NYC_311

June 2, 2020

[1]:
```
pwd()
```

[1]:
```
'/home/labsuser/MAY-11'
```

[2]:
```
import numpy as np
import pandas as pd
```

[3]:
```
#Import required libraries
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

[4]:
```
# Question 1.) Import a 311 NYC service request.
```

[5]:
```
df_Nyc = pd.read_csv("311_Service_Requests_from_2010_to_Present.csv")
```

```
/usr/local/lib/python3.7/site-packages/IPython/core/interactiveshell.py:3063:
DtypeWarning: Columns (48,49) have mixed types.Specify dtype option on import or
set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

[6]:
```
# We are keeping one orig copy also with us
df_orig = pd.read_csv("311_Service_Requests_from_2010_to_Present.csv")
```

[7]:
```
df_Nyc.head()
```

[7]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type
0	32310363	12/31/2015 11:59:45 PM	01-01-16 0:55	NYPD	New York City Police Department	Noise - Street/Sidewalk
1	32309934	12/31/2015 11:59:44 PM	01-01-16 1:26	NYPD	New York City Police Department	Blocked Driveway
2	32309159	12/31/2015 11:59:29 PM	01-01-16 4:51	NYPD	New York City Police Department	Blocked Driveway
3	32305098	12/31/2015 11:57:46 PM	01-01-16 7:43	NYPD	New York City Police Department	Illegal Parking
4	32306529	12/31/2015 11:56:58 PM	01-01-16 3:24	NYPD	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]
```

[8]:
```
df_Nyc.shape
```

[8]:
```
(300698, 53)
```

[9]:
```
# See columns
df_Nyc.columns
```

[9]:
```
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')
```

[10]:
```
# First we should check which column has how many missing values
df_Nyc.isnull().sum()
```

[10]:
```
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
dtype: int64
```

[11]:
```
# As we seen Closed Date is important column and have many missing values
df_Nyc[df_Nyc['Closed Date'].isnull()]
```

[11]:
```
        Unique Key            Created Date Closed Date Agency  \
416       32305700  12/31/2015 02:16:04 PM         NaN   NYPD
611       32309308  12/31/2015 09:58:06 AM         NaN   NYPD
1648      32303348  12/30/2015 05:13:42 AM         NaN   NYPD
1816      32294519  12/29/2015 10:44:50 PM         NaN   NYPD
1965      32296487  12/29/2015 07:09:13 PM         NaN   NYPD
…              …                       …         …     …
300273    30287350  03/29/2015 02:40:19 PM         NaN   NYPD
300492    30284963  03/29/2015 08:50:15 AM         NaN   NYPD
300496    30285492  03/29/2015 08:44:13 AM         NaN   NYPD
300620    30282717  03/29/2015 01:55:35 AM         NaN   NYPD
300693    30281872  03/29/2015 12:33:41 AM         NaN   NYPD

                            Agency Name           Complaint Type  \
416     New York City Police Department          Illegal Parking
611     New York City Police Department  Noise - Street/Sidewalk
```

```
1648    New York City Police Department          Illegal Parking
1816    New York City Police Department         Derelict Vehicle
1965    New York City Police Department         Derelict Vehicle
…                                  …                        …
300273  New York City Police Department         Blocked Driveway
300492  New York City Police Department                  Vending
300496  New York City Police Department                  Vending
300620  New York City Police Department       Noise - Commercial
300693  New York City Police Department       Noise - Commercial

                         Descriptor        Location Type  Incident Zip  \
416     Posted Parking Sign Violation      Street/Sidewalk           NaN
611                 Loud Music/Party      Street/Sidewalk           NaN
1648     Commercial Overnight Parking      Street/Sidewalk           NaN
1816              With License Plate      Street/Sidewalk           NaN
1965              With License Plate      Street/Sidewalk           NaN
…                                 …                   …           …
300273                    No Access      Street/Sidewalk           NaN
300492                   Unlicensed      Street/Sidewalk           NaN
300496                   Unlicensed      Street/Sidewalk           NaN
300620               Loud Music/Party  Club/Bar/Restaurant           NaN
300693               Loud Music/Party  Club/Bar/Restaurant           NaN

           Incident Address  … Bridge Highway Name Bridge Highway Direction  \
416       5426-5526 90TH ST  …                  NaN                       NaN
611              30 STREET  …                  NaN                       NaN
1648    21600-2169 91ST AVE  …                  NaN                       NaN
1816            127 STREET  …                  NaN                       NaN
1965     5201-5299 68TH ST  …                  NaN                       NaN
…                      …  …                    …                         …
300273   3801-3999 23RD AVE  …                  NaN                       NaN
300492          COOPER AVE  …                  NaN                       NaN
300496           80 STREET  …                  NaN                       NaN
300620     CRESCENT AVENUE  …                  NaN                       NaN
300693     CRESCENT AVENUE  …                  NaN                       NaN

       Road Ramp Bridge Highway Segment Garage Lot Name Ferry Direction  \
416          NaN                    NaN             NaN             NaN
611          NaN                    NaN             NaN             NaN
1648         NaN                    NaN             NaN             NaN
1816         NaN                    NaN             NaN             NaN
1965         NaN                    NaN             NaN             NaN
…            …                      …               …               …
300273       NaN                    NaN             NaN             NaN
300492       NaN                    NaN             NaN             NaN
300496       NaN                    NaN             NaN             NaN
300620       NaN                    NaN             NaN             NaN
```

```
300693       NaN                     NaN             NaN             NaN

       Ferry Terminal Name Latitude Longitude Location
416                    NaN      NaN       NaN      NaN
611                    NaN      NaN       NaN      NaN
1648                   NaN      NaN       NaN      NaN
1816                   NaN      NaN       NaN      NaN
1965                   NaN      NaN       NaN      NaN
…                      …        …         …        …
300273                 NaN      NaN       NaN      NaN
300492                 NaN      NaN       NaN      NaN
300496                 NaN      NaN       NaN      NaN
300620                 NaN      NaN       NaN      NaN
300693                 NaN      NaN       NaN      NaN

[2164 rows x 53 columns]
```

[12]:
```
# For our future exploration on Closed Date
# column we have noted down one row by its unique key column
# to check changes everytime we do something for Closed Date or related column
df_Nyc[df_Nyc['Unique Key'] == 32305700]
```

[12]:
```
     Unique Key            Created Date Closed Date Agency  \
416    32305700  12/31/2015 02:16:04 PM         NaN   NYPD

                          Agency Name   Complaint Type  \
416  New York City Police Department  Illegal Parking

                        Descriptor    Location Type  Incident Zip  \
416  Posted Parking Sign Violation  Street/Sidewalk           NaN

      Incident Address  … Bridge Highway Name Bridge Highway Direction  \
416  5426-5526 90TH ST  …                 NaN                       NaN

    Road Ramp Bridge Highway Segment Garage Lot Name Ferry Direction  \
416       NaN                   NaN             NaN             NaN

    Ferry Terminal Name Latitude Longitude Location
416                 NaN      NaN       NaN      NaN

[1 rows x 53 columns]
```

[13]:
```
# We check data type of each column
df_Nyc.dtypes
```

[13]:
```
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                             object
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                  object
Bridge Highway Direction             object
Road Ramp                            object
Bridge Highway Segment               object
Garage Lot Name                     float64
Ferry Direction                      object
```

```
Ferry Terminal Name                object
Latitude                          float64
Longitude                         float64
Location                           object
dtype: object
```

[14]:
```
# Question 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)

# Solution 2
```

[15]:
```
import datetime as dt
import time, datetime
```

[16]:
```
# Convert "Closed Date" to datetime dtype
df_Nyc['Closed Date'] = pd.to_datetime(df_Nyc['Closed Date'])
df_Nyc['Closed Date'].dtype
```

[16]:
```
dtype('<M8[ns]')
```

[17]:
```
# Convert "Created Date" to datetime dtype
df_Nyc['Created Date'] = pd.to_datetime(df_Nyc['Created Date'])
df_Nyc['Created Date'].dtype
```

[17]:
```
dtype('<M8[ns]')
```

[18]:
```
# Create new column Request_Closing_Time with time taken to close complain
df_Nyc["Request_Closing_Time"] = df_Nyc["Closed Date"]-df_Nyc["Created Date"]
df_Nyc["Request_Closing_Time"].head()
```

[18]:
```
0   00:55:15
1   01:26:16
2   04:51:31
3   07:45:14
4   03:27:02
Name: Request_Closing_Time, dtype: timedelta64[ns]
```

[19]:
```
# Question 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.
```

[20]:
```
# Solution 3
# From here starting Insight
# Insight - 1 - Categorize Request_Closing_Time as follows -
```

```
# Below 2 hours - Fast,
# Between 2 to 4 hours - Acceptable,
# Between 4 to 6 - Slow,
# More than 6 hours - Very Slow
# For this, first will create new column Request_Closing_In_Hr and
# then create new column - Request_Closing_Time_Category

# Function to convert TimeDelta in Hour
def toHr(timeDel):
    days = timeDel.days
    hours = round(timeDel.seconds/3600, 2)
    result = (days * 24) + hours
    #print(days)
    #print(hours)
    return result
    #return round(pd.Timedelta(timeDel).seconds / 3600, 2)
```

[21]:
```
# Testing of function with days
test_days = df_Nyc[df_Nyc['Unique Key'] == 32122264]['Request_Closing_Time']
print(toHr(test_days[27704]))
print(test_days[27704])
print(test_days.dtype)
```

```
145.08
6 days 01:05:00
timedelta64[ns]
```

[22]:
```
# Apply this function to every row of column Request_Closing_Time
df_Nyc['Request_Closing_In_Hr'] = df_Nyc['Request_Closing_Time'].apply(toHr)

df_Nyc['Request_Closing_In_Hr'].head()
```

[22]:
```
0    0.92
1    1.44
2    4.86
3    7.75
4    3.45
Name: Request_Closing_In_Hr, dtype: float64
```

[23]:
```
import math
```

[24]:
```
# Function to categorize hours - Less than 2 hours - Fast,
# Between 2 to 4 hours - Acceptable,
# Between 4 to 6 - Slow,
# More than 6 hours - Very Slow
```

[25]:
```
def hrToCategory(hr):
    if (math.isnan(hr)):
        return 'Unspecified'
    elif (hr < 2.0):
        return 'Fast'
    elif (4.0 > hr >= 2.0):
        return 'Acceptable'
    elif (6.0 > hr >= 4.0):
        return 'Slow'
    else:
        return 'Very Slow'
```

[26]:
```
# Testing function
print(hrToCategory(1.99))

# Create new column Request_Closing_Time_Category and apply function on column␣
 ↪Request_Closing_In_Hr

df_Nyc['Request_Closing_Time_Category'] = df_Nyc['Request_Closing_In_Hr'].
 ↪apply(hrToCategory)

df_Nyc['Request_Closing_Time_Category'].head()
```

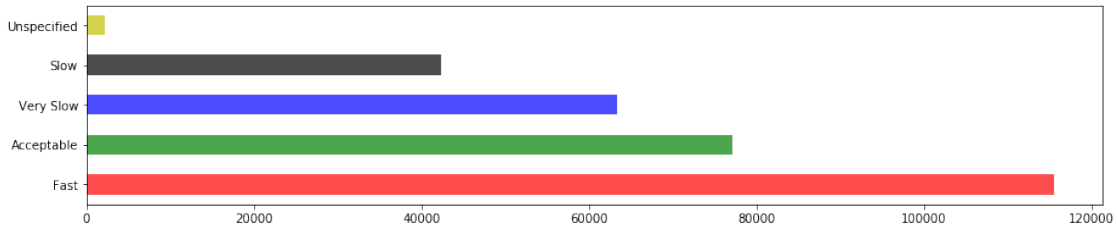
```
Fast
```

[26]:
```
0          Fast
1          Fast
2          Slow
3     Very Slow
4    Acceptable
Name: Request_Closing_Time_Category, dtype: object
```

[27]:
```
df_Nyc['Request_Closing_Time_Category'].value_counts()
```

[27]:
```
Fast           115550
Acceptable      77195
Very Slow       63388
Slow            42401
Unspecified      2164
Name: Request_Closing_Time_Category, dtype: int64
```

[28]:
```
# Create Bar plot for Request_Closing_Time_Category to check frequency in␣
 ↪Request_Closing_Time_Category and it prove Most count is in Fast category␣
 ↪means closed less than 2 hours
df_Nyc['Request_Closing_Time_Category'].value_counts().plot(kind="barh",␣
 ↪color=list('rgbkymc'), alpha=0.7, figsize=(15,3))
plt.show()
```



[29]:
```
df_Nyc.head()
```

[29]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	…	Bridge Highway Segment	Garage Lot Name	Ferry Direction	Ferry Terminal Name	Latitude	Longitude
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	…	NaN	NaN	NaN	NaN	40.865682	-73.923501
1	32309934	2015-12-31 23:59:44	2016-01-01 01:26:00	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	11105.0	27-07 23 AVENUE	…	NaN	NaN	NaN	NaN	40.775945	-73.915094
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	…	NaN	NaN	NaN	NaN	40.870325	-73.888525
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	…	NaN	NaN	NaN	NaN	40.835994	-73.828379
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	87-14 57 ROAD	…	NaN	NaN	NaN	NaN	40.733060	-73.874170

```
                                  Location Request_Closing_Time  \
0   (40.86568153633767, -73.92350095571744)              00:55:15
1  (40.775945312321085, -73.91509393898605)              01:26:16
2  (40.870324522111424, -73.88852464418646)              04:51:31
3   (40.83599404683083, -73.82837939584206)              07:45:14
4  (40.733059618956815, -73.87416975810375)              03:27:02

  Request_Closing_In_Hr Request_Closing_Time_Category
0                  0.92                          Fast
1                  1.44                          Fast
2                  4.86                          Slow
3                  7.75                     Very Slow
4                  3.45                    Acceptable

[5 rows x 56 columns]
```

[30]:
```
# Insight 2 - To check with Month have Complain creation most and least

# We will create one column with Create_Month name

# Created Series for months in text format
monthSeries = pd.Series({1: 'Jan', 2: 'Feb', 3: 'Mar', 4: 'Apr', 5: 'May', 6:␣
 ↪'Jun', 7: 'Jul', 8: 'Aug', 9: 'Sep', 10: 'Oct', 11: 'Nov', 12: 'Dec'})
print(monthSeries)
print(monthSeries[12])
```

```
1     Jan
2     Feb
3     Mar
4     Apr
5     May
6     Jun
7     Jul
8     Aug
9     Sep
10    Oct
11    Nov
12    Dec
dtype: object
Dec
```

[31]:
```
df_Nyc['Created Date'].dtype

# Function to fetch month from Created Date column

def getMonth(cDate):
    a = str(cDate)
```

```
    datee = datetime.datetime.strptime(a, "%Y-%m-%d %H:%M:%S")
    return monthSeries[datee.month]

# Test function getMonth
print(df_Nyc['Created Date'][0])
print(getMonth(df_Nyc['Created Date'][0]))
```

```
2015-12-31 23:59:45
Dec
```

[32]:
```
# Created new column Created_Month and kept all text format months in that column

df_Nyc['Created_Month'] = df_Nyc['Created Date'].apply(getMonth)
df_Nyc['Created_Month']
```

[32]:
```
0         Dec
1         Dec
2         Dec
3         Dec
4         Dec
         ...
300693    Mar
300694    Mar
300695    Mar
300696    Mar
300697    Mar
Name: Created_Month, Length: 300698, dtype: object
```

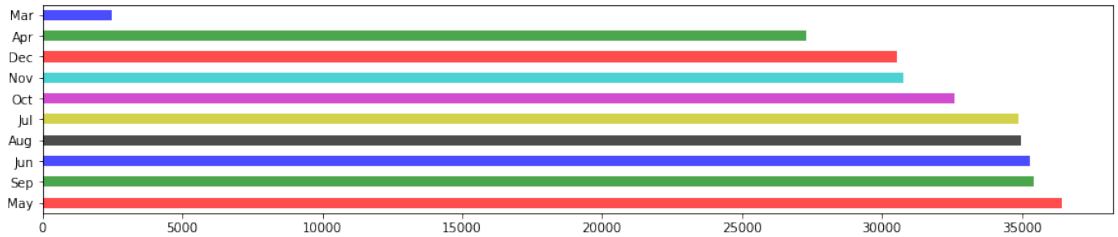
[33]:
```
df_Nyc.head()
```

[33]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip
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
1	32309934	2015-12-31 23:59:44	2016-01-01 01:26:00	NYPD	New York City Police Department	Blocked Driveway			
2	32309159	2015-12-31 23:59:29	2016-01-01 04:51:00	NYPD	New York City Police Department	Blocked Driveway			
3	32305098	2015-12-31 23:57:46	2016-01-01 07:43:00	NYPD	New York City Police Department	Illegal Parking			
4	32306529	2015-12-31 23:56:58	2016-01-01 03:24:00	NYPD	New York City Police Department	Illegal Parking			

```
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  … Garage Lot Name Ferry 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

  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 Request_Closing_Time  \
0   (40.86568153633767, -73.92350095571744)              00:55:15
1  (40.775945312321085, -73.91509393898605)              01:26:16
2  (40.870324522111424, -73.88852464418646)              04:51:31
3   (40.83599404683083, -73.82837939584206)              07:45:14
4  (40.733059618956815, -73.87416975810375)              03:27:02

  Request_Closing_In_Hr Request_Closing_Time_Category Created_Month
0                  0.92                          Fast           Dec
1                  1.44                          Fast           Dec
2                  4.86                          Slow           Dec
3                  7.75                     Very Slow           Dec
4                  3.45                    Acceptable           Dec

[5 rows x 57 columns]
```

[34]:
```
df_Nyc['Created_Month'].value_counts()

# Create Bar plot for Complain Created Month to check frequency and it prove␣
 ↪Most count is in May month and least is in March and in January there is no␣
 ↪any complain
df_Nyc['Created_Month'].value_counts().plot(kind="barh", color=list('rgbkymc'),␣
 ↪alpha=0.7, figsize=(15,3))
plt.show()
```



[35]:
```
# To confirm doubt of January doesn't have any value, we used original␣
 ↪dataframe and check if any entry for Jan month
df_orig[df_orig['Created Date'].str.startswith('01/')]
```

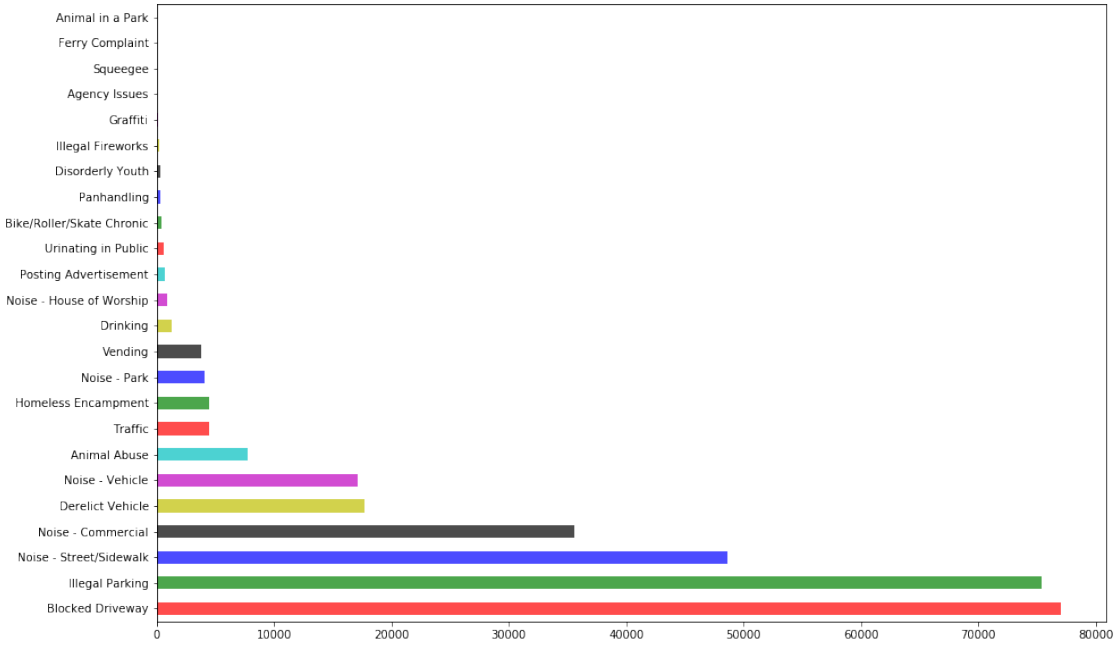
[35]:
```
Empty DataFrame
Columns: [Unique Key, Created Date, Closed Date, Agency, Agency Name, Complaint
Type, Descriptor, Location Type, Incident Zip, Incident Address, Street Name,
Cross Street 1, Cross Street 2, Intersection Street 1, Intersection Street 2,
Address Type, City, Landmark, Facility 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]
Index: []

[0 rows x 53 columns]
```

[36]:
```
# Insight - 3
# Check count in each complain type - sorted decreasing order
df_Nyc['Complaint Type'].value_counts()
```

[36]:
```
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
```

[37]:
```
# Create Bar plot for complain type to check frequency in Complain Type
df_Nyc['Complaint Type'].value_counts().plot(kind="barh",␣
 ↪color=list('rgbkymc'), alpha=0.7, figsize=(15,10))
plt.show()
```

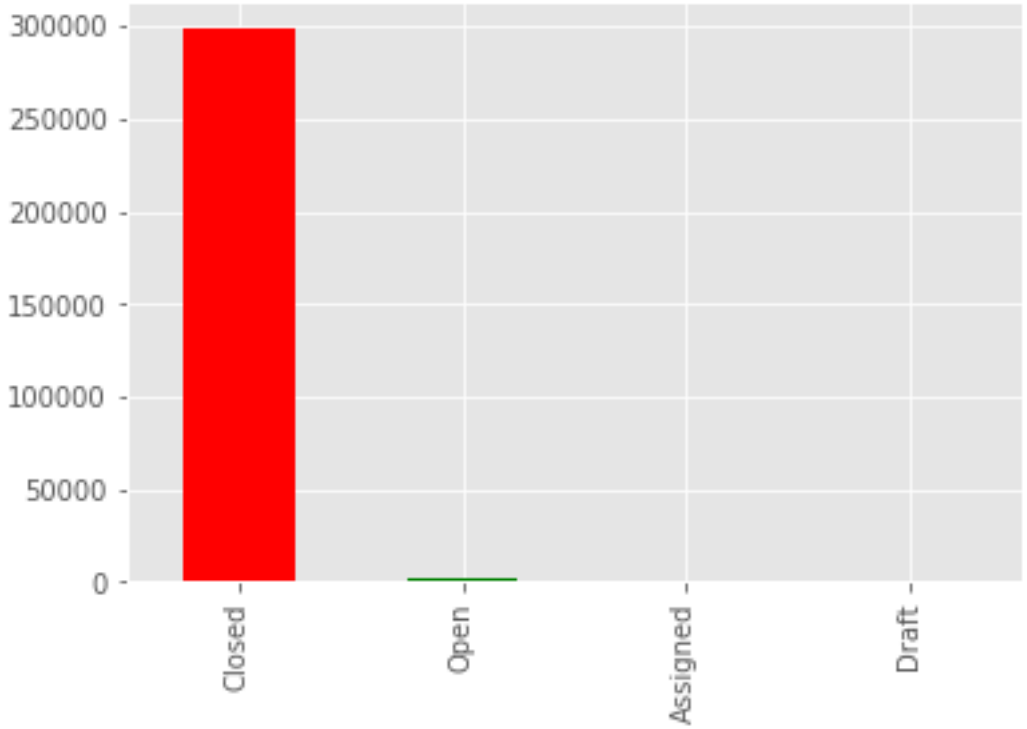


[38]:
```
# Insight 4
# Let's check count for status type
df_Nyc['Status'].value_counts()
```

[38]:
```
Closed      298471
Open          1439
Assigned       786
```

```
Draft            2
Name: Status, dtype: int64
```

[39]:
```
# Draw Bar lot for Status
from matplotlib import style
style.use('ggplot')
df_Nyc['Status'].value_counts().plot(kind='bar', color=list('rgbkymc'))
plt.show()
```



[40]:
```
# Question 4.: Order the complaint types based on the average␣
 ↪'Request_Closing_Time', grouping them for different locations.

# Solution 4:

# For location we can choose here City, so first check if there is missing␣
 ↪values there
df_Nyc['City'].isnull().sum()
```

[40]:
```
2614
```

[41]:
```
# Fill all missing value with some default value here i used - Not Available
df_Nyc['City'].fillna('Not Available', inplace=True)
```

[42]:
```
df_Nyc['City'].head()
```

[42]:
```
0    NEW YORK
1     ASTORIA
2       BRONX
3       BRONX
4    ELMHURST
Name: City, dtype: object
```

[43]:
```
df_Nyc['City']
```

[43]:
```
0              NEW YORK
1               ASTORIA
2                 BRONX
3                 BRONX
4              ELMHURST
                 …
300693    Not Available
300694    RICHMOND HILL
300695         BROOKLYN
300696            BRONX
300697         NEW YORK
Name: City, Length: 300698, dtype: object
```

[44]:
```
# Group them for City (location) first and Complain Type in that
df_Nyc_grouped = df_Nyc.groupby(['City', 'Complaint Type'])
```

[45]:
```
# get average of this grouped dataframe, and get Request_Closing_Time column␣
 ↪from there
df_Nyc_mean = df_Nyc_grouped.mean()['Request_Closing_In_Hr']
df_Nyc_mean.isnull().sum()
```

[45]:
```
4
```

[46]:
```
# Group by City(location) first and then Complain Type and showing average of␣
 ↪Request Closing in Hour
df_Nyc_grouped = df_Nyc.groupby(['City','Complaint Type']).
 ↪agg({'Request_Closing_In_Hr': 'mean'})
df_Nyc_grouped
```

[46]:

		Request_Closing_In_Hr
City	Complaint Type	
ARVERNE	Animal Abuse	2.153158
	Blocked Driveway	2.526000
	Derelict Vehicle	2.968889
	Disorderly Youth	3.595000
	Drinking	0.240000

```
…                                                    …
Woodside Blocked Driveway                        6.405455
         Derelict Vehicle                        4.965000
         Illegal Parking                         5.219500
         Noise - Commercial                      2.390000
         Noise - Street/Sidewalk                 3.410000

[782 rows x 1 columns]
```

[47]:
```
# Check if any value is NaN
df_Nyc_grouped[df_Nyc_grouped['Request_Closing_In_Hr'].isnull()]
```

[47]:
```
                                        Request_Closing_In_Hr
City          Complaint Type
Not Available Ferry Complaint                             NaN
              Noise - House of Worship                    NaN
              Panhandling                                 NaN
              Posting Advertisement                       NaN
```

[48]:
```
# Check total rows
print(df_Nyc_grouped)
```

```
                                   Request_Closing_In_Hr
City     Complaint Type
ARVERNE  Animal Abuse                           2.153158
         Blocked Driveway                       2.526000
         Derelict Vehicle                       2.968889
         Disorderly Youth                       3.595000
         Drinking                               0.240000
…                                                    …
Woodside Blocked Driveway                       6.405455
         Derelict Vehicle                       4.965000
         Illegal Parking                        5.219500
         Noise - Commercial                     2.390000
         Noise - Street/Sidewalk                3.410000

[782 rows x 1 columns]
```

[49]:
```
# drop null values from this group
df_Nyc_grouped_withoutna = df_Nyc_grouped.dropna()
```

[50]:
```
# verify if new group has null values
df_Nyc_grouped_withoutna.isnull().sum()
```

[50]:
```
Request_Closing_In_Hr    0
dtype: int64
```

[51]:
```
# verify number of rows after dropping null values
print(df_Nyc_grouped_withoutna)
```

```
                                  Request_Closing_In_Hr
City     Complaint Type
ARVERNE  Animal Abuse                          2.153158
         Blocked Driveway                      2.526000
         Derelict Vehicle                      2.968889
         Disorderly Youth                      3.595000
         Drinking                              0.240000
…                                                   …
Woodside Blocked Driveway                      6.405455
         Derelict Vehicle                      4.965000
         Illegal Parking                       5.219500
         Noise - Commercial                    2.390000
         Noise - Street/Sidewalk               3.410000

[778 rows x 1 columns]
```

[52]:
```
# Sorting by column - Request_Closing_In_Hr for City on grouped
df_Nyc_sorted = df_Nyc_grouped_withoutna.sort_values(['City',␣
 ↪'Request_Closing_In_Hr'])
df_Nyc_sorted
```

[52]:
```
                                  Request_Closing_In_Hr
City     Complaint Type
ARVERNE  Drinking                              0.240000
         Vending                               0.480000
         Urinating in Public                   0.690000
         Panhandling                           1.030000
         Noise - Park                          1.285000
…                                                   …
Woodside Noise - Commercial                    2.390000
         Noise - Street/Sidewalk               3.410000
         Derelict Vehicle                      4.965000
         Illegal Parking                       5.219500
         Blocked Driveway                      6.405455

[778 rows x 1 columns]
```

[ ]:
```
# Question 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?
```

[53]:
```
import scipy.stats as stats
from math import sqrt
```

[54]:
```
##### Try ANOVA for first one

# HO : All Complain Types average response time mean is similar
# H1 : Not similar

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

[54]:
```
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
```

[55]:
```
top5_complaints_type = df_Nyc['Complaint Type'].value_counts()[:5]
top5_complaints_type
```

[55]:
```
Blocked Driveway          77044
Illegal Parking           75361
Noise - Street/Sidewalk   48612
Noise - Commercial        35577
```

```
Derelict Vehicle           17718
Name: Complaint Type, dtype: int64
```

[56]:
```
top5_complaints_type_names = top5_complaints_type.index
top5_complaints_type_names
```

[56]:
```
Index(['Blocked Driveway', 'Illegal Parking', 'Noise - Street/Sidewalk',
       'Noise - Commercial', 'Derelict Vehicle'],
      dtype='object')
```

[58]:
```
sample_data = df_Nyc.loc[df_Nyc['Complaint Type'].
↪isin(top5_complaints_type_names), ['Complaint Type',␣
↪'Request_Closing_In_Hr']]
sample_data.head()
```

[58]:

	Complaint Type	Request_Closing_In_Hr
0	Noise - Street/Sidewalk	0.92
1	Blocked Driveway	1.44
2	Blocked Driveway	4.86
3	Illegal Parking	7.75
4	Illegal Parking	3.45

[59]:
```
sample_data.shape
```

[59]:
```
(254312, 2)
```

[60]:
```
sample_data.isnull().sum()
```

[60]:
```
Complaint Type              0
Request_Closing_In_Hr    2059
dtype: int64
```

[61]:
```
#sample_data[~sample_data.isin(['NaN', 'NaT']).any(axis=1)]
#sample_data[sample_data.isnull()]

sample_data.dropna(how='any', inplace=True)
sample_data.isnull().sum()
# sample_data_without_null[sample_data_without_null.isnull()]
```

[61]:
```
Complaint Type           0
Request_Closing_In_Hr    0
dtype: int64
```

[62]:
```
sample_data.shape
```

[62]:
```
(252253, 2)
```

[63]:
```
s1 = sample_data[sample_data['Complaint Type'] ==␣
↪top5_complaints_type_names[0]].Request_Closing_In_Hr
s1.head()
```

[63]:
```
1     1.44
2     4.86
7     1.80
9     1.38
10    7.80
Name: Request_Closing_In_Hr, dtype: float64
```

[64]:
```
s2 = sample_data[sample_data['Complaint Type'] ==␣
↪top5_complaints_type_names[1]].Request_Closing_In_Hr
s2.head()
```

[64]:
```
3    7.75
4    3.45
5    1.89
6    1.96
8    8.55
Name: Request_Closing_In_Hr, dtype: float64
```

[65]:
```
s3 = sample_data[sample_data['Complaint Type'] ==␣
↪top5_complaints_type_names[2]].Request_Closing_In_Hr
s3.head()
```

[65]:
```
0     0.92
12    2.48
19    0.78
38    0.49
54    1.50
Name: Request_Closing_In_Hr, dtype: float64
```

[66]:
```
s4 = sample_data[sample_data['Complaint Type'] ==␣
↪top5_complaints_type_names[3]].Request_Closing_In_Hr
s4.head()
```

[66]:
```
17    0.85
18    2.93
22    1.26
29    2.50
30    1.99
Name: Request_Closing_In_Hr, dtype: float64
```

[67]:
```
s5 = sample_data[sample_data['Complaint Type'] ==␣
↪top5_complaints_type_names[4]].Request_Closing_In_Hr
s5.head()
```

[67]:
```
14      10.49
151      3.95
255      1.36
256      4.13
295      0.75
Name: Request_Closing_In_Hr, dtype: float64
```

[68]:
```
print(s1.isnull().sum())
print(s2.isnull().sum())
print(s3.isnull().sum())
print(s4.isnull().sum())
print(s5.isnull().sum())
```

```
0
0
0
0
0
```

[69]:
```
stats.f_oneway(s1, s2, s3, s4, s5)
```

[69]:
```
F_onewayResult(statistic=1799.598683238952, pvalue=0.0)
```

[ ]:
```
# We can see pvalue is less than 0.05 so we reject null hypothesis and average␣
↪response time is not same.
```

[ ]:
```
### Try ChiSquare Test for second one - # Are the type of complaint or service␣
↪requested and location related?

# H0 : 2 categories - Complain Type and Location is independent means not␣
↪related
# Ha : 2 categories - Complain Type and Location is dependent means related
```

[70]:
```
top5_location = df_Nyc['City'].value_counts()[:5]
top5_location
```

[70]:
```
BROOKLYN         98307
NEW YORK         65994
BRONX            40702
STATEN ISLAND    12343
JAMAICA           7296
Name: City, dtype: int64
```

[71]:
```
top5_location_names = top5_location.index
top5_location_names
```

[71]:
```
Index(['BROOKLYN', 'NEW YORK', 'BRONX', 'STATEN ISLAND', 'JAMAICA'],
dtype='object')
```

[74]:
```
sample_data_location_c_type = df_Nyc.loc[(df_Nyc['Complaint Type'].
↪isin(top5_complaints_type_names)) & (df_Nyc['City'].
↪isin(top5_location_names)), ['Complaint Type', 'City']]
sample_data_location_c_type.head()
```

[74]:

	Complaint Type	City
0	Noise - Street/Sidewalk	NEW YORK
2	Blocked Driveway	BRONX
3	Illegal Parking	BRONX
5	Illegal Parking	BROOKLYN
6	Illegal Parking	NEW YORK

[75]:
```
pd.crosstab(sample_data_location_c_type['Complaint Type'],␣
↪sample_data_location_c_type['City'], margins=True)
```

[75]:

City	BRONX	BROOKLYN	JAMAICA	NEW YORK	STATEN ISLAND	All
Complaint Type						
Blocked Driveway	12755	28148	2818	2072	2142	47935
Derelict Vehicle	1953	5181	954	537	1766	10391
Illegal Parking	7859	27462	1421	12128	4886	53756
Noise - Commercial	2434	11463	429	14550	678	29554
Noise - Street/Sidewalk	8892	13356	339	20433	819	43839
All	33893	85610	5961	49720	10291	185475

[76]:
```
ch2, p_value, df, exp_frq = stats.chi2_contingency(pd.
↪crosstab(sample_data_location_c_type['Complaint Type'],␣
↪sample_data_location_c_type['City']))
```

[77]:
```
print(ch2)
print(p_value)
```

```
40522.79928349593
0.0
```

[ ]:
```
#We can see pvalue is less than 0.05 so we reject null hypothesis means␣
↪complain type and location is not independent.
```

[ ]:
```

```