**Analyzing NYC311 Data**

**Introduction**

This document outlines the plan for analyzing the NYC311 data to answer two primary questions:

1. How did Hurricane Sandy impact New York City base on NYC311 data?
2. What was the impact of COVID-19 on New York City based on NYC311 data?

The analysis involves summarizing and visualizing the data to identify patterns and trends related to these events.

**Plan for Data Analysis**

The analysis will include the following steps:

1. **Data Preprocessing**: Convert date columns to datetime format, handle missing values, and filter data based on the specified time frames.
2. **Data Summarization**: Summarize the total number of service requests and categorize the types of service requests before and after each event.
3. **Data Visualization**: Plot the number of service requests over time, create bar charts for common complaint types, and map the geographical distribution of requests.
4. **Specific Analyses**:
   * Hurricane Sandy: Compare service requests, identify common complaint types, map affected areas, and identify agencies with the highest requests.
   * COVID-19: Compare service requests, identify common complaint types, map affected areas, and identify agencies with the highest requests.

**Sample Code for the Analysis**

`

import pandas as pd

import matplotlib.pyplot as plt

class Read311Data:

def \_\_init\_\_(self, path):

self.path = path

def loadFile(self):

file = pd.read\_csv(self.path)

return file

def main():

# Reads the CSV File

file = Read311Data('./311data.csv')

data = file.loadFile()

# Convert date columns to datetime

data['created\_date'] = pd.to\_datetime(data['created\_date'])

data['closed\_date'] = pd.to\_datetime(data['closed\_date'], errors='coerce')

data['resolution\_date'] = pd.to\_datetime(data['resolution\_date'], errors='coerce')

# Filter data for Hurricane Sandy period

preHurricaneData = data[(data['created\_date'] >= '2012-09-28') & (data['created\_date'] < '2012-10-29')]

postHurricaneData = data[(data['created\_date'] >= '2012-10-29') & (data['created\_date'] <= '2012-11-29')]

# Filter data for COVID-19 period

preCovidData = data[(data['created\_date'] >= '2020-02-01') & (data['created\_date'] < '2020-03-22')]

postCovidData = data[(data['created\_date'] >= '2020-03-22') & (data['created\_date'] <= '2020-06-01')]

# Summarize and visualize data

def summarize\_data(df, period):

if df.empty:

print(f"No data available for {period}.")

return

print(f"Summary for {period}:")

print(df['complaint\_type'].value\_counts().head(10))

df['complaint\_type'].value\_counts().head(10).plot(kind='bar')

plt.title(f"Top 10 Complaint Types - {period}")

plt.show()

# Summarize and visualize Hurricane Sandy data

summarize\_data(preHurricaneData, 'Before Hurricane Sandy')

summarize\_data(postHurricaneData, 'After Hurricane Sandy')

# Summarize and visualize COVID-19 data

summarize\_data(preCovidData, 'Before COVID-19 Lockdown')

summarize\_data(postCovidData, 'After COVID-19 Lockdown')

if \_\_name\_\_ == "\_\_main\_\_":

main()

**`**