Project_2

September 23, 2022

[36]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
import seaborn as sns
from scipy import stats
```

0.1 1. Import a 311 NYC service request.

[37]:
```
df=pd.read_csv("311_Service_Requests_from_2010_to_Present.csv")
```

[38]:
```
df
```

[38]:
```
        Unique Key              Created Date              Closed Date Agency  \
0         32310363  12/31/2015 11:59:45 PM  01/01/2016 12:55:15 AM   NYPD
1         32309934  12/31/2015 11:59:44 PM  01/01/2016 01:26:57 AM   NYPD
2         32309159  12/31/2015 11:59:29 PM  01/01/2016 04:51:03 AM   NYPD
3         32305098  12/31/2015 11:57:46 PM  01/01/2016 07:43:13 AM   NYPD
4         32306529  12/31/2015 11:56:58 PM  01/01/2016 03:24:42 AM   NYPD
…              …                       …                       …    …
364553    29609918  01/01/2015 12:04:44 AM  01/01/2015 10:22:31 AM   NYPD
364554    29608392  01/01/2015 12:04:28 AM  01/01/2015 02:25:02 AM   NYPD
364555    29607589  01/01/2015 12:01:30 AM  01/01/2015 12:20:33 AM   NYPD
364556    29610889  01/01/2015 12:01:29 AM  01/01/2015 02:42:22 AM   NYPD
364557    29611816  01/01/2015 12:00:50 AM  01/01/2015 02:47:50 AM   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
…                                   …                       …
364553  New York City Police Department          Illegal Parking
364554  New York City Police Department          Noise - Vehicle
```

```
364555  New York City Police Department  Noise - Street/Sidewalk
364556  New York City Police Department         Blocked Driveway
364557  New York City Police Department         Blocked Driveway

                        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
…                                …              …           …
364553               Blocked Hydrant  Street/Sidewalk       11421.0
364554                Car/Truck Horn  Street/Sidewalk       10468.0
364555             Loud Music/Party  Street/Sidewalk       10031.0
364556                   No Access  Street/Sidewalk       10466.0
364557                   No Access  Street/Sidewalk       11420.0

             Incident Address  … Bridge Highway Name  \
0         71 VERMILYEA AVENUE  …                 NaN
1             27-07 23 AVENUE  …                 NaN
2       2897 VALENTINE AVENUE  …                 NaN
3         2940 BAISLEY AVENUE  …                 NaN
4               87-14 57 ROAD  …                 NaN
…                         …  …                 …
364553          84-25 85 ROAD  …                 NaN
364554   2555 SEDGWICK AVENUE  …                 NaN
364555    508 WEST 139 STREET  …                 NaN
364556    931 EAST 226 STREET  …                 NaN
364557      123-19 135 STREET  …                 NaN

       Bridge Highway Direction Road Ramp Bridge Highway Segment  \
0                           NaN       NaN                    NaN
1                           NaN       NaN                    NaN
2                           NaN       NaN                    NaN
3                           NaN       NaN                    NaN
4                           NaN       NaN                    NaN
…                           …         …                      …
364553                      NaN       NaN                    NaN
364554                      NaN       NaN                    NaN
364555                      NaN       NaN                    NaN
364556                      NaN       NaN                    NaN
364557                      NaN       NaN                    NaN

       Garage Lot Name Ferry Direction Ferry Terminal Name   Latitude  \
0                  NaN             NaN                 NaN  40.865682
1                  NaN             NaN                 NaN  40.775945
2                  NaN             NaN                 NaN  40.870325
```

```
3                    NaN             NaN                  NaN  40.835994
4                    NaN             NaN                  NaN  40.733060
…                    …               …                    …      …
364553               NaN             NaN                  NaN  40.695145
364554               NaN             NaN                  NaN  40.867830
364555               NaN             NaN                  NaN  40.821647
364556               NaN             NaN                  NaN  40.886361
364557               NaN             NaN                  NaN  40.674212

        Longitude                                    Location
0      -73.923501   (40.86568153633767, -73.92350095571744)
1      -73.915094  (40.775945312321085, -73.91509393898605)
2      -73.888525  (40.870324522111424, -73.88852464418646)
3      -73.828379   (40.83599404683083, -73.82837939584206)
4      -73.874170  (40.733059618956815, -73.87416975810375)
…           …                                          …
364553 -73.860949   (40.69514470265117, -73.86094888534394)
364554 -73.907178   (40.86782963689454, -73.90717786644662)
364555 -73.950873  (40.821646626438095, -73.95087342885292)
364556 -73.853290   (40.88636077906953, -73.85329048666742)
364557 -73.803585  (40.674211762243935, -73.80358548685278)

[364558 rows x 53 columns]
```

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

[39]:
```
# Converting the data into datetime format
df["Created Date"]=pd.to_datetime(df["Created Date"])
df["Closed Date"]=pd.to_datetime(df["Closed Date"])
```

[40]:
```
# Print the data type of Create Date & Closed Date
create_date=df["Created Date"]
print("Data type of Create Date : " , np.dtype(create_date))
closed_date=df["Closed Date"]
print("Data type of Closed Date : " , np.dtype(closed_date))
```

```
Data type of Create Date :  datetime64[ns]
Data type of Closed Date :  datetime64[ns]
```

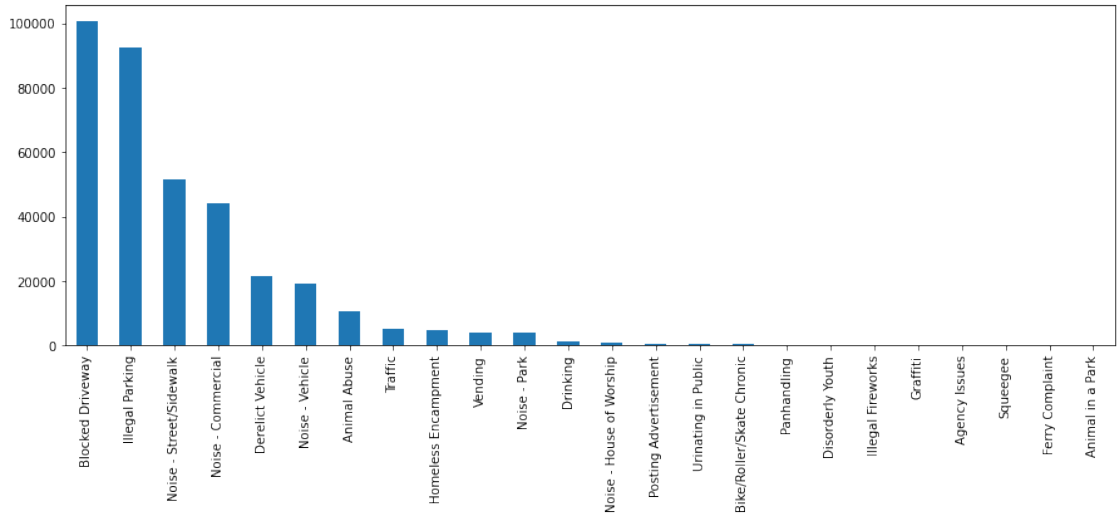
[41]:
```
#Creating the new column that the time elapsed between request creation and␣
 ↪request closing
df["Request_Closing_Time"]=(df["Closed Date"]-df["Created Date"])
```

```
print(df["Request_Closing_Time"])
```

```
0         0 days 00:55:30
1         0 days 01:27:13
2         0 days 04:51:34
3         0 days 07:45:27
4         0 days 03:27:44
                 ...
364553    0 days 10:17:47
364554    0 days 02:20:34
364555    0 days 00:19:03
364556    0 days 02:40:53
364557    0 days 02:47:00
Name: Request_Closing_Time, Length: 364558, dtype: timedelta64[ns]
```

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

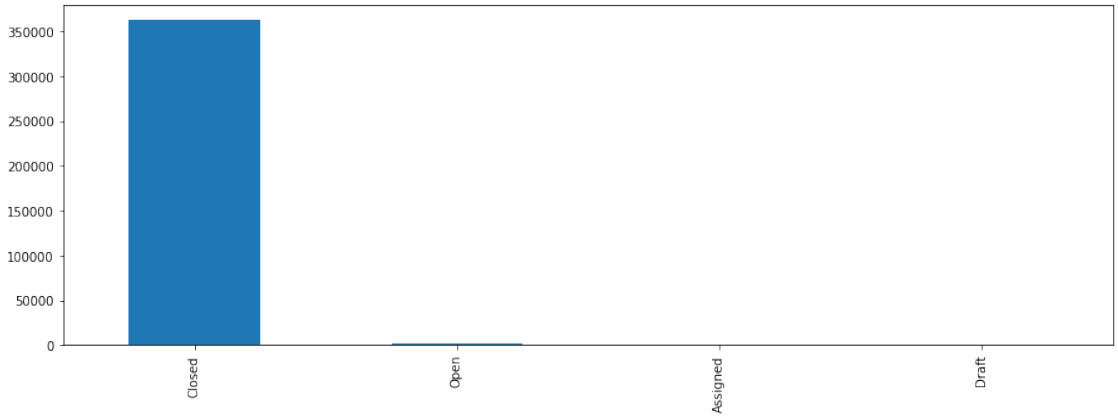
[42]:
```
# (A) Count plot to understand the type of the complaint raised
df['Complaint Type'].value_counts().plot(kind='bar',alpha=1,figsize=(15,5))
plt.show()
```



0.4 Maximum complaint requests belongs Blocked driveway.

[43]:
```
# (B) Count plot to know the status of the requests
Status_Count=df['Status'].value_counts()
Status_Count_Percentage=Status_Count/Status_Count.sum()*100
print(Status_Count_Percentage)
df['Status'].value_counts().plot(kind='bar',alpha=1,figsize=(15,5))
plt.show()
```

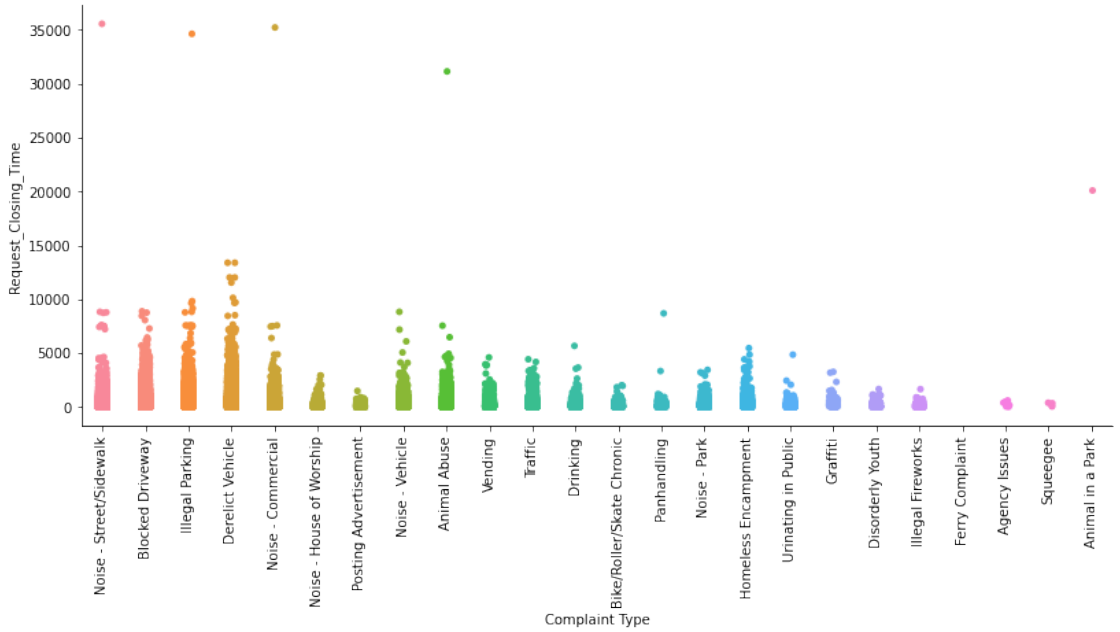
```
Closed      99.329599
Open         0.449860
Assigned     0.219992
Draft        0.000549
Name: Status, dtype: float64
```



0.5 As the above result 99.33% of the cases are closed state.

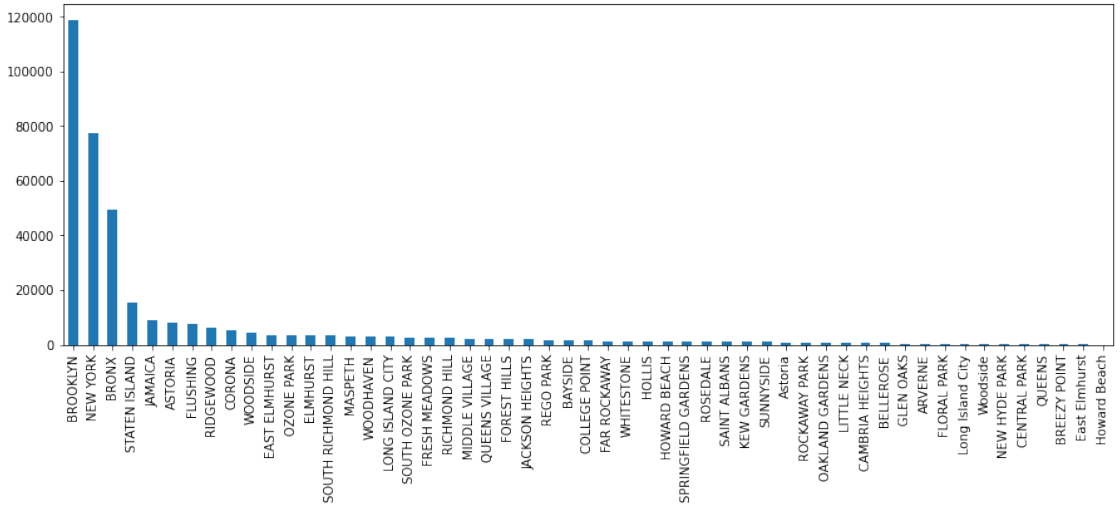
[44]:
```
# (C) Categorical Scatter Plot to understand which type of complaints are␣
 ↪taking more time to get resolved

Request_Closing_Time=[]
for x in (df["Closed Date"]-df["Created Date"]):
    close=x.total_seconds()/60
    Request_Closing_Time.append(close)

df["Request_Closing_Time"]=Request_Closing_Time


Complaint_Closing_Time=sns.catplot(x='Complaint Type',␣
 ↪y="Request_Closing_Time",data=df)
Complaint_Closing_Time.fig.set_figwidth(15)
```

```
Complaint_Closing_Time.fig.set_figheight(5)
plt.xticks(rotation=90)
plt.show()
```



0.6 As the above result, We found "Derelict Vehicle" Complaint Type take more time to resolve.

[45]:
```
# (D) Count plot to understand no. of the complaint raised in differtnt City.
df['City'].value_counts().plot(kind='bar',alpha=1,figsize=(15,5))
plt.show()
```



0.7 Brooklyn city have maximum no of complaint and least no, of complaint in Howard Beach City.

[ ]:

0.8 4. Order the complaint types based on the average 'Request_Closing_Time', grouping them for different locations.

[46]:
```
new_df= df[["Complaint Type","Location Type","Request_Closing_Time"]].copy()
new_df
pivot = new_df.groupby (['Complaint Type','Location Type']).agg({'mean'})
pivot
g = pivot['Request_Closing_Time'].groupby('Complaint Type', group_keys=False)

sort_pivot = g.apply(lambda x: x.sort_values(['mean'],ascending=True))
sort_pivot
```

[46]:

		mean
Complaint Type	Location Type	
Animal Abuse	Subway Station	183.967949
	Park/Playground	223.850117
	Store/Commercial	262.884451
	Residential Building	267.260350
	Commercial	270.649846
…		…
Urinating in Public	Club/Bar/Restaurant	426.132667
Vending	Park/Playground	210.227987
	Store/Commercial	238.316396
	Street/Sidewalk	239.744137
	Residential Building/House	251.681811

[71 rows x 1 columns]

0.9 5. Perform a statistical test for the following:

Please note: For the below statements you need to state the Null and Alternate and then provide a statistical test to accept or reject the Null Hypothesis along with the corresponding 'p-value'.

Whether the average response time across complaint types is similar or not (overall)

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

[47]:
```
# Whether the average response time across complaint types is similar or not␣
↪(overall)
```

0.10 H0: there is no significant different in mean of Request_Closing_Time for different Complaint

0.11 H1:there is signficant different in mean of Request_Closing_Time for different Complaint

[48]:
```
H_df=pd.DataFrame()
H_df["Request_Closing_Time"]=df["Request_Closing_Time"]
H_df["Complaint Type"]=df["Complaint Type"]
H_df.dropna(inplace=True)
H_df.head()
```

[48]:

	Request_Closing_Time	Complaint Type
0	55.500000	Noise - Street/Sidewalk
1	87.216667	Blocked Driveway
2	291.566667	Blocked Driveway
3	465.450000	Illegal Parking
4	207.733333	Illegal Parking

[49]:
```
from scipy import stats
from scipy.stats import chi2_contingency

data_crosstab = pd.crosstab( H_df["Request_Closing_Time"],H_df["Complaint␣
↪Type"])
stat, p, dof, expected = chi2_contingency(data_crosstab)

alpha = 0.001
if p <= alpha:
    print('Independent (H0 holds true)')
else:
    print('Dependent (reject H0)')
```

```
Independent (H0 holds true)
```

0.12 From above result H1:there is signficant different in mean of Request_Closing_Time for different Complaint

[50]:
```
# Are the type of complaint or service requested and location related?
```

0.13 H0:Complaint Type and Location Type are independent

0.14 H1:Complaint Type and Location Type are related

[51]:
```
L_df=pd.DataFrame()
L_df["Location Type"]=df["Location Type"]
L_df["Complaint Type"]=df["Complaint Type"]
L_df.dropna(inplace=True)
L_df.head()
```

[51]:

	Location Type	Complaint Type
0	Street/Sidewalk	Noise - Street/Sidewalk
1	Street/Sidewalk	Blocked Driveway
2	Street/Sidewalk	Blocked Driveway
3	Street/Sidewalk	Illegal Parking
4	Street/Sidewalk	Illegal Parking

[52]:
```
data_crosstab = pd.crosstab( L_df["Location Type"],L_df["Complaint Type"])
stat, p, dof, expected = chi2_contingency(data_crosstab)

alpha = 0.05
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (H0 holds true)')
```

```
Dependent (reject H0)
```

0.15 From Above Result H1:Complaint Type and Location Type are relate