

# Práctica de laboratorio - Delitos en San Francisco

Litzy Yulissa Nevarez Garcia

19040372

## Objetivos

Demuestre sus conocimientos acerca del ciclo activo del análisis de datos mediante un conjunto de datos determinado, herramientas, Python y la libreta de anotaciones de Jupyter.

Parte 1: Importación de paquetes Python

Parte 2: Carga de datos

Parte 3: Preparación de datos

Parte 4: Análisis de datos

Parte 5: Visualización de datos

## Contexto/Situación

En esta práctica de laboratorio, importará algunos paquetes de Python que nos necesarios para analizar un conjunto de datos que contiene información sobre delitos en San Francisco. Luego utilizará Python y Jupyter Notebook para preparar estos datos para su análisis, analizarlos, graficarlos y comunicar sus conclusiones.

## Recursos necesarios

- 1 PC con acceso a Internet
- Raspberry Pi versión 2 o superior
- Bibliotecas de Python: pandas, numpy, matplotlib, folium, datetime y csv
- Archivos de datos: Map-Crime\_Incidents-Previous\_Three\_Months.csv

## Parte 1: Importación de paquetes de Python

En esta sección, importará los paquetes de Python que son necesarios para el resto de esta práctica de laboratorio.

**numpy**

NumPy es el paquete fundamental para computación científica con Python. Contiene entre otras cosas: un potente objeto de matriz N dimensional y funciones sofisticadas (difusión).

### pandas

Pandas es un código abierto, una biblioteca con licencia BSD autorizada que proporciona un alto rendimiento, estructuras de datos fáciles de usar y herramientas de análisis de datos para el lenguaje de programación de Python.

### matplotlib

Matplotlib es una biblioteca gráfica para el lenguaje de programación de Python y su extensión matemática numérica NumPy.

### folium

Folium es una biblioteca para crear mapas interactivos.

```
In [1]: # Code cell 1
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import folium
```

## Parte 2: Carga de datos

En esta sección, cargará el conjunto de datos de delitos en San Francisco y los paquetes de Python necesarios para analizarlo y visualizarlo.

### Paso 1: Cargar los datos delitos en San Francisco en un marco de datos

En este paso, se importarán los datos de delitos en San Francisco de un archivo de valores separado por comas (csv) en una trama de datos.

```
In [2]: # code cell 2
# This should be a local path
dataset_path = 'C:/Users/yulis/Analitica de los Datos en las organizaciones/Data/Map-Crime-Incidents-Previous-Three-Months.csv'
# read the original dataset (in comma separated values format) into a DataFrame
SF = pd.read_csv(dataset_path)
```

Para ver las primeras cinco líneas del archivo csv, se utilizará el comando head de Linux.

```
In [3]: # code cell 3
!head -n 5 ./Data/Map-Crime-Incidents-Previous-Three-Months.csv
SF.head()
```

"head" no se reconoce como un comando interno o externo,  
programa o archivo por lotes ejecutable.

Out[3]:

	IncidentNum	Category	Descript	DayOfWeek	Date	Time	PdDistrict	Resoluti
0	NaN	LARCENY/THEFT	GRAND THEFT FROM UNLOCKED AUTO	Sunday	08/31/2014 07:00:00 AM +0000	20:30	CENTRAL	NO
1	NaN	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Sunday	08/31/2014 07:00:00 AM +0000	14:30	CENTRAL	NO
2	NaN	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Sunday	08/31/2014 07:00:00 AM +0000	11:30	CENTRAL	NO
3	NaN	DRUG/NARCOTIC	POSSESSION OF METH- AMPHETAMINE	Sunday	08/31/2014 07:00:00 AM +0000	17:49	MISSION	ARRE BOOK
4	NaN	DRUG/NARCOTIC	POSSESSION OF COCAINE	Sunday	08/31/2014 07:00:00 AM +0000	18:05	NORTHERN	ARRE BOOK

## Paso 2: Ver los datos importados

A) Al ingresar el nombre del marco de datos en una celda, puede visualizar las filas superior e inferior de manera estructurada.

