In [1]:
```
#Importing relevant packages
import numpy as np
import pandas as pd
import time
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()

import warnings
warnings.filterwarnings("ignore")

from scipy import stats
from scipy.stats import chi2_contingency

import statsmodels.api as sm
from statsmodels.formula.api import ols
```

In [2]:
```
#Importing the dataset of New york city calls
df = pd.read_csv("311_Service_Requests_from_2010_to_Present.csv")
```

In [3]:
```
#Exploring the data
df.head()
```

Out[3]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	...	Bridge Highway Name	Bridge Highway Direction	Road Ramp	Bridge Highway Segment	Garage Lot Name	Ferry Direction	Ferry Terminal Name	Latitude	Longituc
0	32310363	12/31/2015 11:59:45 PM	01-01-16 0:55	NYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk	10034.0	71 VERMILYEA AVENUE	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	40.865682	-73.92350
1	32309934	12/31/2015 11:59:44 PM	01-01-16 1:26	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	11105.0	27-07 23 AVENUE	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	40.775945	-73.91509
2	32309159	12/31/2015 11:59:29 PM	01-01-16 4:51	NYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	10458.0	2897 VALENTINE AVENUE	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	40.870325	-73.88852
3	32305098	12/31/2015 11:57:46 PM	01-01-16 7:43	NYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Sidewalk	10461.0	2940 BAISLEY AVENUE	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	40.835994	-73.82837
4	32306529	12/31/2015 11:56:58 PM	01-01-16 3:24	NYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street/Sidewalk	11373.0	87-14 57 ROAD	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	40.733060	-73.87417

5 rows × 53 columns

In [4]:
```
df.describe()
```

Out[4]:

	Unique Key	Incident Zip	X Coordinate (State Plane)	Y Coordinate (State Plane)	School or Citywide Complaint	Vehicle Type	Taxi Company Borough	Taxi Pick Up Location	Garage Lot Name	Latitude	Longitude
count	3.006980e+05	298083.000000	2.971580e+05	297158.000000	0.0	0.0	0.0	0.0	0.0	297158.000000	297158.000000
mean	3.130054e+07	10848.888645	1.004854e+06	203754.534416	NaN	NaN	NaN	NaN	NaN	40.725885	-73.925630
std	5.738547e+05	583.182081	2.175338e+04	29880.183529	NaN	NaN	NaN	NaN	NaN	0.082012	0.078454
min	3.027948e+07	83.000000	9.133570e+05	121219.000000	NaN	NaN	NaN	NaN	NaN	40.499135	-74.254937
25%	3.080118e+07	10310.000000	9.919752e+05	183343.000000	NaN	NaN	NaN	NaN	NaN	40.669796	-73.972142
50%	3.130436e+07	11208.000000	1.003158e+06	201110.500000	NaN	NaN	NaN	NaN	NaN	40.718661	-73.931781
75%	3.178446e+07	11238.000000	1.018372e+06	224125.250000	NaN	NaN	NaN	NaN	NaN	40.781840	-73.876805
max	3.231065e+07	11697.000000	1.067173e+06	271876.000000	NaN	NaN	NaN	NaN	NaN	40.912869	-73.700760

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300698 entries, 0 to 300697
Data columns (total 53 columns):
 #   Column                          Non-Null Count   Dtype  
---  ------                          --------------   -----  
 0   Unique Key                      300698 non-null  int64  
 1   Created Date                    300698 non-null  object 
 2   Closed Date                     298534 non-null  object 
 3   Agency                          300698 non-null  object 
 4   Agency Name                     300698 non-null  object 
 5   Complaint Type                  300698 non-null  object 
 6   Descriptor                      294784 non-null  object 
 7   Location Type                   300567 non-null  object 
 8   Incident Zip                    298083 non-null  float64
 9   Incident Address                256288 non-null  object 
 10  Street Name                     256288 non-null  object 
 11  Cross Street 1                  251419 non-null  object 
 12  Cross Street 2                  250919 non-null  object 
 13  Intersection Street 1           43858 non-null   object 
 14  Intersection Street 2           43362 non-null   object 
 15  Address Type                    297883 non-null  object 
 16  City                            298084 non-null  object 
 17  Landmark                        349 non-null     object 
 18  Facility Type                   298527 non-null  object 
 19  Status                          300698 non-null  object 
 20  Due Date                        300695 non-null  object 
 21  Resolution Description          300698 non-null  object 
 22  Resolution Action Updated Date  298511 non-null  object 
 23  Community Board                 300698 non-null  object 
 24  Borough                         300698 non-null  object 
 25  X Coordinate (State Plane)      297158 non-null  float64
 26  Y Coordinate (State Plane)      297158 non-null  float64
 27  Park Facility Name              300698 non-null  object 
 28  Park Borough                    300698 non-null  object 
 29  School Name                     300698 non-null  object 
 30  School Number                   300698 non-null  object 
 31  School Region                   300697 non-null  object 
 32  School Code                     300697 non-null  object 
 33  School Phone Number             300698 non-null  object 
 34  School Address                  300698 non-null  object 
 35  School City                     300698 non-null  object 
 36  School State                    300698 non-null  object 
 37  School Zip                      300697 non-null  object 
 38  School Not Found                300698 non-null  object 
 39  School or Citywide Complaint    0 non-null       float64
 40  Vehicle Type                    0 non-null       float64
 41  Taxi Company Borough            0 non-null       float64
 42  Taxi Pick Up Location           0 non-null       float64
 43  Bridge Highway Name             243 non-null     object 
 44  Bridge Highway Direction        243 non-null     object 
 45  Road Ramp                       213 non-null     object 
 46  Bridge Highway Segment          213 non-null     object 
 47  Garage Lot Name                 0 non-null       float64
 48  Ferry Direction                 1 non-null       object 
 49  Ferry Terminal Name             2 non-null       object 
 50  Latitude                        297158 non-null  float64
```

```
 51  Longitude                       297158 non-null  float64
 52  Location                        297158 non-null  object
dtypes: float64(10), int64(1), object(42)
memory usage: 121.6+ MB
```

In [6]:
```
#1. Identifying the shape of the dataset
df.shape
```

Out[6]:
```
(300698, 53)
```

In [7]:
```
#2. Identifying variables with null values
df.isnull().sum()
```

Out[7]:
```
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
```

We can observe a Lots of Missing Values in the data.

In [8]:
```
#Checking the data type of Date Columns
df[['Created Date','Closed Date','Due Date']].dtypes
```

Out[8]:
```
Created Date    object
Closed Date     object
Due Date        object
dtype: object
```

In [9]:
```
#Converting the data into datetime format
df['Created Date'] = pd.to_datetime(df['Created Date'])
df['Closed Date'] = pd.to_datetime(df['Closed Date'])
```

In [10]:
```
# Creating a new column that consist the amount of time taken to resolve the complaint
df["Request_Closing_Time"] = (df["Closed Date"]-df["Created Date"])
```

In [11]:
```
#Converting the data in Request_Closing_Time column in Minutes
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
```

In [12]:
```
df.head()
```

Out[12]:

ency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	...	Bridge Highway Direction	Road Ramp	Bridge Highway Segment	Garage Lot Name	Ferry Direction	Ferry Terminal Name	Latitude	Longitude	Location	Request_Closing_Time
IYPD	New York City Police Department	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk	10034.0	71 VERMILYEA AVENUE	...	NaN	NaN	NaN	NaN	NaN	NaN	40.865682	-73.923501	(40.86568153633767, -73.92350095571744)	55.250000
IYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	11105.0	27-07 23 AVENUE	...	NaN	NaN	NaN	NaN	NaN	NaN	40.775945	-73.915094	(40.775945312321085, -73.91509393898605)	86.266667
IYPD	New York City Police Department	Blocked Driveway	No Access	Street/Sidewalk	10458.0	2897 VALENTINE AVENUE	...	NaN	NaN	NaN	NaN	NaN	NaN	40.870325	-73.888525	(40.870324522111424, -73.88852464418646)	291.516667
IYPD	New York City Police Department	Illegal Parking	Commercial Overnight Parking	Street/Sidewalk	10461.0	2940 BAISLEY AVENUE	...	NaN	NaN	NaN	NaN	NaN	NaN	40.835994	-73.828379	(40.83599404683083, -73.82837939584206)	465.233333
IYPD	New York City Police Department	Illegal Parking	Blocked Sidewalk	Street/Sidewalk	11373.0	87-14 57 ROAD	...	NaN	NaN	NaN	NaN	NaN	NaN	40.733060	-73.874170	(40.733059618956815, -73.87416975810375)	207.033333

In [13]:
```
#Rounding off "Request Closing Time" to 2 decimal places
df["Request_Closing_Time"] = round(df["Request_Closing_Time"],2)
```

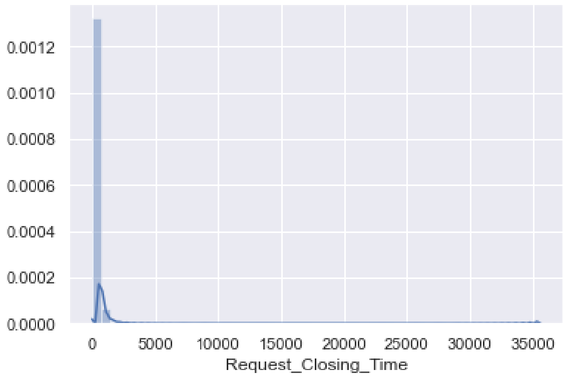
In [14]:
```
#Finding the number of Unique Data in Agency Column
df["Agency"].unique()
```

Out[14]:
```
array(['NYPD'], dtype=object)
```

Findings 1: All of the data belongs to NYPD

In [15]:
```
# Univariate Distribution plot for request closing time
sns.distplot(df["Request_Closing_Time"])
plt.show
```

Out[15]:
```
<function matplotlib.pyplot.show(*args, **kw)>
```

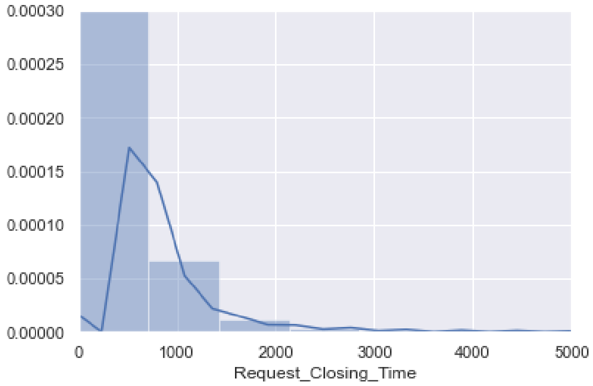


In [16]:
```
print("Total Number of Concerns: ",len(df),"\n")
print("Percentage of Requests took less than 100 hour to get solved :", round((len(df)-(df["Request_Closing_Time"]>100).sum())/len(df)*100,2),"%")
print("Percentage of Requests took less than 1000 hour to get solved :", round((len(df)-(df["Request_Closing_Time"]>1000).sum())/len(df)*100,2),"%")
```

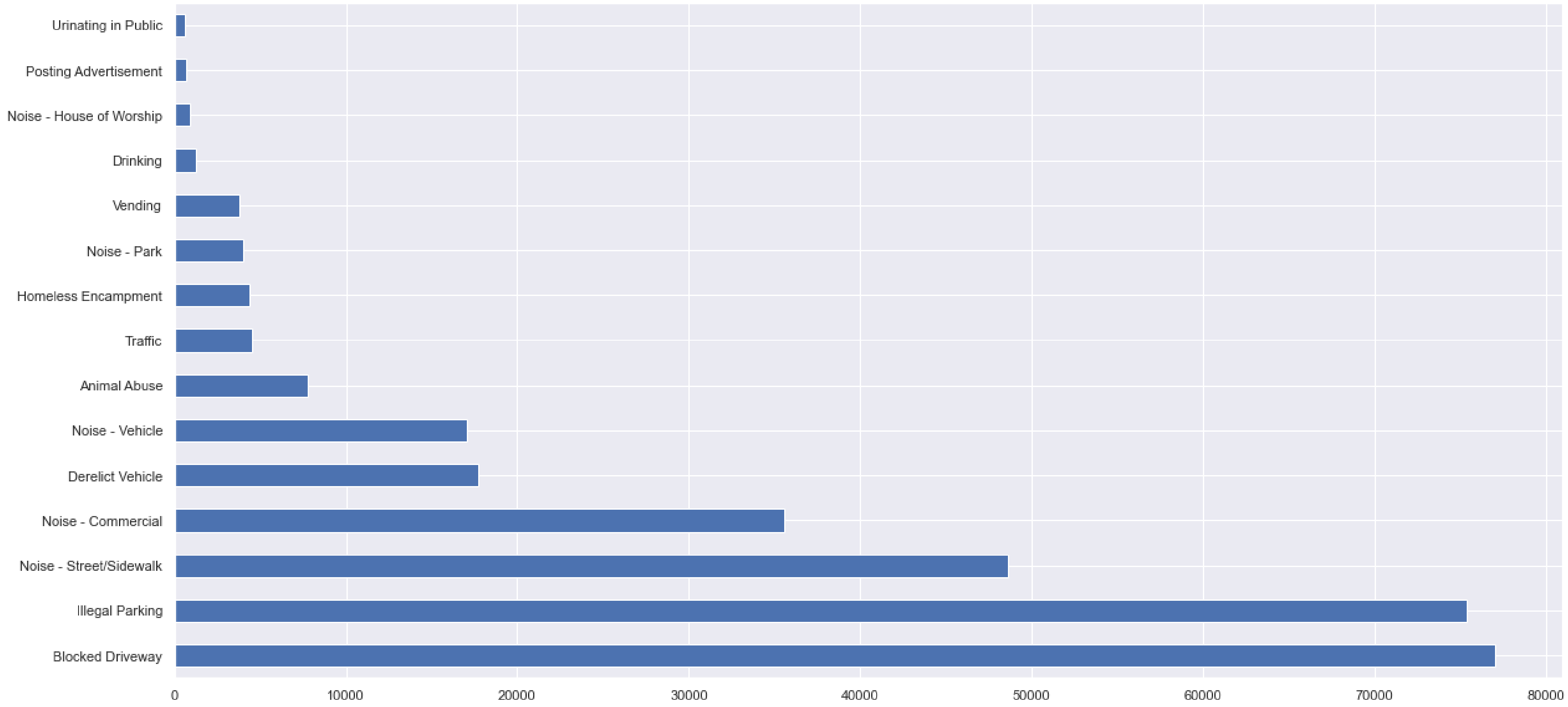
```
Total Number of Concerns:  300698

Percentage of Requests took less than 100 hour to get solved : 33.32 %
Percentage of Requests took less than 1000 hour to get solved : 97.19 %
```

Findings 2: We can see that the data is heavily skewed because of outliers. We may obseve that almost 97% of the requests are getting resolved in less than 1000 hours. Plotting a graph below to visualize the same

In [17]:
```
# Univariate Distribution plot for Request Closing Time
sns.distplot(df["Request_Closing_Time"])
plt.xlim((0,5000))
plt.ylim((0,0.0003))
plt.show()
```



In [18]:
```
# Count plot to understand the type of the complaint raised
df["Complaint Type"].value_counts()[:15].plot(kind="barh",alpha=1.0,figsize=(20,10))
plt.show()
```



In [19]:
```
g = df.groupby("Complaint Type").size()
```

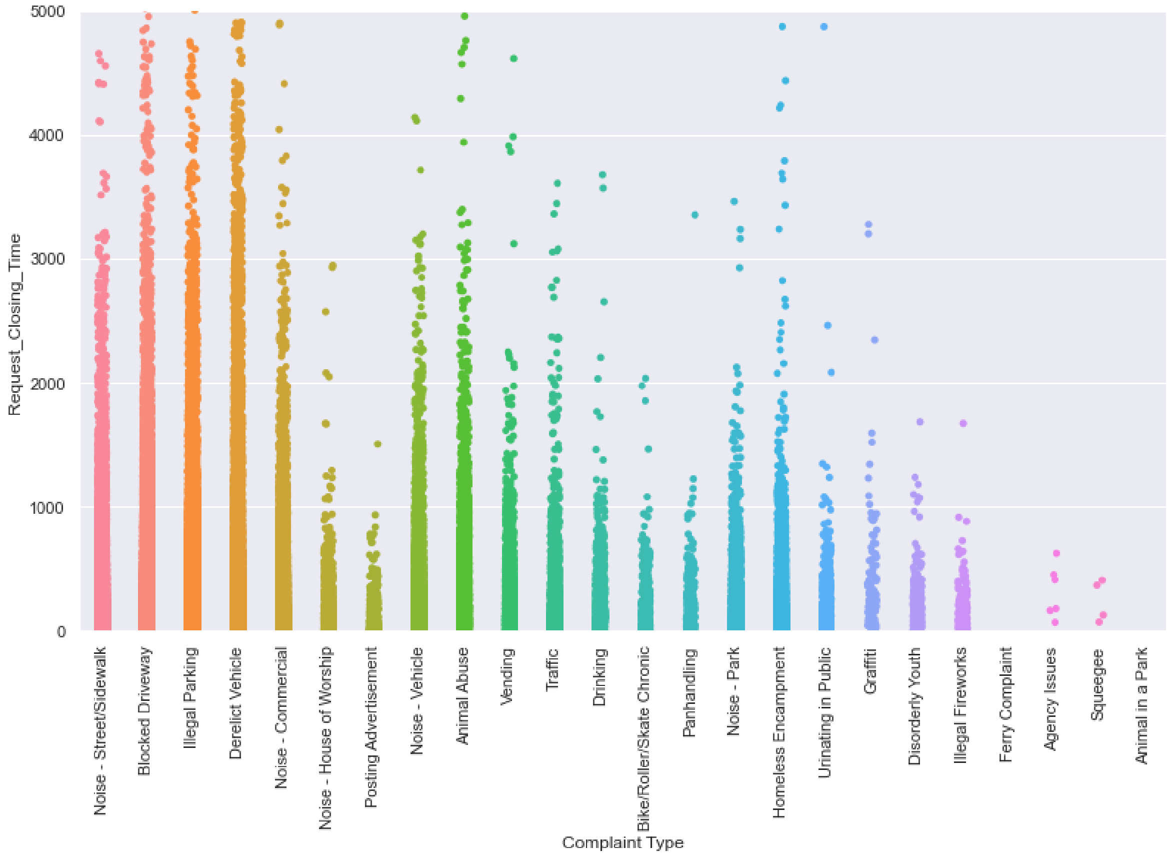
In [20]:
```
top_comp = g.sort_values(ascending = False).head(8)
```

In [21]:
```
top_comp.head(8)
```

Out[21]:
```
Complaint Type
Blocked Driveway           77044
Illegal Parking            75361
Noise - Street/Sidewalk    48612
Noise - Commercial         35577
Derelict Vehicle           17718
Noise - Vehicle            17083
Animal Abuse                7778
Traffic                     4498
dtype: int64
```

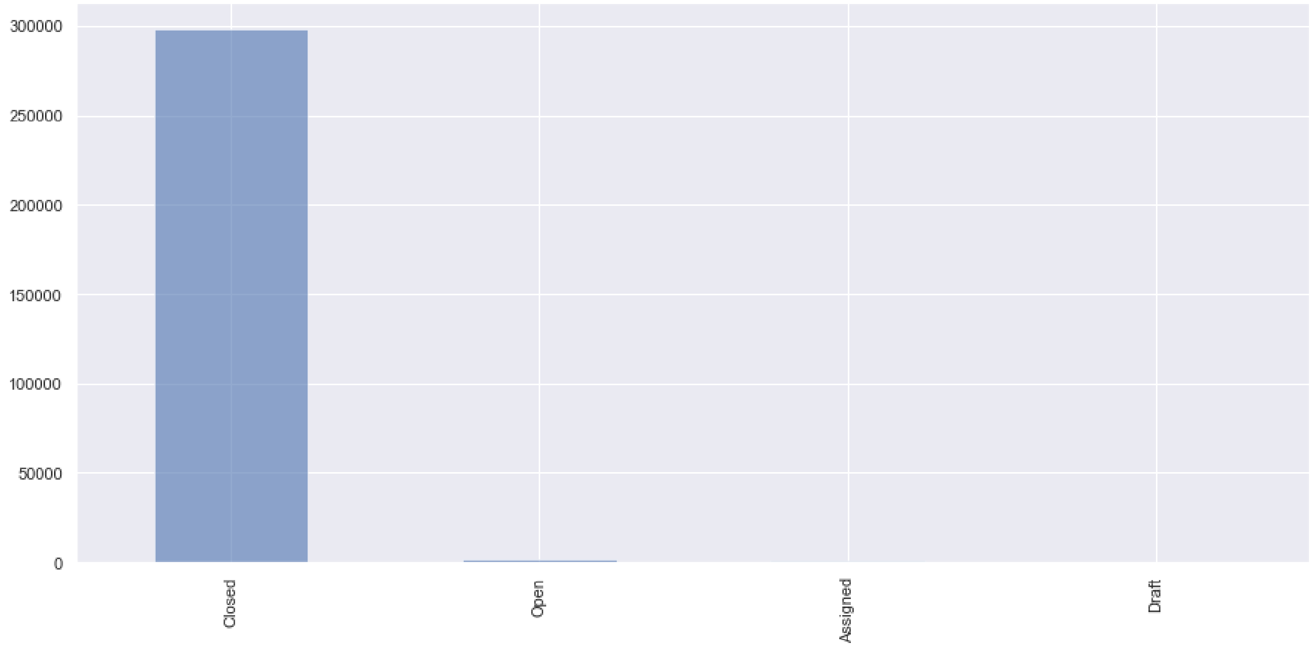
Findings 3 - We can observe that almost 60% of the requests are related to Transport

In [22]:
```
# Categorical scatter plot to understand which type of complaints are taking more time to get resolved
s = sns.catplot(x = "Complaint Type",y="Request_Closing_Time",data = df)
s.fig.set_figwidth(15)
s.fig.set_figheight(7)
plt.xticks(rotation = 90)
plt.ylim((0,5000))
plt.show()
```



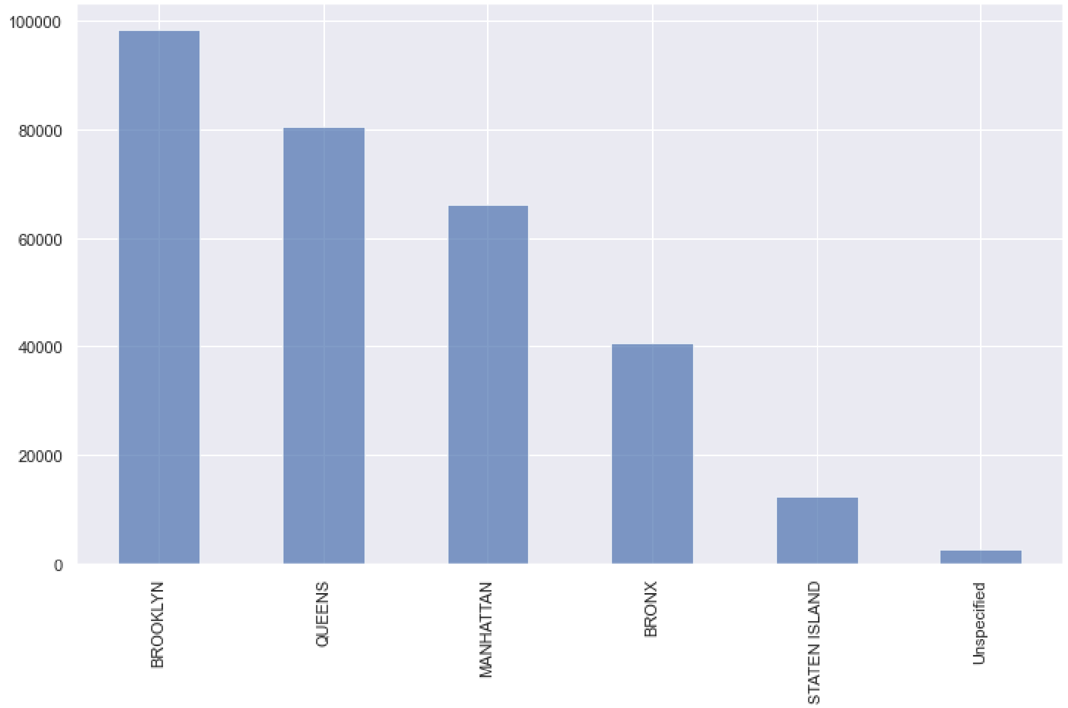
Type *Markdown* and LaTeX: α^2

In [23]:
```
# count plot to know the status of the requests
df["Status"].value_counts().plot(kind = "bar",alpha = 0.6,figsize = (15,7))

plt.show()
```



Findings 4 :98% of the requests are closed

In [24]:
```
# Count plot for Column borough
plt.figure(figsize = (12,7))
df["Borough"].value_counts().plot(kind = "bar",alpha=0.7)
plt.show()
```



In [25]:
```
# Percentage of cases in each borough
for x in df["Borough"].unique():
    print("Percentage of Request from",x,"Division:", round((df["Borough"]==x).sum()/len(df)*100,2))
```

```
Percentage of Request from MANHATTAN Division: 21.99
Percentage of Request from QUEENS Division: 26.82
Percentage of Request from BRONX Division: 13.54
Percentage of Request from BROOKLYN Division: 32.69
Percentage of Request from Unspecified Division: 0.86
Percentage of Request from STATEN ISLAND Division: 4.1
```

In [26]:
```
# Unique Location Types
df["Location Type"].unique()
```

Out[26]:
```
array(['Street/Sidewalk', 'Club/Bar/Restaurant', 'Store/Commercial',
       'House of Worship', 'Residential Building/House',
       'Residential Building', 'Park/Playground', 'Vacant Lot',
       'House and Store', 'Highway', 'Commercial', 'Roadway Tunnel',
       'Subway Station', 'Parking Lot', 'Bridge', 'Terminal', nan,
       'Ferry', 'Park'], dtype=object)
```

In [27]:
```
#Request Closing time for all location type sorted in ascending order
pd.DataFrame(df.groupby("Location Type")["Request_Closing_Time"].mean()).sort_values("Request_Closing_Time",ascending=False)
```

Out[27]:

	Request_Closing_Time
Location Type	
Park	20210.080000
Vacant Lot	448.434935
Commercial	320.565806
Parking Lot	320.130171
Residential Building/House	309.505639
House and Store	300.795269
Residential Building	289.089824
Street/Sidewalk	268.515306
Roadway Tunnel	266.525714
Bridge	229.160000
Highway	223.424346
Park/Playground	207.137142
Store/Commercial	198.089098
House of Worship	191.833323
Club/Bar/Restaurant	186.074345
Subway Station	142.251471
Ferry	NaN
Terminal	NaN

Findings 5 - We see that maximum (mean) time to resolve the complaint is taken in Park, Vacant Lot and Commercial areas whereas the cases in the subway stations and restaurent are resolved in very less time

In [28]:
```
# Request Closing Time for all city sorted in ascending order
pd.DataFrame(df.groupby("City")["Request_Closing_Time"].mean()).sort_values("Request_Closing_Time")
```

Out[28]:

City	Request_Closing_Time
ARVERNE	135.895727
ROCKAWAY PARK	139.133772
LITTLE NECK	154.660447
OAKLAND GARDENS	157.853303
BAYSIDE	160.760123
FAR ROCKAWAY	167.399686
NEW YORK	178.357375
FLUSHING	181.081878
FOREST HILLS	193.449011
CORONA	193.670482
WHITESTONE	194.688807
FRESH MEADOWS	195.843223
COLLEGE POINT	196.417918
JACKSON HEIGHTS	196.419828
CENTRAL PARK	197.658557
ELMHURST	198.631074
REGO PARK	207.665720
BREEZY POINT	209.790333
EAST ELMHURST	214.659777
STATEN ISLAND	232.796707
Howard Beach	241.750000
BROOKLYN	242.878854
Long Island City	246.045448
Astoria	251.076285
RIDGEWOOD	266.507594
ASTORIA	275.934782
SAINT ALBANS	283.252014
KEW GARDENS	302.578716
Woodside	312.083083
JAMAICA	312.606079
SOUTH OZONE PARK	319.678569
MIDDLE VILLAGE	323.097501
RICHMOND HILL	329.658538
WOODHAVEN	335.728713

City	Request_Closing_Time
MASPETH	335.985815
SOUTH RICHMOND HILL	337.049099
OZONE PARK	340.863731
HOLLIS	345.610168
East Elmhurst	362.868571
BRONX	365.769719
HOWARD BEACH	369.652331
LONG ISLAND CITY	392.351437
SUNNYSIDE	411.120346
WOODSIDE	413.606002
NEW HYDE PARK	453.365714
GLEN OAKS	528.943725
SPRINGFIELD GARDENS	551.145096
ROSEDALE	601.867581
CAMBRIA HEIGHTS	607.426415
BELLEROSE	633.386720
QUEENS VILLAGE	654.411279
FLORAL PARK	703.171250
QUEENS	815.586250

Handling Missing Values

In [29]:
```
# Finding the Percentage of Missing Value in each column
pd.DataFrame((df.isnull().sum()/df.shape[0]*100)).sort_values(0,ascending = False)[:20]
```

Out[29]:

	0
School or Citywide Complaint	100.000000
Garage Lot Name	100.000000
Vehicle Type	100.000000
Taxi Pick Up Location	100.000000
Taxi Company Borough	100.000000
Ferry Direction	99.999667
Ferry Terminal Name	99.999335
Road Ramp	99.929165
Bridge Highway Segment	99.929165
Bridge Highway Direction	99.919188
Bridge Highway Name	99.919188
Landmark	99.883937
Intersection Street 2	85.579552
Intersection Street 1	85.414602
Cross Street 2	16.554483
Cross Street 1	16.388203
Street Name	14.768971
Incident Address	14.768971
Descriptor	1.966757
Latitude	1.177261

We can observe that certain columns have no values at all and some have 50% of the values missing. I am going to remove those columns as they can't help us in our further analysis

In [30]:
```
ndf = df.loc[:,(df.isnull().sum()/df.shape[0]*100)<=50]
```

In [31]:
```
ndf.shape
```

Out[31]: (300698, 40)

In [32]:
```
rem = []
for x in ndf.columns.tolist():
    if ndf[x].nunique()<=3:
        print(x+"  ",ndf[x].unique())
        rem.append(x)
```

```
Agency   ['NYPD']
Agency Name   ['New York City Police Department' 'NYPD' 'Internal Affairs Bureau']
Facility Type   ['Precinct' nan]
Park Facility Name   ['Unspecified' 'Alley Pond Park - Nature Center']
School Name   ['Unspecified' 'Alley Pond Park - Nature Center']
School Number   ['Unspecified' 'Q001']
School Region   ['Unspecified' nan]
School Code   ['Unspecified' nan]
School Phone Number   ['Unspecified' '7182176034']
School Address   ['Unspecified' 'Grand Central Parkway, near the soccer field']
School City   ['Unspecified' 'QUEENS']
School State   ['Unspecified' 'NY']
School Zip   ['Unspecified' nan]
School Not Found   ['N']
```

We can observe that above columns are not detailed. We may proceed to remove these columns as well to simplify our Analysis?

In [33]:
```
ndf.drop(rem, axis = 1, inplace = True)
```

In [34]:
```
ndf.shape
```

Out[34]:
```
(300698, 26)
```

In [35]:
```
# Remove columns that are not needed for our analysis
rem1 = ["Unique Key","Incident Address","Descriptor","Street Name","Cross Street 1","Cross Street 2","Due Date",
        "Resolution Description","Resolution Action Updated Date","Community Board","X Coordinate (State Plane)",
        "Y Coordinate (State Plane)","Park Borough","Latitude","Longitude","Location"]
```

In [36]:
```
ndf.drop(rem1, axis = 1, inplace = True)
```

In [37]:
```
ndf.head()
```

Out[37]:

	Created Date	Closed Date	Complaint Type	Location Type	Incident Zip	Address Type	City	Status	Borough	Request_Closing_Time
0	2015-12-31 23:59:45	2016-01-01 00:55:00	Noise - Street/Sidewalk	Street/Sidewalk	10034.0	ADDRESS	NEW YORK	Closed	MANHATTAN	55.25
1	2015-12-31 23:59:44	2016-01-01 01:26:00	Blocked Driveway	Street/Sidewalk	11105.0	ADDRESS	ASTORIA	Closed	QUEENS	86.27
2	2015-12-31 23:59:29	2016-01-01 04:51:00	Blocked Driveway	Street/Sidewalk	10458.0	ADDRESS	BRONX	Closed	BRONX	291.52
3	2015-12-31 23:57:46	2016-01-01 07:43:00	Illegal Parking	Street/Sidewalk	10461.0	ADDRESS	BRONX	Closed	BRONX	465.23
4	2015-12-31 23:56:58	2016-01-01 03:24:00	Illegal Parking	Street/Sidewalk	11373.0	ADDRESS	ELMHURST	Closed	QUEENS	207.03

Type *Markdown* and LaTeX: α^2

Statistical Test - Whether the average response time across complaint types are similar or not.

1) Null Hypothesis - There is no significant difference in request closing time for different Complaint type

2) Alternate Hypothesis - There is significant difference in request closing time for different complaint type

In [38]:
```
anova_df = pd.DataFrame()
anova_df["Request_Closing_Time"] = ndf["Request_Closing_Time"]
anova_df["Complaint"] = ndf["Complaint Type"]

#Removing missing values if any
anova_df.dropna(inplace = True)
anova_df.head()
```

Out[38]:

	Request_Closing_Time	Complaint
0	55.25	Noise - Street/Sidewalk
1	86.27	Blocked Driveway
2	291.52	Blocked Driveway
3	465.23	Illegal Parking
4	207.03	Illegal Parking

In [39]:
```
anova_df.shape
```

Out[39]: (298534, 2)

In [40]:
```
lm = ols("Request_Closing_Time~Complaint", data = anova_df).fit()
table = sm.stats.anova_lm(lm)
table
```

Out[40]:

	df	sum_sq	mean_sq	F	PR(>F)
Complaint	22.0	1.455049e+09	6.613859e+07	514.176958	0.0
Residual	298511.0	3.839747e+10	1.286300e+05	NaN	NaN

Since p Value for the Complaint is less thant 0.01, we can accept the alternate hypothesis. There is significant difference in the request closing time for different complaint types

Statistical test 2 - Are the type of complaint or service requested and location related?

Null Hypothesis - Complaint Type and Location Type are independent

Alternate Hypothesis - Complaint Type and Location Type are related

In [41]:
```
chi_sq = pd.DataFrame()
chi_sq["Complaint_Type"] = ndf["Complaint Type"]
chi_sq["Location_Type"] = ndf["Location Type"]
chi_sq.dropna(inplace = True)
```

In [42]:
```
chi_sq.head()
```

Out[42]:

	Complaint_Type	Location_Type
0	Noise - Street/Sidewalk	Street/Sidewalk
1	Blocked Driveway	Street/Sidewalk
2	Blocked Driveway	Street/Sidewalk
3	Illegal Parking	Street/Sidewalk
4	Illegal Parking	Street/Sidewalk

In [43]:
```
# Performing cross tabulation
d_cross = pd.crosstab(chi_sq["Location_Type"],chi_sq["Complaint_Type"])
d_cross
```

Out[43]:

Complaint_Type Location_Type	Animal Abuse	Animal in a Park	Bike/Roller/Skate Chronic	Blocked Driveway	Derelict Vehicle	Disorderly Youth	Drinking	Ferry Complaint	Graffiti	Homeless Encampment	...	Noise - House of Worship	Noise - Park	Noise - Street/Sidewalk	Noise - Vehicle	Panhandling	Posting Advertisement	Squee
Bridge	0	0	0	0	0	0	0	0	0	2	...	0	0	0	0	0	0	
Club/Bar/Restaurant	0	0	0	0	0	0	366	0	0	0	...	0	0	0	0	0	0	
Commercial	62	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
Ferry	0	0	0	0	0	0	0	1	0	0	...	0	0	0	0	0	0	
Highway	0	0	0	0	14	0	0	0	0	15	...	0	0	0	0	0	0	
House and Store	93	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
House of Worship	0	0	0	0	0	0	0	0	0	0	...	929	0	0	0	0	0	
Park	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
Park/Playground	123	0	0	0	0	0	98	0	0	353	...	0	4041	0	0	6	0	
Parking Lot	110	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	7	
Residential Building	227	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
Residential Building/House	5085	0	26	0	0	77	291	0	56	983	...	0	0	0	0	16	54	
Roadway Tunnel	0	0	0	0	5	0	0	0	0	1	...	0	0	0	0	0	0	
Store/Commercial	522	0	53	0	0	8	90	0	32	512	...	0	0	0	0	60	6	
Street/Sidewalk	1531	0	348	77007	17614	201	434	0	25	2541	...	0	0	48601	17080	225	582	
Subway Station	22	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
Terminal	0	0	0	0	0	0	0	1	0	0	...	0	0	0	0	0	0	
Vacant Lot	0	0	0	0	77	0	0	0	0	0	...	0	0	0	0	0	0	

18 rows × 23 columns

In [44]:
```
stat, p, dof, expected = chi2_contingency(d_cross)

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

```
Dependent (reject H0)
```

Since p value for the chi square test is less than 0.05 (LOS) we can reject the null hypothesis;

It can be concluded that Complaint type and Location Type are related