

In [1]:

```
#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'.

#1.Whether the average response time across complaint types is similar or not (overall)
#2.Are the type of complaint or service requested and location related?
```

In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
data=pd.read_csv('~/.work/datasets/311_Service_Requests_from_2010_to_Present.csv')
data.head(2)
```

Out[2]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location T
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/Sidev
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/Sidev

2 rows × 53 columns

In [3]:

```
data.rename({'Created Date':'Created_date','Closed Date':'Closed_date','Complaint Type':'Complaint_type','Location Type':'Location_type'},axis=1,inplace=True)
data.head(1)
```

Out[3]:

	Unique Key	Created_date	Closed_date	Agency	Agency Name	Complaint_type	Descriptor	L
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	St

1 rows × 53 columns

In [4]:

```
x=pd.to_datetime(data.Created_date)
y=pd.to_datetime(data.Closed_date)
```

In [5]:

```
x
```

Out[5]:

```
0      2015-12-31 23:59:45
1      2015-12-31 23:59:44
2      2015-12-31 23:59:29
3      2015-12-31 23:57:46
4      2015-12-31 23:56:58
5      2015-12-31 23:56:30
6      2015-12-31 23:55:32
7      2015-12-31 23:54:05
8      2015-12-31 23:53:58
9      2015-12-31 23:53:58
10     2015-12-31 23:52:58
11     2015-12-31 23:50:57
12     2015-12-31 23:48:03
13     2015-12-31 23:47:58
14     2015-12-31 23:47:37
15     2015-12-31 23:47:30
16     2015-12-31 23:47:02
17     2015-12-31 23:44:52
18     2015-12-31 23:40:59
19     2015-12-31 23:40:55
20     2015-12-31 23:40:43
21     2015-12-31 23:38:51
22     2015-12-31 23:34:18
23     2015-12-31 23:32:46
24     2015-12-31 23:31:40
25     2015-12-31 23:30:28
26     2015-12-31 23:26:41
27     2015-12-31 23:26:35
28     2015-12-31 23:25:56
29     2015-12-31 23:25:01
...
1637   2015-12-30 06:09:52
1638   2015-12-30 06:08:21
1639   2015-12-30 06:06:18
1640   2015-12-30 06:05:02
1641   2015-12-30 06:01:07
1642   2015-12-30 05:59:17
1643   2015-12-30 05:57:14
1644   2015-12-30 05:53:54
1645   2015-12-30 05:41:08
1646   2015-12-30 05:35:20
1647   2015-12-30 05:25:57
1648   2015-12-30 05:13:42
1649   2015-12-30 05:12:57
1650   2015-12-30 04:59:16
1651   2015-12-30 04:59:16
1652   2015-12-30 04:57:55
1653   2015-12-30 04:57:27
1654   2015-12-30 04:49:43
1655   2015-12-30 04:39:30
1656   2015-12-30 04:38:49
1657   2015-12-30 04:34:11
1658   2015-12-30 04:26:19
1659   2015-12-30 04:16:28
1660   2015-12-30 04:08:11
1661   2015-12-30 04:03:57
1662   2015-12-30 04:02:30
1663   2015-12-30 03:51:43
1664   2015-12-30 03:49:33
```

```
1665    2015-12-30 03:34:52
```

```
1666    2015-12-30 03:34:39
```

```
Name: Created_date, Length: 1667, dtype: datetime64[ns]
```

In [6]:

```
y
```

Out[6]:

```

0      2016-01-01 00:55:00
1      2016-01-01 01:26:00
2      2016-01-01 04:51:00
3      2016-01-01 07:43:00
4      2016-01-01 03:24:00
5      2016-01-01 01:50:00
6      2016-01-01 01:53:00
7      2016-01-01 01:42:00
8      2016-01-01 08:27:00
9      2016-01-01 01:17:00
10     2016-01-01 07:41:00
11     2016-01-01 10:58:00
12     2016-01-01 02:17:00
13     2016-01-01 08:18:00
14     2016-01-01 10:17:00
15     2016-01-01 15:20:00
16     2016-01-01 04:39:00
17     2016-01-01 00:36:00
18     2016-01-01 02:37:00
19     2016-01-01 00:28:00
20     2016-01-01 04:12:00
21     2016-01-01 09:11:00
22     2016-01-01 00:50:00
23     2016-01-01 00:25:00
24     2016-01-03 16:22:00
25     2016-01-01 02:47:00
26     2015-12-31 23:53:31
27     2016-01-01 05:07:00
28     2016-01-01 01:12:00
29     2016-01-01 01:55:00
...
1637   2015-12-30 06:58:18
1638   2015-12-31 00:00:34
1639   2015-12-30 06:48:10
1640   2015-12-30 07:52:05
1641   2015-12-30 07:46:31
1642   2015-12-30 08:05:57
1643   2015-12-30 08:24:48
1644   2015-12-30 19:58:18
1645   2015-12-30 07:43:26
1646   2015-12-30 06:46:04
1647   2015-12-30 06:23:29
1648                                     NaT
1649   2015-12-30 05:37:48
1650   2015-12-30 07:47:07
1651   2015-12-30 07:52:08
1652   2015-12-31 17:05:55
1653   2015-12-30 11:54:10
1654   2015-12-30 05:16:22
1655   2015-12-30 06:57:38
1656   2015-12-30 17:43:33
1657   2015-12-30 07:43:52
1658   2015-12-30 04:51:43
1659   2015-12-30 05:35:38
1660   2015-12-30 05:09:50
1661   2015-12-30 15:40:05
1662   2015-12-31 17:05:54
1663   2015-12-30 06:35:05
1664   2015-12-30 04:44:27

```

```
1665    2015-12-30 05:09:53
```

```
1666    2015-12-31 11:13:47
```

```
Name: Closed_date, Length: 1667, dtype: datetime64[ns]
```


In [7]:

```
z=y-x  
z
```

Out[7]:

```

0      0 days 00:55:15
1      0 days 01:26:16
2      0 days 04:51:31
3      0 days 07:45:14
4      0 days 03:27:02
5      0 days 01:53:30
6      0 days 01:57:28
7      0 days 01:47:55
8      0 days 08:33:02
9      0 days 01:23:02
10     0 days 07:48:02
11     0 days 11:07:03
12     0 days 02:28:57
13     0 days 08:30:02
14     0 days 10:29:23
15     0 days 15:32:30
16     0 days 04:51:58
17     0 days 00:51:08
18     0 days 02:56:01
19     0 days 00:47:05
20     0 days 04:31:17
21     0 days 09:32:09
22     0 days 01:15:42
23     0 days 00:52:14
24     2 days 16:50:20
25     0 days 03:16:32
26     0 days 00:26:50
27     0 days 05:40:25
28     0 days 01:46:04
29     0 days 02:29:59
...
1637   0 days 00:48:26
1638   0 days 17:52:13
1639   0 days 00:41:52
1640   0 days 01:47:03
1641   0 days 01:45:24
1642   0 days 02:06:40
1643   0 days 02:27:34
1644   0 days 14:04:24
1645   0 days 02:02:18
1646   0 days 01:10:44
1647   0 days 00:57:32
1648           NaT
1649   0 days 00:24:51
1650   0 days 02:47:51
1651   0 days 02:52:52
1652   1 days 12:08:00
1653   0 days 06:56:43
1654   0 days 00:26:39
1655   0 days 02:18:08
1656   0 days 13:04:44
1657   0 days 03:09:41
1658   0 days 00:25:24
1659   0 days 01:19:10
1660   0 days 01:01:39
1661   0 days 11:36:08
1662   1 days 13:03:24
1663   0 days 02:43:22
1664   0 days 00:54:54

```

```
1665    0 days 01:35:01
1666    1 days 07:39:08
Length: 1667, dtype: timedelta64[ns]
```

In [8]:

```
data.insert(3, 'Request_closing_Time', z)
```

In [9]:

```
data.head(2)
```

Out[9]:

	Unique Key	Created_date	Closed_date	Request_closing_Time	Agency	Agency Name	Complai
0	32310363	12/31/2015 11:59:45 PM	01-01-16 0:55	00:55:15	NYPD	New York City Police Department	Street/S
1	32309934	12/31/2015 11:59:44 PM	01-01-16 1:26	01:26:16	NYPD	New York City Police Department	I D

2 rows x 54 columns

In [10]:

```
p=data.Request_closing_Time.dt.total_seconds()
data.insert(4, 'Total_Seconds', p)
```

In [11]:

```
data.head(2)
```

Out[11]:

	Unique Key	Created_date	Closed_date	Request_closing_Time	Total_Seconds	Agency	A
0	32310363	12/31/2015 11:59:45 PM	01-01-16 0:55	00:55:15	3315.0	NYPD	Ne City Depa
1	32309934	12/31/2015 11:59:44 PM	01-01-16 1:26	01:26:16	5176.0	NYPD	Ne City Depa

2 rows x 55 columns

In [12]:

```
pd.crosstab(data.City,data.Descriptor).style.background_gradient(cmap = 'Blues')
```

Out[12]:

Descriptor	After Hours - Licensed Est	Banging/Pounding	Blocked Hydrant	Blocked Sidewalk	Car/Truck Horn	Car/Truck Music	Chained
City							
ASTORIA	0	0	1	2	0	1	0
BAYSIDE	0	0	0	0	0	0	0
BELLEROSE	0	0	0	0	0	0	0
BRONX	0	2	13	7	2	0	0
BROOKLYN	2	12	50	27	5	6	0
CAMBRIA HEIGHTS	0	0	0	0	0	0	0
COLLEGE POINT	0	0	0	0	0	0	0
CORONA	0	1	1	2	0	0	0
EAST ELMHURST	0	0	0	1	0	0	0
ELMHURST	0	0	0	2	0	0	0
FAR ROCKAWAY	0	0	0	0	0	0	0
FLUSHING	0	1	2	0	0	0	0
FOREST HILLS	0	0	0	1	0	0	0
FRESH MEADOWS	0	0	1	0	0	1	0
GLEN OAKS	0	0	1	0	0	0	0
HOLLIS	0	0	0	0	0	0	0
HOWARD BEACH	0	0	1	1	0	0	0
JACKSON HEIGHTS	0	1	0	0	0	0	0
JAMAICA	0	0	1	1	0	0	0
KEW GARDENS	0	0	0	0	0	0	0
LITTLE NECK	0	0	0	0	0	0	0
LONG ISLAND CITY	0	0	0	0	1	0	0
MASPETH	0	1	1	1	0	0	0
MIDDLE VILLAGE	0	0	2	1	0	0	0
NEW YORK	0	2	6	10	6	7	0
OAKLAND GARDENS	0	0	0	1	0	0	0

Descriptor	After Hours - Licensed Est	Banging/Pounding	Blocked Hydrant	Blocked Sidewalk	Car/Truck Horn	Car/Truck Music	Chained
City							
OZONE PARK	0	0	1	1	0	0	0
QUEENS VILLAGE	0	0	1	0	0	2	0
REGO PARK	0	0	0	0	0	0	0
RICHMOND HILL	0	0	1	1	0	0	0
RIDGEWOOD	0	0	14	2	0	0	0
ROCKAWAY PARK	0	0	0	0	0	0	0
ROSEDALE	0	0	1	0	0	0	0
SAINT ALBANS	0	0	0	0	0	0	0
SOUTH OZONE PARK	0	0	1	2	0	1	0
SOUTH RICHMOND HILL	0	3	1	0	0	0	0
SPRINGFIELD GARDENS	0	0	0	0	0	0	0
STATEN ISLAND	1	0	2	6	0	0	0
SUNNYSIDE	0	0	0	0	0	0	0
WOODHAVEN	0	0	2	1	1	0	0
WOODSIDE	0	0	1	1	0	0	0

In [16]:

```
#Brooklyn is the city with the most reported crimes.
```

In [13]:

```
g=data.groupby(['Total_Seconds','City'])  
g.first()
```

Out[13]:

		Unique Key	Created_date	Closed_date	Request_closing_Time	Ag
Total_Seconds	City					
209.0	NEW YORK	32306100	12/31/2015 10:14:17 AM	12/31/2015 10:17:46 AM	0 days 00:03:29	↑
218.0	NEW YORK	32309005	12/31/2015 11:25:13 AM	12/31/2015 11:28:51 AM	0 days 00:03:38	↑
230.0	RIDGEWOOD	32307296	12/31/2015 06:54:39 PM	12/31/2015 06:58:29 PM	0 days 00:03:50	↑
241.0	BROOKLYN	32308746	12/31/2015 11:10:03 PM	12/31/2015 11:14:04 PM	0 days 00:04:01	↑
251.0	BROOKLYN	32306603	12/31/2015 09:34:42 PM	12/31/2015 09:38:53 PM	0 days 00:04:11	↑
253.0	NEW YORK	32304230	12/30/2015 10:28:19 PM	12/30/2015 10:32:32 PM	0 days 00:04:13	↑
265.0	BROOKLYN	32310164	12/31/2015 07:45:00 PM	12/31/2015 07:49:25 PM	0 days 00:04:25	↑
273.0	NEW YORK	32307932	12/31/2015 10:21:05 PM	12/31/2015 10:25:38 PM	0 days 00:04:33	↑
279.0	BROOKLYN	32308024	12/31/2015 11:09:25 PM	12/31/2015 11:14:04 PM	0 days 00:04:39	↑
287.0	NEW YORK	32306903	12/31/2015 10:54:50 AM	12/31/2015 10:59:37 AM	0 days 00:04:47	↑
295.0	NEW YORK	32307354	12/31/2015 11:11:18 PM	12/31/2015 11:16:13 PM	0 days 00:04:55	↑
317.0	RICHMOND HILL	32307708	12/31/2015 05:27:07 PM	12/31/2015 05:32:24 PM	0 days 00:05:17	↑
330.0	NEW YORK	32309006	12/31/2015 11:23:22 AM	12/31/2015 11:28:52 AM	0 days 00:05:30	↑
353.0	NEW YORK	32309431	12/31/2015 10:18:33 PM	12/31/2015 10:24:26 PM	0 days 00:05:53	↑
365.0	NEW YORK	32307316	12/31/2015 05:51:30 AM	12/31/2015 05:57:35 AM	0 days 00:06:05	↑
382.0	NEW YORK	32299374	12/30/2015 10:59:48 AM	12/30/2015 11:06:10 AM	0 days 00:06:22	↑

		Unique Key	Created_date	Closed_date	Request_closing_Time	Ag
Total_Seconds	City					
404.0	NEW YORK	32306133	12/31/2015 10:12:07 AM	12/31/2015 10:18:51 AM	0 days 00:06:44	↑
409.0	BROOKLYN	32308662	12/31/2015 10:45:29 PM	12/31/2015 10:52:18 PM	0 days 00:06:49	↑
415.0	BROOKLYN	32306587	12/31/2015 10:48:38 PM	12/31/2015 10:55:33 PM	0 days 00:06:55	↑
418.0	NEW YORK	32306615	12/31/2015 10:24:58 PM	12/31/2015 10:31:56 PM	0 days 00:06:58	↑
428.0	BROOKLYN	32305555	12/31/2015 09:31:44 PM	12/31/2015 09:38:52 PM	0 days 00:07:08	↑
433.0	BAYSIDE	32306482	12/31/2015 10:20:55 AM	12/31/2015 10:28:08 AM	0 days 00:07:13	↑
444.0	NEW YORK	32306889	12/31/2015 11:12:26 AM	12/31/2015 11:19:50 AM	0 days 00:07:24	↑
450.0	BROOKLYN	32310586	12/31/2015 08:40:13 PM	12/31/2015 08:47:43 PM	0 days 00:07:30	↑
453.0	NEW YORK	32308118	12/31/2015 11:08:41 PM	12/31/2015 11:16:14 PM	0 days 00:07:33	↑
459.0	NEW YORK	32302643	12/30/2015 03:23:28 PM	12/30/2015 03:31:07 PM	0 days 00:07:39	↑
	RIDGEWOOD	32307210	12/31/2015 05:42:38 PM	12/31/2015 05:50:17 PM	0 days 00:07:39	↑
462.0	BRONX	32299529	12/31/2015 12:27:16 AM	12/31/2015 12:34:58 AM	0 days 00:07:42	↑
475.0	NEW YORK	32299517	12/31/2015 01:00:01 AM	12/31/2015 01:07:56 AM	0 days 00:07:55	↑
483.0	NEW YORK	32310200	12/31/2015 10:58:24 PM	12/31/2015 11:06:27 PM	0 days 00:08:03	↑
...	
60828.0	NEW YORK	32309521	12/31/2015 10:50:12 PM	01-01-16 15:44	0 days 16:53:48	↑
64333.0	BROOKLYN	32304725	12/30/2015 06:08:21 AM	12/31/2015 12:00:34 AM	0 days 17:52:13	↑
64490.0	JACKSON HEIGHTS	32307040	12/31/2015 12:38:10 PM	01-01-16 6:33	0 days 17:54:50	↑

		Unique Key	Created_date	Closed_date	Request_closing_Time	Ag
Total_Seconds	City					
66667.0	BRONX	32302281	12/30/2015 07:11:10 PM	12/31/2015 01:42:17 PM	0 days 18:31:07	↑
68116.0	BRONX	32304128	12/30/2015 03:32:55 PM	12/31/2015 10:28:11 AM	0 days 18:55:16	↑
69075.0	BRONX	32299414	12/30/2015 08:22:49 PM	12/31/2015 03:34:04 PM	0 days 19:11:15	↑
69239.0	OZONE PARK	32304753	12/30/2015 04:46:54 PM	12/31/2015 12:00:53 PM	0 days 19:13:59	↑
69758.0	OZONE PARK	32299036	12/30/2015 03:34:40 PM	12/31/2015 10:57:18 AM	0 days 19:22:38	↑
70439.0	BROOKLYN	32300050	12/30/2015 09:09:26 PM	12/31/2015 04:43:25 PM	0 days 19:33:59	↑
71186.0	BRONX	32305576	12/31/2015 07:07:34 AM	01-01-16 2:54	0 days 19:46:26	↑
72613.0	EAST ELMHURST	32302286	12/30/2015 02:58:47 PM	12/31/2015 11:09:00 AM	0 days 20:10:13	↑
74076.0	BROOKLYN	32301335	12/30/2015 07:09:13 PM	12/31/2015 03:43:49 PM	0 days 20:34:36	↑
83293.0	NEW YORK	32300225	12/30/2015 11:06:37 PM	12/31/2015 10:14:50 PM	0 days 23:08:13	↑
84401.0	BROOKLYN	32297974	12/30/2015 05:09:17 PM	12/31/2015 04:35:58 PM	0 days 23:26:41	↑
89396.0	BRONX	32298498	12/30/2015 09:27:42 AM	12/31/2015 10:17:38 AM	1 days 00:49:56	↑
90177.0	BRONX	32303420	12/30/2015 09:39:27 AM	12/31/2015 10:42:24 AM	1 days 01:02:57	↑
90787.0	BROOKLYN	32298512	12/30/2015 03:22:50 PM	12/31/2015 04:35:57 PM	1 days 01:13:07	↑
91269.0	BRONX	32301881	12/30/2015 08:55:25 AM	12/31/2015 10:16:34 AM	1 days 01:21:09	↑
92302.0	EAST ELMHURST	32303317	12/30/2015 02:20:15 PM	12/31/2015 03:58:37 PM	1 days 01:38:22	↑
92643.0	NEW YORK	32303265	12/30/2015 03:22:58 PM	12/31/2015 05:07:01 PM	1 days 01:44:03	↑

		Unique Key	Created_date	Closed_date	Request_closing_Time	Ag
Total_Seconds	City					
99164.0	BRONX	32304750	12/30/2015 10:09:35 AM	12/31/2015 01:42:19 PM	1 days 03:32:44	↑
103166.0	BROOKLYN	32302748	12/30/2015 11:56:29 AM	12/31/2015 04:35:55 PM	1 days 04:39:26	↑
104533.0	BROOKLYN	32299497	12/30/2015 11:33:42 AM	12/31/2015 04:35:55 PM	1 days 05:02:13	↑
106516.0	BRONX	32301698	12/30/2015 08:08:06 AM	12/31/2015 01:43:22 PM	1 days 05:35:16	↑
112160.0	NEW YORK	32300432	12/30/2015 09:56:38 AM	12/31/2015 05:05:58 PM	1 days 07:09:20	↑
112448.0	NEW YORK	32300180	12/30/2015 09:51:49 AM	12/31/2015 05:05:57 PM	1 days 07:14:08	↑
113948.0	BRONX	32299842	12/30/2015 03:34:39 AM	12/31/2015 11:13:47 AM	1 days 07:39:08	↑
130080.0	NEW YORK	32297995	12/30/2015 04:57:55 AM	12/31/2015 05:05:55 PM	1 days 12:08:00	↑
133404.0	NEW YORK	32301730	12/30/2015 04:02:30 AM	12/31/2015 05:05:54 PM	1 days 13:03:24	↑
233420.0	BRONX	32308423	12/31/2015 11:31:40 PM	01-03-16 16:22	2 days 16:50:20	↑

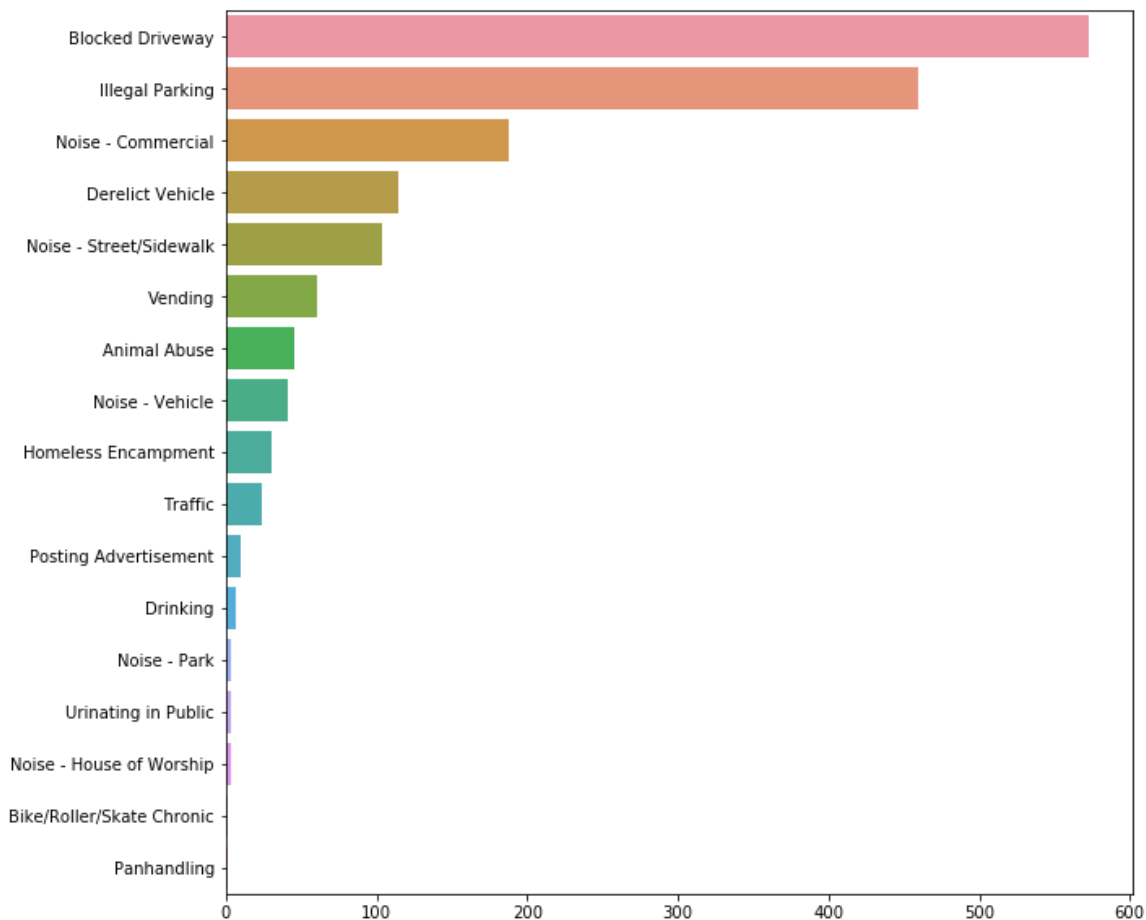
1653 rows × 53 columns

In [22]:

```
#the lowest time taken to solve an issue is 209 secs for newyork city and vending(in Prohibited area)-Complaint_type
#the highest time taken to solve an issue is 233420 secs which is for Bronx City and Blocked Driveway(no Access)-Complaint_type
```

In [14]:

```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(10,10))
d=data.Complaint_type.value_counts()
fig=sns.barplot(y=d.index,x=d.values)
plt.show()
```

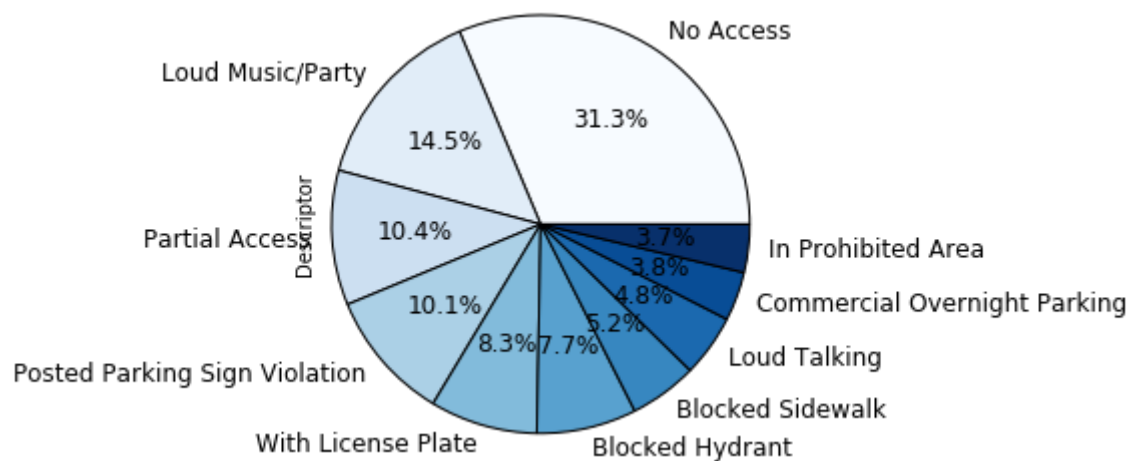


In []:

```
#we have received Blocked Driveway Complaints the most.
```

In [15]:

```
data.Descriptor.value_counts()[ :10].plot.pie(radius=1.2,
                                              fontsize=12,autopct='% .1f%%',cmap='Blues',wedgeprops={'Edgecolor':'Black'})
plt.show()
```



In []:

```
#if we take the first 10 descriptors we can see that people are complaining the most about "blocked Drive away" with "no Access"
```

In [16]:

```
g=pd.crosstab([data.Descriptor,data.Complaint_type],data.City[:30])[:10]
g
```

Out[16]:

	City	ASTORIA	BRONX	BROOKLYN	ELMHURST	JACKSON HEIGHTS	JAMAICA
Descriptor	Complaint_type						
Blocked Hydrant	Illegal Parking	0	0	0	0	0	
Blocked Sidewalk	Illegal Parking	0	0	0	1	0	
Commercial Overnight Parking	Illegal Parking	0	1	0	0	0	
Double Parked Blocking Vehicle	Illegal Parking	0	0	0	0	0	
Loud Music/Party	Noise - Commercial	0	0	4	0	0	
	Noise - House of Worship	0	0	0	0	0	
	Noise - Street/Sidewalk	0	1	0	0	0	
No Access	Blocked Driveway	1	4	1	0	1	
Partial Access	Blocked Driveway	0	0	0	0	0	
Posted Parking Sign Violation	Illegal Parking	0	0	2	0	0	

In [21]:

```
#if we take a sample of the 30 cities along with first 10 complaint_types and De
scriptor, we can see the most reported event is
#No Access-Blocked Driveway and Loud Music/Party -Noise-Commercial
```

In [17]:

```
f=data.Total_Seconds.mean()  
f
```

Out[17]:

14679.462139423076

In [18]:

```
Order=data.sort_values(['Complaint_type','Request_closing_Time'],ascending=[False,True])  
Order
```


Out[18]:

	Unique Key	Created_date	Closed_date	Request_closing_Time	Total_Seconds	Agency	
596	32306100	12/31/2015 10:14:17 AM	12/31/2015 10:17:46 AM	00:03:29	209.0	NYPD	(D
546	32309005	12/31/2015 11:25:13 AM	12/31/2015 11:28:51 AM	00:03:38	218.0	NYPD	(D
572	32306903	12/31/2015 10:54:50 AM	12/31/2015 10:59:37 AM	00:04:47	287.0	NYPD	(D
547	32309006	12/31/2015 11:23:22 AM	12/31/2015 11:28:52 AM	00:05:30	330.0	NYPD	(D
599	32306133	12/31/2015 10:12:07 AM	12/31/2015 10:18:51 AM	00:06:44	404.0	NYPD	(D
557	32306889	12/31/2015 11:12:26 AM	12/31/2015 11:19:50 AM	00:07:24	444.0	NYPD	(D
1246	32302643	12/30/2015 03:23:28 PM	12/30/2015 03:31:07 PM	00:07:39	459.0	NYPD	(D
571	32309746	12/31/2015 10:58:40 AM	12/31/2015 11:07:48 AM	00:09:08	548.0	NYPD	(D
561	32310477	12/31/2015 11:10:33 AM	12/31/2015 11:19:51 AM	00:09:18	558.0	NYPD	(D
475	32309808	12/31/2015 12:38:40 PM	12/31/2015 01:08:21 PM	00:29:41	1781.0	NYPD	(D
1332	32298890	12/30/2015 01:35:19 PM	12/30/2015 02:05:21 PM	00:30:02	1802.0	NYPD	(D
393	32309060	12/31/2015 02:48:41 PM	12/31/2015 03:20:35 PM	00:31:54	1914.0	NYPD	(D
616	32309807	12/31/2015 09:56:07 AM	12/31/2015 10:38:52 AM	00:42:45	2565.0	NYPD	(D
1273	32300867	12/30/2015 02:46:08 PM	12/30/2015 03:31:06 PM	00:44:58	2698.0	NYPD	(D
486	32306824	12/31/2015 12:22:30 PM	12/31/2015 01:08:22 PM	00:45:52	2752.0	NYPD	(D
488	32310473	12/31/2015 12:21:07 PM	12/31/2015 01:08:23 PM	00:47:16	2836.0	NYPD	(D
1277	32298019	12/30/2015 02:43:23 PM	12/30/2015 03:31:05 PM	00:47:42	2862.0	NYPD	(D

	Unique Key	Created_date	Closed_date	Request_closing_Time	Total_Seconds	Agency	
491	32310452	12/31/2015 12:19:34 PM	12/31/2015 01:08:24 PM	00:48:50	2930.0	NYPD	(D
1193	32302642	12/30/2015 04:41:38 PM	12/30/2015 05:30:47 PM	00:49:09	2949.0	NYPD	(D
1281	32299912	12/30/2015 02:41:16 PM	12/30/2015 03:31:04 PM	00:49:48	2988.0	NYPD	(D
495	32310491	12/31/2015 12:17:43 PM	12/31/2015 01:08:24 PM	00:50:41	3041.0	NYPD	(D
1282	32303599	12/30/2015 02:39:35 PM	12/30/2015 03:31:02 PM	00:51:27	3087.0	NYPD	(D
1292	32297960	12/30/2015 02:28:42 PM	12/30/2015 03:20:47 PM	00:52:05	3125.0	NYPD	(D
497	32309806	12/31/2015 12:15:29 PM	12/31/2015 01:08:26 PM	00:52:57	3177.0	NYPD	(D
1284	32302640	12/30/2015 02:37:25 PM	12/30/2015 03:31:03 PM	00:53:38	3218.0	NYPD	(D
500	32308969	12/31/2015 12:13:52 PM	12/31/2015 01:08:28 PM	00:54:36	3276.0	NYPD	(D
1288	32304537	12/30/2015 02:30:53 PM	12/30/2015 03:29:59 PM	00:59:06	3546.0	NYPD	(D
1061	32300836	12/30/2015 08:16:17 PM	12/30/2015 09:25:47 PM	01:09:30	4170.0	NYPD	(D
428	32310451	12/31/2015 01:58:41 PM	12/31/2015 03:21:50 PM	01:23:09	4989.0	NYPD	(D
537	32306132	12/31/2015 11:34:14 AM	12/31/2015 01:09:33 PM	01:35:19	5719.0	NYPD	(D
...	
382	32309352	12/31/2015 03:09:34 PM	12/31/2015 04:59:29 PM	01:49:55	6595.0	NYPD	(D
1117	32303159	12/30/2015 06:54:44 PM	12/30/2015 08:47:50 PM	01:53:06	6786.0	NYPD	(D
1510	32300390	12/30/2015 09:46:05 AM	12/30/2015 11:48:04 AM	02:01:59	7319.0	NYPD	(D
164	32310094	12/31/2015 08:50:22 PM	12/31/2015 11:05:17 PM	02:14:55	8095.0	NYPD	(D

	Unique Key	Created_date	Closed_date	Request_closing_Time	Total_Seconds	Agency	
1309	32303198	12/30/2015 02:10:32 PM	12/30/2015 04:39:19 PM	02:28:47	8927.0	NYPD	(D
1213	32301387	12/30/2015 04:16:33 PM	12/30/2015 06:47:02 PM	02:30:29	9029.0	NYPD	(D
391	32306488	12/31/2015 02:51:09 PM	12/31/2015 05:21:52 PM	02:30:43	9043.0	NYPD	(D
189	32307992	12/31/2015 07:54:38 PM	12/31/2015 10:34:06 PM	02:39:28	9568.0	NYPD	(D
1437	32301382	12/30/2015 11:28:53 AM	12/30/2015 02:11:02 PM	02:42:09	9729.0	NYPD	(D
531	32310189	12/31/2015 11:38:58 AM	12/31/2015 02:22:12 PM	02:43:14	9794.0	NYPD	(D
1490	32297549	12/30/2015 10:09:42 AM	12/30/2015 12:54:07 PM	02:44:25	9865.0	NYPD	(D
1460	32300439	12/30/2015 10:59:04 AM	12/30/2015 01:51:11 PM	02:52:07	10327.0	NYPD	(D
485	32306547	12/31/2015 12:23:20 PM	12/31/2015 03:27:22 PM	03:04:02	11042.0	NYPD	(D
570	32308676	12/31/2015 11:00:27 AM	12/31/2015 02:36:13 PM	03:35:46	12946.0	NYPD	(D
1255	32304177	12/30/2015 03:10:40 PM	12/30/2015 06:47:03 PM	03:36:23	12983.0	NYPD	(D
1312	32302273	12/30/2015 02:07:52 PM	12/30/2015 05:49:54 PM	03:42:02	13322.0	NYPD	(D
329	32309359	12/31/2015 04:21:08 PM	12/31/2015 08:03:17 PM	03:42:09	13329.0	NYPD	(D
1484	32301388	12/30/2015 10:17:40 AM	12/30/2015 02:05:20 PM	03:47:40	13660.0	NYPD	(D
1167	32303167	12/30/2015 05:15:23 PM	12/30/2015 09:13:03 PM	03:57:40	14260.0	NYPD	(D
576	32309364	12/31/2015 10:47:10 AM	12/31/2015 02:56:33 PM	04:09:23	14963.0	NYPD	(D
1268	32302395	12/30/2015 02:51:05 PM	12/30/2015 07:00:39 PM	04:09:34	14974.0	NYPD	(D
307	32310128	12/31/2015 04:59:34 PM	12/31/2015 09:14:28 PM	04:14:54	15294.0	NYPD	(D

	Unique Key	Created_date	Closed_date	Request_closing_Time	Total_Seconds	Agency	
1253	32298461	12/30/2015 03:12:59 PM	12/30/2015 07:35:54 PM	04:22:55	15775.0	NYPD	(D
1418	32303263	12/30/2015 11:59:33 AM	12/30/2015 04:48:03 PM	04:48:30	17310.0	NYPD	(D
1516	32300440	12/30/2015 09:38:09 AM	12/30/2015 04:34:56 PM	06:56:47	25007.0	NYPD	(D
373	32306461	12/31/2015 03:26:41 PM	12/31/2015 10:53:23 PM	07:26:42	26802.0	NYPD	(D
1485	32303216	12/30/2015 10:16:22 AM	12/30/2015 06:50:14 PM	08:33:52	30832.0	NYPD	(D
1334	32297589	12/30/2015 01:32:44 PM	12/30/2015 10:12:02 PM	08:39:18	31158.0	NYPD	(D
1110	32300487	12/30/2015 07:02:14 PM	12/31/2015 05:17:50 AM	10:15:36	36936.0	NYPD	(D
551	32305075	12/31/2015 11:17:32 AM	12/31/2015 10:56:36 PM	11:39:04	41944.0	NYPD	(D

1667 rows × 55 columns

In []:

```
#Whether the average (response time) across complaint types is similar or not (o
verall)
#H0:response_time is similar across complaint_types
#H1:response_time is not similar across complaint_types
```

In [19]:

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

In [23]:

```
mod=ols('Total_Seconds~Complaint_type',data=data).fit()
tab=sm.stats.anova_lm(mod)
```

In [24]:

tab

Out[24]:

	df	sum_sq	mean_sq	F	PR(>F)
Complaint_type	16.0	1.955111e+10	1.221944e+09	4.478596	8.357868e-09
Residual	1647.0	4.493690e+11	2.728409e+08	NaN	NaN

In []:

```
#Since the Pvalue is less than significance value 0.05 thats why we will reject the null hypothesis
#From this test, we can say that average response time is not similar for overal 1 complaint Types.
```

In [25]:

```
gh=pd.crosstab(data.Complaint_type,data.Request_closing_Time.mean())
gh
```

Out[25]:

col_0	0 days 04:04:39.462139
Complaint_type	
Animal Abuse	45
Bike/Roller/Skate Chronic	1
Blocked Driveway	573
Derelict Vehicle	114
Drinking	7
Homeless Encampment	30
Illegal Parking	459
Noise - Commercial	188
Noise - House of Worship	3
Noise - Park	3
Noise - Street/Sidewalk	104
Noise - Vehicle	41
Panhandling	1
Posting Advertisement	10
Traffic	24
Urinating in Public	3
Vending	61

In []:

```
#Are the type of complaint or service requested and location related?
#H0:Complaint/Service Requested are significantly based on location
#H1:Complaint/Service Requested are not significantly based on location
```

In [26]:

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

In [27]:

```
LocvsCom=pd.crosstab(data.Complaint_type,data.Location_type)
LocvsCom
```

Out[27]:

Location_type	Club/Bar/Restaurant	House and Store	House of Worship	Park/Playground	Residential Building	Res Building
Complaint_type						
Animal Abuse	0	2	0	0	4	
Bike/Roller/Skate Chronic	0	0	0	0	0	
Blocked Driveway	0	0	0	0	0	
Derelict Vehicle	0	0	0	0	0	
Drinking	5	0	0	0	0	
Homeless Encampment	0	0	0	1	0	
Illegal Parking	0	0	0	0	0	
Noise - Commercial	63	0	0	0	0	
Noise - House of Worship	0	0	3	0	0	
Noise - Park	0	0	0	3	0	
Noise - Street/Sidewalk	0	0	0	0	0	
Noise - Vehicle	0	0	0	0	0	
Panhandling	0	0	0	0	0	
Posting Advertisement	0	0	0	0	0	
Traffic	0	0	0	0	0	
Urinating in Public	0	0	0	0	0	
Vending	0	0	0	0	0	

In [28]:

```
from scipy import stats  
stats.chi2_contingency(LocvsCom)
```

Out[28]:

```

(5757.0541426006475,
 0.0,
 128,
 array([[1.83563287e+00, 5.39892022e-02, 8.09838032e-02, 1.07978404e
-01,
        1.07978404e-01, 9.71805639e-01, 3.72525495e+00, 3.80623875e
+01,
        5.39892022e-02],
 [4.07918416e-02, 1.19976005e-03, 1.79964007e-03, 2.39952010e
-03,
        2.39952010e-03, 2.15956809e-02, 8.27834433e-02, 8.45830834e
-01,
        1.19976005e-03],
 [2.33737253e+01, 6.87462507e-01, 1.03119376e+00, 1.37492501e
+00,
        1.37492501e+00, 1.23743251e+01, 4.74349130e+01, 4.84661068e
+02,
        6.87462507e-01],
 [4.65026995e+00, 1.36772645e-01, 2.05158968e-01, 2.73545291e
-01,
        2.73545291e-01, 2.46190762e+00, 9.43731254e+00, 9.64247151e
+01,
        1.36772645e-01],
 [2.85542891e-01, 8.39832034e-03, 1.25974805e-02, 1.67966407e
-02,
        1.67966407e-02, 1.51169766e-01, 5.79484103e-01, 5.92081584e
+00,
        8.39832034e-03],
 [1.22375525e+00, 3.59928014e-02, 5.39892022e-02, 7.19856029e
-02,
        7.19856029e-02, 6.47870426e-01, 2.48350330e+00, 2.53749250e
+01,
        3.59928014e-02],
 [1.87234553e+01, 5.50689862e-01, 8.26034793e-01, 1.10137972e
+00,
        1.10137972e+00, 9.91241752e+00, 3.79976005e+01, 3.88236353e
+02,
        5.50689862e-01],
 [7.66886623e+00, 2.25554889e-01, 3.38332334e-01, 4.51109778e
-01,
        4.51109778e-01, 4.05998800e+00, 1.55632873e+01, 1.59016197e
+02,
        2.25554889e-01],
 [1.22375525e-01, 3.59928014e-03, 5.39892022e-03, 7.19856029e
-03,
        7.19856029e-03, 6.47870426e-02, 2.48350330e-01, 2.53749250e
+00,
        3.59928014e-03],
 [1.22375525e-01, 3.59928014e-03, 5.39892022e-03, 7.19856029e
-03,
        7.19856029e-03, 6.47870426e-02, 2.48350330e-01, 2.53749250e
+00,
        3.59928014e-03],
 [4.24235153e+00, 1.24775045e-01, 1.87162567e-01, 2.49550090e
-01,
        2.49550090e-01, 2.24595081e+00, 8.60947810e+00, 8.79664067e
+01,
        1.24775045e-01],
 [1.67246551e+00, 4.91901620e-02, 7.37852430e-02, 9.83803239e

```



```

-02,
    9.83803239e-02, 8.85422915e-01, 3.39412118e+00, 3.46790642e
+01,
    4.91901620e-02],
[4.07918416e-02, 1.19976005e-03, 1.79964007e-03, 2.39952010e
-03,
    2.39952010e-03, 2.15956809e-02, 8.27834433e-02, 8.45830834e
-01,
    1.19976005e-03],
[4.07918416e-01, 1.19976005e-02, 1.79964007e-02, 2.39952010e
-02,
    2.39952010e-02, 2.15956809e-01, 8.27834433e-01, 8.45830834e
+00,
    1.19976005e-02],
[9.79004199e-01, 2.87942412e-02, 4.31913617e-02, 5.75884823e
-02,
    5.75884823e-02, 5.18296341e-01, 1.98680264e+00, 2.02999400e
+01,
    2.87942412e-02],
[1.22375525e-01, 3.59928014e-03, 5.39892022e-03, 7.19856029e
-03,
    7.19856029e-03, 6.47870426e-02, 2.48350330e-01, 2.53749250e
+00,
    3.59928014e-03],
[2.48830234e+00, 7.31853629e-02, 1.09778044e-01, 1.46370726e
-01,
    1.46370726e-01, 1.31733653e+00, 5.04979004e+00, 5.15956809e
+01,
    7.31853629e-02]]))

```

In [29]:

```
chisquare,p_value,df,frequency=stats.chi2_contingency(LocvsCom)
```

In [30]:

```
p_value
```

Out[30]:

```
0.0
```

In []:

```

#As pvalue is less 0.05(significant level), so we have to reject the null Hypoth
esis
#From this test, we can say that Complaint/Service Requested are not significant
ly based on location

```