In [4]:

```
# Code cell 4
pd.set_option('display.max_rows', 10) #Visualize 10 rows
SF
```

Out[4]:

	IncidentNum	Category	Descript	DayOfWeek	Date	Time	PdDistrict	Resolution
0	NaN	LARCENY/THEFT	GRAND THEFT FROM UNLOCKED AUTO	Sunday	08/31/2014 07:00:00 AM +0000	20:30	CENTRAL	
1	NaN	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Sunday	08/31/2014 07:00:00 AM +0000	14:30	CENTRAL	
2	NaN	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Sunday	08/31/2014 07:00:00 AM +0000	11:30	CENTRAL	
3	NaN	DRUG/NARCOTIC	POSSESSION OF METH-AMPHETAMINE	Sunday	08/31/2014 07:00:00 AM +0000	17:49	MISSION	
4	NaN	DRUG/NARCOTIC	POSSESSION OF COCAINE	Sunday	08/31/2014 07:00:00 AM +0000	18:05	NORTHERN	
...	...	...	...	...	...	...	...	...
30755	NaN	LARCENY/THEFT	PETTY THEFT SHOPLIFTING	Sunday	06/01/2014 07:00:00 AM +0000	15:30	SOUTHERN	
30756	NaN	OTHER OFFENSES	DRIVERS LICENSE, SUSPENDED OR REVOKED	Sunday	06/01/2014 07:00:00 AM +0000	16:00	NORTHERN	
30757	NaN	ASSAULT	BATTERY	Sunday	06/01/2014 07:00:00 AM +0000	15:00	TENDERLOIN	
30758	NaN	ASSAULT	ASSAULT WITH CAUSTIC CHEMICALS	Sunday	06/01/2014 07:00:00 AM +0000	15:20	CENTRAL	
30759	NaN	OTHER OFFENSES	DRIVERS LICENSE, SUSPENDED OR REVOKED	Sunday	06/01/2014 07:00:00 AM +0000	13:15	INGLESIDE	

30760 rows × 12 columns

b) Utilice la función columns para ver el nombre de las variables en el marco de datos.

In [5]:

```
# Code cell 5
SF.columns
```

Out[5]:

```
Index(['IncidentNum', 'Category', 'Descript', 'DayOfWeek', 'Date', 'Time',
      'PdDistrict', 'Resolution', 'Address', 'X', 'Y', 'Location'],
      dtype='object')
```

¿Cuántas variables se incluyen en el marco de datos de SF (ignore el índice)?

## 12 Variables

c) Utilice la función len para determinar la cantidad de filas en el conjunto de datos.

```
In [6]: # Code cell 6
len(SF)
```

```
Out[6]: 30760
```

## Parte 3: Preparación de datos

Ahora que cuenta con los datos cargados en el entorno de trabajo y determinó el análisis que desea realizar, es momento de preparar los datos para el análisis.

**Paso 1: Extraer el mes y el día del campo de Date (Fecha)** lambda es una palabra clave de Python que define las reconocidas funciones anónimas. lambda le permite especificar una función en una línea de código sin utilizar def y sin definir un nombre específico para ella. La sintaxis para una expresión lambda es:

*Parámetros lambda: expresión.*

En este caso, la función lambda se utiliza para crear una función en línea que seleccione solo los dígitos del mes de la variable Date (Fecha) e int para convertir una representación de secuencia en un valor entero. Luego, la función pandas apply se utiliza para aplicar esta función a una columna entera (en práctica, apply define implícitamente un bucle for y pasa una por una las filas a la función lambda). El mismo procedimiento se puede hacer para el día.

```
In [7]: # Code cell 7
SF['Month'] = SF['Date'].apply(lambda row: int(row[0:2]))
SF['Day'] = SF['Date'].apply(lambda row: int(row[3:5]))
```

Para verificar que estas dos variables se agreguen al marco de datos de SF, use la función print para imprimir algunos de los valores de estas columnas y type para verificar que estas columnas nuevas contengan, de hecho, valores numéricos.

```
In [8]: # Code cell 8
print(SF['Month'][0:5])
print(SF['Day'][0:5])

0      8
1      8
2      8
3      8
4      8
Name: Month, dtype: int64
0     31
1     31
2     31
3     31
4     31
Name: Day, dtype: int64
```

```
In [9]: # Code cell 9
print(type(SF['Month'][0]))

<class 'numpy.int64'>
```

## Paso 2: Eliminar las variables del marco de datos de SF

a) La columna IncidntNum contiene varias celdas con NaN. En este caso, faltan los datos. Además, IncidntNum no proporciona ningún valor el análisis. La columna se puede descartar del marco de datos. Una manera de eliminar variables que no desea en un marco de datos es mediante la función del.

```
In [10]: # Code cell 10
del SF['IncidntNum']
```

b) De manera similar, el atributo Location no se incluirá en este análisis. Puede descartarse del marco de datos.

O bien, puede usar la función drop en el marco de datos y especificar que el eje es el 1 (0 para las filas) y que el comando no requiere una asignación a otro valor para almacenar el resultado (inplace = True ).

```
In [11]: # Code cell 11
SF.drop('Location', axis=1, inplace=True )
```

c) Verifique que se hayan eliminado las columnas.

```
In [12]: # Code cell 12
SF.columns
```

```
Out[12]: Index(['Category', 'Descript', 'DayOfWeek', 'Date', 'Time', 'PdDistrict',
              'Resolution', 'Address', 'X', 'Y', 'Month', 'Day'],
              dtype='object')
```

## Parte 4: Análisis de datos

Ahora que el marco de datos se ha elaborado con datos, es momento de analizar los datos.

### Paso 1: Resumir las variables para obtener información estadística

a) Utilice la función value\_counts para resumir la cantidad de delitos cometidos por tipo; luego seleccione print para visualizar los contenidos de la variable CountCategory.

```
In [13]: # Code cell 13
CountCategory = SF['Category'].value_counts()
print(CountCategory)
```

LARCENY/THEFT	8205
OTHER OFFENSES	4004
NON-CRIMINAL	3653
ASSAULT	2518
VEHICLE THEFT	1885

...

SEX OFFENSES, NON FORCIBLE	5
BAD CHECKS	3
GAMBLING	1
PORNOGRAPHY/OBSCENE MAT	1
BRIBERY	1

Name: Category, Length: 36, dtype: int64

b) De manera predeterminada, los conteos se ordenan de forma descendente. El valor del parámetro opcional ascendente se puede configurar en True para invertir este comportamiento.

```
In [14]: # Code cell 14
SF['Category'].value_counts(ascending=True)
```

```
Out[14]: PORNOGRAPHY/OBSCENE MAT      1
          GAMBLING                    1
          BRIBERY                     1
          BAD CHECKS                   3
          SEX OFFENSES, NON FORCIBLE  5
          ...
          VEHICLE THEFT               1885
          ASSAULT                     2518
          NON-CRIMINAL                3653
          OTHER OFFENSES              4004
          LARCENY/THEFT               8205
          Name: Category, Length: 36, dtype: int64
```

¿Qué tipo de delitos se cometió más?

### Se cometieron mas crímenes de tipo Robo

c) Al jerarquizar las dos funciones en un comando, puede lograr el mismo resultado con una línea de código.

```
In [15]: # Code cell 15
print(SF['Category'].value_counts(ascending=True))
```

```
PORNOGRAPHY/OBSCENE MAT      1
          GAMBLING                    1
          BRIBERY                     1
          BAD CHECKS                   3
          SEX OFFENSES, NON FORCIBLE  5
          ...
          VEHICLE THEFT               1885
          ASSAULT                     2518
          NON-CRIMINAL                3653
          OTHER OFFENSES              4004
          LARCENY/THEFT               8205
          Name: Category, Length: 36, dtype: int64
```

**Pregunta de desafío:** ¿Qué PdDistrict presentaba la mayoría de los incidentes de delitos informados? Proporcione los comandos de Python utilizados para respaldar su respuesta.

*Escriba su respuesta aquí*

```
In [16]: print(SF['PdDistrict'].value_counts(ascending=True))
```

```
RICHMOND      1622
PARK           1800
TARAVAL       2038
TENDERLOIN    2449
INGLESIDE     2613
BAYVIEW       2970
NORTHERN      3205
CENTRAL       3867
MISSION       4011
SOUTHERN      6185
Name: PdDistrict, dtype: int64
```

### **Paso 2: Crear subconjuntos de datos y organizarlos en marcos de datos más pequeños**

a) La indexación lógica se puede utilizar para seleccionar únicamente las filas en las cuales se cumple una condición específica. Por ejemplo, el código siguiente recupera sólo los delitos cometidos en agosto y guarda el resultado en un nuevo marco de datos.

```
In [17]: # Code cell 16
AugustCrimes = SF[SF['Month'] == 8]
AugustCrimes
```



Out[17]:

	Category	Descript	DayOfWeek	Date	Time	PdDistrict	Resolution	
0	LARCENY/THEFT	GRAND THEFT FROM UNLOCKED AUTO	Sunday	08/31/2014 07:00:00 AM +0000	20:30	CENTRAL	NONE	H CAL
1	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Sunday	08/31/2014 07:00:00 AM +0000	14:30	CENTRAL	NONE	CO JACI
2	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Sunday	08/31/2014 07:00:00 AM +0000	11:30	CENTRAL	NONE	SUT STI
3	DRUG/NARCOTIC	POSSESSION OF METH-AMPHETAMINE	Sunday	08/31/2014 07:00:00 AM +0000	17:49	MISSION	ARREST, BOOKED	1 MIS
4	DRUG/NARCOTIC	POSSESSION OF COCAINE	Sunday	08/31/2014 07:00:00 AM +0000	18:05	NORTHERN	ARREST, BOOKED	LAF O
...	...	...	...	...	...	...	...	...
9715	NON-CRIMINAL	AIDED CASE, MENTAL DISTURBED	Friday	08/01/2014 07:00:00 AM +0000	19:55	MISSION	NONE	110I of P
9716	OTHER OFFENSES	MISCELLANEOUS INVESTIGATION	Friday	08/01/2014 07:00:00 AM +0000	22:47	RICHMOND	NONE	150I BRO
9717	ASSAULT	THREATS AGAINST LIFE	Friday	08/01/2014 07:00:00 AM +0000	23:55	BAYVIEW	NONE	40I of
9718	DRIVING UNDER THE INFLUENCE	DRIVING WHILE UNDER THE INFLUENCE OF ALCOHOL	Friday	08/01/2014 07:00:00 AM +0000	23:38	NORTHERN	ARREST, BOOKED	1 C LAC
9719	SEX OFFENSES, FORCIBLE	ASSAULT TO RAPE WITH BODILY FORCE	Friday	08/01/2014 07:00:00 AM +0000	00:01	MISSION	NONE	100I of P

9720 rows × 12 columns



¿Cuántos incidentes de delitos hubo en agosto?

*Escriba su respuesta aquí*In [18]: `AugustCrimes.count() #9720`

```
Out[18]: Category      9720
          Descript     9720
          DayOfWeek     9720
          Date          9720
          Time          9720
          ...
          Address      9720
          X            9720
          Y            9720
          Month        9720
          Day          9720
          Length: 12, dtype: int64
```

¿Cuántos robos se informaron en agosto?

*Escriba su respuesta aquí*

```
In [19]: AugustCrimes["Category"].value_counts() #2697
```

```
Out[19]: LARCENY/THEFT          2697
          OTHER OFFENSES      1297
          NON-CRIMINAL        1164
          ASSAULT              756
          VEHICLE THEFT       566
          ...
          FAMILY OFFENSES      3
          LOITERING            3
          SEX OFFENSES, NON FORCIBLE 3
          PROSTITUTION         2
          GAMBLING             1
          Name: Category, Length: 32, dtype: int64
```

```
In [20]: # code cell 17
          # Possible code for the question: How many burglaries were reported in the month of Au
          AugustCrimes = SF[SF['Month'] == 8]
          AugustCrimesB = SF[SF['Category'] == 'BURGLARY']
          len(AugustCrimesB)
```

```
Out[20]: 1257
```

B) Para crear un subconjunto del marco de datos de SF para un día específico, use la función query para comparar el mes y el día al mismo tiempo.

```
In [21]: # Code cell 18
          Crime0704 = SF.query('Month == 7 and Day == 4')
          Crime0704
```

Out[21]:

	Category	Descript	DayOfWeek	Date	Time	PdDistrict	Resolution
<b>19087</b>	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Friday	07/04/2014 07:00:00 AM +0000	22:30	SOUTHERN	NONE
<b>19088</b>	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Friday	07/04/2014 07:00:00 AM +0000	18:15	SOUTHERN	NONE
<b>19089</b>	BURGLARY	BURGLARY,RESIDENCE UNDER CONSTRT, FORCIBLE ENTRY	Friday	07/04/2014 07:00:00 AM +0000	00:50	TARAVAl	NONE
<b>19090</b>	NON-CRIMINAL	LOST PROPERTY	Friday	07/04/2014 07:00:00 AM +0000	19:00	PARK	NONE
<b>19091</b>	ASSAULT	BATTERY	Friday	07/04/2014 07:00:00 AM +0000	21:00	NORTHERN	NONE
...	...	...	...	...	...	...	...
<b>19423</b>	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Friday	07/04/2014 07:00:00 AM +0000	19:25	SOUTHERN	NONE
<b>19424</b>	OTHER OFFENSES	LOST/STOLEN LICENSE PLATE	Friday	07/04/2014 07:00:00 AM +0000	11:00	INGLESIDE	NONE
<b>19425</b>	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Friday	07/04/2014 07:00:00 AM +0000	20:30	SOUTHERN	NONE
<b>19426</b>	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Friday	07/04/2014 07:00:00 AM +0000	08:00	SOUTHERN	NONE
<b>19427</b>	LARCENY/THEFT	PETTY THEFT OF PROPERTY	Friday	07/04/2014 07:00:00 AM +0000	15:30	RICHMOND	NONE

341 rows × 12 columns

In [22]:

```
# Code cell 19
SF.columns
```

Out[22]:

```
Index(['Category', 'Descript', 'DayOfWeek', 'Date', 'Time', 'PdDistrict',
      'Resolution', 'Address', 'X', 'Y', 'Month', 'Day'],
      dtype='object')
```

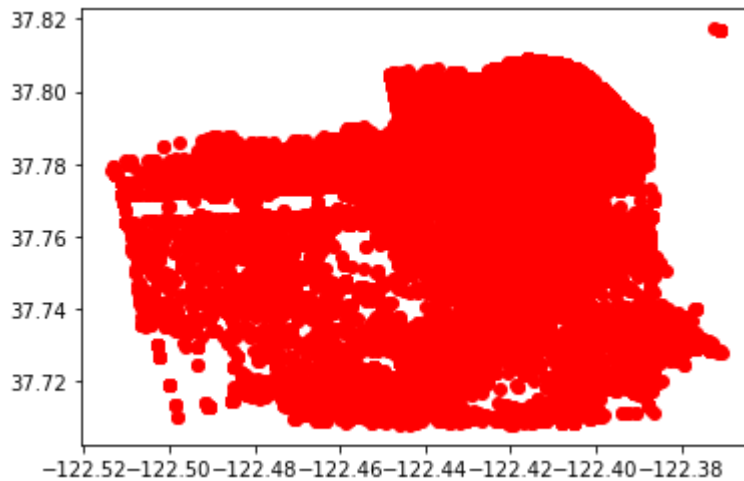
## Parte 5: Presentación de datos

La visualización y presentación de datos proporciona una descripción general inmediata que puede no ser evidente simplemente observando los datos sin procesar. El marco de datos de SF contiene las coordenadas de longitud y latitud que se pueden utilizar para graficar los datos.

**Paso 1: Graficar el marco de datos de SF con las mismas variables X e Y**

a) Utilice la función plot() para graficar el marco de datos de SF. Utilice el parámetro opcional para crear el gráfico en rojo y configurar la forma del marcador en un círculo utilizando ro.

```
In [23]: # Code cell 20
plt.plot(SF['X'],SF['Y'], 'ro')
plt.show()
```



b) Identifique la cantidad de distritos de departamentos policiales y luego cree el diccionario pd\_districts para asociar la secuencia a un valor entero.

```
In [24]: # Code cell 21
pd_districts = np.unique(SF['PdDistrict'])
pd_districts_levels = dict(zip(pd_districts, range(len(pd_districts))))
pd_districts_levels
```

```
Out[24]: {'BAYVIEW': 0,
          'CENTRAL': 1,
          'INGLESIDE': 2,
          'MISSION': 3,
          'NORTHERN': 4,
          'PARK': 5,
          'RICHMOND': 6,
          'SOUTHERN': 7,
          'TARAVAL': 8,
          'TENDERLOIN': 9}
```

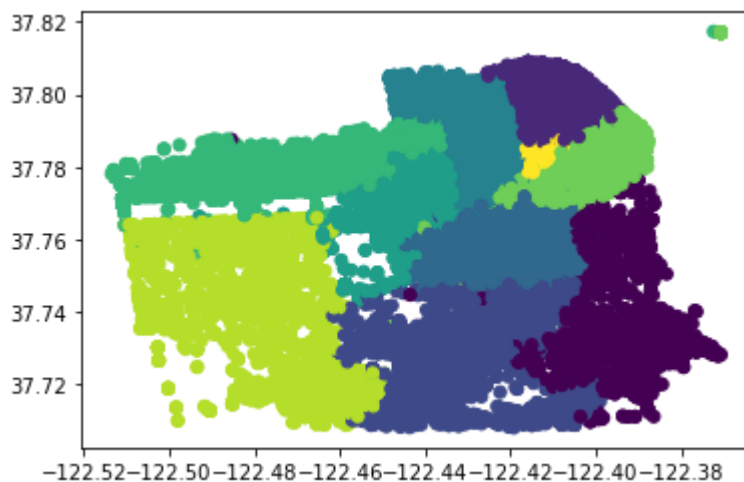
c) Utilice apply y lambda para agregar la ID del valor entero del departamento policial a una nueva columna para el marco de datos.

```
In [25]: # Code cell 22
SF['PdDistrictCode'] = SF['PdDistrict'].apply(lambda row: pd_districts_levels[row])
SF['PdDistrictCode']
```

```
Out[25]:
0      1
1      1
2      1
3      3
4      4
..
30755   7
30756   4
30757   9
30758   1
30759   2
Name: PdDistrictCode, Length: 30760, dtype: int64
```

d) Utilice el PdDistrictCode creado recientemente para cambiar automáticamente el color.

```
In [26]: # Code cell 23
#plt.scatter(SF['X'], SF['Y'], c=['blue'])
plt.scatter(SF['X'], SF['Y'], c=SF['PdDistrictCode'])
plt.show()
```



## Paso 2: Agregar los paquetes de mapas para mejorar el gráfico

En el paso 1, creó un gráfico simple que muestra dónde ocurrieron los incidentes de delitos en el condado de SF. Este gráfico es útil, pero folium ofrece funciones adicionales que permiten superponer este gráfico en un mapa de OpenStreet.

a) Folium requiere que se especifique el color del marcador mediante un valor hexadecimal. Por este motivo, utilizamos el paquete colores y seleccionamos los colores necesarios.

```
In [27]: # Code cell 24
from matplotlib import colors
districts = np.unique(SF['PdDistrict'])
print(list(colors.cnames.values())[0:len(districts)])

['#F0F8FF', '#FAEBD7', '#00FFFF', '#7FFFD4', '#F0FFFF', '#F5F5DC', '#FFE4C4', '#000000', '#FFEB3D', '#0000FF']
```

b) Cree un diccionario de colores para cada distrito del departamento policial.

```
In [28]: # Code cell 25
color_dict = dict(zip(districts, list(colors.cnames.values())[0:-1:len(districts)]))
```

```
color_dict
```

```
Out[28]: {'BAYVIEW': '#F0F8FF',
          'CENTRAL': '#8A2BE2',
          'INGLESIDE': '#00FFFF',
          'MISSION': '#FF8C00',
          'NORTHERN': '#FF1493',
          'PARK': '#F8F8FF',
          'RICHMOND': '#4B0082',
          'SOUTHERN': '#FAFAD2',
          'TARAVAL': '#B0C4DE',
          'TENDERLOIN': '#9370DB'}
```

c) Cree el mapa con las coordenadas centrales de los datos de SF para centrar el mapa (utilice mean). Para reducir el tiempo de cómputo, se utiliza plotEvery para limitar la cantidad de datos graficados. Establezca este valor a 1 para graficar todas las filas (puede llevar mucho tiempo visualizar el mapa).

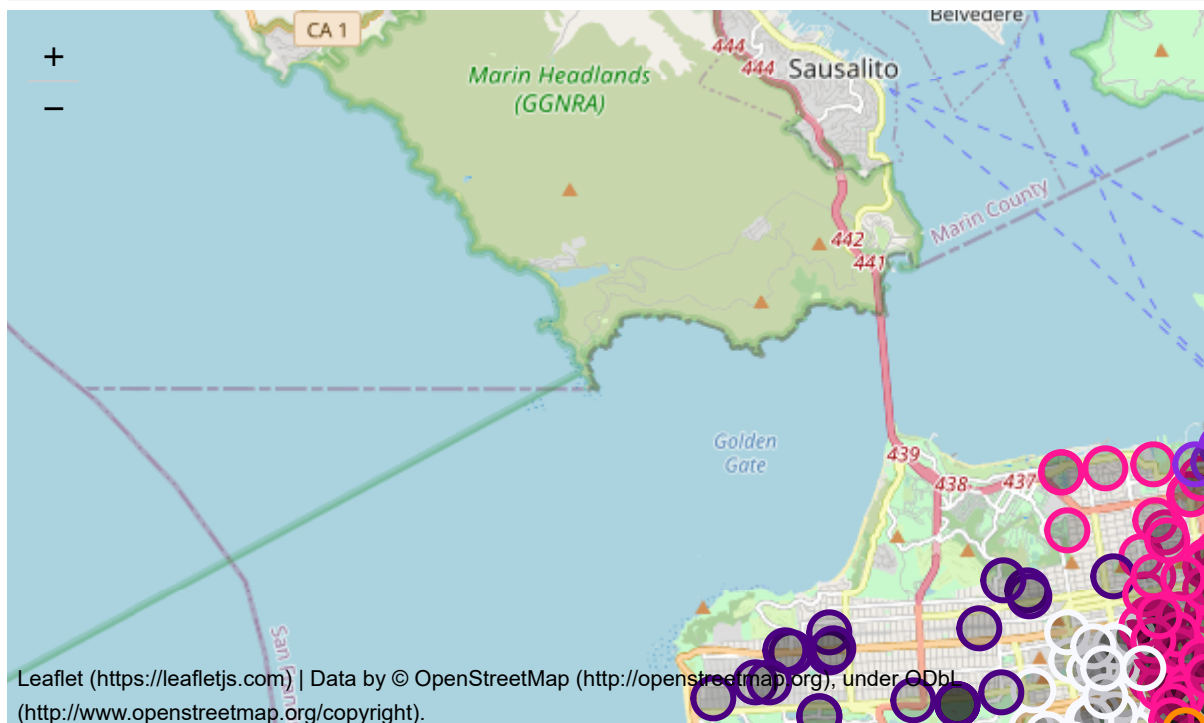
```
In [29]: # Code cell 26
#Create map
map_osm = folium.Map(location=[SF['Y'].mean(), SF['X'].mean()], zoom_start = 12)
plotEvery = 50
obs = list(zip( SF['Y'], SF['X'], SF['PdDistrict']))

for el in obs[0:-1:plotEvery]:

    folium.CircleMarker(el[0:2], color=color_dict[el[2]], fill_color=el[2],radius=10).
```

```
In [30]: # Code cell 27
map_osm
```

```
Out[30]:
```



## Conclusiones

Para primera práctica aprendí la importancia de lo que es aprender a utilizar los datos mediante la estadística, para así poder analizar y poder hacer predicciones o calcular estimaciones de lo que necesitamos y como con tener simples tablas y graficas se pueden llegar a estos tipos de cálculos y analizarlos.

En la segunda práctica aprendí a como aprender a presentar los datos para un mejor análisis, por ejemplo, para poder calcular en que mes se integró una persona o trabajador que el tipo de dato sea de tipo fecha para un mejor manejo de las fórmulas, que en nuestro caso se utilizó Excel. también se presentarán datos que no estén en un formato bien hecho o hayan sido modificados y tenemos que aprender a saber que datos usar y como darles un mejor formato.

En la práctica 3, aprendí sobre las librerías que cuenta python para manejar las fechas y el tiempo, también para importar archivos en formato CSV, y también a generar funciones o subprocessos. también con los datos que utilizamos aprendimos a como poner nuestras propias columnas y ver cuáles son las que necesitamos para así quitar las que no son necesarias y poder mostrar los datos y los usuarios lo puedan entender de manera más fácil.

En la práctica 4, se enfocó más en el uso de una base de datos usando SQLite, que contiene los datos de una agenda sencilla pero que también en python podemos hacer las consultas que se requieran ayudándonos a tener una mejor automatización o manejo de las consultas y sea más fácil consultar.

En la práctica 5, utilizamos una base de datos en que contiene la información del contador de internet de SQL, utilizando esa información hicimos consultas y utilizamos una librería que es matplotlib que sirve para generar gráficos y nos permite personalizarlos y mostrar datos como nosotros necesitamos.

En esta última práctica utilizamos un archivo csv que contiene la información de los crímenes en san francisco, es una práctica que yo considero más realista y con datos que se nos pueden llegar ocurrir en al trabajo, entonces saber analizarlos utilizando las librerías de python como el matplotlib, el pandas que ayuda a que la estructura de los datos sea más fácil, también otra librería es el folium que sirve para crear mapas interactivos, que la utilizamos para mostrar en que distritos de San Francisco, mediante colores, se presentaron diferentes tipos de crímenes en el mes de Agosto.

In [ ]: