**Customer Service Requests Analysis**

**=========================Project 1=======================**

**=========================By Suleyman Muhammed====================**

**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’.
6. - Whether the average response time across complaint types is similar or not (overall)

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

**Dataset Description :**

Field Description

* Unique Key (Plain text) - Unique identifier for the complaints
* Created Date (Date and Time) - The date and time on which the complaint is raised
* Closed Date (Date and Time) - The date and time on which the complaint is closed
* Agency (Plain text) - Agency code
* Agency Name (Plain text) - Name of the agency
* Complaint Type (Plain text) - Type of the complaint
* Descriptor (Plain text) - Complaint type label (Heating - Heat, Traffic Signal Condition -

Controller)

* Location Type (Plain text) - Type of the location (Residential, Restaurant, Bakery, etc)
* Incident Zip (Plain text) - Zip code for the location
* Incident Address (Plain text) - Address of the location
* Street Name (Plain text) - Name of the street
* Cross Street 1 (Plain text) - Detail of cross street
* Cross Street 2 (Plain text) - Detail of another cross street
* Intersection Street 1 (Plain text) - Detail of intersection street if any
* Intersection Street 2 (Plain text) - Detail of another intersection street if any
* Address Type (Plain text) - Categorical (Address or Intersection)
* City (Plain text) - City for the location
* Landmark (Plain text) - Empty field
* Facility Type (Plain text) - N/A
* Status (Plain text) - Categorical (Closed or Pending)
* Due Date (Date and Time) - Date and time for the pending complaints
* Resolution Action Updated Date (Date and Time) - Date and time when the resolution was provided
* Community Board (Plain text) - Categorical field (specifies the community board with its code)
* Borough (Plain text) - Categorical field (specifies the community board)
* X Coordinate (State Plane) (Number)
* Y Coordinate (State Plane) (Number)
* Park Facility Name (Plain text) - Unspecified
* Park Borough (Plain text) - Categorical (Unspecified, Queens, Brooklyn etc)
* School Name (Plain text) - Unspecified
* School Number (Plain text) - Unspecified
* School Region (Plain text) - Unspecified
* School Code (Plain text) - Unspecified
* School Phone Number (Plain text) - Unspecified
* School Address (Plain text) - Unspecified
* School City (Plain text) - Unspecified
* School State (Plain text) - Unspecified
* School Zip (Plain text) - Unspecified
* School Not Found (Plain text) - Empty Field
* School or Citywide Complaint (Plain text) - Empty Field
* Vehicle Type (Plain text) - Empty Field
* Taxi Company Borough (Plain text) - Empty Field
* Taxi Pick Up Location (Plain text) - Empty Field
* Bridge Highway Name (Plain text) - Empty Field
* Bridge Highway Direction (Plain text) - Empty Field
* Road Ramp (Plain text) - Empty Field
* Bridge Highway Segment (Plain text) - Empty Field
* Garage Lot Name (Plain text) - Empty Field
* Ferry Direction (Plain text) - Empty Field
* Ferry Terminal Name (Plain text) - Empty Field
* Latitude (Number) - Latitude of the location
* Longitude (Number) - Longitude of the location

In [3]:

**import** pandas **as** pd

**import** numpy **as** np

**import** scipy.stats **as** stats

**import** matplotlib.pyplot **as** plt

**from** matplotlib **import** style

**import** seaborn **as** sns

**import** warnings

**Step 1**

- Importing the dataset

In [4]:

df **=** pd**.**read\_csv('C:\\Users\\ayman\\Desktop\\DataScience\\Data Science with Python Two\\Cutomer\_Service\_Request\_Analysis\_from\_2010\_to\_Present.csv', low\_memory**=False**)

In [5]:

*# View the first 10 rows of the dataset*

df**.**head(10)

Out[5]:

|  | **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** | **Longitude** | **Location** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 | -73.923501 | (40.86568153633767, -73.92350095571744) |
| **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 | -73.915094 | (40.775945312321085, -73.91509393898605) |
| **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 | -73.888525 | (40.870324522111424, -73.88852464418646) |
| **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 | -73.828379 | (40.83599404683083, -73.82837939584206) |
| **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 | -73.874170 | (40.733059618956815, -73.87416975810375) |
| **5** | 32306554 | 12/31/2015 11:56:30 PM | 01-01-16 1:50 | NYPD | New York City Police Department | Illegal Parking | Posted Parking Sign Violation | Street/Sidewalk | 11215.0 | 260 21 STREET | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 40.660823 | -73.992568 | (40.66082272389114, -73.99256786342693) |
| **6** | 32306559 | 12/31/2015 11:55:32 PM | 01-01-16 1:53 | NYPD | New York City Police Department | Illegal Parking | Blocked Hydrant | Street/Sidewalk | 10032.0 | 524 WEST 169 STREET | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 40.840848 | -73.937375 | (40.840847591440415, -73.9373750864581) |
| **7** | 32307009 | 12/31/2015 11:54:05 PM | 01-01-16 1:42 | NYPD | New York City Police Department | Blocked Driveway | No Access | Street/Sidewalk | 10457.0 | 501 EAST 171 STREET | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 40.837503 | -73.902905 | (40.83750262540012, -73.90290517326568) |
| **8** | 32308581 | 12/31/2015 11:53:58 PM | 01-01-16 8:27 | NYPD | New York City Police Department | Illegal Parking | Posted Parking Sign Violation | Street/Sidewalk | 11415.0 | 83-44 LEFFERTS BOULEVARD | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 40.704977 | -73.832605 | (40.704977164399935, -73.8326047502584) |
| **9** | 32308391 | 12/31/2015 11:53:58 PM | 01-01-16 1:17 | NYPD | New York City Police Department | Blocked Driveway | No Access | Street/Sidewalk | 11219.0 | 1408 66 STREET | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 40.623793 | -73.999539 | (40.623793065806524, -73.99953890121567) |

10 rows × 53 columns

In [6]:

*# shape-up the dataset to understand*

df**.**shape

Out[6]:

(300698, 53)

In [7]:

*# The columns' data info*

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 [8]:

*# Type of complaints*

df['Complaint Type']**.**unique()

Out[8]:

array(['Noise - Street/Sidewalk', 'Blocked Driveway', 'Illegal Parking',

'Derelict Vehicle', 'Noise - Commercial',

'Noise - House of Worship', 'Posting Advertisement',

'Noise - Vehicle', 'Animal Abuse', 'Vending', 'Traffic',

'Drinking', 'Bike/Roller/Skate Chronic', 'Panhandling',

'Noise - Park', 'Homeless Encampment', 'Urinating in Public',

'Graffiti', 'Disorderly Youth', 'Illegal Fireworks',

'Ferry Complaint', 'Agency Issues', 'Squeegee', 'Animal in a Park'],

dtype=object)

In [9]:

*# Descriptor*

df['Descriptor']**.**unique()

Out[9]:

array(['Loud Music/Party', 'No Access', 'Commercial Overnight Parking',

'Blocked Sidewalk', 'Posted Parking Sign Violation',

'Blocked Hydrant', 'With License Plate', 'Partial Access',

'Unauthorized Bus Layover', 'Double Parked Blocking Vehicle',

'Double Parked Blocking Traffic', 'Vehicle', 'Loud Talking',

'Banging/Pounding', 'Car/Truck Music', 'Tortured',

'In Prohibited Area', 'Congestion/Gridlock', 'Neglected',

'Car/Truck Horn', 'In Public', 'Other (complaint details)', nan,

'No Shelter', 'Truck Route Violation', 'Unlicensed',

'Overnight Commercial Storage', 'Engine Idling',

'After Hours - Licensed Est', 'Detached Trailer',

'Underage - Licensed Est', 'Chronic Stoplight Violation',

'Loud Television', 'Chained', 'Building', 'In Car',

'Police Report Requested', 'Chronic Speeding',

'Playing in Unsuitable Place', 'Drag Racing',

'Police Report Not Requested', 'Nuisance/Truant', 'Homeless Issue',

'Language Access Complaint', 'Disruptive Passenger',

'Animal Waste'], dtype=object)

In [10]:

*# Complaints' count by city*

complaints\_type\_by\_city **=** pd**.**DataFrame({'count':df**.**groupby(['Complaint Type','City'])**.**size()})**.**reset\_index()

complaints\_type\_by\_city

Out[10]:

|  | **Complaint Type** | **City** | **count** |
| --- | --- | --- | --- |
| **0** | Animal Abuse | ARVERNE | 38 |
| **1** | Animal Abuse | ASTORIA | 125 |
| **2** | Animal Abuse | BAYSIDE | 37 |
| **3** | Animal Abuse | BELLEROSE | 7 |
| **4** | Animal Abuse | BREEZY POINT | 2 |
| **...** | ... | ... | ... |
| **759** | Vending | STATEN ISLAND | 25 |
| **760** | Vending | SUNNYSIDE | 15 |
| **761** | Vending | WHITESTONE | 1 |
| **762** | Vending | WOODHAVEN | 6 |
| **763** | Vending | WOODSIDE | 15 |

764 rows × 3 columns

In [11]:

df**.**groupby(['Borough','Complaint Type','Descriptor'])**.**size()

Out[11]:

Borough Complaint Type Descriptor

BRONX Animal Abuse Chained 132

In Car 36

Neglected 673

No Shelter 71

Other (complaint details) 311

...

Unspecified Noise - Vehicle Engine Idling 11

Posting Advertisement Vehicle 1

Traffic Truck Route Violation 1

Vending In Prohibited Area 2

Unlicensed 5

Length: 288, dtype: int64

In [12]:

*# Check for duplications*

df**.**duplicated()**.**sum()

Out[12]:

0

In [13]:

df**.**isna()**.**sum()

Out[13]:

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

In [14]:

*# Drop the Not applicable values*

df**.**dropna(subset**=**['City','Latitude'],inplace**=True**)

In [15]:

df**.**to\_csv('Nyc\_Customer\_Cleaned.csv',index**=False**)

**Step 2**

- Convert and read the date represented columns to date datatype

- And Check the elapsed time between the Created and Closed date by

creating a new column

In [16]:

**import** datetime

In [19]:

df**=**pd**.**read\_csv('C:\\Users\\ayman\\Desktop\\DataScience\\Data Science with Python Two\\Cutomer\_Service\_Request\_Analysis\_from\_2010\_to\_Present.csv', parse\_dates**=**["Created Date","Closed Date"], low\_memory**=False**)

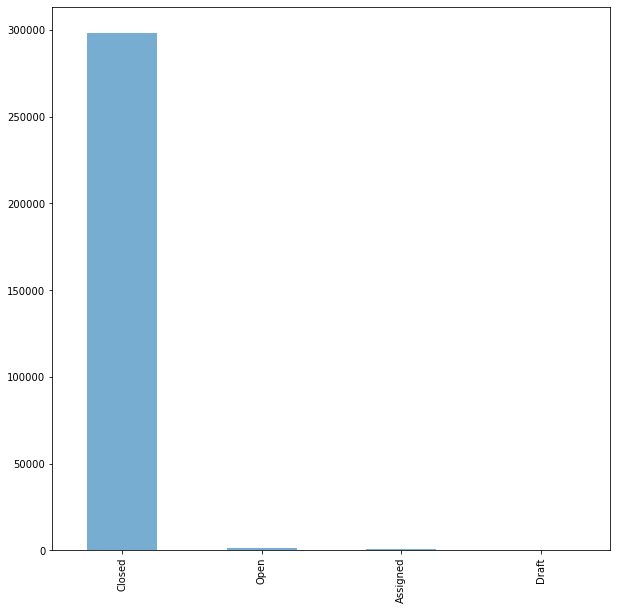
In [20]:

df["Closing\_time"] **=** df["Closed Date"] **-** df["Created Date"]

In [22]:

df['Status']**.**value\_counts()**.**plot(kind**=**'bar',alpha**=**0.6,figsize**=**(10,10))

plt**.**show()



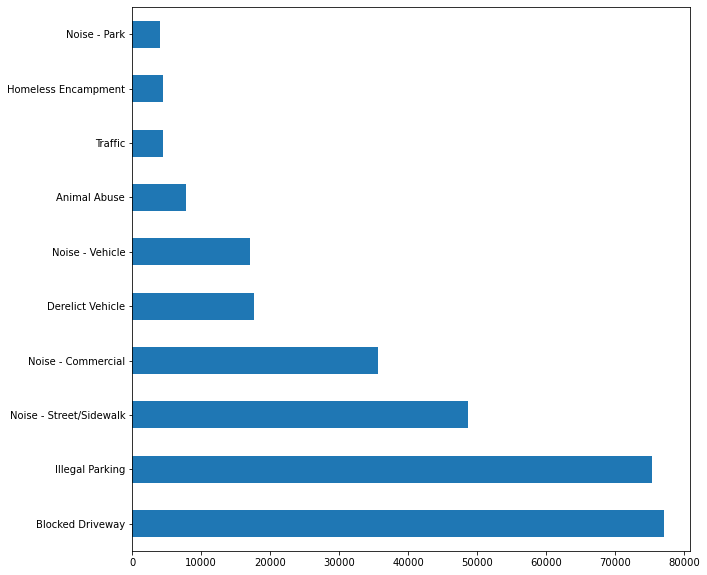
In [23]:

*# Break down the types of complaints so it will be easy to find out most of the complaints type*

df['Complaint Type']**.**value\_counts()**.**head(10)**.**plot(kind**=**'barh',figsize**=**(10,10))

Out[23]:

<AxesSubplot:>



In [28]:

*# Calculating the Request Closing time to minutes*

df['Closing\_time\_min'] **=** df['Closing\_time']**/**np**.**timedelta64(1,'m')

In [29]:

df**.**head()

Out[29]:

|  | **Unique Key** | **Created Date** | **Closed Date** | **Agency** | **Agency Name** | **Complaint Type** | **Descriptor** | **Location Type** | **Incident Zip** | **Incident Address** | **...** | **Road Ramp** | **Bridge Highway Segment** | **Garage Lot Name** | **Ferry Direction** | **Ferry Terminal Name** | **Latitude** | **Longitude** | **Location** | **Closing\_time** | **Closing\_time\_min** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 | NaN | 40.865682 | -73.923501 | (40.86568153633767, -73.92350095571744) | 0 days 00:55:15 | 55.250000 |
| **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 | NaN | 40.775945 | -73.915094 | (40.775945312321085, -73.91509393898605) | 0 days 01:26:16 | 86.266667 |
| **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 | NaN | 40.870325 | -73.888525 | (40.870324522111424, -73.88852464418646) | 0 days 04:51:31 | 291.516667 |
| **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 | NaN | 40.835994 | -73.828379 | (40.83599404683083, -73.82837939584206) | 0 days 07:45:14 | 465.233333 |
| **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 | NaN | 40.733060 | -73.874170 | (40.733059618956815, -73.87416975810375) | 0 days 03:27:02 | 207.033333 |

5 rows × 55 columns

In [30]:

*# Compliants Types grouped by*

df**.**groupby(["Borough","Complaint Type","Descriptor"])**.**size()

Out[30]:

Borough Complaint Type Descriptor

BRONX Animal Abuse Chained 132

In Car 36

Neglected 672

No Shelter 71

Other (complaint details) 311

...

STATEN ISLAND Traffic Congestion/Gridlock 113

Drag Racing 18

Truck Route Violation 13

Vending In Prohibited Area 5

Unlicensed 20

Length: 248, dtype: int64

In [34]:

*# Now Complaint type depend on the number of request*

df['City']**=** df['City']**.**str**.**lower()**.**str**.**replace(" ","\_")

txt**=**{'weight':'bold'}

plt**.**figure(figsize**=**(15,12))

sns**.**countplot(x**=**'Complaint Type',data**=**df,order**=**df['Complaint Type']**.**value\_counts()**.**index)

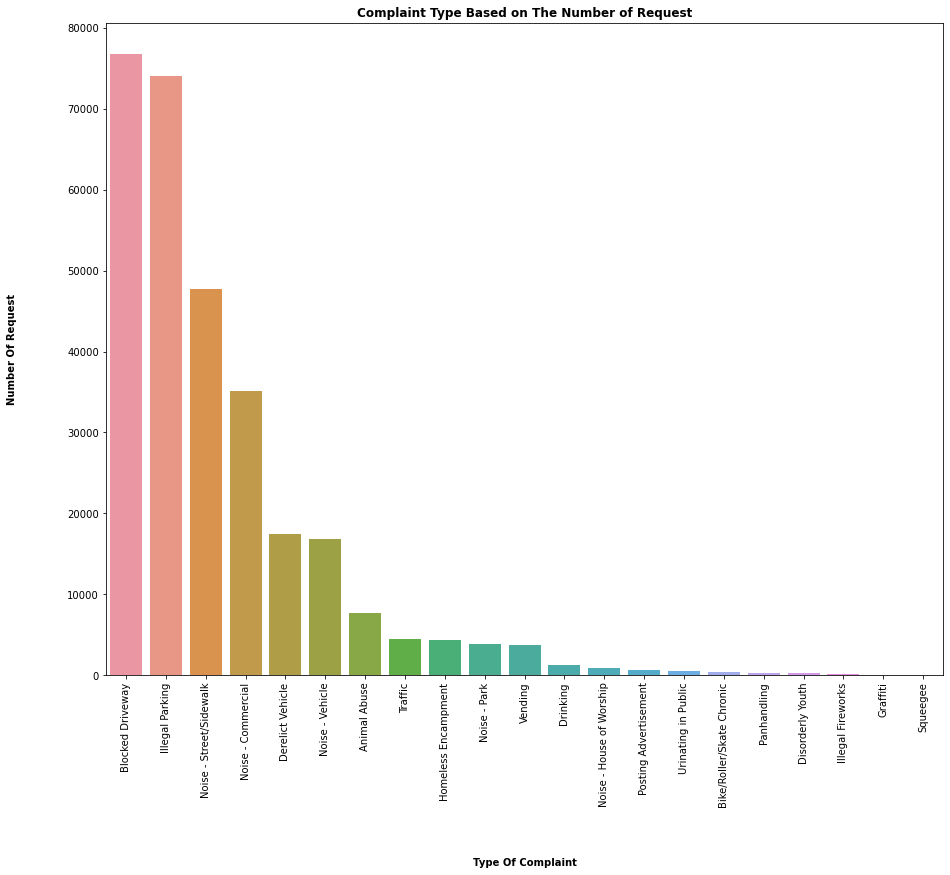
plt**.**xticks(rotation**=**90)

plt**.**title("Complaint Type Based on The Number of Request",fontdict**=**txt)

plt**.**xlabel('Type Of Complaint',fontdict**=**txt,labelpad**=**50)

plt**.**ylabel('Number Of Request',fontdict**=**txt,labelpad**=**50)

plt**.**show()



In [38]:

highComplaints **=** df**.**dropna(subset**=**["Complaint Type"])

highComplaints **=** df**.**groupby("Complaint Type")

*# Sort the complaint type*

complaintTypeSorted **=** highComplaints**.**size()**.**sort\_values(ascending **=** **False**)

complaintTypeSorted **=** complaintTypeSorted**.**to\_frame('Number of Complaint')**.**reset\_index()

complaintTypeSorted

complaintTypeSorted**.**head(20)

Out[38]:

|  | **Complaint Type** | **Number of Complaint** |
| --- | --- | --- |
| **0** | Blocked Driveway | 76722 |
| **1** | Illegal Parking | 74063 |
| **2** | Noise - Street/Sidewalk | 47770 |
| **3** | Noise - Commercial | 35165 |
| **4** | Derelict Vehicle | 17519 |
| **5** | Noise - Vehicle | 16873 |
| **6** | Animal Abuse | 7747 |
| **7** | Traffic | 4473 |
| **8** | Homeless Encampment | 4366 |
| **9** | Noise - Park | 3929 |
| **10** | Vending | 3776 |
| **11** | Drinking | 1271 |
| **12** | Noise - House of Worship | 922 |
| **13** | Posting Advertisement | 649 |
| **14** | Urinating in Public | 592 |
| **15** | Bike/Roller/Skate Chronic | 414 |
| **16** | Panhandling | 301 |
| **17** | Disorderly Youth | 285 |
| **18** | Illegal Fireworks | 163 |
| **19** | Graffiti | 113 |

**Step 3**

**Visualization**

**Conclusion for the Following Complaints:**

**- Number of Complaints by City**

**- Responce time for the Complaints (Average vs Maximum) and (Minimum vs Maximum)**

**- Which Complaints are Responded quick & which ones are not**

**- Number of Complaints Requests by Complaint Type**

**- Number of Complaints Type by City**

In [39]:

df['City']**=** df['City']**.**str**.**lower()**.**str**.**replace(" ","\_")

txt**=**{'weight':'bold'}

plt**.**figure(figsize**=**(15,12))

sns**.**countplot(x**=**'Complaint Type',data**=**df,order**=**df['Complaint Type']**.**value\_counts()**.**index)

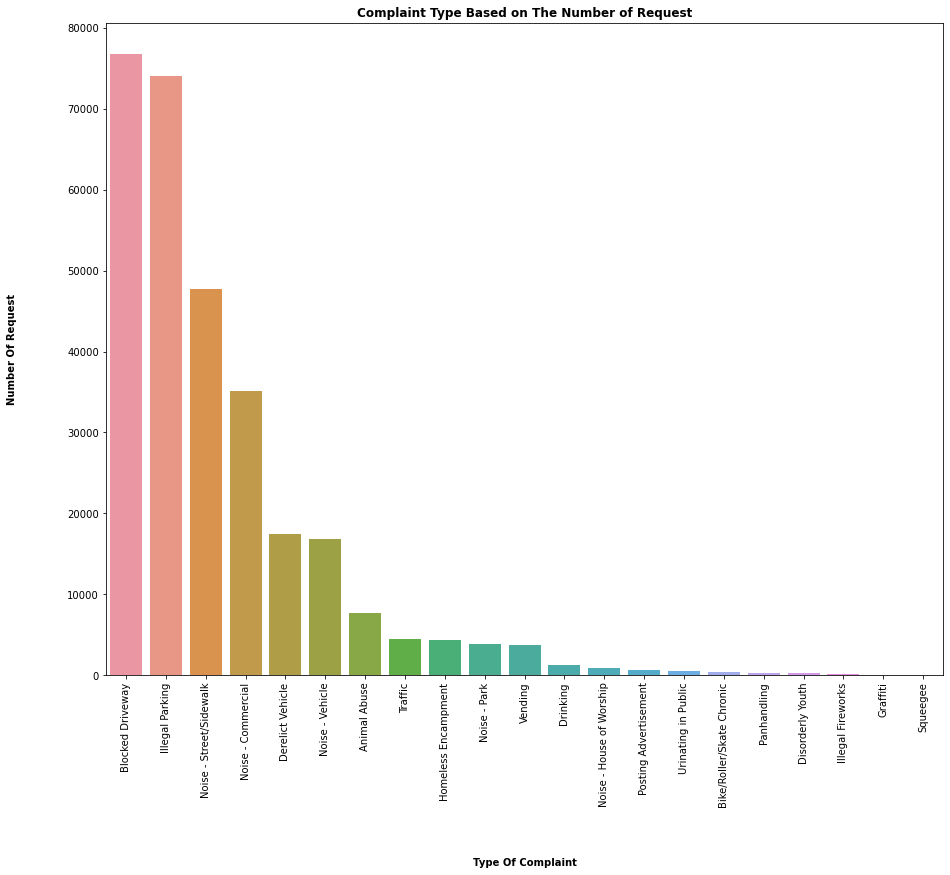
plt**.**xticks(rotation**=**90)

plt**.**title("Complaint Type Based on The Number of Request",fontdict**=**txt)

plt**.**xlabel('Type Of Complaint',fontdict**=**txt,labelpad**=**50)

plt**.**ylabel('Number Of Request',fontdict**=**txt,labelpad**=**50)

plt**.**show()



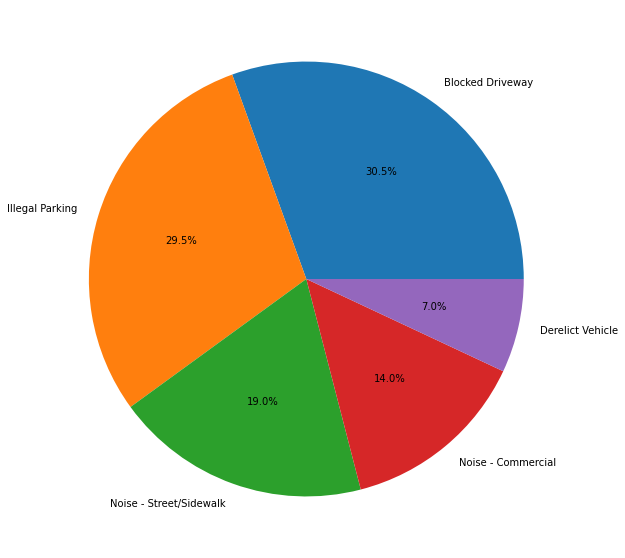
In [40]:

complaintTypeSorted **=** complaintTypeSorted**.**head()

plt**.**figure(figsize**=**(10,10))

plt**.**pie(complaintTypeSorted['Number of Complaint'],labels**=**complaintTypeSorted["Complaint Type"],autopct**=**"%1.1f%%")

plt**.**show()



In [41]:

complaintType\_groupedby **=** df**.**groupby('Complaint Type')

In [43]:

dataGrouped **=** complaintType\_groupedby**.**get\_group('Blocked Driveway')

dataGrouped**.**shape

Out[43]:

(76722, 55)

