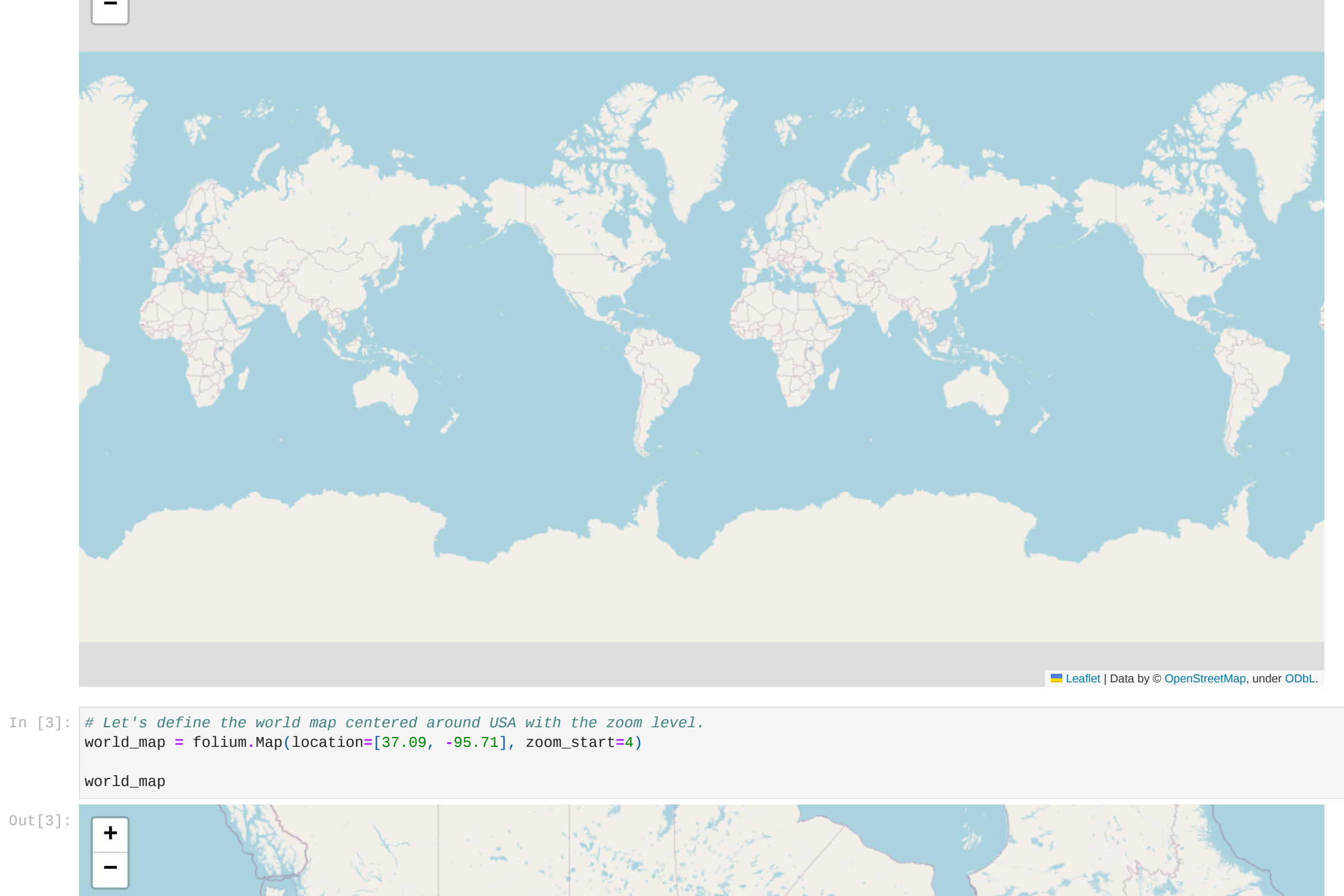


```
In [1]: import pandas as pd
import numpy as np
import folium
```

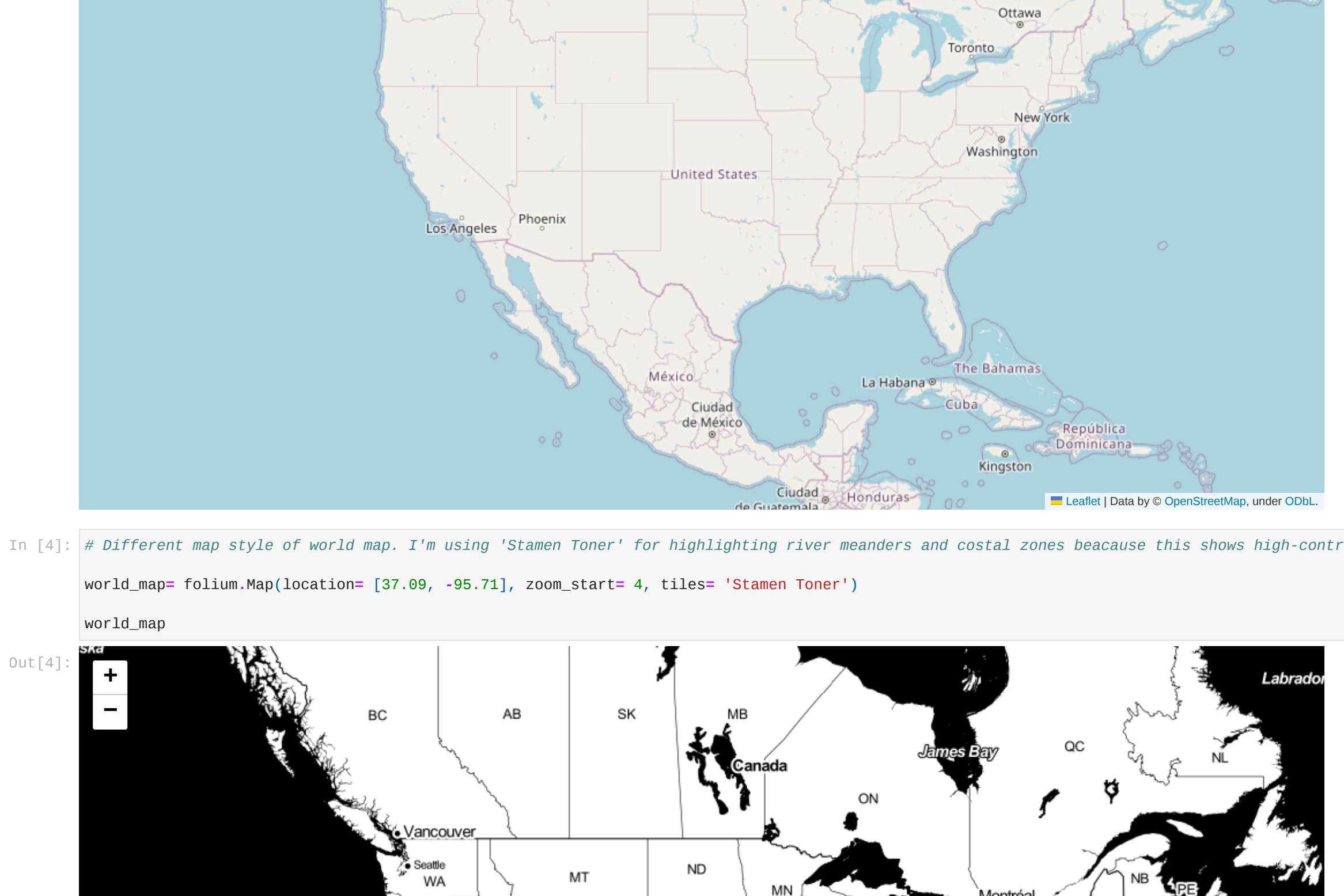
```
In [2]: # Firstly, let's simply create world map
world_map = folium.Map()
```

world_map



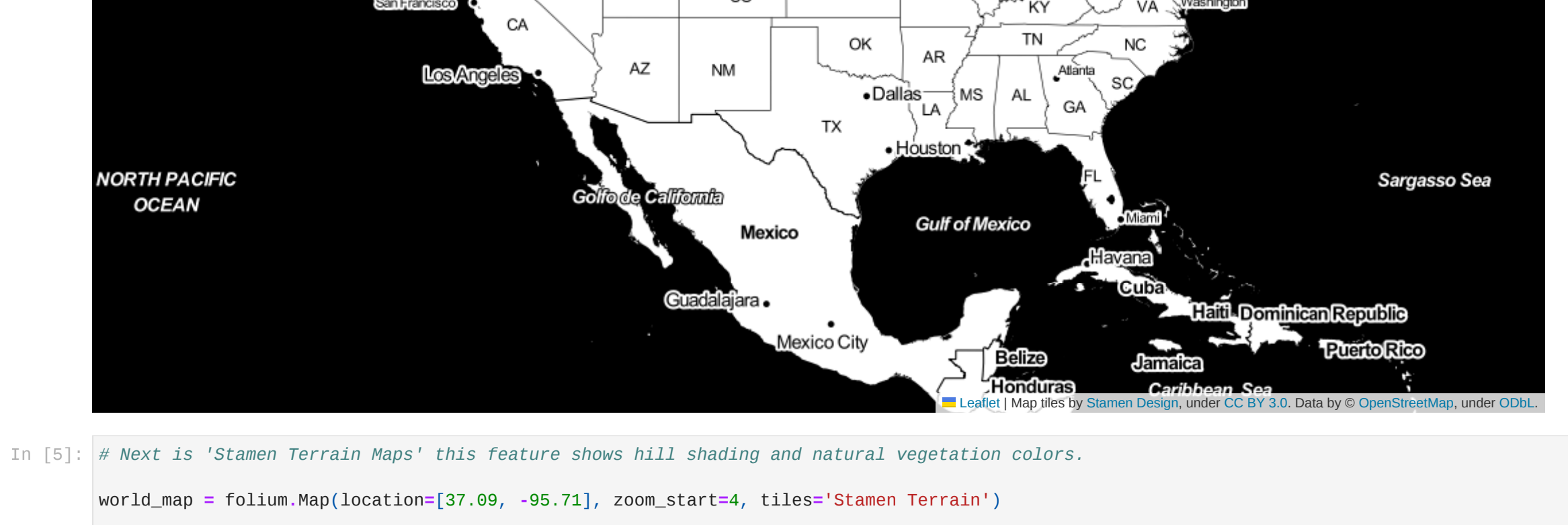
```
In [3]: # Let's define the world map centered around USA with the zoom level.
world_map = folium.Map(location=[37.09, -95.71], zoom_start=4)
```

world_map



```
In [4]: # Different map style of world map. I'm using 'Stamen Toner' for highlighting river meanders and costal zones because this shows high-contrast.
```

world_map



```
In [5]: # Next is 'Stamen Terrain Maps' this feature shows hill shading and natural vegetation colors.
```

world_map = folium.Map(location=[37.09, -95.71], zoom_start=4, tiles='Stamen Terrain')

display map

world_map



Maps with Markers

```
In [6]: df_incidents = pd.read_csv('Police_Department_Incident_Reports__Historical_2003_to_May_2018.csv')
df_incidents.head()
```

Out[6]:

	PdId	IncidentNum	Incident Code	Category	Descript	DayOfWeek	Date	Time	PdDistrict	Resolution	...	Fix It Zones as of 2017 11-06 2 2	DELETE - HSOC Zones 2 2	Fix It Zones as of 2018 02-07 2 2	CBD, BID and CBD Boundaries as of 2017 2 2
0	4133422003074	41334220	3074	ROBBERY	ROBBERY, BODILY FORCE	Monday	11/22/2004	17:50	INGLESIDE	NONE	...	NaN	NaN	NaN	NaN
1	5118535807021	51185358	7021	VEHICLE THEFT	STOLEN AUTOMOBILE	Tuesday	10/18/2005	20:00	PARK	NONE	...	NaN	NaN	NaN	NaN
2	4018830907021	40188309	7021	VEHICLE THEFT	STOLEN AUTOMOBILE	Sunday	02/15/2004	02:00	SOUTHERN	NONE	...	NaN	NaN	NaN	NaN
3	11014543126030	110145431	26030	ARSON	ARSON	Friday	02/18/2011	05:27	INGLESIDE	NONE	...	NaN	NaN	NaN	NaN
4	10108108004134	101081080	4134	ASSAULT	BATTERY	Sunday	11/21/2010	17:00	SOUTHERN	NONE	...	NaN	NaN	NaN	NaN

5 rows × 35 columns

```
In [7]: df_incidents.shape
```

Out[7]: (2129525, 35)

```
In [8]: # limit = 100
df_incidents = df_incidents.iloc[0:limit, :]
```

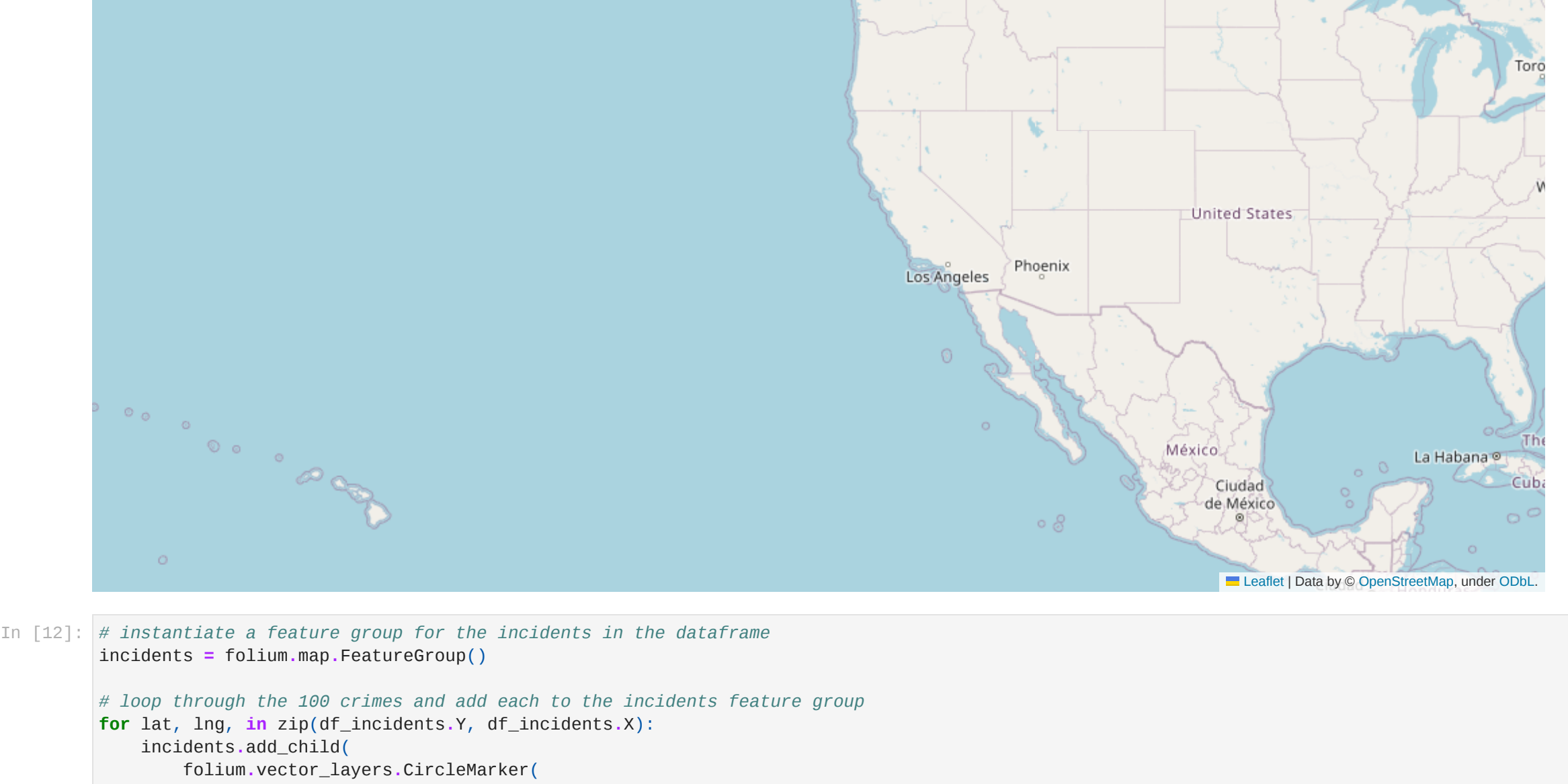
```
In [9]: df_incidents.shape
```

Out[9]: (100, 35)

```
In [10]: # latitude= 37.77
longitude = -122.42
```

```
In [11]: san_map = folium.Map(location=[latitude, longitude], zoom_start=4)
```

san_map



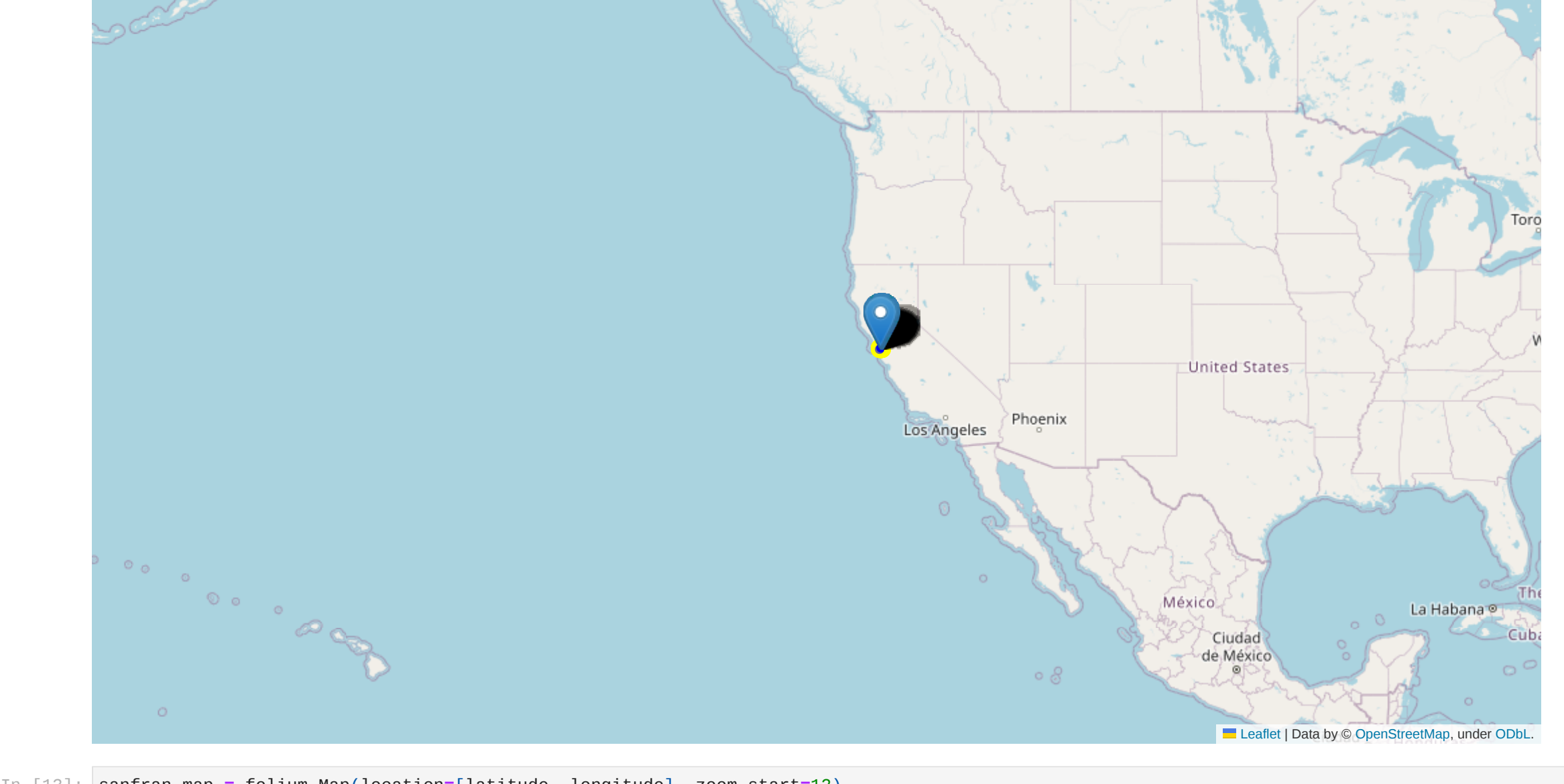
```
In [12]: # instantiate a feature group for the incidents in the dataframe
incidents = folium.map.FeatureGroup()

# loop through the 100 crimes and add each to the incidents feature group
for lat, lng, in zip(df_incidents.Y, df_incidents.X):
    incidents.add_child(
        folium.vector_layers.CircleMarker(
            [lat, lng],
            radius=5, # define how big you want the circle markers to be
            color='yellow',
            fill=True,
            fill_color='blue',
            fill_opacity=0.6
        )
    )

# add pop-up text to each marker on the map
latitudes = list(df_incidents.Y)
longitudes = list(df_incidents.X)
labels = list(df_incidents.Category)

for lat, lng, label in zip(latitudes, longitudes, labels):
    folium.Marker([lat, lng], popup=label).add_to(san_map)

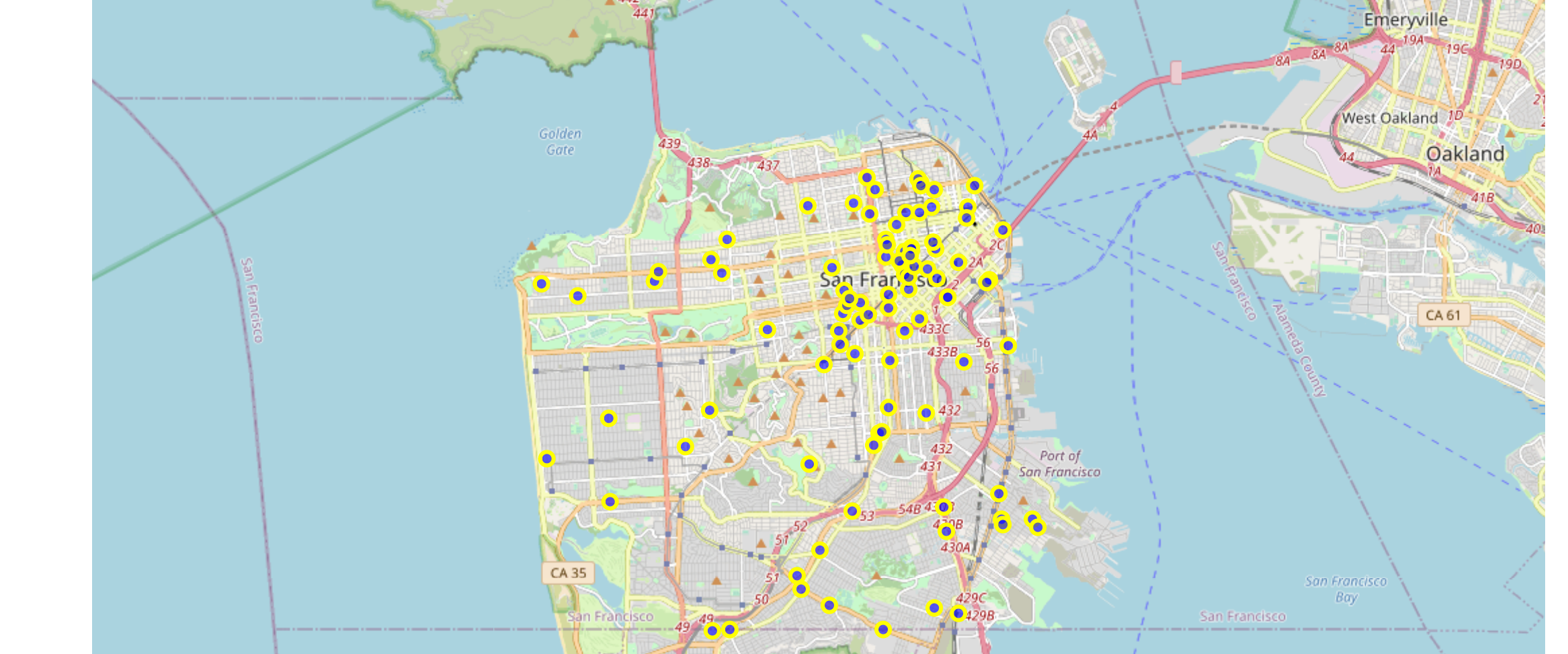
# add incidents to map
san_map.add_child(incidents)
```



```
In [13]: sanfran_map = folium.Map(location=[latitude, longitude], zoom_start=12)
```

```
# loop through the 100 crimes and add each to the map
for lat, lng, label in zip(df_incidents.Y, df_incidents.X, df_incidents.Category):
    folium.vector_layers.CircleMarker(
        [lat, lng],
        radius=5,
        color='yellow',
        fill=True,
        popup=label,
        fill_color='blue',
        fill_opacity=0.6
    ).add_to(sanfran_map)

# show map
sanfran_map
```



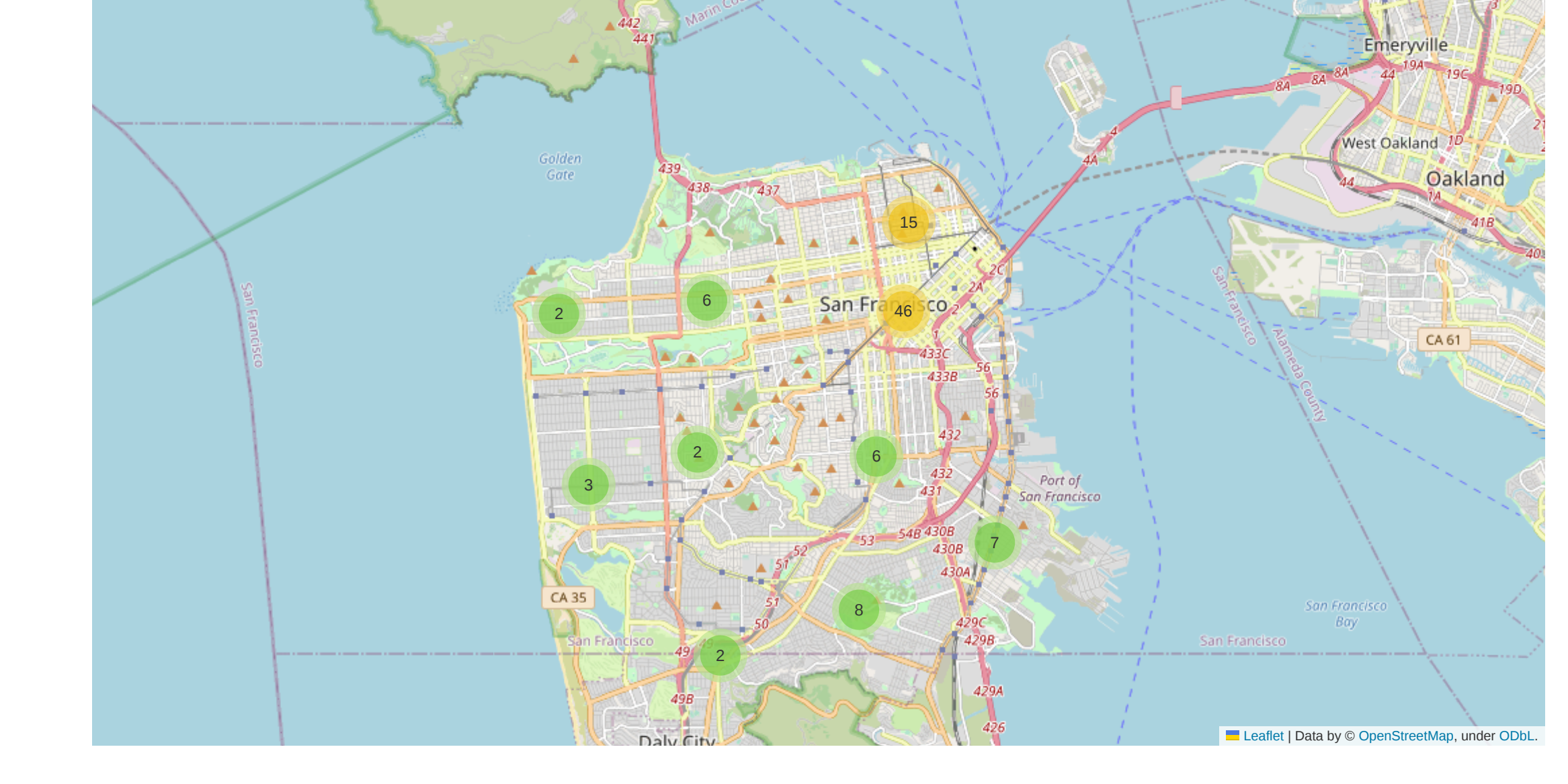
```
In [14]: from folium import plugins
```

```
# let's start again with a clean copy of the map of San Francisco
sanfran_map = folium.Map(location=[latitude, longitude], zoom_start=12)
```

```
incidents = plugins.MarkerCluster().add_to(sanfran_map)
```

```
for lat, lng, label, in zip(df_incidents.Y, df_incidents.X, df_incidents.Category):
    folium.Marker(
        location=[lat, lng],
        icon=None,
        popup=label,
    ).add_to(incidents)
```

```
sanfran_map
```



Choropleth Map Style

```
In [15]: df_canada = pd.read_csv('Dataset of canadian immigrants.csv')
```

```
df_canada.head()
```

Out[15]:

	Country	Continent	Region	DevName	1980	1981	1982	1983	1984	1985	...	2005	2006	2007	2008	2009	2010	2011	2012	2013	Total
0	Alghanistan	Asia	Southern Asia	Developing regions	16	39	39	47	71	340	...	3436	3009	2652	2111	1746	1758	2203	2635	2004	58639
1	Albania	Europe	Southern Europe	Developed regions	1	0	0	0	0	0	...	1223	856	702	560	716	561	539	620	603	15099
2	Algeria	Africa	Northern Africa	Developing regions	80	67	71	69	63	44	...	3626	4807	3623	4005	5393	4752	4325	3774	4331	69439
3	American Samoa	Oceania	Polynesia	Developing regions	0	1	0	0	0	0	...	0	1	0	0	0	0	0	0	0	6
4	Andorra	Europe	Southern Europe	Developed regions	0	0	0	0	0	0	...	0	1	1	0	0	0	0	0	1	15

5 rows × 39 columns

```
In [16]: world_geo='https://raw.githubusercontent.com/python-visualization/folium/main/examples/data/world-countries.json'
```

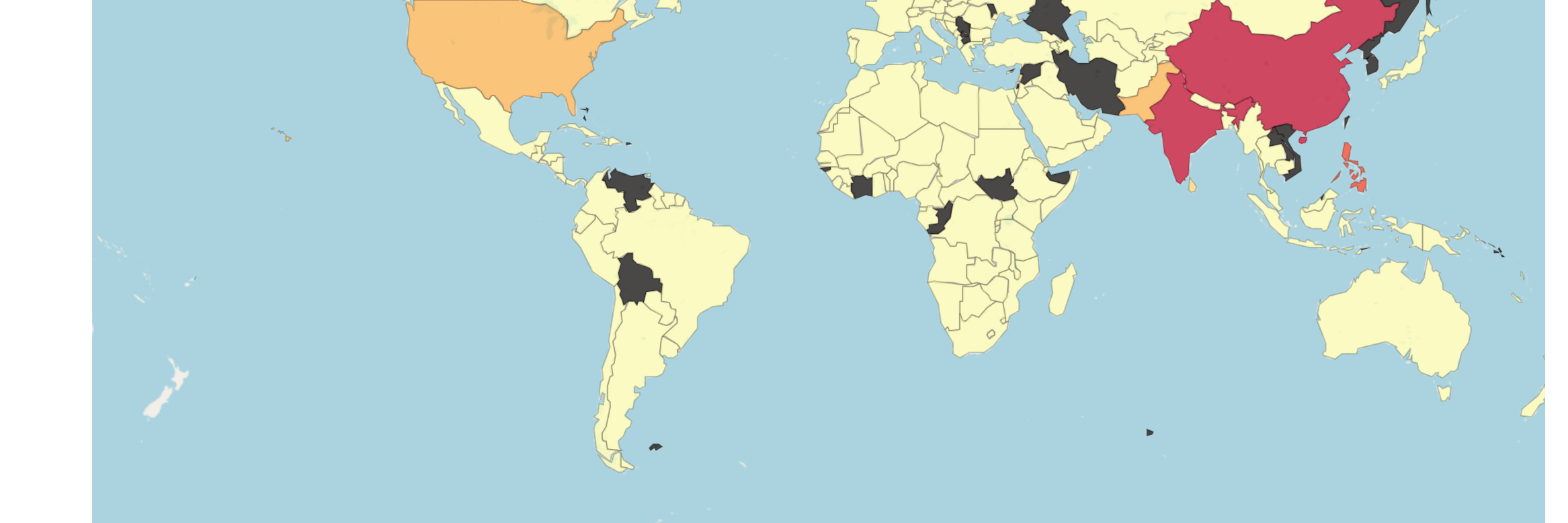
```
world_map = folium.Map(location=[0, 0], zoom_start=2)
```

```
In [17]: world_map.choropleth(
    geo_data=world_geo,
    data=df_canada,
    columns=['Country', 'Total'],
    key_on='feature.properties.name',
    fill_color='yldrd',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Immigration to Canada'
)

# display map
world_map
```

C:\Users\prabh\AppData\Local\Programs\Python\Python310\lib\site-packages\folium\Folium.py:465: FutureWarning: The choropleth method has been deprecated. Instead use the new Choropleth class, which has the same arguments. See the example notebook 'GeoJSON_and_choropleth' for how to do this.

warnings.warn(



```
In [ ]:
```