table
```

```
table = df_city_type
#print(table)
stat, p, dof, expected = chi2_contingency(table)
print('dof=%d' % dof)
print(expected)
# interpret test-statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('probability=%.3f, critical=%.3f, stat=%.3f' % (prob, critical, stat))
if abs(stat) >= critical:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
# interpret p-value
alpha = 1.0 - prob
print('significance=%.3f, p=%.3f' % (alpha, p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
```

```
dof=1040
[[7.54357803e+00 3.38884263e-01 7.20311756e+01 … 3.72127878e+00
  4.59249075e-01 2.99765699e+00]
 [2.32744139e+02 1.04556917e+01 2.22239816e+03 … 1.14813663e+02
  1.41693412e+01 9.24875561e+01]
 [2.63588344e+01 1.18413227e+00 2.51691946e+02 … 1.30029239e+01
  1.60471202e+00 1.04744385e+01]
 …
 [9.03481816e+01 4.05876056e+00 8.62705431e+02 … 4.45691381e+01
  5.50034993e+00 3.59024401e+01]
 [1.26901040e+02 5.70084453e+00 1.21173680e+03 … 6.26008171e+01
  7.72566881e+00 5.04277665e+01]
 [4.83488012e+00 2.17199952e-01 4.61666994e+01 … 2.38506671e+00
  2.94344967e-01 1.92127823e+00]]
probability=0.950, critical=1116.137, stat=131573.935
Dependent (reject H0)
significance=0.050, p=0.000
Dependent (reject H0)
```

[ ]: