```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import style
import seaborn as sns
import datetime
import scipy.stats as stats
from math import sqrt

#1 Import a 311 NYC service request.
service311 = pd.read_csv ('311_Service_Requests_from_2010_to_Present.csv')

#2 Read or convert the columns ‘Created Date’ and Closed Date’ to datetime datatype
service311['Created Date'] = pd.to_datetime(service311['Created Date'])
print(service311['Created Date'].dtype)
service311['Closed Date'] = pd.to_datetime(service311['Closed Date'])
print(service311['Closed Date'].dtype)

#3 create a new column ‘Request_Closing_Time’ as the time elapsed between request creation and request closing.
service311["Request_Closing_Time"] = service311["Closed Date"] - service311["Created Date"]
print(service311['Request_Closing_Time'].head())

#4a Provide major insights/patterns that you can offer in a visual format (tables);
print(service311.head())
print(service311.shape)
print(service311.columns)
print(service311.dtypes)

#4b Provide major insights/patterns that you can offer in a visual format (graphs)

#4b.1 Have a look at the status of tickets
complaintTypecity = pd.DataFrame({'count':service311.groupby(['Complaint Type','City']).size()}).reset_index()
service311.groupby(['Borough','Complaint Type','Descriptor']).size()
service311['Status'].value_counts().plot(kind='bar',alpha=0.6,figsize=(7,7))
plt.show()

#4b.2 Scatter plot displaying all the cities that raised complaint of type 'Blocked Driveway'
service311['City'].dropna(inplace=True)
groupedby_complainttype = service311.groupby('Complaint Type')
grp_data = groupedby_complainttype.get_group('Blocked Driveway')
grp_data['City'].fillna('Unknown City', inplace =True)
plt.figure(figsize=(20, 15))
plt.scatter(grp_data['Complaint Type'],grp_data['City'])
plt.title('Plot showing list of cities that raised complaint of type Blocked Driveway')
plt.show()

#4b.3 & 4b.4 Complaint type Breakdown with bar plot to figure out majority of complaint types and top 10 complai
nts
service311[service311['Closed Date'].isnull()]
service311['Complaint Type'].unique()
service311['Descriptor'].unique()
service311['Complaint Type'].value_counts().head(10).plot(kind='barh',figsize=(5,5));
```

```
service311.groupby(["Borough","Complaint Type","Descriptor"]).size()
majorcomplints=service311.dropna(subset=["Complaint Type"])
majorcomplints=service311.groupby("Complaint Type")
sortedComplaintType = majorcomplints.size().sort_values(ascending = False)
sortedComplaintType = sortedComplaintType.to_frame('count').reset_index()
print(sortedComplaintType.head(10))
sortedComplaintType = sortedComplaintType.head()
plt.figure(figsize=(5,5))
plt.pie(sortedComplaintType['count'],labels=sortedComplaintType["Complaint Type"], autopct="%1.1f%%")
plt.show()

# 5: Perform a statistical test for the following:
# H0 : All Complain Types average response time mean is similar
# H1 : Not similar

top5_complaints_type = service311['Complaint Type'].value_counts()[:5]
print(top5_complaints_type)
top5_complaints_type_names = top5_complaints_type.index

print(top5_complaints_type_names)
print(service311['Complaint Type'].isin(top5_complaints_type_names))

sample_data = service311.loc[service311['Complaint Type'].isin(top5_complaints_type_names), ['Complaint Type', 'Request_Closing_Time']]
sample_data.head()
sample_data.dropna(how='any', inplace=True)
sample_data.isnull().sum()

s1 = sample_data[sample_data['Complaint Type'] == top5_complaints_type_names[0]].Request_Closing_Time
print(s1.head())

s2 = sample_data[sample_data['Complaint Type'] == top5_complaints_type_names[1]].Request_Closing_Time
print(s2.head())

s3 = sample_data[sample_data['Complaint Type'] == top5_complaints_type_names[2]].Request_Closing_Time
print(s3.head())

s4 = sample_data[sample_data['Complaint Type'] == top5_complaints_type_names[3]].Request_Closing_Time
print(s4.head())

s5 = sample_data[sample_data['Complaint Type'] == top5_complaints_type_names[4]].Request_Closing_Time
print(s5.head())
print(s1.isnull().sum())
print(s2.isnull().sum())
print(s3.isnull().sum())
print(s4.isnull().sum())
print(s5.isnull().sum())
print(stats.f_oneway(s1, s2, s3, s4, s5))

# We can see pvalue is less than 0.05 so we reject null hypothesis and average response time is not same.
```