Customer Service Requests Analysis Project 1

DESCRIPTION

Background of Problem Statement :

NYC 311's mission is to provide the public with quick and easy access to all New York City government services and information while offering the best customer service. Each day, NYC311 receives thousands of requests related to several hundred types of non-emergency services, including noise complaints, plumbing issues, and illegally parked cars. These requests are received by NYC311 and forwarded to the relevant agencies such as the police, buildings, or transportation. The agency responds to the request, addresses it, and then closes it.

Problem Objective :

Perform a service request data analysis of New York City 311 calls. You will focus on the data wrangling techniques to understand the pattern in the data and also visualize the major complaint types. Domain: Customer Service

Analysis Tasks to be performed:

(Perform a service request data analysis of New York City 311 calls)

1. Import a 311 NYC service request.
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)
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. Order the complaint types based on the average 'Request_Closing_Time', grouping them for different locations.
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?

########################## **Solution**

#

1. Import a 311 NYC service request.

#

In [2]:
```
import pandas as pd
import numpy as np
#import matplotlib as mpl
from matplotlib import pyplot as plt
```

```
Nyc311DataFrame = pd.read_csv('Data/311_Service_Requests_from_2010_to_Present.csv',
                              parse_dates=['Created Date', 'Closed Date', 'Resolution Action U
```

In [3]:
```
Nyc311DataFrame.describe()
```

Out[3]:

	Incident Zip	X Coordinate (State Plane)	Y Coordinate (State Plane)	School or Citywide Complaint	Vehicle Type	Taxi Company Borough	Taxi Pick Up Location	Garag Lo Nam
count	298083.000000	2.971580e+05	297158.000000	0.0	0.0	0.0	0.0	0.
mean	10848.888645	1.004854e+06	203754.534416	NaN	NaN	NaN	NaN	Nal
std	583.182081	2.175338e+04	29880.183529	NaN	NaN	NaN	NaN	Nal
min	83.000000	9.133570e+05	121219.000000	NaN	NaN	NaN	NaN	Nal
25%	10310.000000	9.919752e+05	183343.000000	NaN	NaN	NaN	NaN	Nal
50%	11208.000000	1.003158e+06	201110.500000	NaN	NaN	NaN	NaN	Nal
75%	11238.000000	1.018372e+06	224125.250000	NaN	NaN	NaN	NaN	Nal
max	11697.000000	1.067173e+06	271876.000000	NaN	NaN	NaN	NaN	Nal

#

1. 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)

#

In [4]:
```
Nyc311DataFrame["Request_Closing_Time"] = Nyc311DataFrame['Closed Date'] - Nyc311Dat
```

In [5]:
```
Nyc311DataFrame
```

Out[5]:

Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	I
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	
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	
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	
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	

Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	I
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	
...	...	...	...	...	...	...	...	
30281872	2015-03-29 00:33:41	NaT	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/Restaurant	
30281230	2015-03-29 00:33:28	2015-03-29 02:33:59	NYPD	New York City Police Department	Blocked Driveway	Partial Access	Street/Sidewalk	
30283424	2015-03-29 00:33:03	2015-03-29 03:40:20	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/Restaurant	
30280004	2015-03-29 00:33:02	2015-03-29 04:38:35	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/Restaurant	
30281825	2015-03-29 00:33:01	2015-03-29 04:41:50	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Store/Commercial	

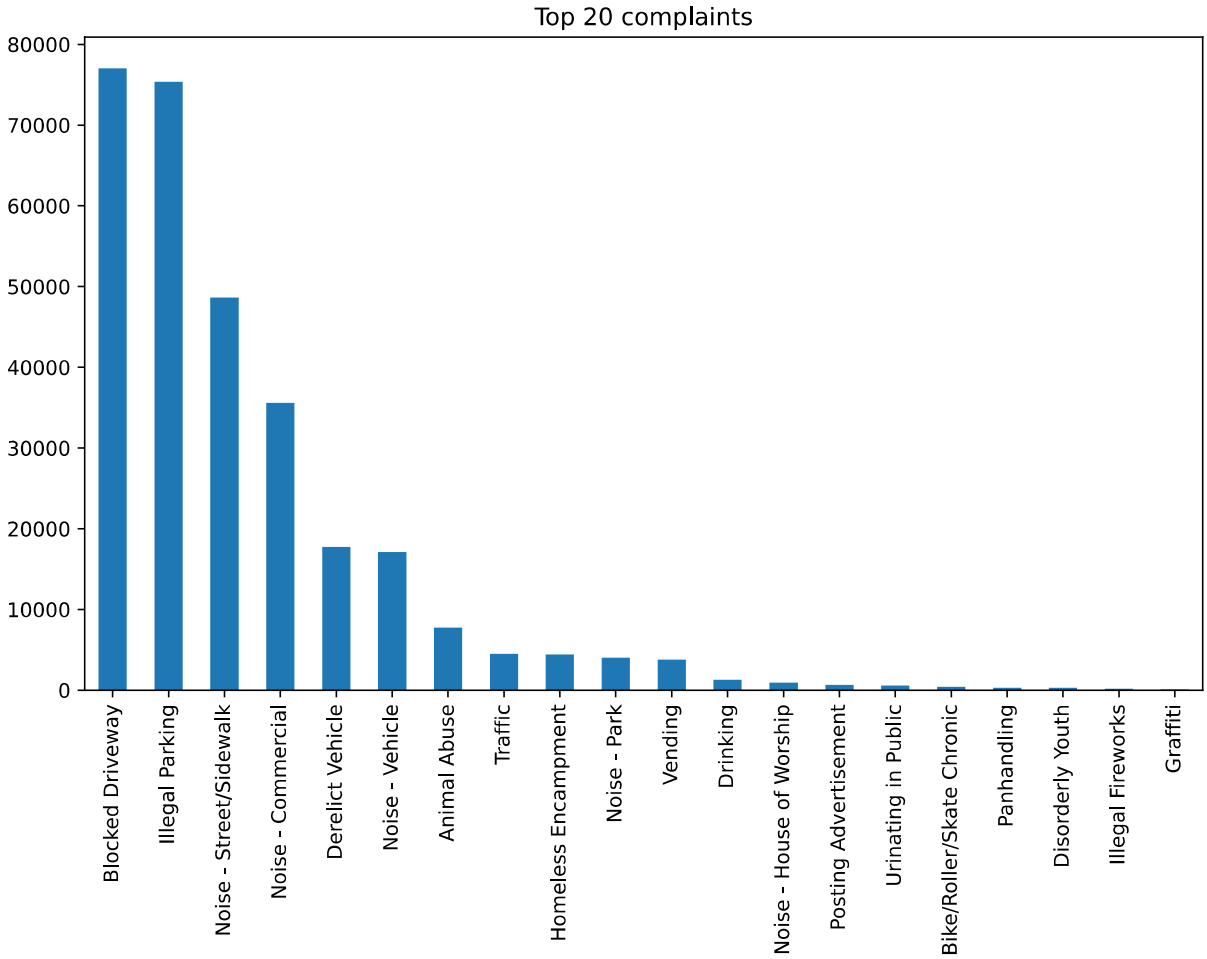
300698 rows × 53 columns

#

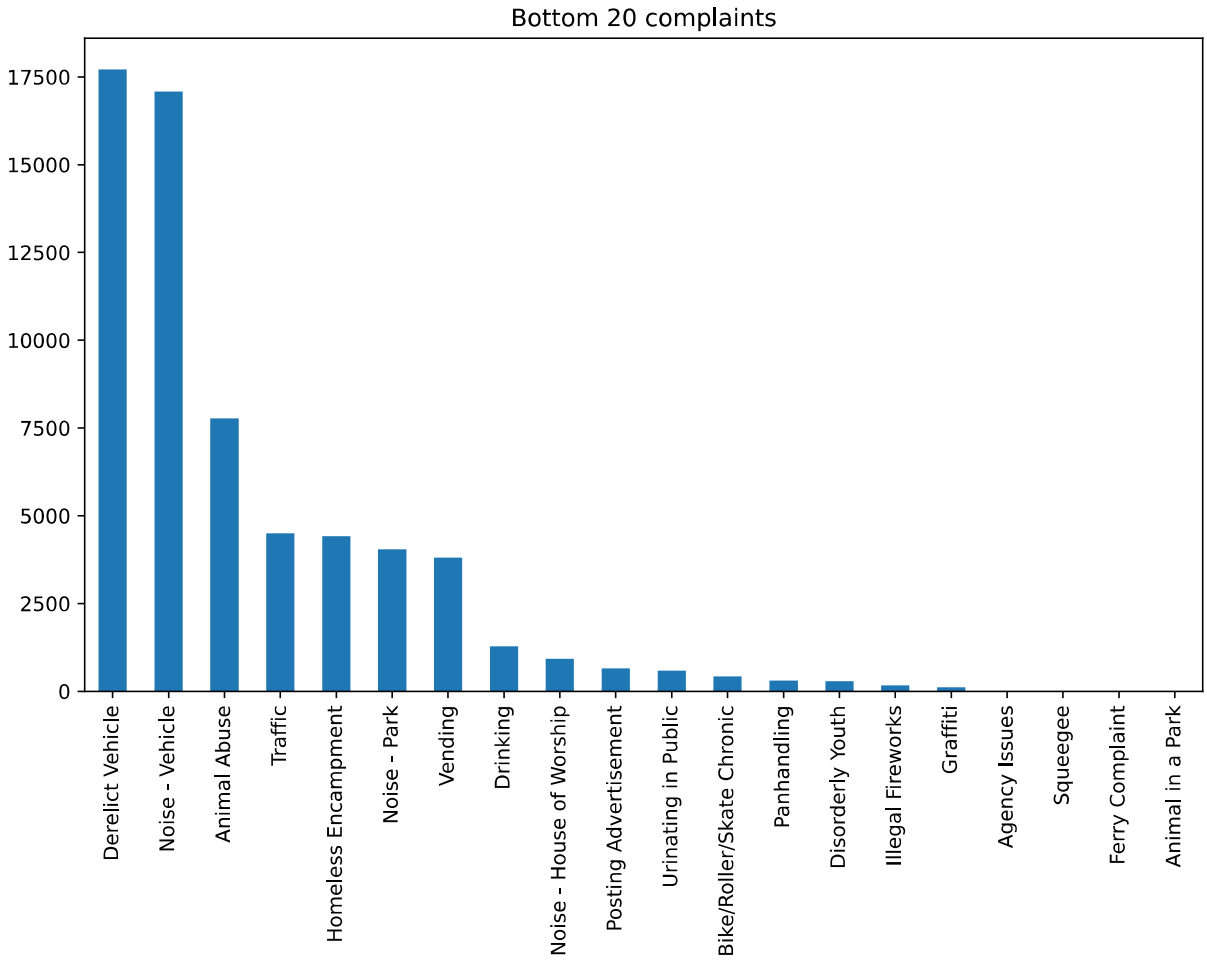
1. Provide major insights/patterns that you can offer in a visual format (graphs or tables); at least 4 major conclusions that you can come up with after generic data mining.

#

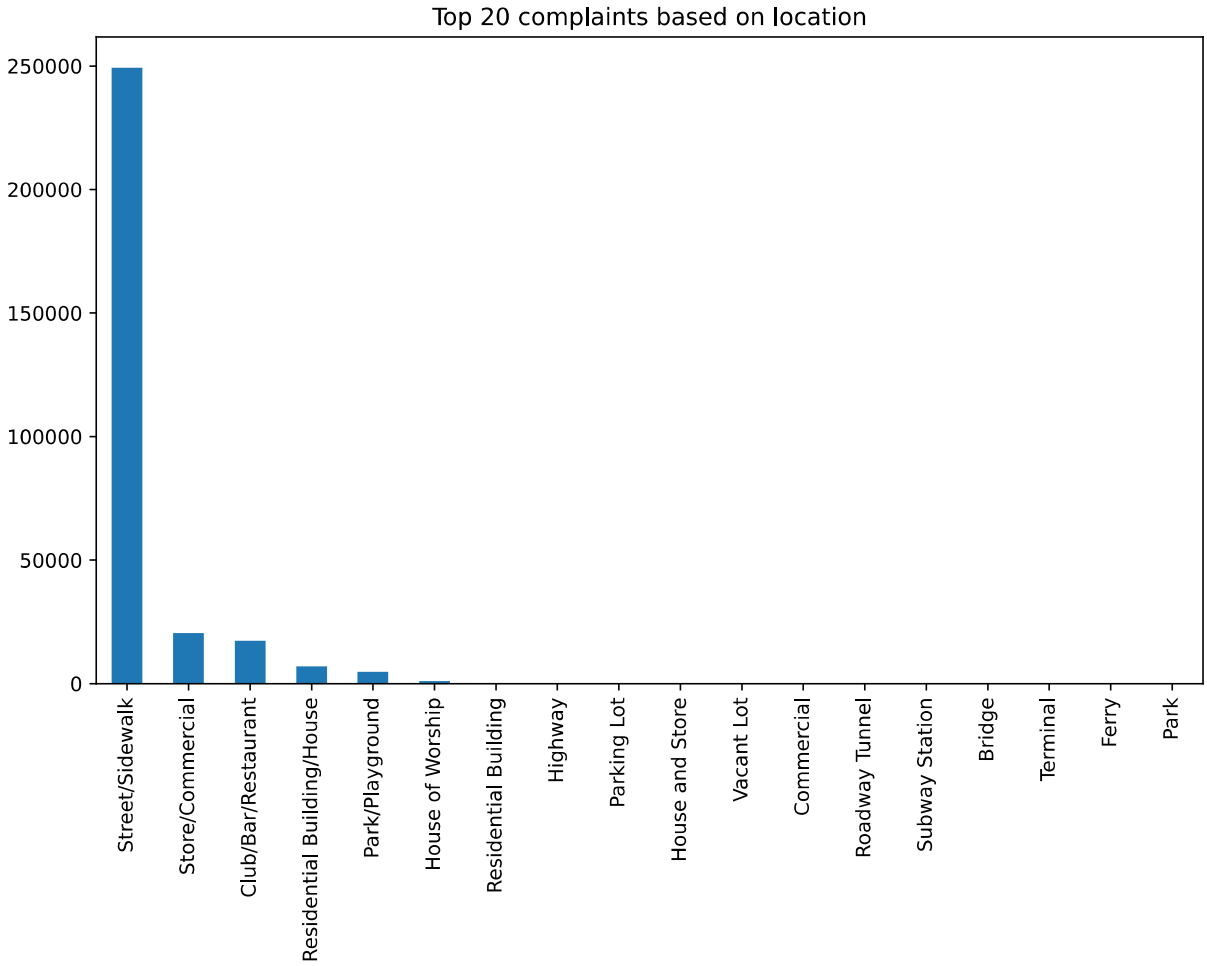
In [6]:
```
_=(Nyc311DataFrame["Complaint Type"].value_counts()).head(20).plot(kind="bar",figsiz
```



In [7]:
```
_=(Nyc311DataFrame["Complaint Type"].value_counts()).tail(20).plot(kind="bar",figsiz
```

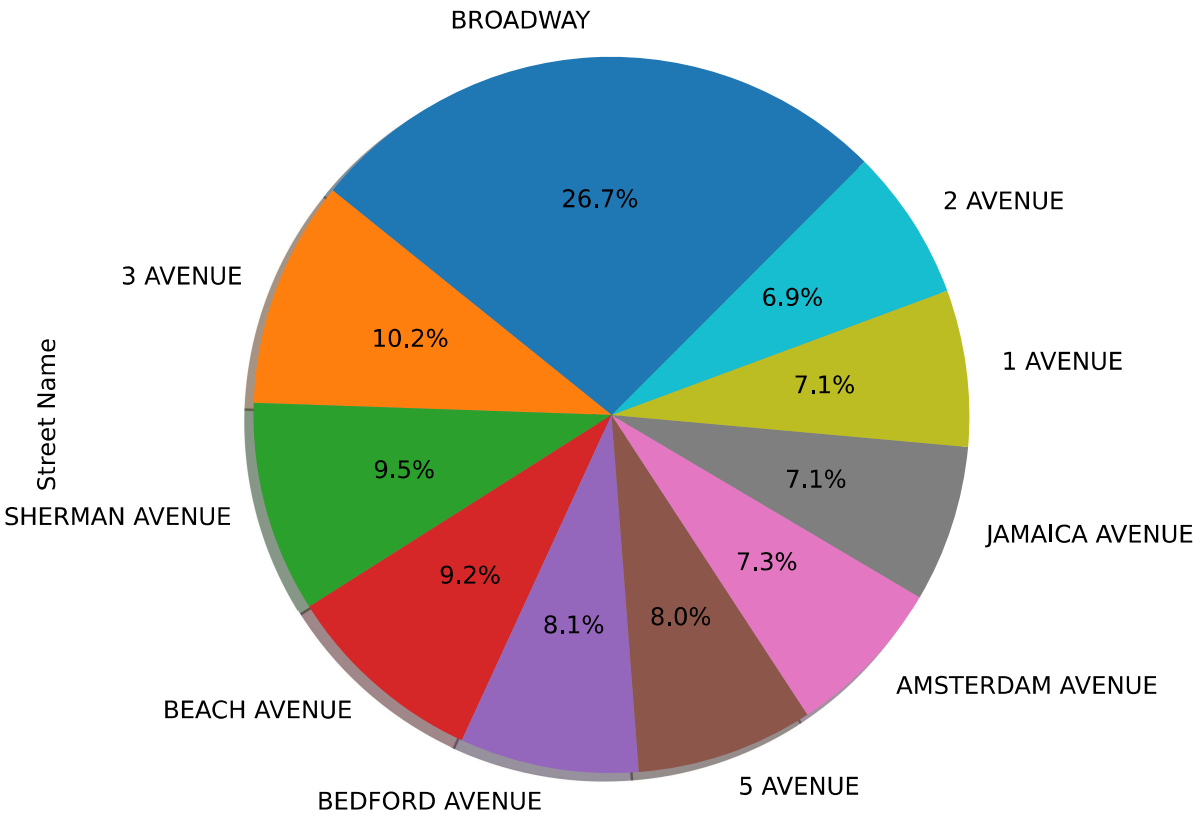


In [8]:
```
_=(Nyc311DataFrame["Location Type"].value_counts()).head(20).plot(kind="bar",figsize
```



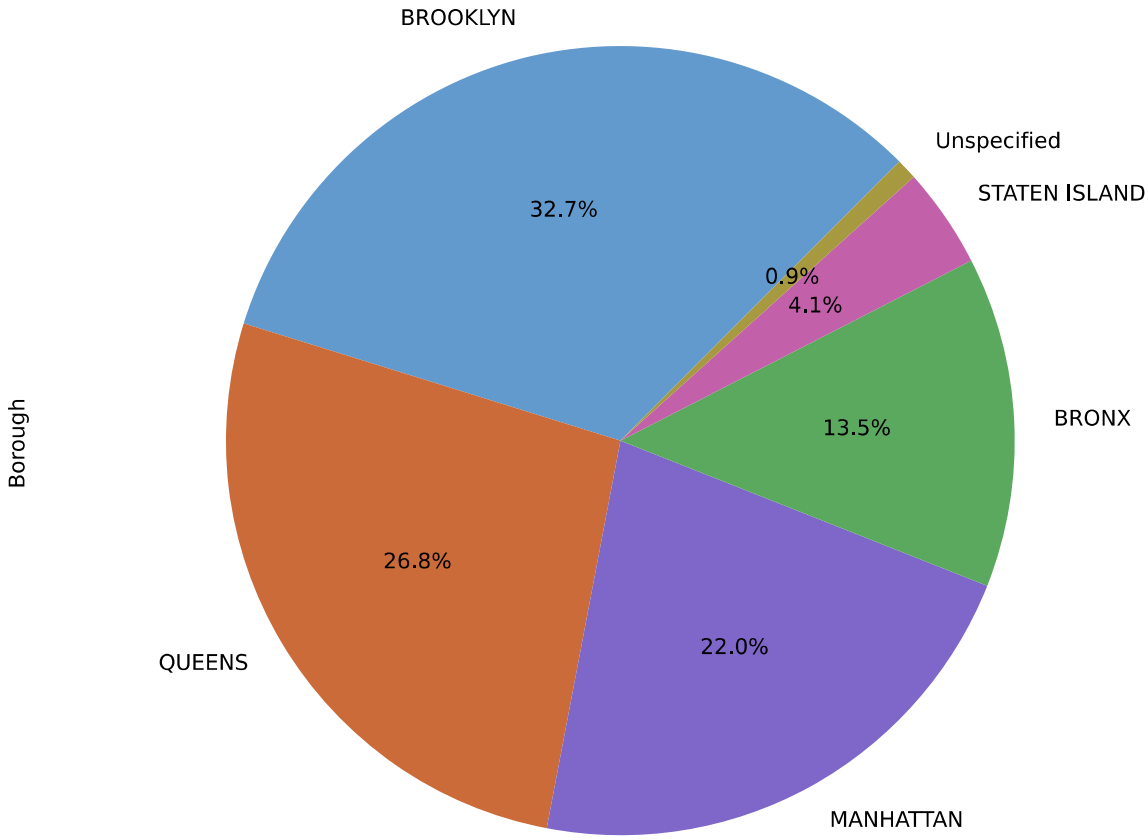
In [16]:
```
#_=(Nyc311DataFrame["Street Name"].value_counts()).head(10).plot(kind='pie',figsize

(Nyc311DataFrame["Street Name"].value_counts()).head(10).plot(kind='pie',autopct='%1
plt.title ="Complaint percentage based on street"
#plt.figure(figsize=(16,8))
plt.axis('equal')
#plt.legend(Nyc311DataFrame["Street Name"], loc="upper right")
plt.show()
```



In [13]:

```
colors = ['#639ace','#ca6b39','#7f67ca','#5ba85f','#c360aa','#a7993f','#cc566a']
Nyc311DataFrame['Borough'].value_counts().plot(kind='pie',autopct='%1.1f%%',
                          startangle=45, shadow=False, colors = colors,
                         figsize = (8,6))
#plt.legend(title='BOROUGH', loc='upper right', bbox_to_anchor=(1.5,1))
plt.axis('equal')
plt.title='# complaints distribution across Boroughs\n'
plt.tight_layout()
plt.show()
```



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

In [26]:
```
GroupByCityandTypeDataFrame = Nyc311DataFrame.groupby(['City','Complaint Type']).Req
#dfGroupByCityandType.sort_values('Complaint Type',ascending=False)
GroupByCityandTypeDataFrame.head(20)
```

Out[26]:
```
City              Complaint Type
MASPETH           Noise - Commercial        -2014 days +00:18:29.908400448
RICHMOND HILL     Derelict Vehicle           -639 days +03:55:52.162546256
JACKSON HEIGHTS   Illegal Parking            -581 days +23:27:26.979050128
Astoria           Noise - Commercial         -408 days +16:43:12.073073376
STATEN ISLAND     Noise - Street/Sidewalk    -391 days +02:10:03.844243800
                  Posting Advertisement      -207 days +04:19:48.573924852
                  Noise - Commercial         -158 days +16:09:52.433842512
LONG ISLAND CITY  Illegal Parking            -134 days +00:45:21.987604054
RICHMOND HILL     Blocked Driveway           -123 days +18:55:05.987551862
EAST ELMHURST     Illegal Parking            -122 days +09:59:49.384430130
JAMAICA           Derelict Vehicle           -112 days +10:21:12.697217216
NEW YORK          Blocked Driveway           -103 days +02:32:10.195603498
WOODHAVEN         Blocked Driveway           -101 days +12:29:29.535986061
NEW YORK          Noise - Park                -89 days +23:17:45.820103405
                  Traffic                     -69 days +04:38:47.756065348
RIDGEWOOD         Blocked Driveway            -63 days +03:34:56.825941691
BRONX             Derelict Vehicle            -55 days +17:22:13.523883883
BROOKLYN          Noise - Commercial          -47 days +13:27:19.581237558
NEW YORK          Vending                     -45 days +15:20:23.382303137
                  Noise - Commercial          -44 days +02:13:09.573805591
Name: Request_Closing_Time, dtype: timedelta64[ns]
```

1. 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?

In [27]:
```
AvgResponseTimeDF = Nyc311DataFrame.groupby(['Complaint Type']).Request_Closing_Time
#dfGroupByCityandType.sort_values('Complaint Type',ascending=False)
AvgResponseTimeDF.head(20)
```

Out[27]:
```
Complaint Type
Noise - Street/Sidewalk     -1177 days +02:03:10.339419072
Illegal Parking             -1175 days +20:56:24.201594864
Noise - Commercial           -991 days +22:26:41.994179504
Derelict Vehicle             -783 days +01:07:36.358837296
Bike/Roller/Skate Chronic    -750 days +03:24:15.310153800
Panhandling                  -696 days +17:28:45.163812536
Noise - Park                    -529 days +22:15:53.176869
Drinking                     -417 days +03:50:51.272442284
Posting Advertisement        -329 days +14:44:44.738908384
Blocked Driveway             -325 days +23:12:34.845438740
Noise - Vehicle              -313 days +16:45:20.583402284
Noise - House of Worship     -230 days +19:19:27.769377496
Vending                      -197 days +14:55:38.952923874
Animal Abuse                 -138 days +23:14:21.166424820
Traffic                       -48 days +16:15:10.560091251
Illegal Fireworks                0 days 02:45:40.101190476
Disorderly Youth                 0 days 03:33:30.902097902
Urinating in Public              0 days 03:37:35.991554054
Squeegee                            0 days 04:02:44.250000
Homeless Encampment              0 days 04:21:56.052536231
Name: Request_Closing_Time, dtype: timedelta64[ns]
```

From the above data null hypothesis can be rejected