1. Understand the dataset:

```
import pandas as pd
import numpy as np

customer_service=pd.read_csv('311_Service_Requests_from_2010_to_Presen
t.csv')

customer_service.head()

customer_service.tail()

customer_service.info()
```

Identify the shape of the dataset

```
customer_service.shape
```

Identify variables with null values

```
customer_service.isnull().any()
```

2. Perform basic data exploratory analysis:

Utilize missing value treatment

```
customer_service.drop_duplicates(inplace=True)

customer_service

# filling empty cells
customer_service.fillna(0,inplace=True) # fill nan with 0

customer_service.isna().any()
```

Analyze the date column and remove the entries if it has an incorrect timeline

```
customer_service.columns

customer_service['Created Date']

customer_service['Created
Date']=pd.to_datetime(customer_service['Created Date'])

customer_service['Created Date']

customer_service['Closed Date']

customer_service['Closed
Date']=pd.to_datetime(customer_service['Closed Date'])

customer_service['Closed Date']

customer_service['Due Date']
```

```
customer_service['Due Date']=pd.to_datetime(customer_service['Due
Date'])

customer_service['Due Date']

customer_service['Resolution Action Updated Date']

customer_service['Resolution Action Updated
Date']=pd.to_datetime(customer_service['Resolution Action Updated
Date'])

customer_service['Resolution Action Updated Date']
```

Draw a frequency plot for city-wise complaints

```
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.colors as mcolors
from collections import Counter

df=Counter(customer_service['City'])

City=pd.DataFrame.from_dict(df,orient='index')
City

City.plot()
plt.xlabel("Cities")
plt.ylabel('no. of complaint')
plt.title('City-Wise Complaints')
plt.show()
```

Draw scatter and hexbin plots for complaint concentration across Brooklyn

```
df2=Counter(customer_service['Borough'])

Borough =pd.DataFrame.from_dict(df2,orient='index')
Borough

Brooklyn= customer_service[customer_service['Borough']=='BROOKLYN']
Brooklyn.columns

Brooklyn.plot(kind='scatter',x='X Coordinate (State Plane)',y='Y
Coordinate (State Plane)')
plt.title('complaint concentration across Brooklyn')
plt.show()

Brooklyn.plot(kind='hexbin',x='X Coordinate (State Plane)',y='Y
Coordinate (State Plane)',gridsize=30)
plt.title('complaint concentration across Brooklyn')
plt.show()
```

3. Find major types of complaints:

Plot a bar graph of count vs. complaint types

```
df3=Counter(customer_service['Complaint Type'])

Complaint=pd.DataFrame.from_dict(df3,orient='index')
Complaint

Complaint.plot(kind='barh',color='r')
plt.xlabel('Counts')
plt.ylabel('complaints type')
plt.title('bar graph of count vs. complaint type')
plt.show()
```

Find the top 10 types of complaints

```
df4=Counter(customer_service['Descriptor'])

Descriptor=pd.DataFrame.from_dict(df4,orient='index')
Descriptor

Descriptor.head(10).plot(kind='barh',color='g')
plt.xlabel('Counts')
plt.ylabel('Descriptor')
plt.title('top 10 types of complaints')
plt.show()
```

Display the types of complaints in each city in a separate dataset

```
Borough.info()

Brooklyn= customer_service[customer_service['Borough']=='BROOKLYN']
Brooklyn.head()

Manhattan= customer_service[customer_service['Borough']=='MANHATTAN']
Manhattan.head()

Queens= customer_service[customer_service['Borough']=='QUEENS']
Queens.head()

Bronx= customer_service[customer_service['Borough']=='BRONX']
Bronx.head()

Staten_Island=customer_service[customer_service['Borough']=='STATEN
ISLAND']
Staten_Island.head()

Unspecified=customer_service[customer_service['Borough']=='Unspecified
']
Unspecified.head()
```

4. Visualize the major types of complaints in each city

```
(Brooklyn['Complaint Type'].value_counts()).plot(kind='bar',
          figsize=(10,6),title ='Complaints in Brooklyn')
plt.show()

(Manhattan['Complaint Type'].value_counts()).plot(kind='bar',
          figsize=(10,6),title ='Complaints in Manhattan')
plt.show()

(Queens['Complaint Type'].value_counts()).plot(kind='bar',
          figsize=(10,6),title ='Complaints in Queens')
plt.show()

(Bronx['Complaint Type'].value_counts()).plot(kind='bar',
          figsize=(10,6),title ='Complaints in Bronx')
plt.show()

(Unspecified['Complaint Type'].value_counts()).plot(kind='bar',
          figsize=(10,6),title ='Complaints in Unspecified')
plt.show()
```

5. Check if the average response time across various types of complaints

```
# add response time column by deriving it
customer_service['Response Time']=customer_service['Resolution Action
Updated Date']-customer_service['Created Date']
response_time=customer_service['Response Time']

df5=Counter(customer_service['Response Time'])

response_time=pd.DataFrame.from_dict(df5,orient='index')
response_time

plt.scatter(x="Complaint",y="response_time")
plt.title('average response time')
plt.show()
```

6. Identify significant variables by performing a statistical analysis using p-values and chi-square values (Optional)

```
fs=['Unique Key','Created Date','Closed Date','Agency Name',
    'Complaint Type','Descriptor','Location Type','Incident
Zip','City',
    'Status','Due Date']

analysis=customer_service[fs]

analysis.isna().sum()
analysis.dtypes

analysis.head()

analysis=pd.get_dummies(analysis)

analysis

analysis.corr()
```