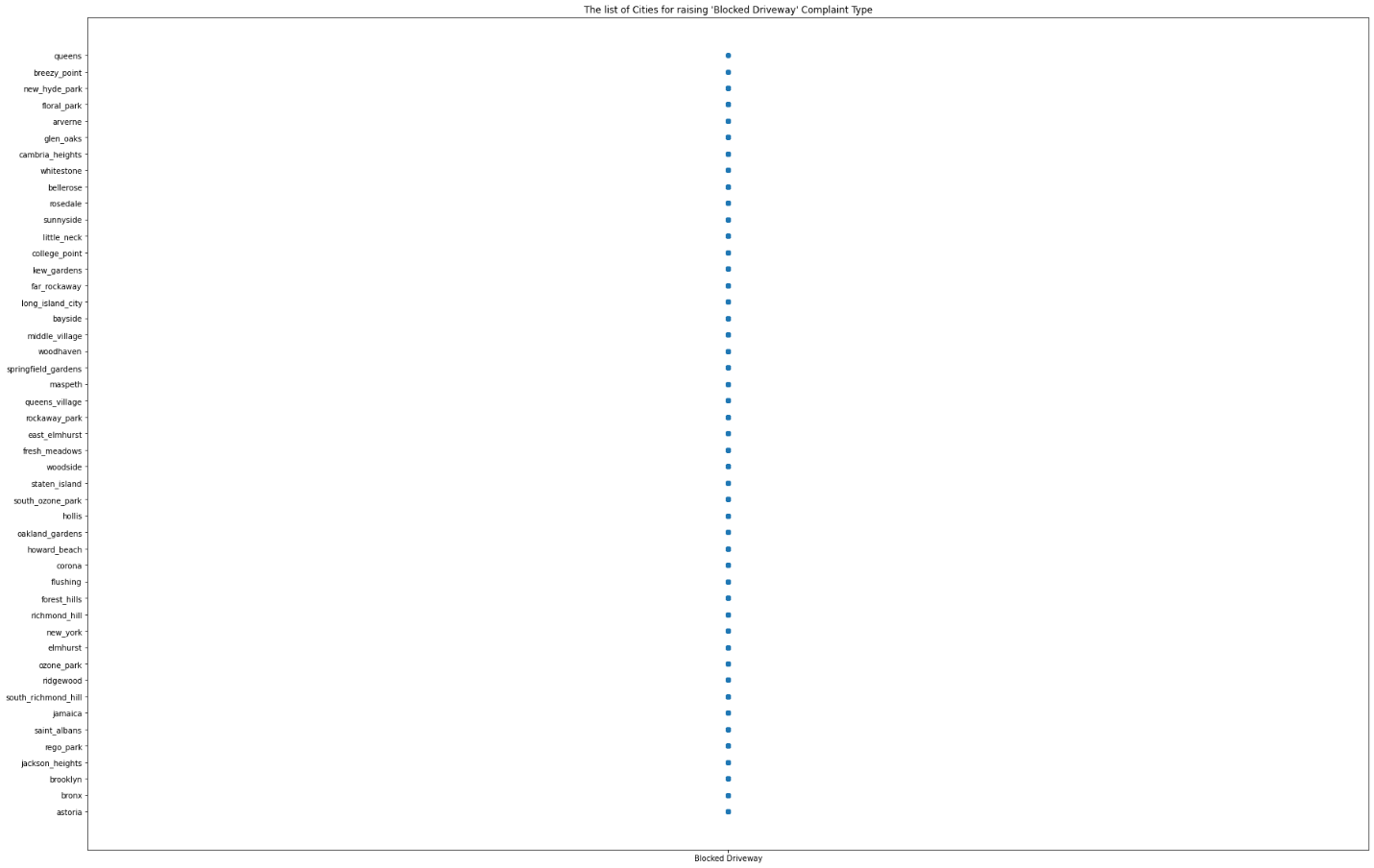
In [52]:

plt**.**figure(figsize**=**(30,20))

plt**.**scatter(dataGrouped['Complaint Type'],dataGrouped['City'])

plt**.**title("The list of Cities for raising 'Blocked Driveway' Complaint Type")

plt**.**show()



**Conclusion #1**

- Based on the above visualization, Blocked Driveway Complaint Type is ranked first for Number of Complaints

- And Illegal Parking Complaint Type is ranked in second place

In [44]:

*# Number of Requests for the Top 20 Cities*

top20\_cities **=** df['City']**.**value\_counts()**.**head(20)

top20 **=** top20\_cities**.**index

txt**=**{'weight':'bold'}

plt**.**figure(figsize**=**(20,10))

sns**.**countplot(x**=**'City',data**=**df[df**.**City**.**isin(top20)], order**=**top20)

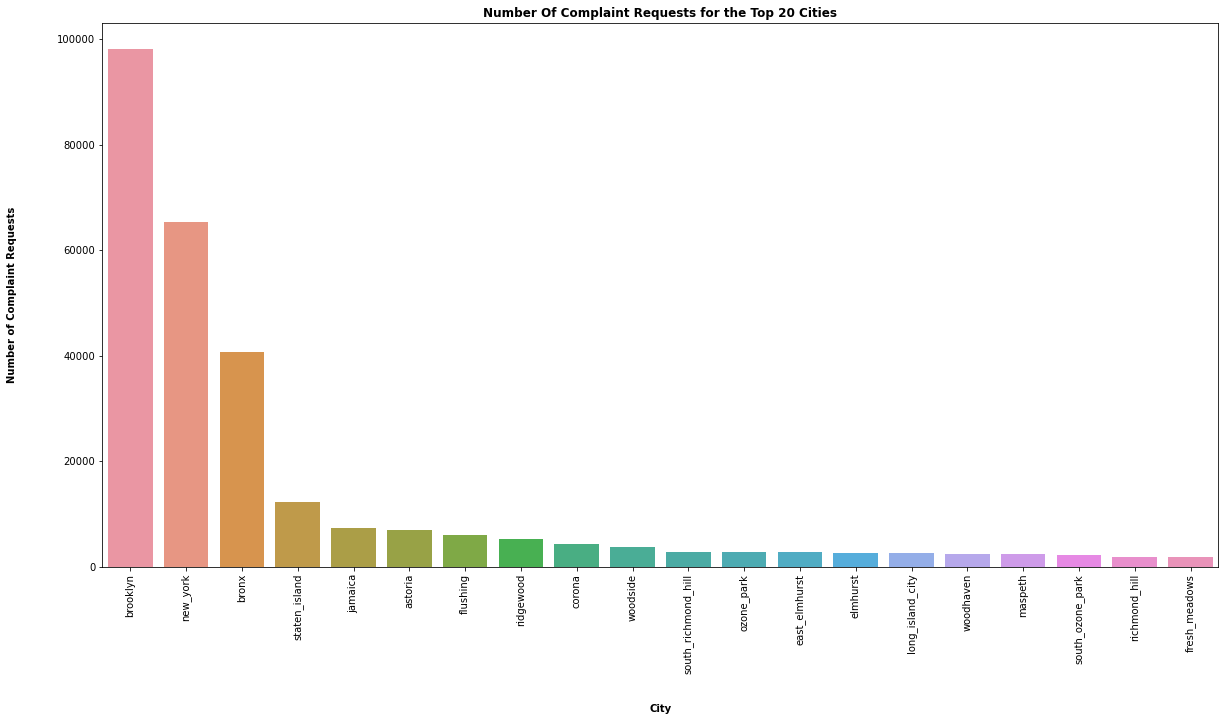
plt**.**title("Number Of Complaint Requests for the Top 20 Cities",fontdict**=**txt)

plt**.**xlabel('City',fontdict**=**txt,labelpad**=**30)

plt**.**xticks(rotation**=**90)

plt**.**ylabel('Number of Complaint Requests',fontdict**=**txt,labelpad**=**40)

plt**.**show()



**Conclusion #2**

**Number of Complaints by City**

- The city of Brooklyn is ranked 1st for the Number of Complaint Request

In [45]:

*# Average Response time for the Complaints*

dsply **=** df[['City','Closing\_time\_min']]

cal **=**dsply**.**groupby('City')['Closing\_time\_min']**.**mean()**.**to\_frame()

cal **=** cal**.**sort\_values('Closing\_time\_min')

cal['City'] **=** cal**.**index

txt**=**{'weight':'bold'}

plt**.**figure(figsize**=**(15,25))

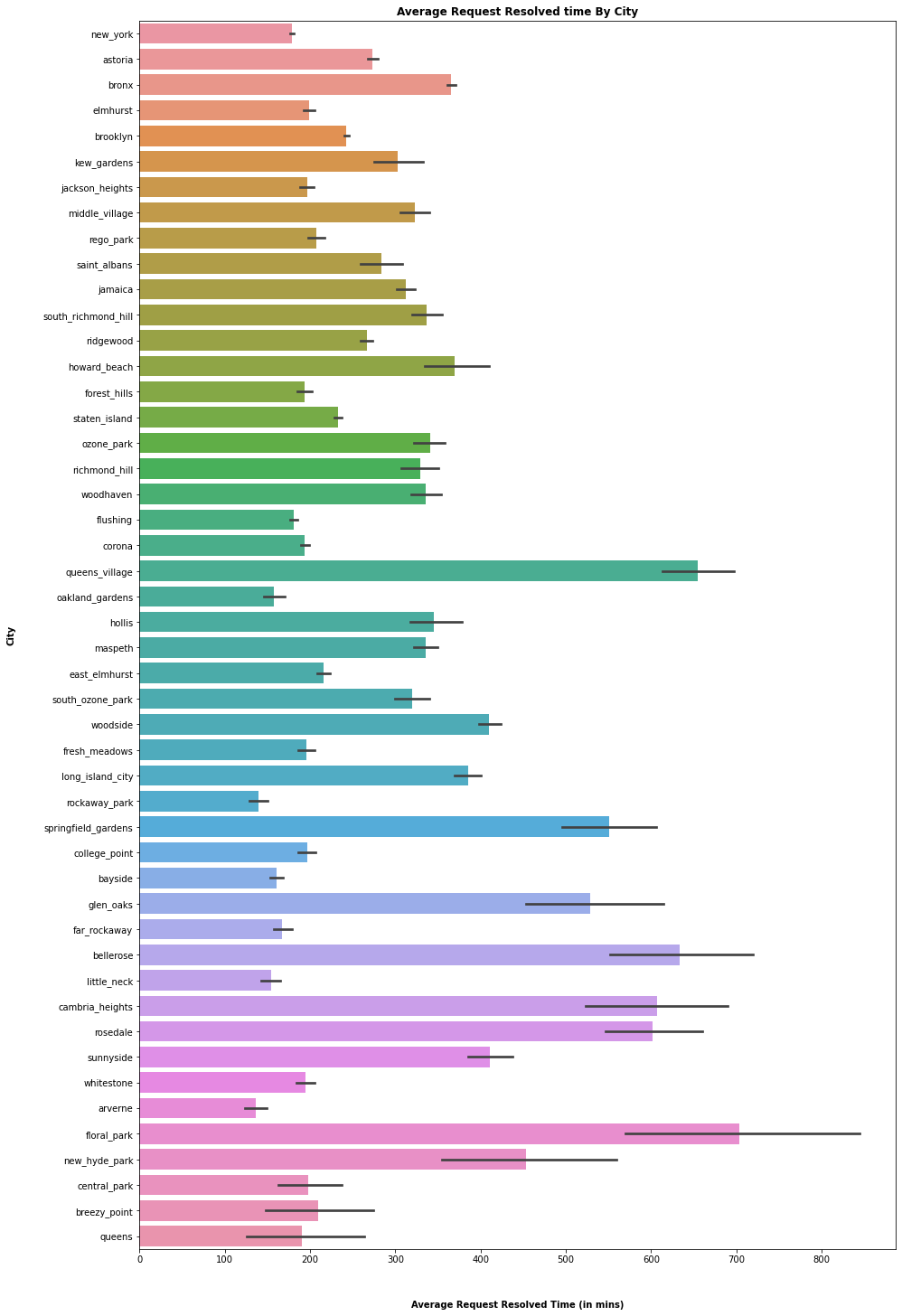
sns**.**barplot(y**=**'City',x**=**'Closing\_time\_min',data**=**dsply)

plt**.**title("Average Request Resolved time By City",fontdict**=**txt)

plt**.**ylabel("City",fontdict**=**txt,labelpad**=**30)

plt**.**xlabel("Average Request Resolved Time (in mins)",fontdict**=**txt,labelpad**=**40)

plt**.**show()



In [49]:

*# Average response time of Boroughs*

txt**=**{'weight':'bold'}

plt**.**figure(figsize**=**(20,15))

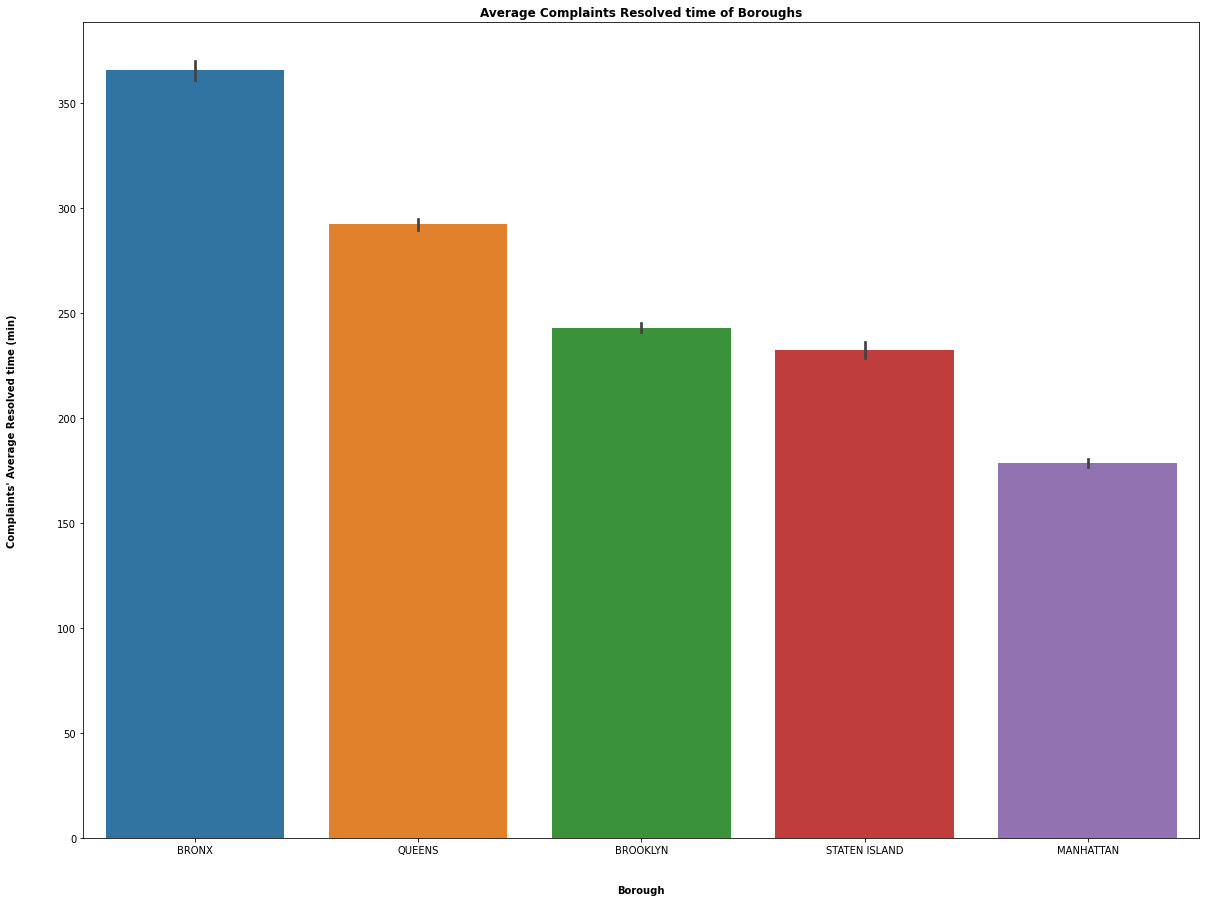
sns**.**barplot(x**=**'Borough',y**=**'Closing\_time\_min',data**=**df,order**=**['BRONX','QUEENS','BROOKLYN','STATEN ISLAND','MANHATTAN'])

plt**.**title("Average Complaints Resolved time of Boroughs",fontdict**=**txt)

plt**.**xlabel("Borough",fontdict**=**txt,labelpad**=**30)

plt**.**ylabel("Complaints' Average Resolved time (min)",fontdict**=**txt,labelpad**=**40)

plt**.**show()



**Conclusion #3**

**Responce time for the Complaints (Average vs Maximum) and (Minimum vs Maximum)**

- When Flora Park scored a maximum number of respond time for the complaint request, Arverne scored minimum

- Based on the average respond time of the complaint request, Bronx is ranked 1st with maximum and Manhattan scored

minimum

In [51]:

*# Average Response time for the Complaints*

dsply2 **=** df[['Complaint Type','Closing\_time\_min']]

cal2 **=**dsply2**.**groupby('Complaint Type')['Closing\_time\_min']**.**mean()**.**to\_frame()

cal2 **=** cal2**.**sort\_values('Closing\_time\_min')

cal2['Complaint Type'] **=** cal2**.**index

txt**=**{'weight':'bold'}

plt**.**figure(figsize**=**(25,15))

sns**.**barplot(x**=**'Complaint Type',y**=**'Closing\_time\_min',data**=**cal2)

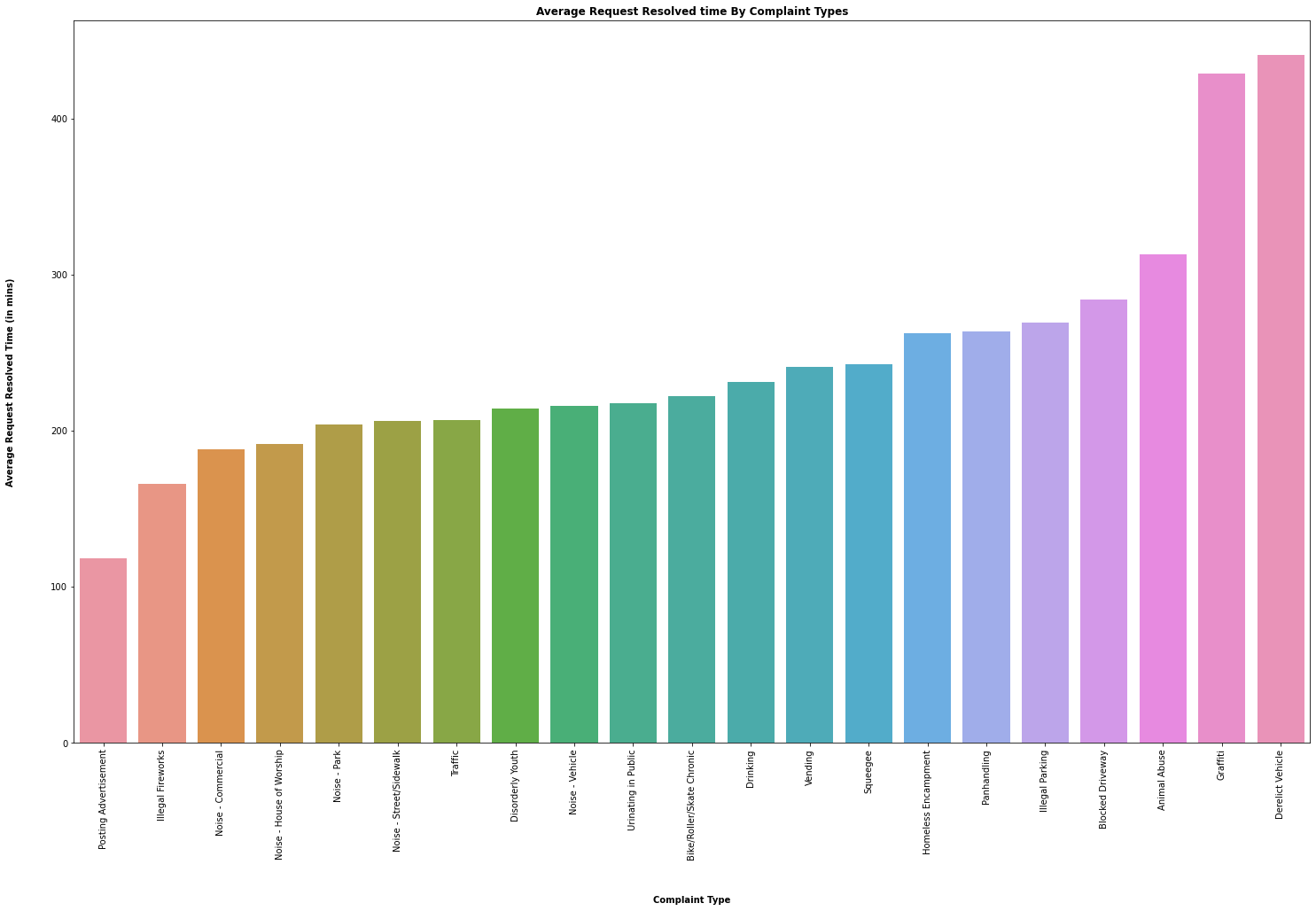
plt**.**title("Average Request Resolved time By Complaint Types",fontdict**=**txt)

plt**.**xlabel("Complaint Type",fontdict**=**txt,labelpad**=**40)

plt**.**ylabel("Average Request Resolved Time (in mins)",fontdict**=**txt,labelpad**=**40)

plt**.**xticks(rotation**=**90)

plt**.**show()



**Conclusion #4**

**Which Complaints are Responded quick & which ones are not**

- Derelict Vehicle & Posting Advertistment complaints scored the slowest and quickest respond time. In this case,

The Respond time for Derelict Vehicle complaints was slower and the Posting Advertistment was Fastest

In [53]:

*# The Complaint Types and number of compalint requests by City*

complaintType\_by\_city **=** pd**.**crosstab(index**=**df['City'],columns**=**df['Complaint Type'])

txt**=**{'weight':'bold'}

plt**.**figure(figsize**=**(25,15))

complaintType\_by\_city**.**plot(kind**=**'barh',figsize**=**(20,30),stacked**=True**)

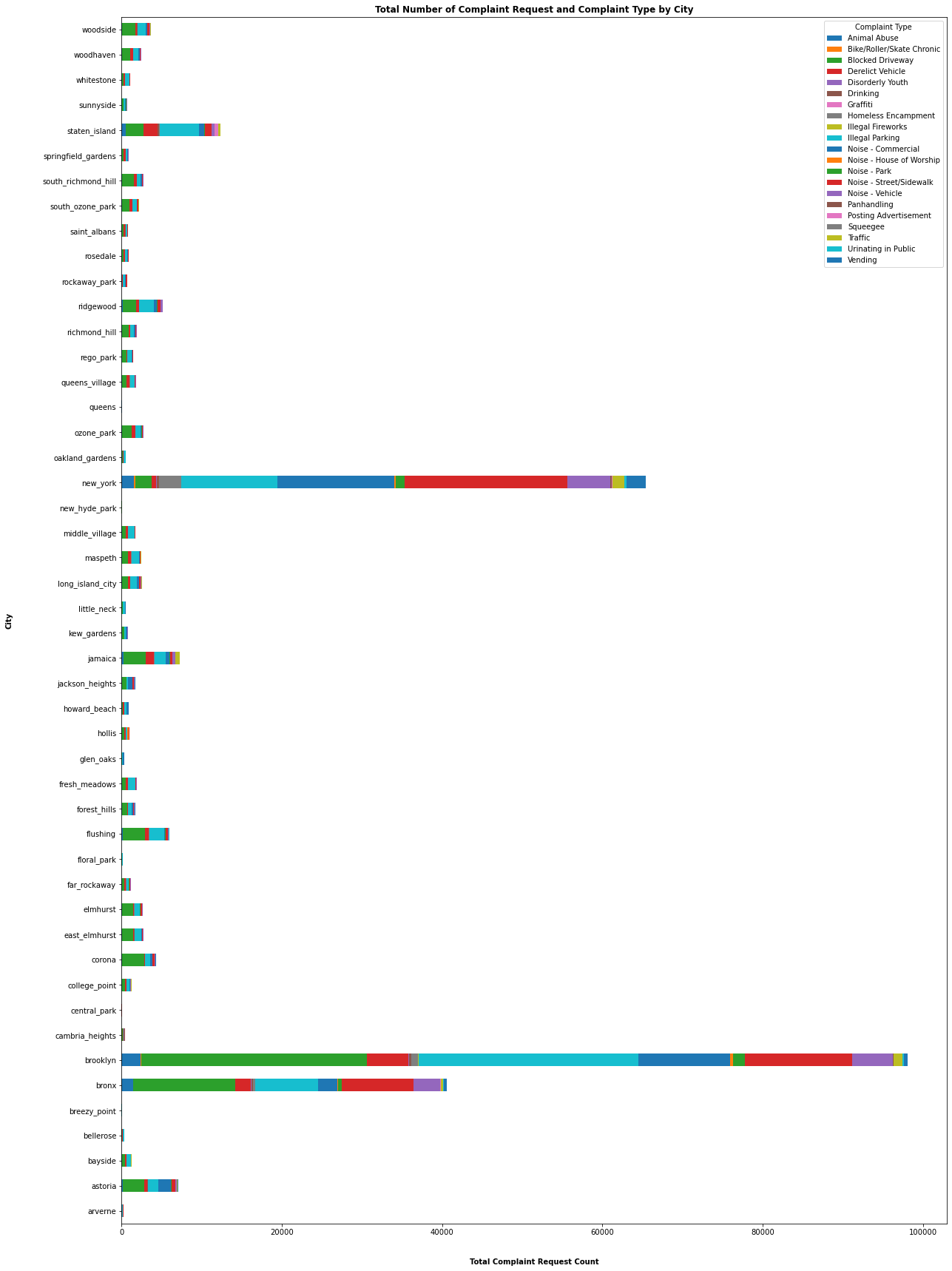
plt**.**title("Total Number of Complaint Request and Complaint Type by City",fontdict**=**txt)

plt**.**xlabel("Total Complaint Request Count",fontdict**=**txt,labelpad**=**30)

plt**.**ylabel("City",fontdict**=**txt,labelpad**=**40)

plt**.**show()

<Figure size 1800x1080 with 0 Axes>



**Conclusion #5**

**Number of Complaints Requests by Complaint Type**

- For both Number of Complaint Request and Complaint Types, Brooklyn scored the maximum number than the other cities.

In [ ]: