**NYC 311 Customer Service Requests Analysis**

**Description:**

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)

Import a 311 NYC service request.  
Read or convert the columns ‘Created Date’ and Closed Date’ to datetime data type and create a new column ‘Request\_Closing\_Time’ as the time elapsed between request creation and request closing. (Hint: Explore the package/module datetime)  
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.  
Order the complaint types based on the average ‘Request\_Closing\_Time’, grouping them for different locations.  
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?

* *DATA INFORMATION AFTER CLEANING AND PRE-PROCESSING*

<class 'pandas.core.frame.DataFrame'>

Int64Index: 297646 entries, 0 to 300697

Data columns (total 14 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Unique Key 297646 non-null int64

1 Created Date 297646 non-null datetime64[ns]

2 Closed Date 297646 non-null datetime64[ns]

3 Complaint Type 297646 non-null object

4 Location Type 297646 non-null object

5 Incident Zip 297646 non-null float64

6 City 297646 non-null object

7 Status 297646 non-null object

8 Due Date 297646 non-null datetime64[ns]

9 Resolution Description 297646 non-null object

10 Resolution Action Updated Date 297646 non-null datetime64[ns]

11 Latitude 297646 non-null float64

12 Longitude 297646 non-null float64

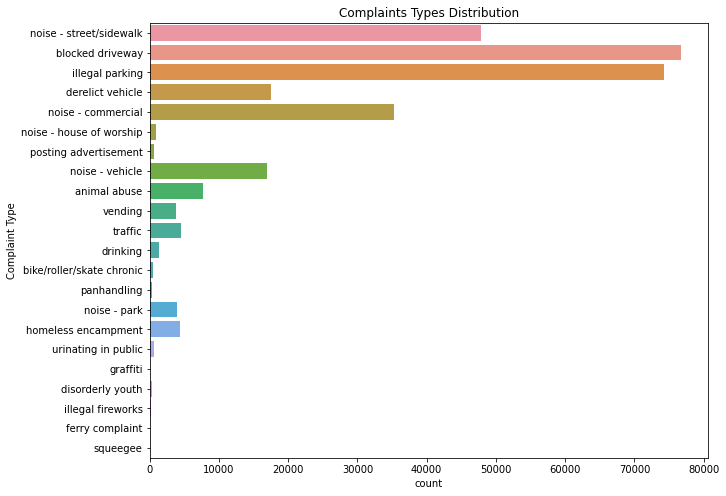
13 Request Closing Time(in hours) 297646 non-null float64

dtypes: datetime64[ns](4), float64(4), int64(1), object(5)

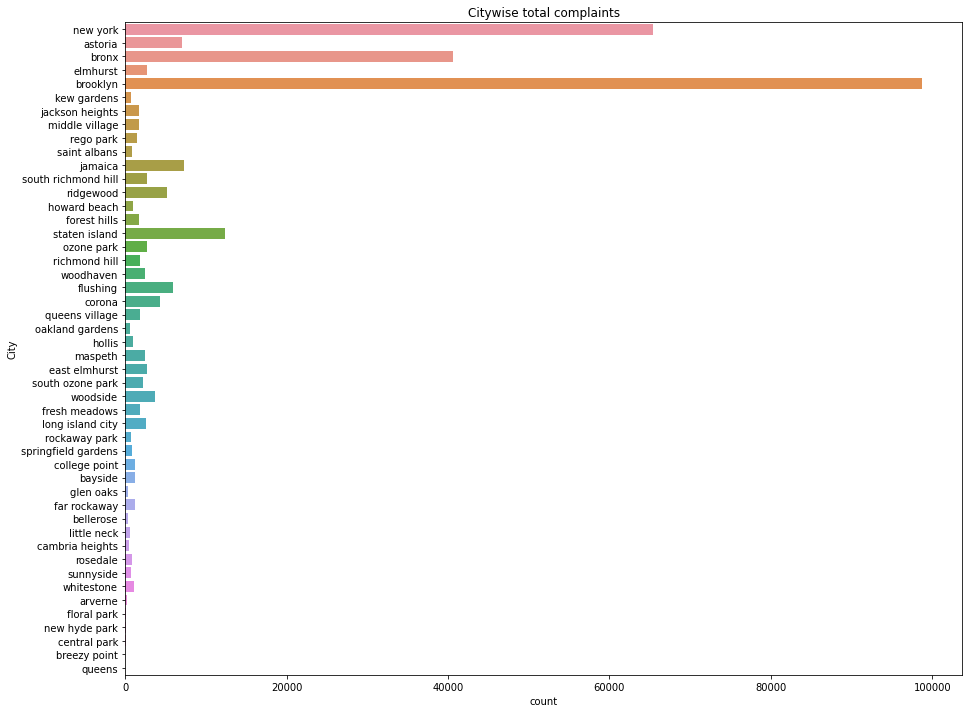
memory usage: 34.1+ MB

* *BARPLOT*

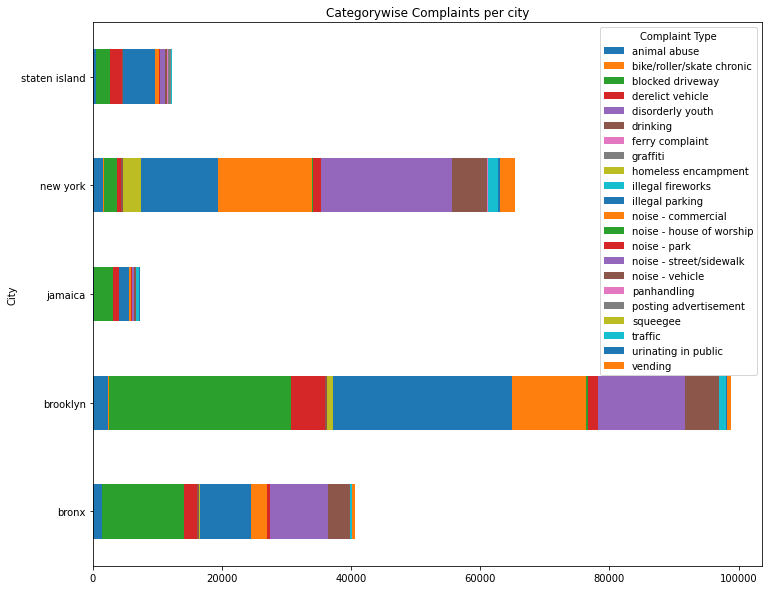
*#total complaints based on category*



*#citywise complaint counts (total)*



*#citywise complaint counts (typewise)*



* *Hypothesis testing*

\* Whether the average response time across complaint types is similar or not (overall)

> When your experiment is trying to draw a comparison or find the difference between one categorical (with more than two categories) and another continuous variable, then you use the ANOVA (Analysis of Variance) test.

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

> Complaint Type (CAT), Location (CAT) : Chi-Square Test

**# new\_ds = ds.dr**

> complaintTypes = df['Complaint Type'].unique()

> for i in range(len(complaintTypes)):

exec("c{} = df.loc[(df['Complaint Type'] == '{}'),'Request Closing Time(in hours)']".format(i+1,complaintTypes[i]))

> fscore,pvalue = stats.f\_oneway(c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c15,c16)

> print(fscore, pvalue)

**Output Value: 541.0488206762245 0.0**

**pvalue of 0 (<0.5) mean we reject null hypothesis(i.e complaint type doesn't affect on request closing time).**

**# Chi square test: Complaint Type (CAT), Location(CAT)**

> ctabDF = pd.crosstab(df['Complaint Type'],df['City'])

> stat, p, dof, expected = chi2\_contingency(ctabDF)

> print('Chi-Square Statistic Value: ',stat)

> print('p value: ',p)

> print('degrees of freedom: ',dof)

**Output Value: Chi-Square Statistic Value: 109359.08964937081**

**p value: 0.0**

**degrees of freedom: 987**