

## ▼ Install folium

```
!pip install folium
```

```
Requirement already satisfied: folium in /usr/local/lib/python3.7/dist-packages (0.8.3)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from folium)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.7/dist-packages (from folium)
Requirement already satisfied: branca>=0.3.0 in /usr/local/lib/python3.7/dist-packages (from folium)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from folium)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from folium)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/dist-packages (from folium)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from folium)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from folium)
Requirement already satisfied: urllib3!=1.25.0,!>=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from folium)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from folium)
```



## ▼ Import the Libraries

```
import folium
import pandas as pd
import numpy as np

import json
import requests

#check the folium version
print(folium.__version__)
```

0.8.3

## ▼ Simple Folium Map

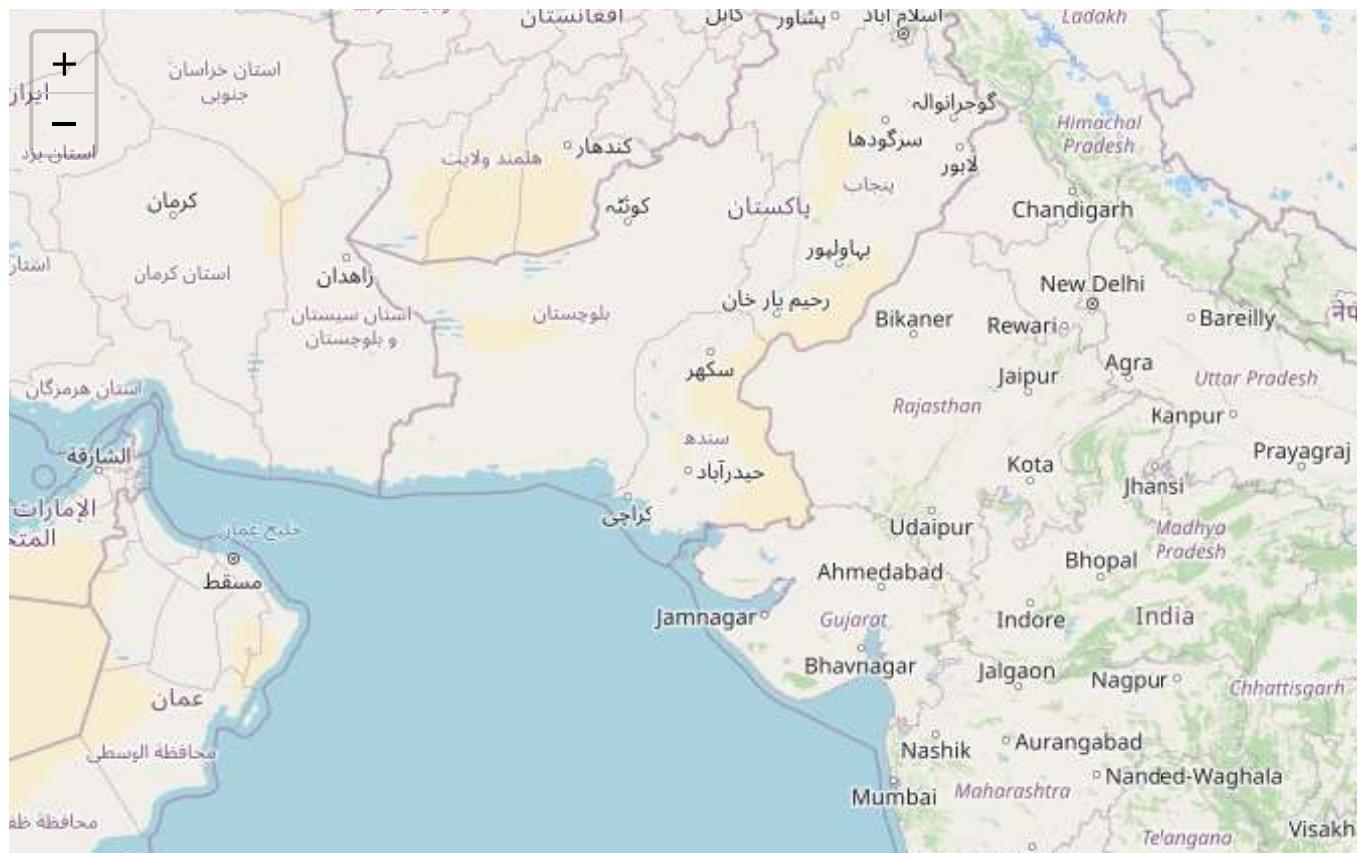
```
world_map = folium.Map()
world_map
```

Make this Notebook Trusted to load map: File -> Trust Notebook



## ▼ Folium Map of India

```
world_map_2 = folium.Map(location=[20.5937,78.9629], zoom_start= 5,tiles='OpenStreetMap')  
world_map_2
```



## ▼ Folium Map with higher zoom level



```
world_map_3 = folium.Map(location=[20.5937,78.9629], zoom_start=10, tiles='OpenStreetMap')
world_map_3
```



```
#Folium map with Stamen Terrain Tile
```

```
world_map_4 = folium.Map(location=[20.5937,78.9629], zoom_start=6, tiles = 'StamenToner')  
world_map_4
```



```
# folium map with Stamen Terrain tile type
world_map_5 = folium.Map(location=[20.5937,78.9629], zoom_start=5, tiles='Stamen Terrain')
world_map_5
```



## ▼ Folium Map for Crime Data

```
# upload the crime dataset
crime_data = pd.read_csv('/content/Police_Department_Incidents_-_Previous_Year__2016_.csv')
crime_data.shape

(150500, 13)
```

```
#print top 5 records
crime_data.head()
```

	IncidntNum	Category	Descript	DayOfWeek	Date	Time	PdDistrict	Resc
0	120058272	WEAPON LAWS	POSS OF PROHIBITED WEAPON	Friday	01/29/2016 12:00:00 AM	11:00	SOUTHERN	A B
1	120058272	WEAPON LAWS	FIREARM, LOADED, IN VEHICLE, POSSESSION OR USE	Friday	01/29/2016 12:00:00 AM	11:00	SOUTHERN	A B
2	141059263	WARRANTS	WARRANT ARREST	Monday	04/25/2016 12:00:00 AM	14:59	BAYVIEW	A B
3	160013662	NON-CRIMINAL	LOST PROPERTY	Tuesday	01/05/2016 12:00:00 AM	23:50	TENDERLOIN	
4	160002740	NON-CRIMINAL	LOST PROPERTY	Friday	01/01/2016 12:00:00 AM	00:30	MISSION	



```
# filtering the top 1000 rows for faster processing
```

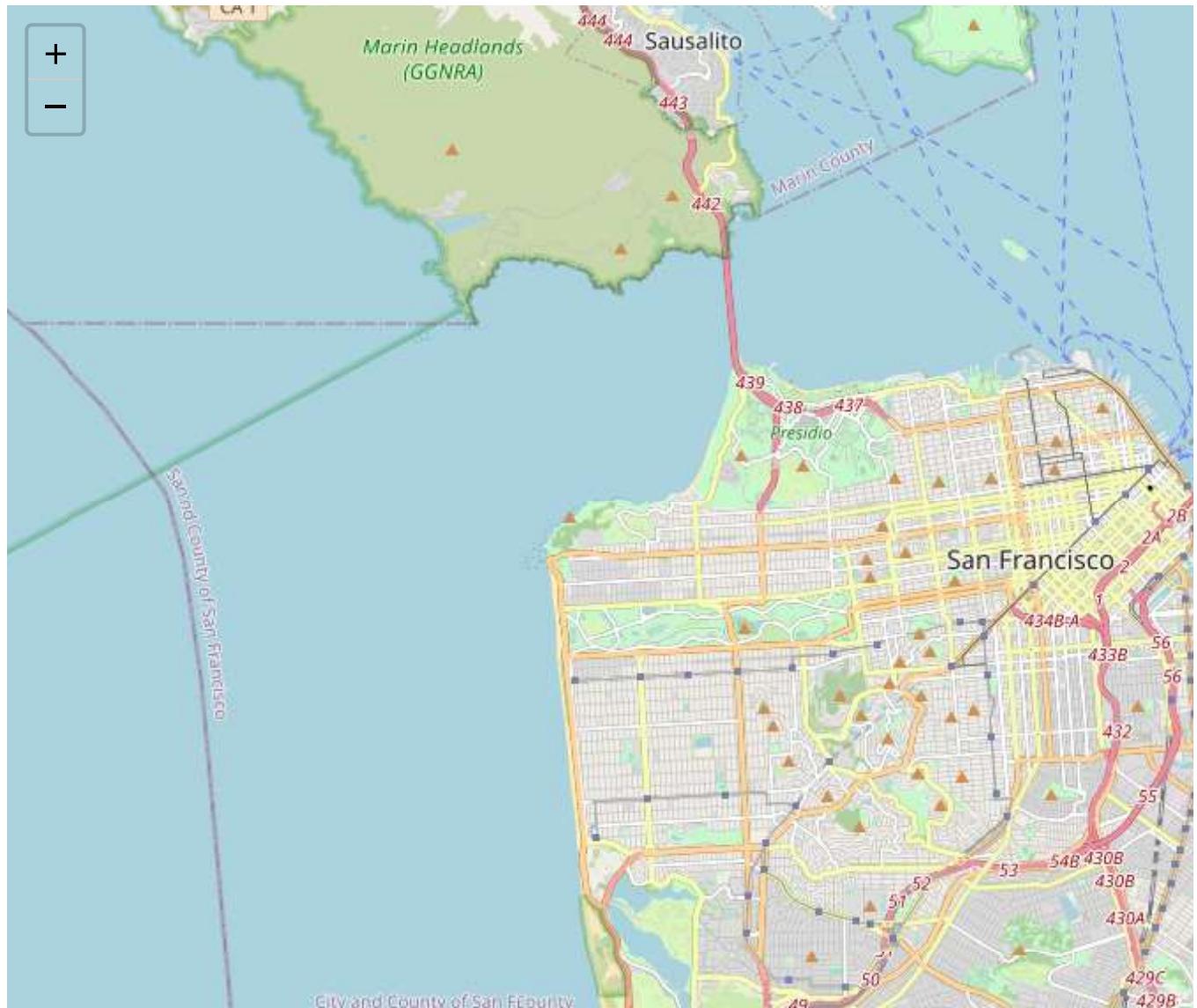
```
crime_data_sub = crime_data.iloc[0:1000,:]
crime_data_sub.shape
```

```
(1000, 13)
```

```
# Co-ordinates for San Francisco
latitude = 37.77
longitude = -122.42
```

```
# display map of San Francisco
```

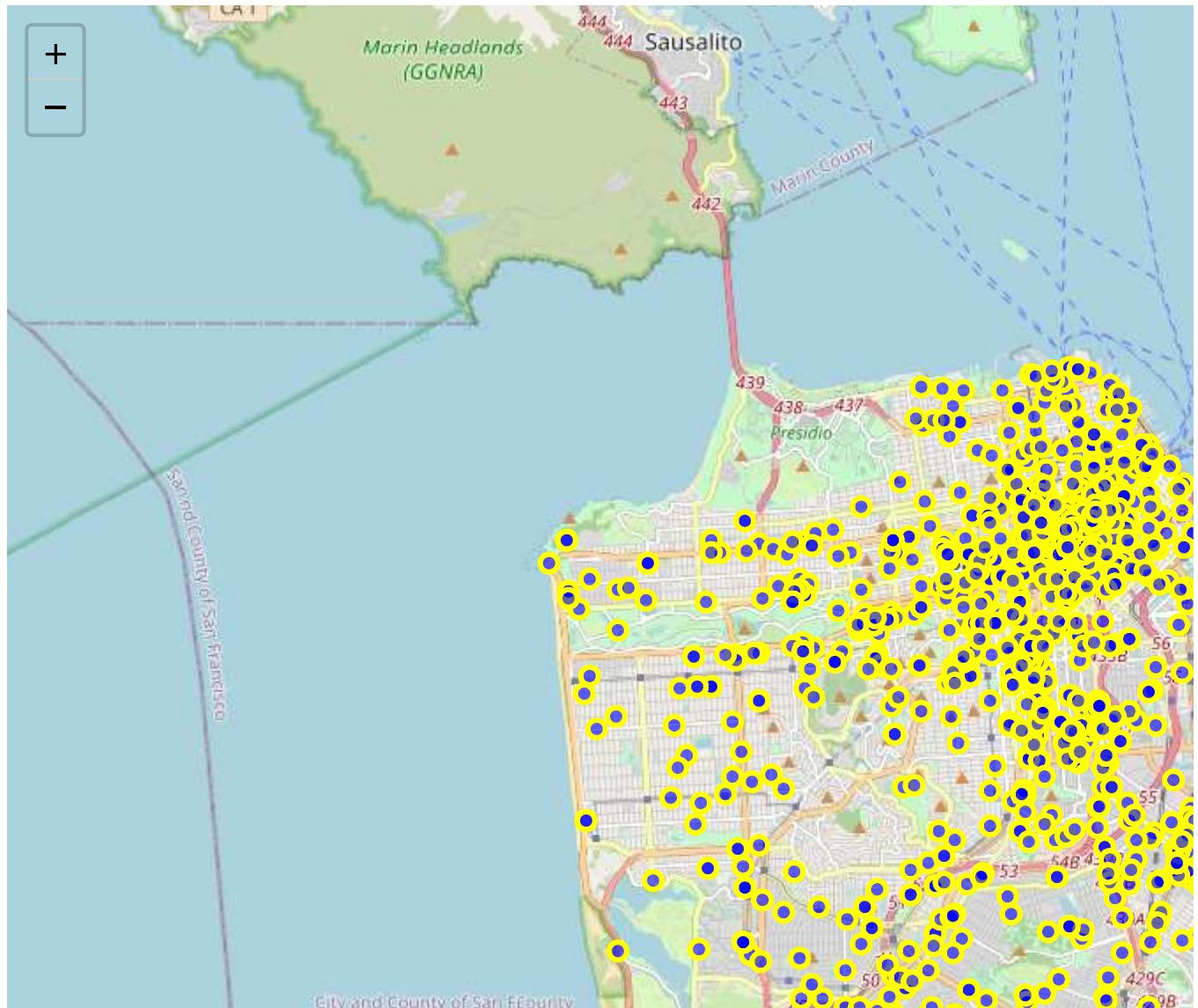
```
sf_map = folium.Map(location = [latitude, longitude], zoom_start = 12)
sf_map
```



```
#display the map with Markers
incidents = folium.map.FeatureGroup()

for lat, lng, in zip(crime_data_sub.Y, crime_data_sub.X):
    incidents.add_child(
        folium.CircleMarker(
            [lat, lng],
            radius=5,
            color='yellow',
```

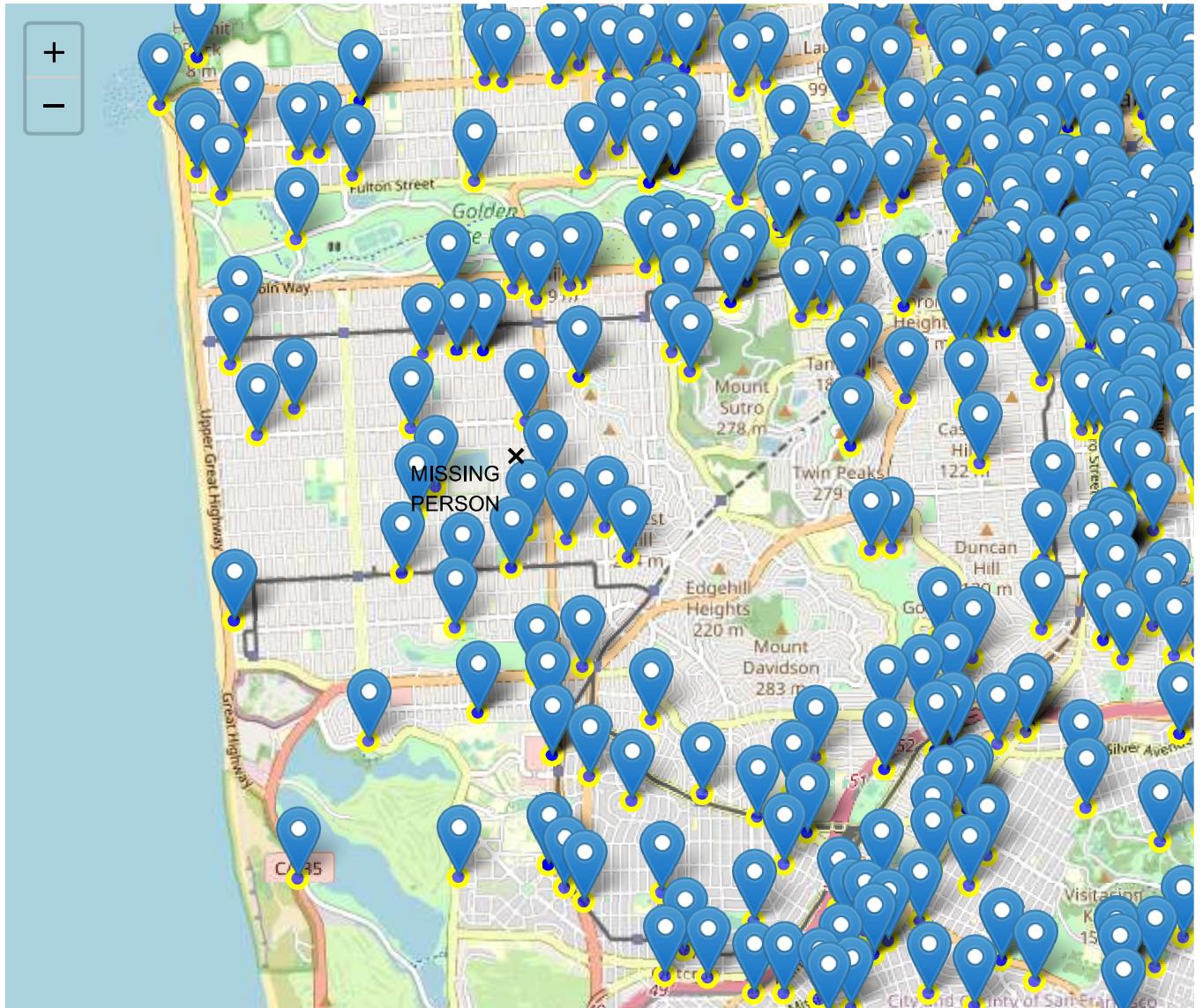
```
        fill=True,  
        fill_color='blue',  
        fill_opacity=0.6  
)  
)  
  
sf_map.add_child(incidents)
```



```
# add pop up to the locations  
latitudes = list(crime_data_sub.Y)  
longitudes = list(crime_data_sub.X)  
labels = list(crime_data_sub['Category'])
```

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

sf_map.add_child(incidents)
```



## ▼ Make Folium Map with Markers

```
#import the plugin
from folium import plugins

# folium map of crime data with the Markers
# taking top 5k records. full data cannot be processed due to hardware limitations
https://colab.research.google.com/drive/1LD3hfmwXUPqi-e5R_sWXRJCwWIXvCis6#scrollTo=dR0W8MWjSkyU&printMode=true
```

```
sf_map_2 = folium.Map(location=[latitude, longitude], zoom_start=12)

incidents_2 = plugins.MarkerCluster().add_to(sf_map_2)

for lat,lng,label in zip(crime_data.iloc[:5000,:].Y, crime_data.iloc[:5000,:].X , crime_data.

    folium.Marker(
        location=[lat,lng],
        icon=None,
        popup = label
    ).add_to(incidents_2)

sf_map_2

# we can click on the circles to zoom and futher get breakdown
```



## ▼ Chloropeth Map

```
#uploading the Unemployment Data for US
us_unemployment_data = pd.read_csv('/content/US_Unemployment_Oct2012.csv')
us_unemployment_data.shape
```

```
(50, 2)
```

```
us_unemployment_data.head()
```

	State	Unemployment	⊕
0	AL	7.1	
1	AK	6.8	
2	AZ	8.1	
3	AR	7.2	
4	CA	10.1	

```
us_latitude = 48
us_longitude=-102
```

```
#drawing the choropeth map
```

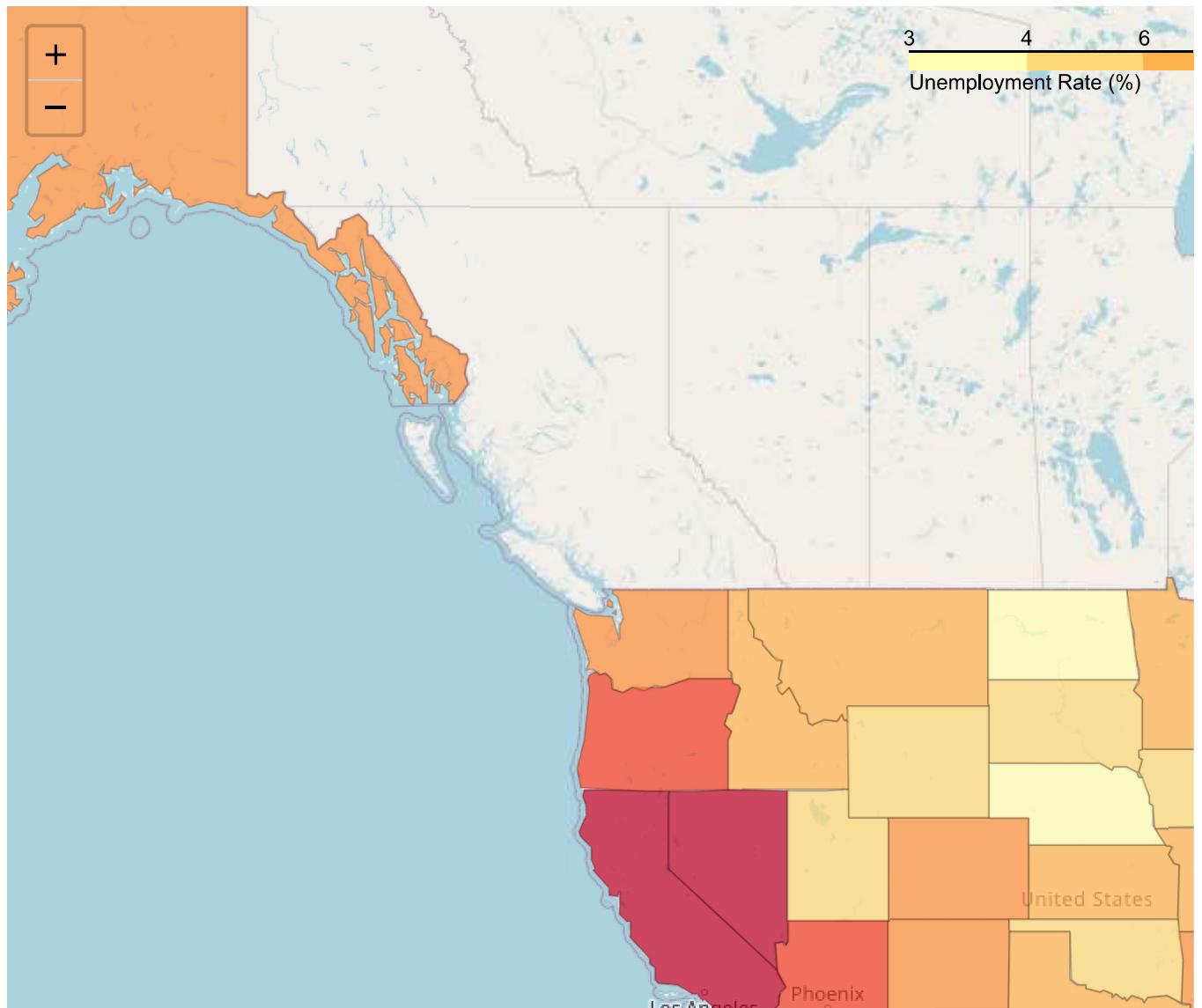
```
state_geo = f"/content/us-states.json"
```

```
map = folium.Map(location=[us_latitude, us_longitude], zoom_start=4)
```

```
folium.Choropleth(
    geo_data=state_geo,
    name="choropleth",
    data=us_unemployment_data,
    columns=["State", "Unemployment"],
    key_on="feature.id",
    fill_color="YlOrRd",
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name="Unemployment Rate (%)",
).add_to(map)
```

```
folium.LayerControl().add_to(map)
```

```
map
```



#### ▼ Chloropeth Map of India Accidents Data

```
#india Accidents data  
india_accidents_data = pd.read_csv('/content/india_accidents_data.csv')
```

```
india_accidents_data.shape
```

```
(37, 67)
```

```
india_accidents_data.head()
```

S. No.	State/ UT	Total Acc. - 2014	Total Acc. - 2016	Fine - Total Acc. - 2014	Fine - Persons Killed - 2014	Fine - Persons Injured - 2014	Mist/fog - Total Acc. - 2014	Mist/fog - Persons Killed - 2014	Mist/fog - Person Injure - 201
0	1 Andhra Pradesh	24440	24846	14591.0	4586	17065.0	724.0	219	925.
1	2 Arunachal Pradesh	205	249	71.0	30	110.0	14.0	10	26.
2	3 Assam	7144	7399	3575.0	1318	3216.0	494.0	150	368.
3	4 Bihar	9556	7640	2343.0	1218	1626.0	1713.0	881	1081.
4	5 Chhattisgarh	13821	13448	5000.0	1354	4584.0	382.0	149	376.



```
india_accidents_df = india_accidents_data[['State/ UT', 'Total Acc. - 2014', 'Total Acc. - 2016']]
```

```
india_accidents_df.shape
```

```
(37, 3)
```

```
india_accidents_df.head()
```

	State/ UT	Total Acc. - 2014	Total Acc. - 2016	edit
0	Andhra Pradesh	24440	24846	
1	Arunachal Pradesh	205	249	
2	Assam	7144	7399	
3	Bihar	9556	7640	
4	Chhattisgarh	13821	13448	

```
india_latitude = 20.5937
india_longitude = 78.9629
```

```
state_geo = f"/content/Indian_States.json"
```

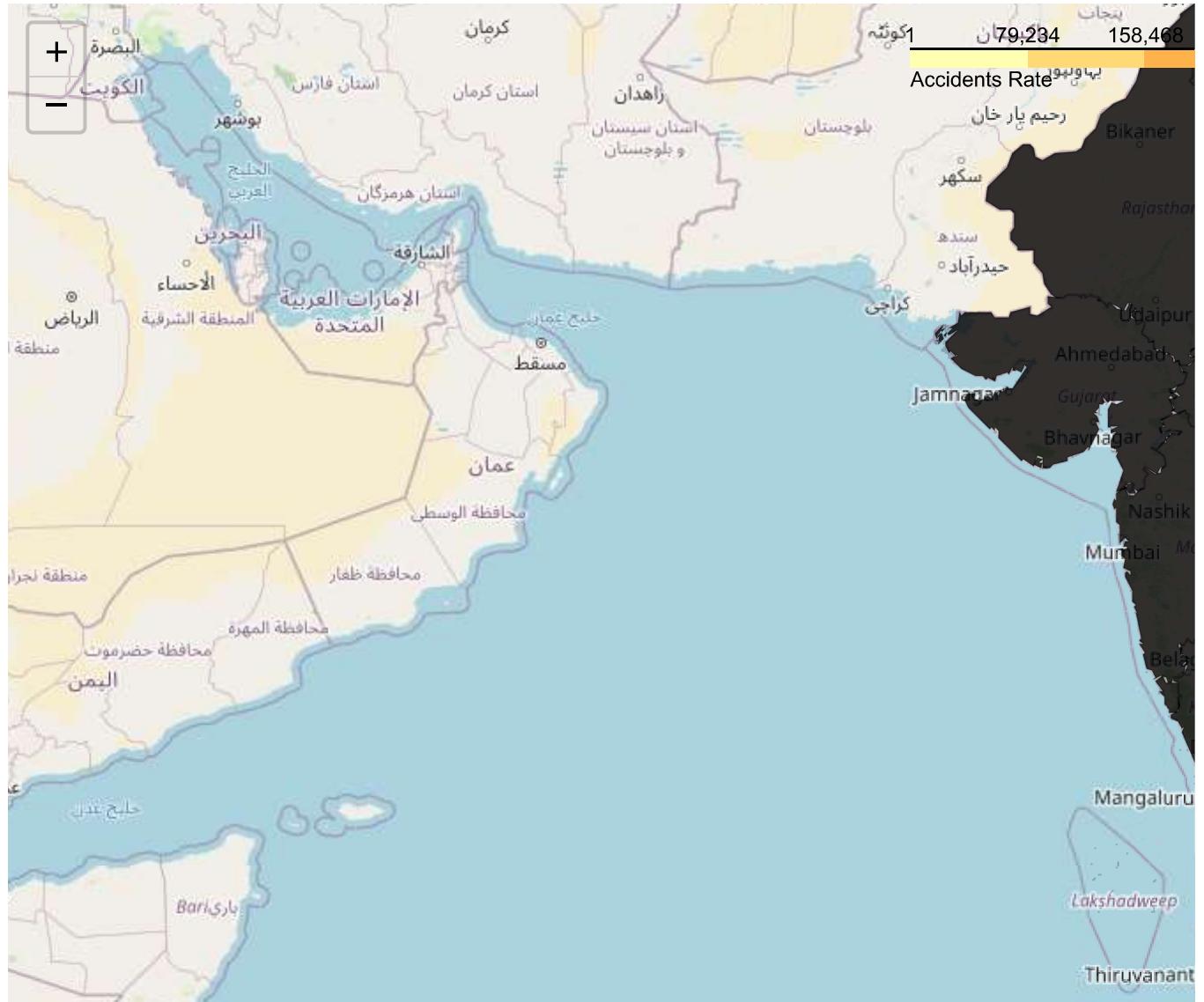
```
map = folium.Map(location=[india_latitude, india_longitude], zoom_start=4)
```

```
folium.Choropleth(
    geo_data=state_geo,
```

```
name="choropleth",
data=india_accidents_df,
columns=["State/ UT", "Total Acc. - 2016"],
key_on="feature.properties.name",
fill_color="YlOrRd",
fill_opacity=0.8,
line_opacity=0.2,
legend_name="Accidents Rate",
).add_to(map)

folium.LayerControl().add_to(map)
```

map



## ▼ Topo Json Visualization

```
antarctic_ice_shelf_topo = f"https://raw.githubusercontent.com/python-visualization/folium/master/examples/data/antarctic_ice_shelf_topo.json"
antarctic_ice_edge = f"https://raw.githubusercontent.com/python-visualization/folium/master/examples/data/antarctic_ice_shelf.json"

antartica_map = folium.Map(
    location=[-59.1759, -11.6016],
    tiles="cartodbpositron",
    zoom_start=2,
)
folium.GeoJson(antarctic_ice_edge, name="geojson").add_to(antartica_map)

folium.TopoJson(
    json.loads(requests.get(antarctic_ice_shelf_topo).text),
    # json.loads(topo_json),
    "objects.antarctic_ice_shelf",
    name="topojson",
).add_to(antartica_map)

folium.LayerControl().add_to(antartica_map)

display(antartica_map)
```

Make this Notebook Trusted to load map: File -> Trust Notebook



---

✓ 10s completed at 4:49 PM

● ✕