In [151]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
```

Task 1

Import NYC service request

In [2]:

```python
df = pd.read_csv(r'D:\DS\Py\Datasets\Projects\311_Service_Requests_from_2010_to_Present.csv',parse_dates = ['Cr
df
```

Out[2]:

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	...	Bridge Highway Name
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
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
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
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
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
...	...	...	...	...	...	...	...	...	...	...	...	...
300693	30281872	2015-03-29 00:33:41	NaT	NYPD	New York City Police Department	Noise - Commercial	Loud Music/Party	Club/Bar/Restaurant	NaN	CRESCENT AVENUE	...	NaN
300694	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	11418.0	100-17 87 AVENUE	...	NaN
300695	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	11206.0	162 THROOP AVENUE	...	NaN
300696	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	10461.0	3151 EAST TREMONT AVENUE	...	NaN
300697	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	10036.0	251 WEST 48 STREET	...	NaN

300698 rows × 53 columns

In [193...

```python
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  datetime64[ns]
 2   Closed Date                     298534 non-null  datetime64[ns]
 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: datetime64[ns](2), float64(10), int64(1), object(40)
memory usage: 121.6+ MB
```

In [194...

```python
df.isna().sum()*100/len(df)
```

Out[194...

```
Unique Key                          0.000000
Created Date                        0.000000
Closed Date                         0.719659
Agency                              0.000000
Agency Name                         0.000000
Complaint Type                      0.000000
Descriptor                          1.966757
Location Type                       0.043565
Incident Zip                        0.869643
Incident Address                   14.768971
Street Name                        14.768971
Cross Street 1                     16.388203
Cross Street 2                     16.554483
Intersection Street 1              85.414602
Intersection Street 2              85.579552
Address Type                        0.936155
City                                0.869311
Landmark                           99.883937
Facility Type                       0.721967
Status                              0.000000
Due Date                            0.000998
Resolution Description              0.000000
Resolution Action Updated Date      0.727308
Community Board                     0.000000
Borough                             0.000000
X Coordinate (State Plane)          1.177261
Y Coordinate (State Plane)          1.177261
Park Facility Name                  0.000000
Park Borough                        0.000000
School Name                         0.000000
School Number                       0.000000
School Region                       0.000333
School Code                         0.000333
School Phone Number                 0.000000
School Address                      0.000000
School City                         0.000000
School State                        0.000000
School Zip                          0.000333
School Not Found                    0.000000
School or Citywide Complaint      100.000000
Vehicle Type                      100.000000
Taxi Company Borough              100.000000
Taxi Pick Up Location             100.000000
Bridge Highway Name                99.919188
Bridge Highway Direction           99.919188
Road Ramp                          99.929165
Bridge Highway Segment             99.929165
Garage Lot Name                   100.000000
Ferry Direction                    99.999667
Ferry Terminal Name                99.999335
Latitude                            1.177261
Longitude                           1.177261
Location                            1.177261
dtype: float64
```

In [195...

```python
#dropping columns with more than 85% missing values
df_clean = df.drop(columns = ['Intersection Street 1','Intersection Street 2','Landmark','School or Citywide Co
```

In [196...

```python
df_clean['City'] = df_clean['City'].str.lower()
df_clean['City']
```

Out[196...

```
0               new york
1                astoria
2                  bronx
3                  bronx
4               elmhurst
               ...
300693               NaN
300694     richmond hill
300695          brooklyn
300696             bronx
300697          new york
Name: City, Length: 300698, dtype: object
```

In [197...

```python
df_clean.isna().sum()*100/len(df_clean)
```

Out[197...

```
Unique Key                         0.000000
Created Date                       0.000000
Closed Date                        0.719659
Agency                             0.000000
Agency Name                        0.000000
Complaint Type                     0.000000
Descriptor                         1.966757
Location Type                      0.043565
Incident Zip                       0.869643
Incident Address                  14.768971
Street Name                       14.768971
Cross Street 1                    16.388203
Cross Street 2                    16.554483
Address Type                       0.936155
City                               0.869311
Facility Type                      0.721967
Status                             0.000000
Due Date                           0.000998
Resolution Description             0.000000
Resolution Action Updated Date     0.727308
Community Board                    0.000000
Borough                            0.000000
X Coordinate (State Plane)         1.177261
Y Coordinate (State Plane)         1.177261
Park Facility Name                 0.000000
Park Borough                       0.000000
School Name                        0.000000
School Number                      0.000000
School Region                      0.000333
School Code                        0.000333
School Phone Number                0.000000
School Address                     0.000000
School City                        0.000000
School State                       0.000000
School Zip                         0.000333
School Not Found                   0.000000
Latitude                           1.177261
Longitude                          1.177261
Location                           1.177261
dtype: float64
```

In [199...

```python
df_clean = df_clean.dropna(subset = ['City','Latitude'])
```

In [200...

```python
df_clean.isna().sum()*100/len(df_clean)
```

Out[200...

```
Unique Key                         0.000000
Created Date                       0.000000
Closed Date                        0.018848
Agency                             0.000000
Agency Name                        0.000000
Complaint Type                     0.000000
Descriptor                         1.965556
Location Type                      0.041061
Incident Zip                       0.001010
Incident Address                  14.935194
Street Name                       14.935194
Cross Street 1                    15.872535
Cross Street 2                    15.873881
Address Type                       0.080103
City                               0.000000
Facility Type                      0.018848
Status                             0.000000
Due Date                           0.000337
Resolution Description             0.000000
Resolution Action Updated Date     0.026925
Community Board                    0.000000
Borough                            0.000000
X Coordinate (State Plane)         0.000000
Y Coordinate (State Plane)         0.000000
Park Facility Name                 0.000000
Park Borough                       0.000000
School Name                        0.000000
School Number                      0.000000
School Region                      0.000000
School Code                        0.000000
School Phone Number                0.000000
School Address                     0.000000
School City                        0.000000
School State                       0.000000
School Zip                         0.000000
School Not Found                   0.000000
Latitude                           0.000000
Longitude                          0.000000
Location                           0.000000
dtype: float64
```

In [201...

```python
df_clean.head()
```

Out[201...

	Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Location Type	Incident Zip	Incident Address	...	School Code	School Phone Number
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	...	Unspecified	Unspecified
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	...	Unspecified	Unspecified
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	...	Unspecified	Unspecified
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	...	Unspecified	Unspecified
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	...	Unspecified	Unspecified

5 rows × 39 columns

Task 2

- Convert the columns 'Created Date' and 'Closed Date' to datetime datatype
- Create a new column 'Request_Closing_Time' as the time elapsed between request creation and request closing

In [185...

```python
df_clean['Created Date'] = pd.to_datetime(df_clean['Created Date'])
df_clean['Closed Date'] = pd.to_datetime(df_clean['Closed Date'])
```

In [205...

```python
df_clean['Request Closing Time'] = df_clean['Closed Date'].values - df_clean['Created Date'].values
df_clean['Request Closing Time']
```

Out[205...

```
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
              ...
300692   0 days 00:38:29
300694   0 days 02:00:31
300695   0 days 03:07:17
300696   0 days 04:05:33
300697   0 days 04:08:49
Name: Request Closing Time, Length: 297117, dtype: timedelta64[ns]
```

In [206...

```python
df_clean['Request Closing Time'] = df_clean['Request Closing Time']/np.timedelta64(1,'m')
```

Task 3

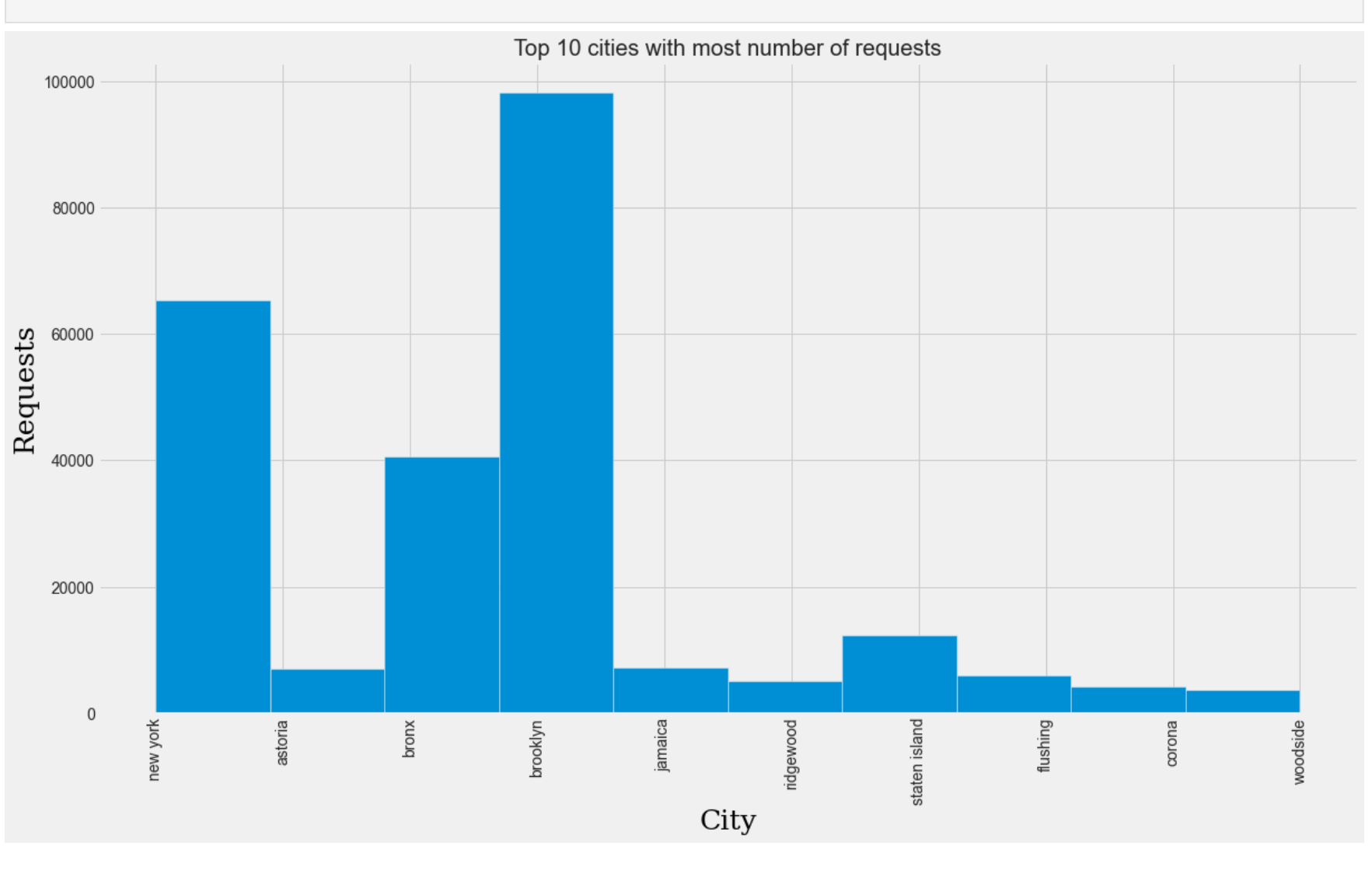
Visualization

In [116...

```python
from matplotlib import style
style.use('fivethirtyeight')
```

In [149...

```python
top10 = df_clean['City'].value_counts().head(10)
top10 = top10.index
plt.figure(figsize = (18,10))
plt.hist(data = df_clean[df_clean.City.isin(top10)], x = 'City',bins = 10)
plt.xlabel('City',color = 'black', fontsize = 25,fontfamily = 'Serif')
plt.ylabel('Requests',color = 'black', fontsize = 25, fontfamily = 'Serif')
plt.title('Top 10 cities with most number of requests')
plt.xticks(rotation = 90)
plt.show()
```

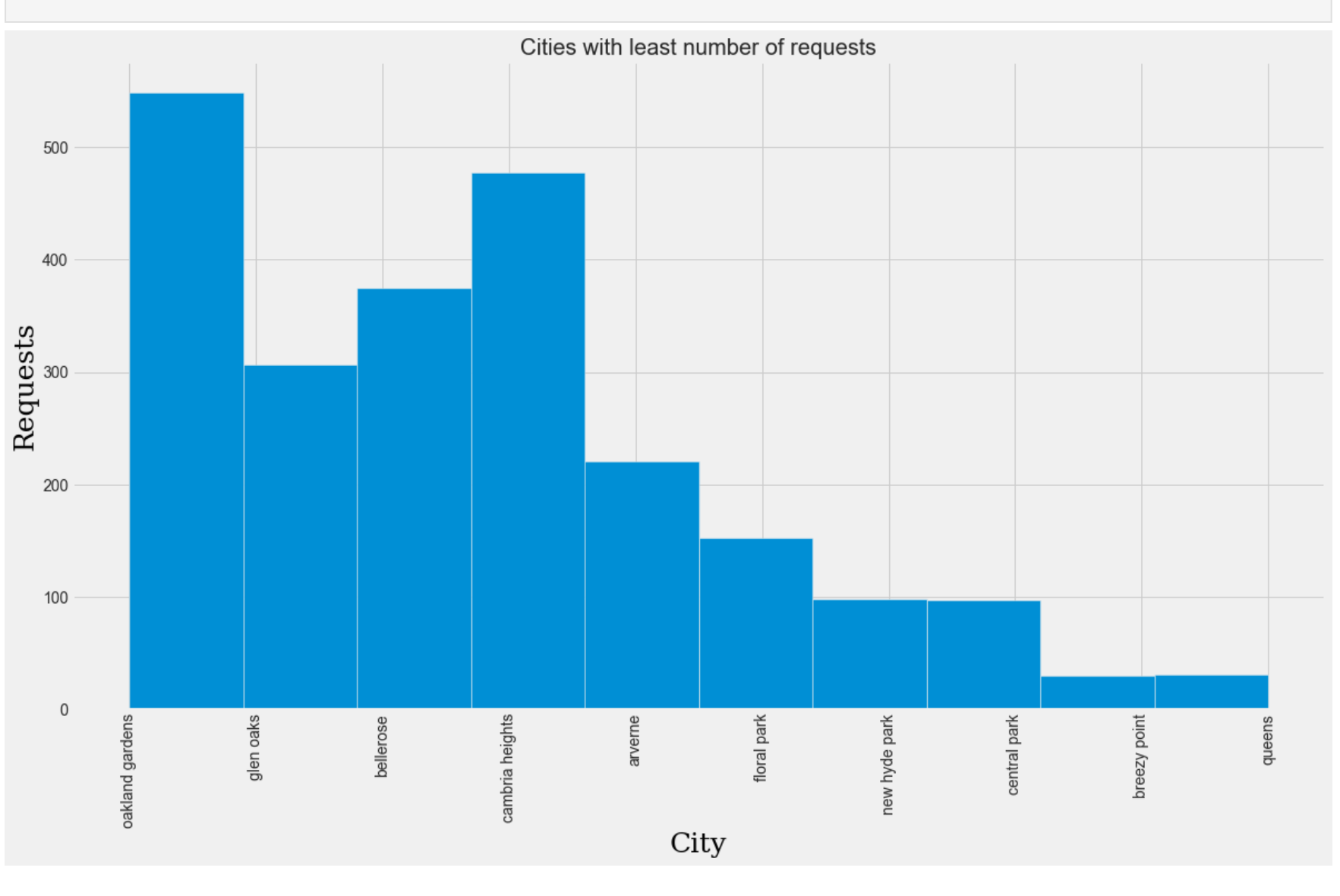


Conclusion

- Most number of complaints are received from Brooklyn

In [150...

```python
least10 = df_clean['City'].value_counts().tail(10)
least10 = least10.index
plt.figure(figsize = (18,10))
plt.hist(data = df_clean[df_clean.City.isin(least10)], x = 'City',bins = 10)
plt.xlabel('City',color = 'black', fontsize = 25,fontfamily = 'Serif')
plt.ylabel('Requests',color = 'black', fontsize = 25, fontfamily = 'Serif')
plt.title('Cities with least number of requests')
plt.xticks(rotation = 90)
plt.show()
```



Conclusion

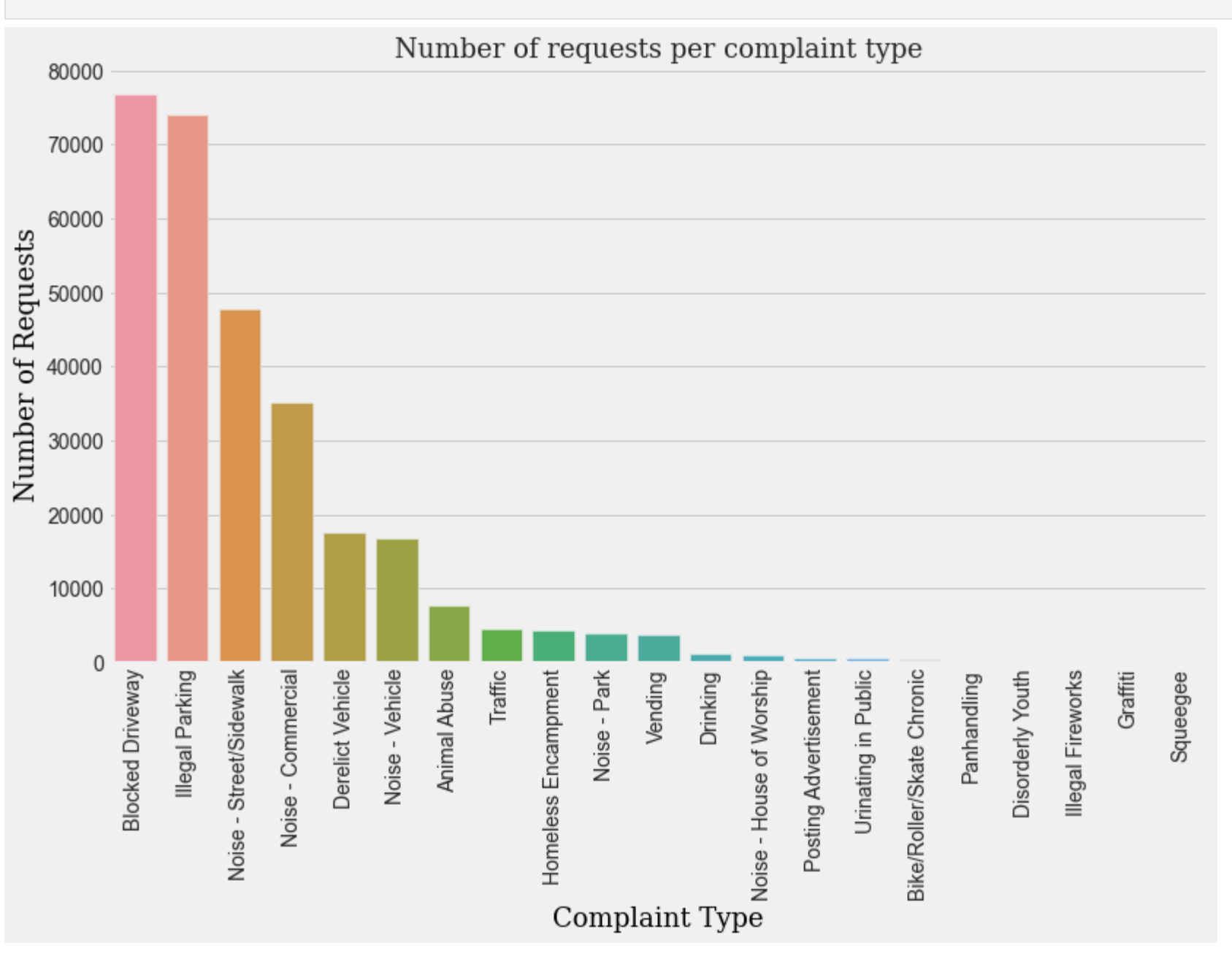
- Least number of requests were made from breezy point and queens

In [151...

```python
import seaborn as sns
```

In [152...

```python
plt.figure(figsize=(12,7))
sns.countplot(x = 'Complaint Type', data = df_clean, order = df_clean['Complaint Type'].value_counts().index)
plt.ylabel('Number of Requests', color = 'black',fontsize = 18, fontfamily = 'Serif')
plt.xlabel('Complaint Type', color = 'black',fontsize = 18, fontfamily = 'Serif')
plt.title('Number of requests per complaint type',fontsize = 18, fontfamily = 'Serif')
plt.xticks(rotation = 90)
plt.show()
```



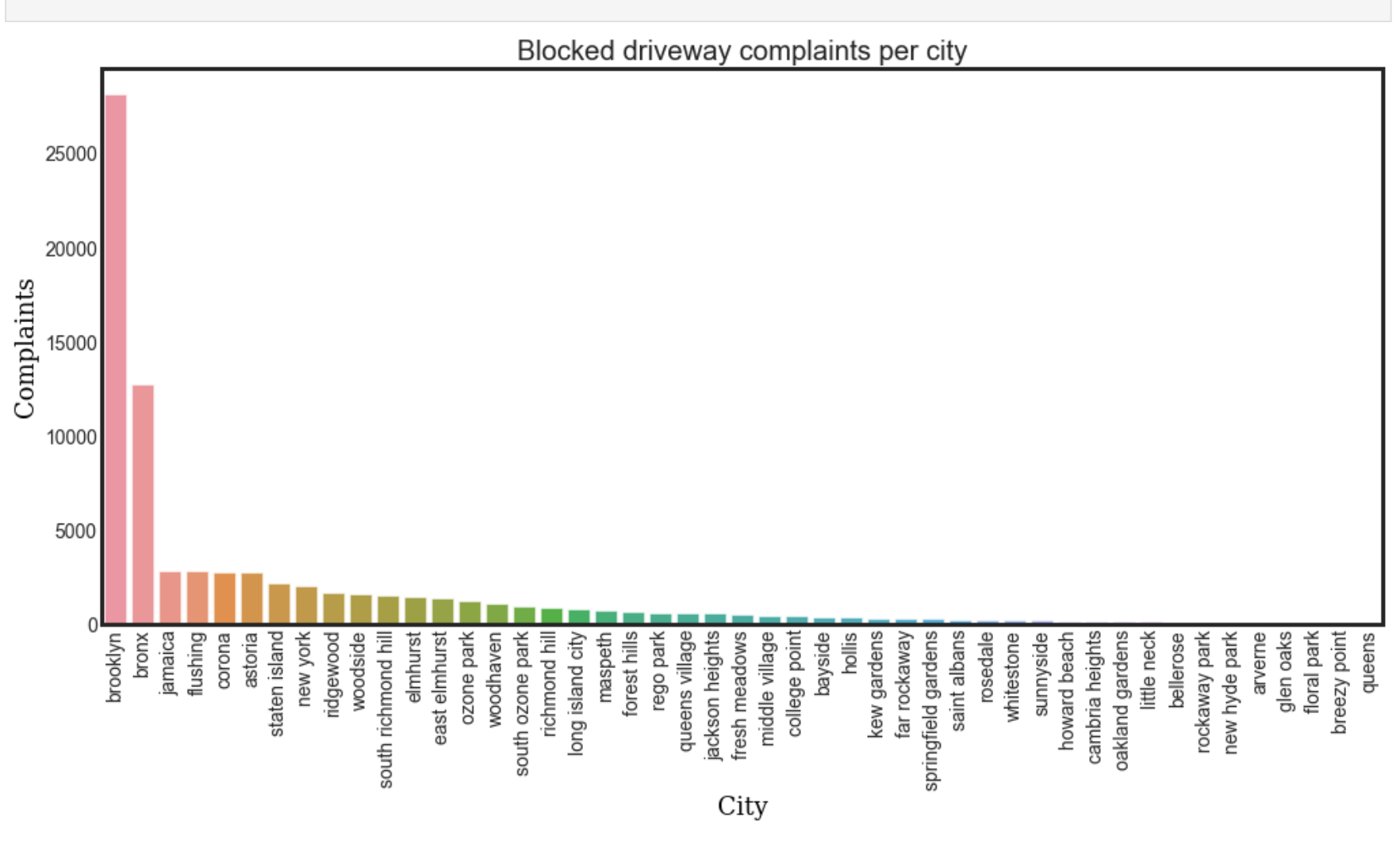
Conclusion

- Most number of complaints are recieved for blocked driveways and the least for Squeegee

In [ ]:

In [153...

```python
city = df_clean[['City','Complaint Type']]
city = city[city['Complaint Type']=='Blocked Driveway']
sns.set_style('ticks')
plt.figure(figsize=(15,7))
sns.countplot(x = 'City', data = city, order = city['City'].value_counts().index)
plt.xlabel('City',color = 'black', fontsize = 18,fontfamily = 'Serif')
plt.ylabel('Complaints',color = 'black', fontsize = 18, fontfamily = 'Serif')
plt.title('Blocked driveway complaints per city')
plt.xticks(rotation = 90)
plt.show()
```



Conclusion

- Highest number of blocked driveway complaints are from Brooklyn
- Majority of blocked driveway complaints are from Brooklyn and Bronx

In [154...

```python
df_clean['Status'].unique()
```

Out[154...

```
array(['Closed', 'Assigned', 'Open', 'Draft'], dtype=object)
```

In [155...

```python
status = df_clean[['Status','City']]
status = status[status['Status']=='Open']
new_status = status.reset_index()
new_status = new_status.loc[:,'Status':'City']
new_status
```

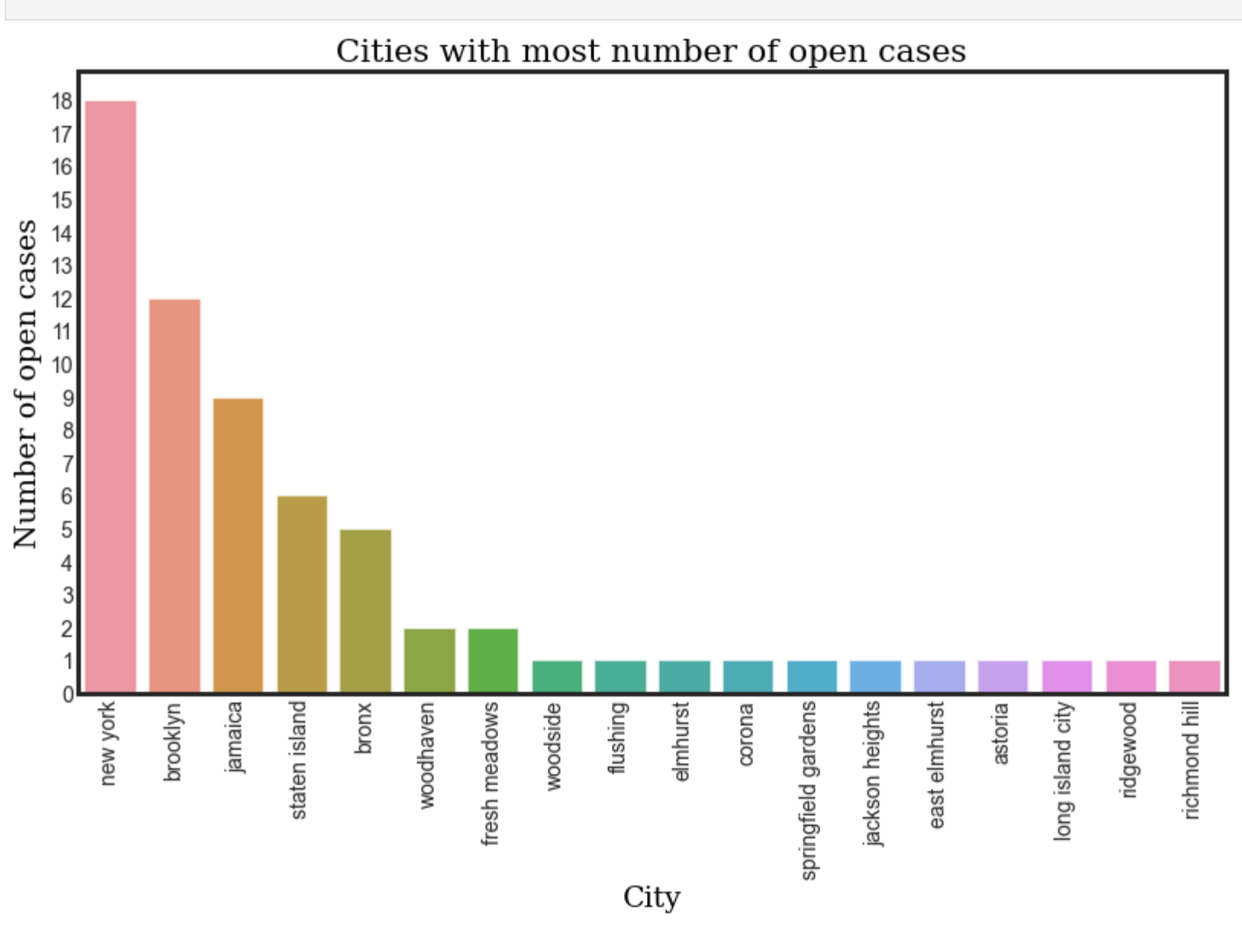
Out[155...

	Status	City
0	Open	new york
1	Open	bronx
2	Open	bronx
3	Open	new york
4	Open	richmond hill
...	...	...
60	Open	brooklyn
61	Open	astoria
62	Open	new york
63	Open	brooklyn
64	Open	new york

65 rows × 2 columns

In [156...

```python
plt.figure(figsize=(12,7))
sns.countplot(x = 'City', data = new_status, order = new_status['City'].value_counts().index)
plt.xticks(rotation = 90)
plt.yticks(range(0,19))
plt.xlabel('City',color = 'black', fontsize = 19, fontfamily = 'Serif')
plt.ylabel('Number of open cases', color = 'black', fontsize = 19, fontfamily = 'Serif')
plt.title('Cities with most number of open cases',color = 'black', fontsize = 21, fontfamily = 'Serif')
plt.show()
```



Conclusion

- Highest number of open cases are in New York
- Even though most number of cases are reported from Brooklyn, highest number of open cases are in New York

Task 4

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

In [214...

```python
#average response time in minutes for complaint type for different cities
complainttype_grouped = df_clean.groupby(['City','Complaint Type'])['Request Closing Time'].mean()
complainttype_grouped.head(25)
```

Out[214...

```
City     Complaint Type
arverne  Animal Abuse                 129.217544
         Blocked Driveway             151.558095
         Derelict Vehicle             178.093210
         Disorderly Youth             215.475000
         Drinking                      14.316667
         Graffiti                      92.000000
         Homeless Encampment          108.887500
         Illegal Parking              138.981897
         Noise - Commercial           137.233333
         Noise - House of Worship      93.731818
         Noise - Park                  77.000000
         Noise - Street/Sidewalk      119.543678
         Noise - Vehicle              111.592857
         Panhandling                   62.000000
         Urinating in Public           41.516667
         Vending                       29.000000
astoria  Animal Abuse                 300.010133
         Bike/Roller/Skate Chronic    104.358889
         Blocked Driveway             289.217849
         Derelict Vehicle             574.505877
         Disorderly Youth             174.155556
         Drinking                     283.345238
         Graffiti                     845.704167
         Homeless Encampment          295.055208
         Illegal Fireworks            166.408333
Name: Request Closing Time, dtype: float64
```

In [217...

```python
#avg response time for different complaint types
complainttypes = df_clean.groupby('Complaint Type')['Request Closing Time'].mean()
complainttypes.head(22)
```

Out[217...

```
Complaint Type
Animal Abuse                 313.084009
Bike/Roller/Skate Chronic    222.129469
Blocked Driveway             284.268343
Derelict Vehicle             440.806599
Disorderly Youth             214.128889
Drinking                     231.395974
Graffiti                     429.075074
Homeless Encampment          262.438582
Illegal Fireworks            165.726483
Illegal Parking              269.029318
Noise - Commercial           188.219341
Noise - House of Worship     191.747777
Noise - Park                 204.266807
Noise - Street/Sidewalk      206.525132
Noise - Vehicle              215.890730
Panhandling                  263.373810
Posting Advertisement        118.548200
Squeegee                     242.737500
Traffic                      206.920025
Urinating in Public          217.599859
Vending                      240.687435
Name: Request Closing Time, dtype: float64
```

Task 5

Statistical testing

- F Test

alpha value = 0.05

- Null hypothesis: There is no major difference in average response time for different complaint types
- Alternate Hypothesis: There is a major difference in average response time for different complaint types

In [ ]:

In [234...

```python
df_clean = df_clean.dropna(subset=['Request Closing Time'])
```

In [235...

```python
complaints = df_clean['Complaint Type'].value_counts().index
for i in range(len(complaints)):
    exec("sample{} = df_clean.loc[(df_clean['Complaint Type'] == '{}') , 'Request Closing Time']".format(i+1 ,c
```

In [236...

```python
#for 21 samples(complaint types)
fscore,pvalue = stats.f_oneway(sample1,sample2,sample3,sample4,sample5,sample6,sample7,sample8,sample9,sample1(
print("score : {:.2f} , pvalue : {:.2f}".format(fscore,pvalue))
```

```
score : 407.78 , pvalue : 0.00
```

Since p value is less than alpha value, we reject null hypothesis

- There is a major difference in average response time for different complaint types(overall)

- Chi square test

alpha value = 0.05

- Null hypothesis: There is no major relation between location and complaint type
- Alternate Hypothesis: There is some relation between location and complaint type

In [ ]:

In [254...

```python
comp_by_loc = pd.crosstab(df_clean['Complaint Type'],df_clean['Location'])
```

In [255...

```python
#chi square test
cscore,pval,df,et = stats.chi2_contingency(comp_by_loc)
print("score : {:.2f} , pvalue : {:.2f}".format(cscore,pval))
```

```
score : 4160248.36 , pvalue : 0.00
```

Since p value is less than alpha value we reject null hypothesis

- There is some relation between complaint type and location

In [ ]: