In [1]:

```
import plotly.express as px
import pandas as pd
import json
```

In [2]:

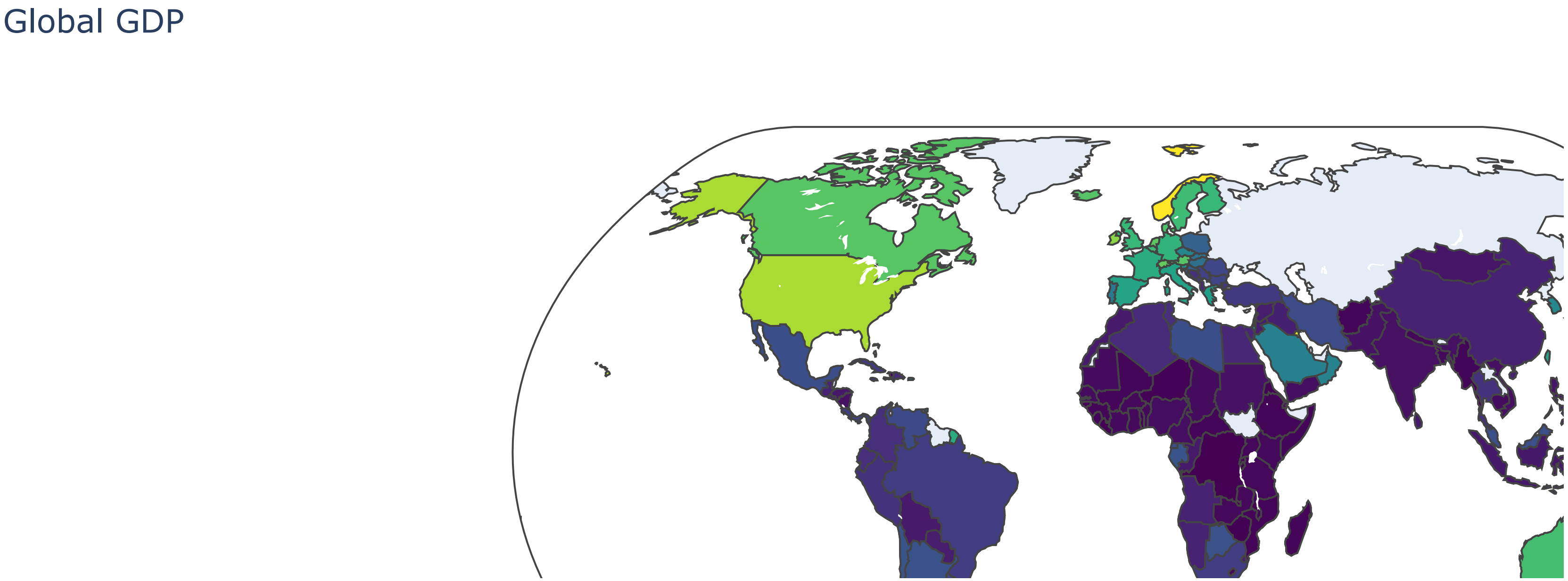
```
df = px.data.gapminder()
df.head()
```

Out[2]:

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
0	Afghanistan	Asia	1952	28.801	8425333	779.445314	AFG	4
1	Afghanistan	Asia	1957	30.332	9240934	820.853030	AFG	4
2	Afghanistan	Asia	1962	31.997	10267083	853.100710	AFG	4
3	Afghanistan	Asia	1967	34.020	11537966	836.197138	AFG	4
4	Afghanistan	Asia	1972	36.088	13079460	739.981106	AFG	4

In [3]:

```
df = px.data.gapminder().query("year == 2007")

fig = px.choropleth(df, locations = 'iso_alpha',
                    color = 'gdpPercap',
                    hover_name = 'country',
                    color_continuous_scale = 'viridis',
                    projection = 'natural earth',
                    title = 'Global GDP')
fig.show()
```



In [4]:

```
import pandas as pd
import plotly.express as px
import json
from urllib.request import urlopen

# Load the GeoJSON file containing US county boundaries
# The 'feature.id' in this GeoJSON is the FIPS code for each county
with urlopen('https://raw.githubusercontent.com/plotly/datasets/master/geojson-counties-fips.json') as response:
    counties = json.load(response)

# Load the data (e.g., unemployment rate by FIPS code)
# The 'fips' column is the key linking the data to the geography
df = pd.read_csv(
    "https://raw.githubusercontent.com/plotly/datasets/master/fips-unemp-16.csv",
    dtype={"fips": str}
)
```

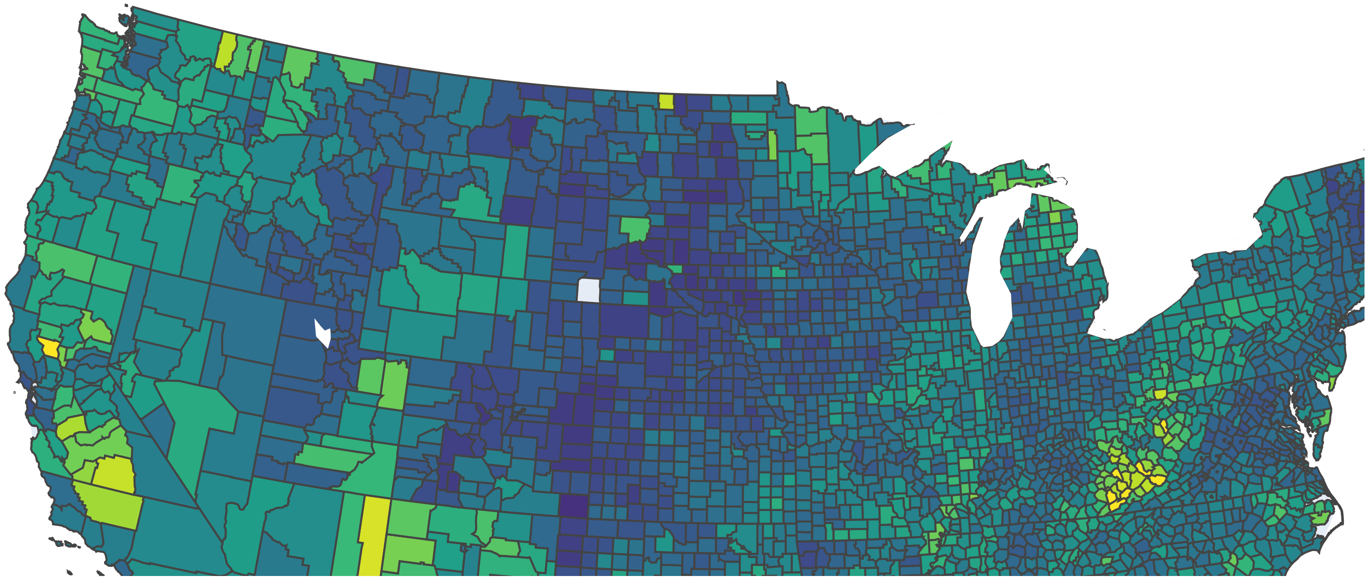
In [5]:

```
# Create the choropleth map
fig = px.choropleth(
    df,                                 # Your DataFrame
    geojson=counties,                   # The GeoJSON data for boundaries
    locations='fips',                   # Column in df that matches GeoJSON IDs
    color='unemp',                      # Column in df to determine the color/shading
    color_continuous_scale="Viridis",   # Color scale to use
    range_color=(0, 12),                # Set the color range (min and max values)
    scope="usa",                        # Restrict the map view to the USA
    labels={'unemp':'Unemployment Rate (%)'}, # Custom label for the colorbar
    title='US County Unemployment Rate (2016)' # Map Title
)
```

In [6]:

```
# Customize the layout
fig.update_layout(
    margin={"r":0,"t":40,"l":0,"b":0} # Adjust margins
)

# Display the figure (this will open an interactive plot in your environment)
fig.show()

# To save the map as an HTML file (preserving interactivity):
# fig.write_html("unemployment_choropleth.html")
```

US County Unemployment Rate (2016)



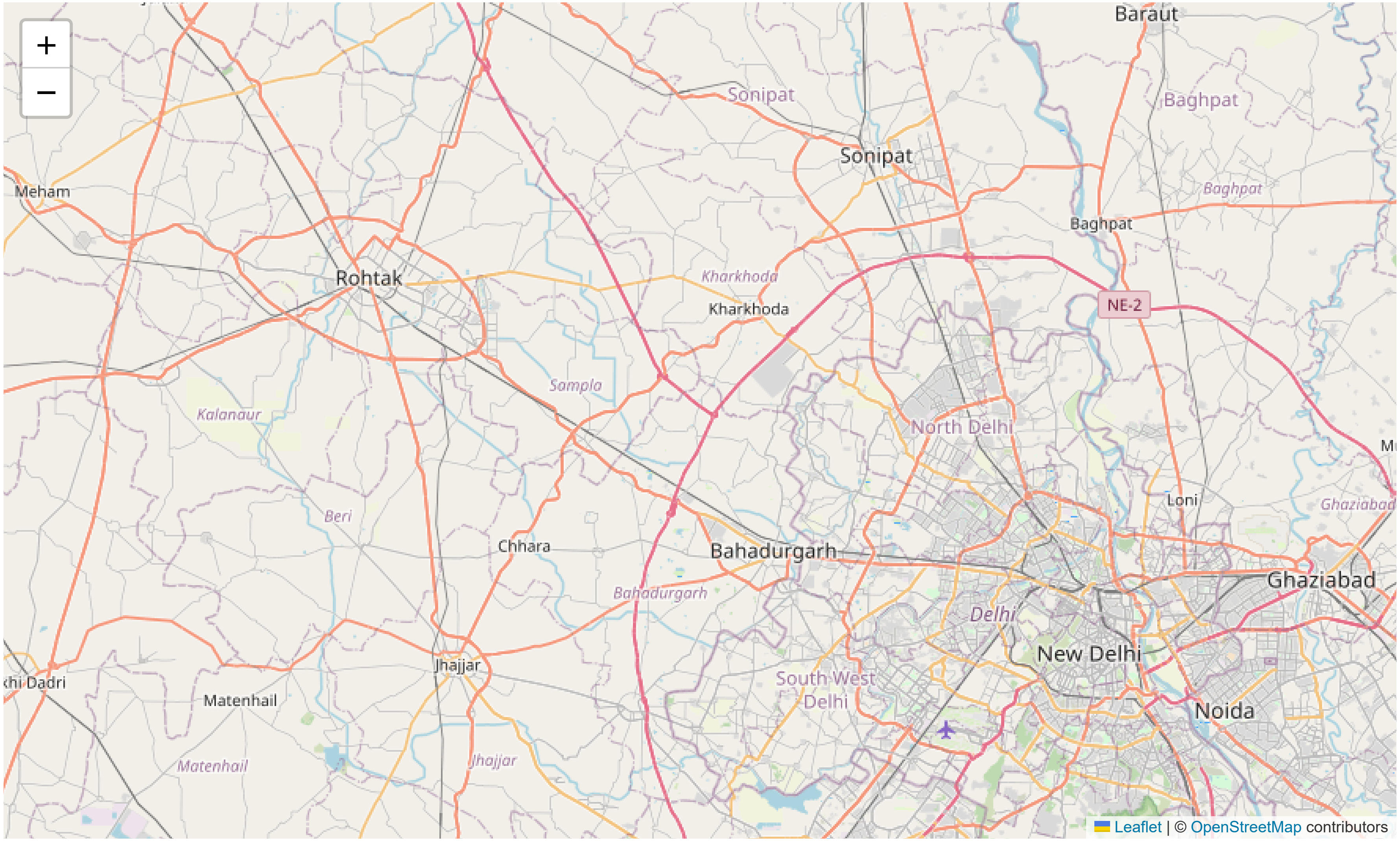
In [7]:
```
pip install folium
```

```
Requirement already satisfied: folium in c:\users\dell\anaconda3\lib\site-packages (0.20.0)
Requirement already satisfied: branca>=0.6.0 in c:\users\dell\anaconda3\lib\site-packages (from folium) (0.8.1)
Requirement already satisfied: jinja2>=2.9 in c:\users\dell\anaconda3\lib\site-packages (from folium) (3.1.3)
Requirement already satisfied: numpy in c:\users\dell\anaconda3\lib\site-packages (from folium) (1.26.4)
Requirement already satisfied: requests in c:\users\dell\anaconda3\lib\site-packages (from folium) (2.31.0)
Requirement already satisfied: xyzservices in c:\users\dell\anaconda3\lib\site-packages (from folium) (2022.9.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\dell\anaconda3\lib\site-packages (from jinja2>=2.9->folium) (2.1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\dell\anaconda3\lib\site-packages (from requests->folium)
(2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\dell\anaconda3\lib\site-packages (from requests->folium) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\dell\anaconda3\lib\site-packages (from requests->folium) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dell\anaconda3\lib\site-packages (from requests->folium) (2024.2.
2)
Note: you may need to restart the kernel to use updated packages.
```

In [8]:

```python
import folium

m = folium.Map(location = [28.6139, 77.2090], zoom_start = 10)
m
```

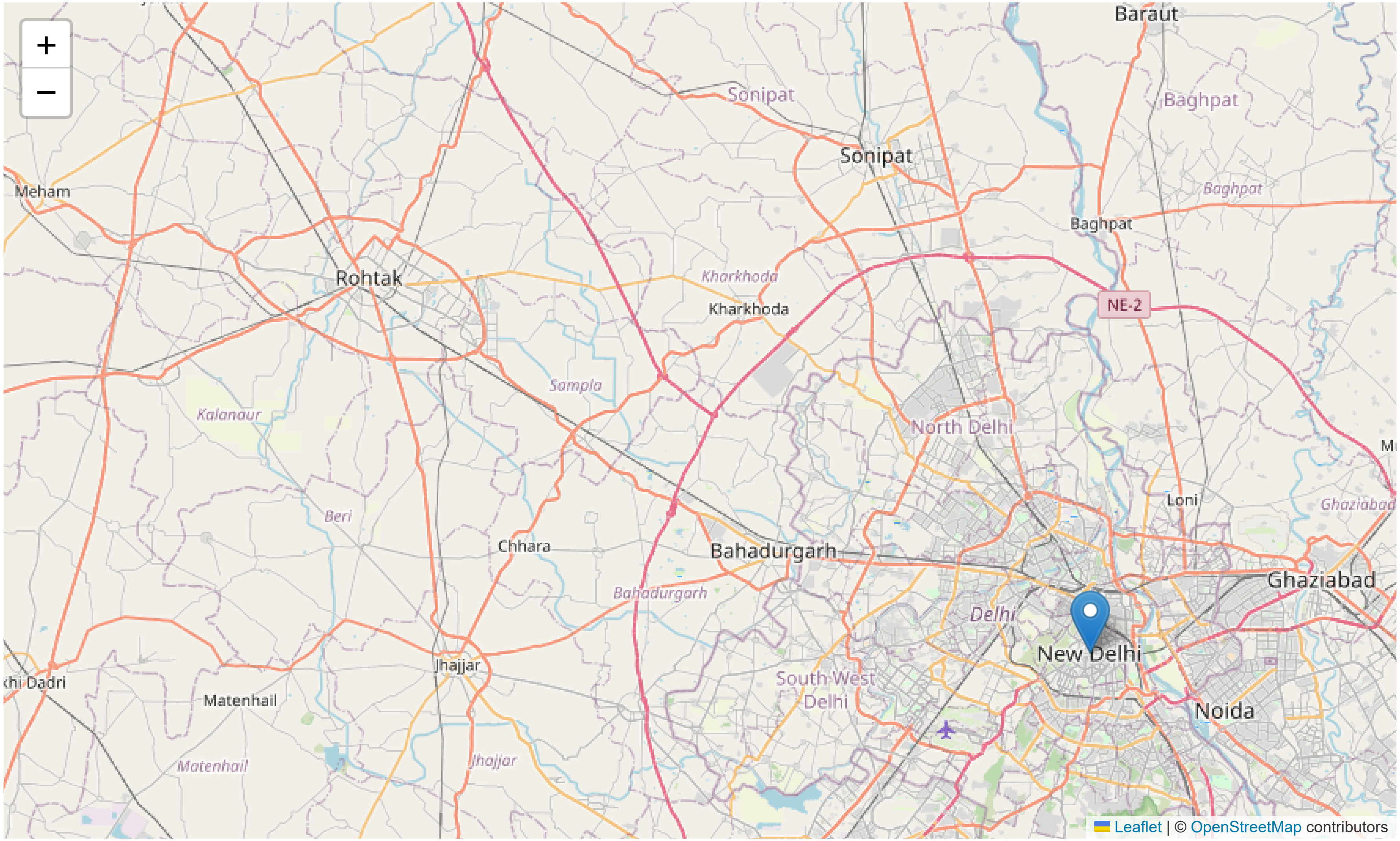
Out[8]:



In [9]:

```
m = folium.Map(location = [28.6139, 77.2090], zoom_start = 10)

folium.Marker(
    location = [28.6139, 77.2090],
    popup = "New Delhi",
    tooltip = "Click for info",
).add_to(m)
m
```

Out[9]:

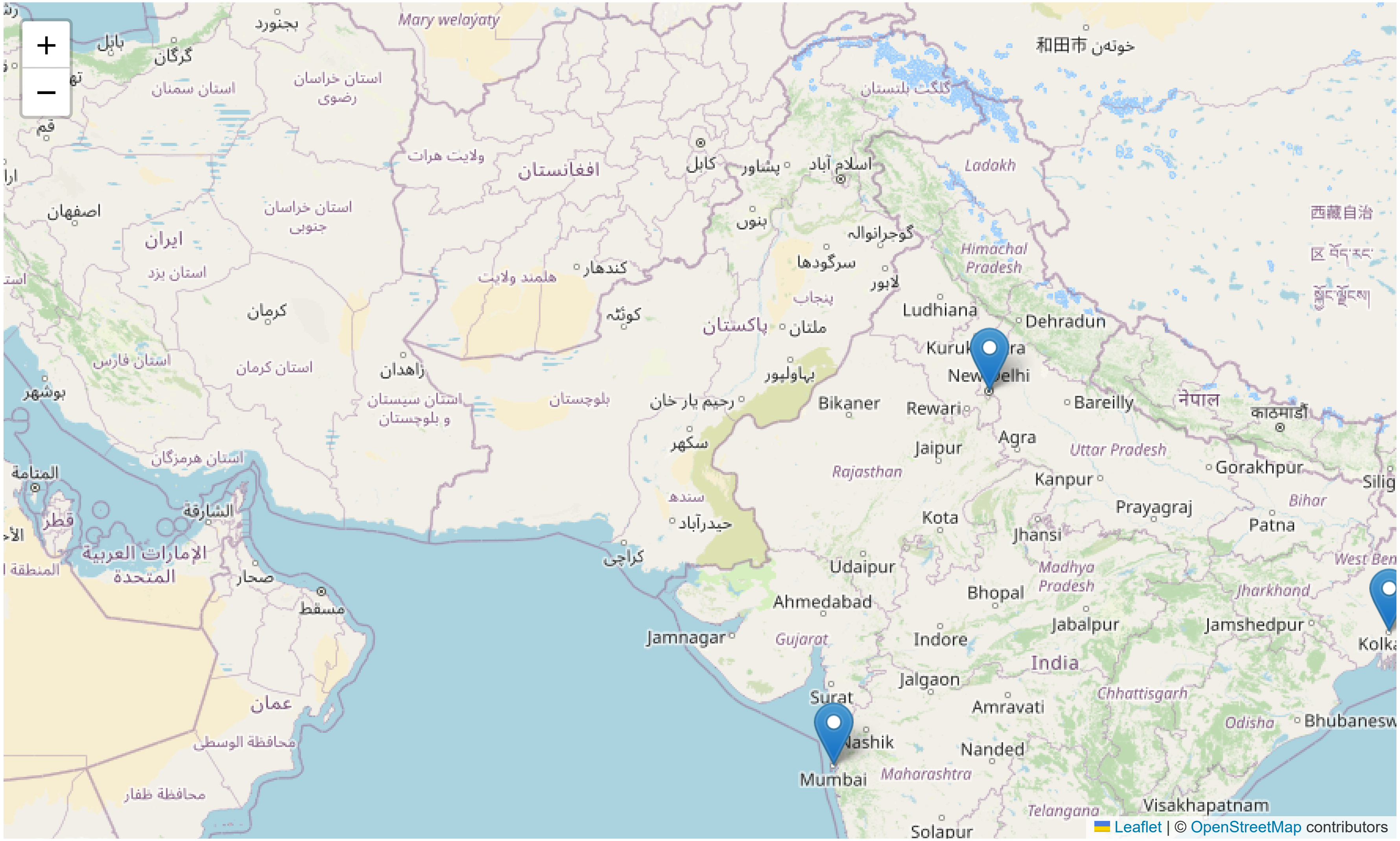


In [10]:

```
locations = [
    ["Delhi", 28.6139, 77.2090],
    ["Mumbai", 19.0760, 72.8777],
    ["Kolkata", 22.5726, 88.3639]
]

m = folium.Map(location = [22.0, 80.0], zoom_start = 5)
```

```
for city, lat,long in locations:
    folium.Marker([lat, long], popup = city).add_to(m)


m
```

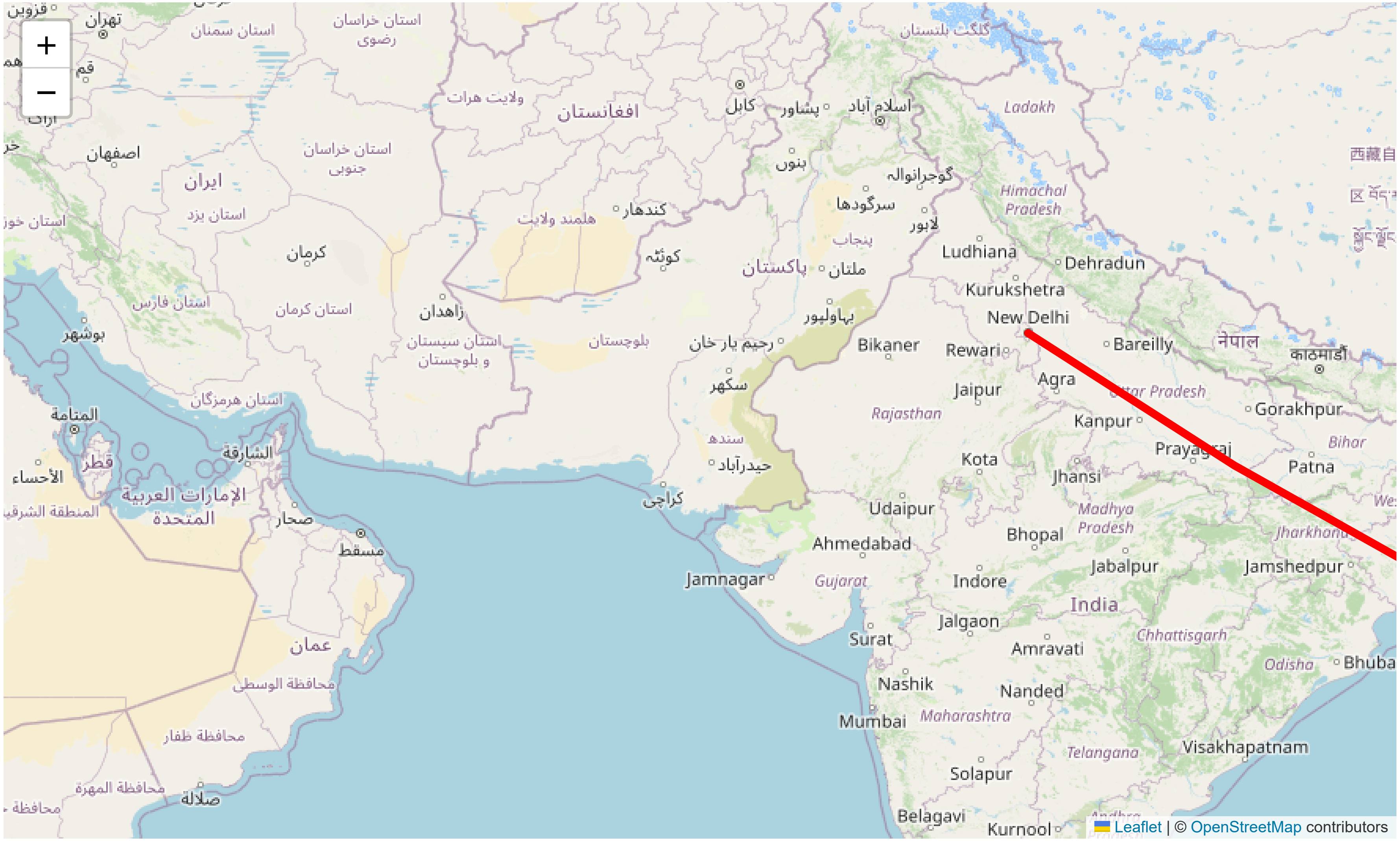
Out[10]:



In [15]:

```
m = folium.Map(location = [20.5, 78.9], zoom_start = 5)

route = [
    [28.6139, 77.2090],
    [25.3176, 82.9739],
    [22.5726, 88.3639]
]

folium.PolyLine(route, color = "red", weight = 5).add_to(m)

m
```

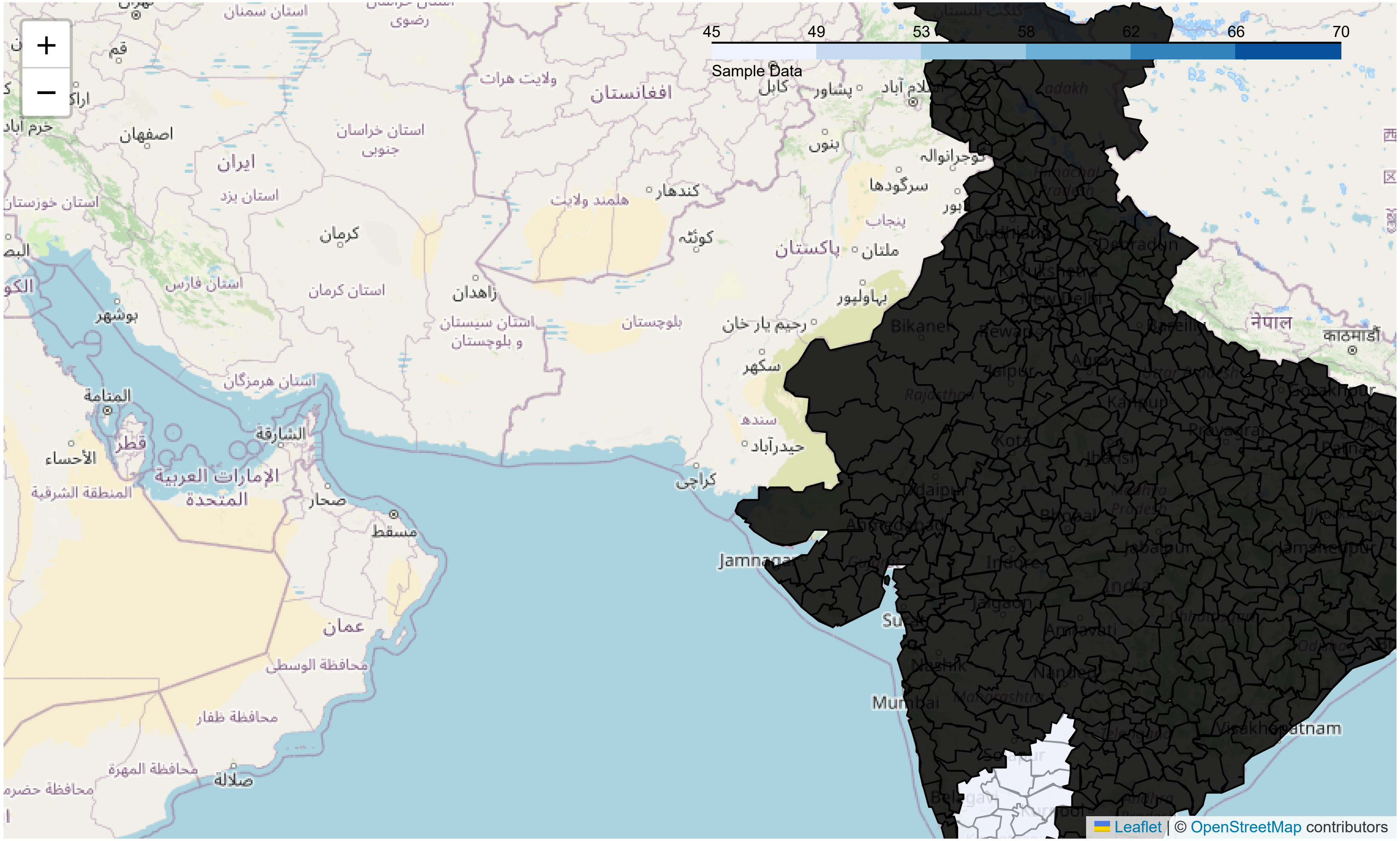
Out[15]:



In [29]:

```
import folium
import pandas as pd

data = {
    "State": ["Kerala", "Tamil Nadu", "Karnataka"],
    "Value" : [70,55,45]
}

df = pd.DataFrame(data)

geo_url = "https://cdn.jsdelivr.net/gh/udit-001/india-maps-data@8d907bc/geojson/india.geojson"

m = folium.Map(location = [20.0, 78.0], zoom_start = 5)

folium.Choropleth(
    geo_data = geo_url,
    data = df,
    columns = ["State", "Value"],
    key_on = "feature.properties.st_nm",
    #fill_color = 'YlGn',
    legend_name = "Sample Data"
).add_to(m)

m
```

Out[29]:


+
−
45
49
53
58
62
66
70
Sample Data
ایران
افغانستان
پاکستان
عمان
نेपाल
काठमाडौं
Jamnagar
Mumbai
Visakhapatnam
Leaflet | © OpenStreetMap contributors

In [ ]: