Project Sugandh

August 31, 2022

[1]:

```
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
from scipy import stats
from scipy.stats import ttest_1samp
```

1 Task 1) Import a 311 NYC service request.

[2]:

```
data=pd.read_csv('311_Service_Requests_from_2010_to_Present.csv')
data.head()
```

[2]:

```
   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]
```

[3]:

```
data.shape
```

[3]:

```
(1667, 53)
```

[4]:

```
data.dtypes
```

[4]:

```
Unique Key                          int64
Created Date                       object
Closed Date                        object
Agency                             object
Agency Name                        object
Complaint Type                     object
Descriptor                         object
Location Type                      object
Incident Zip                      float64
Incident Address                   object
Street Name                        object
```

```
Cross Street 1                     object
Cross Street 2                     object
Intersection Street 1              object
Intersection Street 2              object
Address Type                       object
City                               object
Landmark                          float64
Facility Type                      object
Status                             object
Due Date                           object
Resolution Description             object
Resolution Action Updated Date     object
Community Board                    object
Borough                            object
X Coordinate (State Plane)        float64
Y Coordinate (State Plane)        float64
Park Facility Name                 object
Park Borough                       object
School Name                        object
School Number                      object
School Region                      object
School Code                        object
School Phone Number                object
School Address                     object
School City                        object
School State                       object
School Zip                         object
School Not Found                   object
School or Citywide Complaint      float64
Vehicle Type                      float64
Taxi Company Borough              float64
Taxi Pick Up Location             float64
Bridge Highway Name               float64
Bridge Highway Direction          float64
Road Ramp                         float64
Bridge Highway Segment            float64
Garage Lot Name                   float64
Ferry Direction                   float64
Ferry Terminal Name               float64
Latitude                          float64
Longitude                         float64
Location                           object
dtype: object
```

2 Task 2) Read or convert the columns 'Created Date' and Closed Date' to datetime datatype

[5]:

```
data['Created Date']=pd.to_datetime(data['Created Date'])
data['Closed Date']=pd.to_datetime(data['Closed Date'])
data[['Created Date','Closed Date']].dtypes
```

[5]:

```
Created Date    datetime64[ns]
Closed Date     datetime64[ns]
dtype: object
```

3 Create a new column 'Request_Closing_Time' as the time elapsed between request creation and request closing

[6]:

```
data['Request_Closing_Time']=(data['Closed Date']-data['Created Date']).dt.
 ↪seconds/60/60
data['Request_Closing_Time'].head()
```

[6]:

```
0    0.920833
1    1.437778
2    4.858611
3    7.753889
4    3.450556
Name: Request_Closing_Time, dtype: float64
```

4 Task 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.

4.1 Conclusion 1

4.1.1 Five cities with maximum numbers of complaints

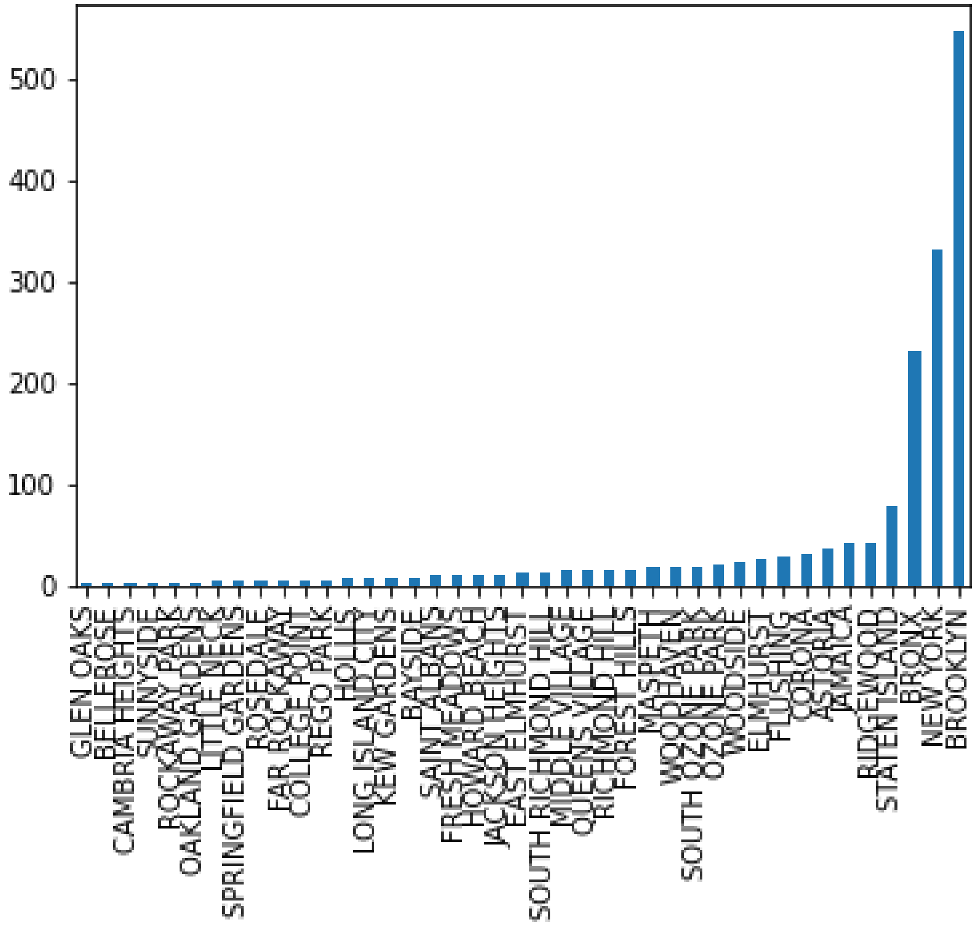
[8]:

```
data.City.value_counts().sort_values(ascending=True).plot.bar()
data.City.value_counts().sort_values(ascending=False).head()
```

[8]:

```
BROOKLYN         547
NEW YORK         331
BRONX            231
STATEN ISLAND     77
RIDGEWOOD         41
```

```
Name: City, dtype: int64
```



4.2 Conclusion 2

4.2.1 In each city maximum no. of complaints is of which type

[9]:
```
newframe1=pd.DataFrame(data.groupby('City')['Complaint Type'].value_counts())
newframe1.groupby('City').head(1)
```

[9]:
```
                                        Complaint Type
City                Complaint Type
ASTORIA             Blocked Driveway                25
BAYSIDE             Derelict Vehicle                 3
BELLEROSE           Derelict Vehicle                 1
BRONX               Blocked Driveway               116
BROOKLYN            Blocked Driveway               191
```

```
CAMBRIA HEIGHTS     Derelict Vehicle                 1
COLLEGE POINT       Blocked Driveway                 3
CORONA              Blocked Driveway                20
EAST ELMHURST       Blocked Driveway                 5
ELMHURST            Blocked Driveway                15
FAR ROCKAWAY        Blocked Driveway                 3
FLUSHING            Blocked Driveway                14
FOREST HILLS        Blocked Driveway                 7
FRESH MEADOWS       Illegal Parking                  4
GLEN OAKS           Illegal Parking                  1
HOLLIS              Blocked Driveway                 3
HOWARD BEACH        Derelict Vehicle                 4
JACKSON HEIGHTS     Blocked Driveway                 4
JAMAICA             Blocked Driveway                20
KEW GARDENS         Blocked Driveway                 4
LITTLE NECK         Blocked Driveway                 1
LONG ISLAND CITY    Blocked Driveway                 3
MASPETH             Blocked Driveway                 7
MIDDLE VILLAGE      Illegal Parking                  7
NEW YORK            Noise - Commercial              79
OAKLAND GARDENS     Blocked Driveway                 1
OZONE PARK          Blocked Driveway                11
QUEENS VILLAGE      Blocked Driveway                 7
REGO PARK           Blocked Driveway                 3
RICHMOND HILL       Blocked Driveway                11
RIDGEWOOD           Illegal Parking                 20
ROCKAWAY PARK       Blocked Driveway                 2
ROSEDALE            Illegal Parking                  3
SAINT ALBANS        Illegal Parking                  5
SOUTH OZONE PARK    Blocked Driveway                 6
SOUTH RICHMOND HILL Blocked Driveway                 9
SPRINGFIELD GARDENS Illegal Parking                  2
STATEN ISLAND       Illegal Parking                 30
SUNNYSIDE           Blocked Driveway                 2
WOODHAVEN           Blocked Driveway                12
WOODSIDE            Blocked Driveway                 8
```

4.3 Conclusion 3

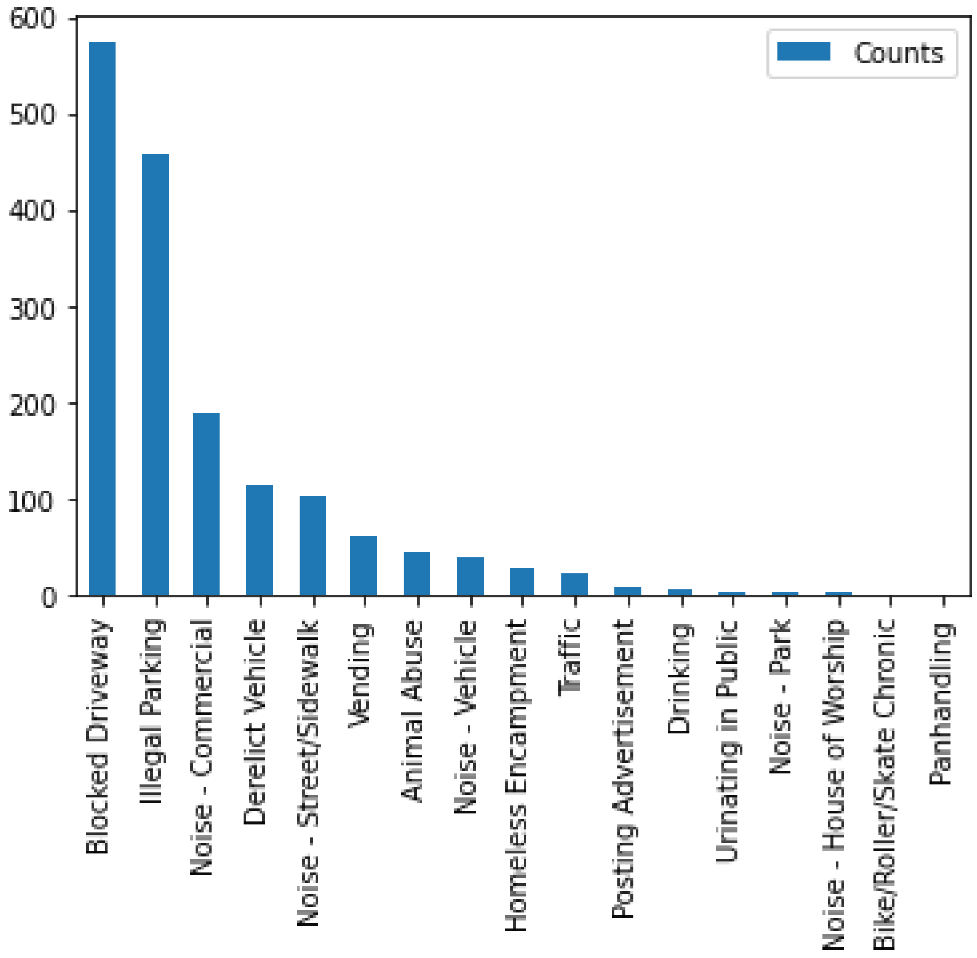
4.3.1 Measuring the frequency of the different complaint

[10]:

```
data['Complaint Type'].value_counts().to_frame().rename(columns={'Complaint␣
↪Type':'Counts'}).plot(kind='bar')
```

[10]:

```
<AxesSubplot:>
```



4.4 Conclusion 4

No. of unique Complaint types

[11]:

```
data['Complaint Type'].nunique()
```

[11]:

```
17
```

5 Task 4) Order the complaint types based on the average‘Request_Closing_Time’, grouping them for different location Type

[12]:

```
data.groupby(['Location Type','Complaint Type'])['Request_Closing_Time'].mean()
```

[12]:

```
Location Type                Complaint Type
Club/Bar/Restaurant          Drinking                     1.706333
                             Noise - Commercial           2.063981
House and Store              Animal Abuse                 1.633889
House of Worship             Noise - House of Worship     1.121111
Park/Playground              Homeless Encampment          3.516667
                             Noise - Park                 4.352685
Residential Building         Animal Abuse                 3.508958
Residential Building/House   Animal Abuse                 3.412000
                             Drinking                     4.989444
                             Homeless Encampment          6.017431
                             Urinating in Public          5.814444
Store/Commercial             Animal Abuse                 2.511944
                             Drinking                     0.776944
                             Homeless Encampment          4.522847
                             Noise - Commercial           2.882218
                             Vending                      3.608611
Street/Sidewalk              Animal Abuse                 2.962882
                             Bike/Roller/Skate Chronic    3.509444
                             Blocked Driveway             4.517120
                             Derelict Vehicle             4.527636
                             Homeless Encampment          3.420688
                             Illegal Parking              4.044109
                             Noise - Street/Sidewalk      2.346090
                             Noise - Vehicle              3.409079
                             Panhandling                  3.536389
                             Posting Advertisement        1.676500
                             Traffic                      2.324525
                             Urinating in Public          2.910833
                             Vending                      2.145129
Vacant Lot                   Derelict Vehicle             5.720139
Name: Request_Closing_Time, dtype: float64
```

6 Task 5) Perform a statistical test

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

[13]:

```
Hnull1='The average response time across complaint types is similar'
Halternate1='The average response time across complaint types is not similar'
```

[14]:

```
newframe3=data[data['Request_Closing_Time'].notnull()]
Sample=newframe3.sample(n=1000)
ttest,pvalue=ttest_1samp(Sample['Request_Closing_Time'],newframe3['Request_Closing_Time'].
 ↪mean())
```

[15]:

```
if(pvalue>0.05):
    print("Null Hypothesis is accepted,",Hnull1)
else:print("Null Hypothesis is Rejected",Halternate1)
```

```
Null Hypothesis is accepted, The average response time across complaint types is
similar
```

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

[16]:

```
Hnull2='The type of complaint or service requested and location are not related'
Halternate2='The type of complaint or service requested and location are␣
 ↪related'
```

[17]:

```
newframe4=pd.crosstab(newframe3['Complaint Type'],newframe3['Location Type'])
observedval=newframe4.values
squarex,pvalue,dof,expected=stats.chi2_contingency(newframe4)
```

[18]:

```
if(pvalue>0.05):print("Null Hypothesis accepted",(Hnull2))
else:(print("Null Hypothesis is rejected,",Halternate2))
```

```
Null Hypothesis is rejected, The type of complaint or service requested and
location are related
```

[ ]:

```

```