

# Assignmnet 2

Arose Niazi (FA18-BSE-010)

## Importing Libraries

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

## Presetup

```
In [2]: pakistan_loc = [30.3753,69.3451]
```

## Creating World Map

```
In [3]: pakistan = folium.Map(
        location=pakistan_loc,
        zoom_start=5
    )
pakistan
```

Out[3]:



## Working on Maps / Adding Multiple Maps

### Adding layers to the map

```
In [4]: tiles = ['stamenwatercolor', 'cartodbpositron', 'openstreetmap', 'stamenterrain']
        for tile in tiles:
            folium.TileLayer(tile).add_to(pakistan)
```

## Making more maps for multiple data

```
In [5]: cases_map = folium.Map(
        location=pakistan_loc,
        tiles='cartodbpositron',
        zoom_start=5
    )

    for tile in tiles:
        folium.TileLayer(tile).add_to(cases_map)

    deaths_map = folium.Map(
        location=pakistan_loc,
        tiles='cartodbpositron',
        zoom_start=5
    )

    for tile in tiles:
        folium.TileLayer(tile).add_to(deaths_map)
```

## Data Collection

### Scrapping Website for locations data

```
In [7]: info = pd.read_html('https://mm-ftw.net/pov.html')
provinces = pd.DataFrame(info[0])
provinces
```

Out[7]:

	IDs	Province	Latitude	Longitude
0	JK	Azad Jammu and Kashmir	33.9259	73.7810
1	BL	Balochistan	28.4907	65.0958
2	FT	F.A.T.A	32.6675	69.8597
3	IS	Islamabad Capital Territory	33.7205	73.0405
4	KP	Khyber Pakhtunkhwa	34.9526	72.3311
5	GB	Gilgit-Baltistan	35.8026	74.9832
6	PU	Punjab	31.1471	72.7097
7	SH	Sindh	25.8943	68.5247

## Scrapping Website for COVID-19 data

```
In [8]: info = pd.read_html("https://news.google.com/covid19/map?hl=en-PK&mid=%2Fm%2F05sb1&gl=PK&ceid=PK%3Aen", match="Cases")
covid19 = pd.DataFrame(info[0])
covid19.head()
```

Out[8]:

	Location	Total cases	New cases (1 day*)	New cases (last 60 days)	Cases per 1 million people	Deaths
0	Worldwide	46860757	No data	NaN	6026	1206069
1	Pakistan	336260	1167	NaN	1534	6849
2	Sindh	146774	443	NaN	3065	2633
3	Punjab	104894	340	NaN	No data	2372
4	Khyber Pakhtunkhwa	39749	100	NaN	1119	1280

## Data Cleaning

```
In [9]: #Convert to DataFrame
covid19 = pd.DataFrame(info[0])
covid19 = covid19.iloc[2:9]
covid19 = covid19.drop(["New cases (1 day*)", "New cases (last 60 days)", "Cases per 1 million people"], axis=1)
covid19 = covid19.rename(columns={'Location': 'Province'})
covid19['Total cases'] = covid19['Total cases'].astype('int')
covid19['Deaths'] = covid19['Deaths'].astype('int')
covid19
```

Out[9]:

	Province	Total cases	Deaths
2	Sindh	146774	2633
3	Punjab	104894	2372
4	Khyber Pakhtunkhwa	39749	1280
5	Islamabad Capital Territory	20243	222
6	Balochistan	15977	152
7	Azad Jammu and Kashmir	4330	98
8	Gilgit-Baltistan	4293	92

## Joining data / Clearning it / Calculating Data

```
In [10]: final_data = pd.merge(provinces, covid19, how='outer', on='Province')
fdata=final_data.sum(axis = 0, skipna = True)
final_data['Deaths'] = final_data['Deaths'].fillna(0)
final_data['Total cases'] = final_data['Total cases'].fillna(0)
final_data['Death Ratio'] = (final_data['Deaths'] / final_data['Total cases']) * 100
final_data['Cases Ratio'] = (final_data['Total cases'] / fdata['Total cases']) * 100
final_data['Death Ratio'] = final_data['Death Ratio'].fillna(0)
final_data
```

Out[10]:

	IDs	Province	Latitude	Longitude	Total cases	Deaths	Death Ratio	Cases Ratio
0	JK	Azad Jammu and Kashmir	33.9259	73.7810	4330.0	98.0	2.263279	1.287694
1	BL	Balochistan	28.4907	65.0958	15977.0	152.0	0.951368	4.751383
2	FT	F.A.T.A	32.6675	69.8597	0.0	0.0	0.000000	0.000000
3	IS	Islamabad Capital Territory	33.7205	73.0405	20243.0	222.0	1.096675	6.020044
4	KP	Khyber Pakhtunkhwa	34.9526	72.3311	39749.0	1280.0	3.220207	11.820912
5	GB	Gilgit-Baltistan	35.8026	74.9832	4293.0	92.0	2.143024	1.276691
6	PU	Punjab	31.1471	72.7097	104894.0	2372.0	2.261330	31.194314
7	SH	Sindh	25.8943	68.5247	146774.0	2633.0	1.793914	43.648962

## Map Data

### Loading map data from GeoJson files

```
In [11]: geoUrl = 'https://raw.githubusercontent.com/Arose-Niazi/Data-Science/master/Assignment%202/Data/PAK_adm1.json'
data_geojson_dict = json.loads(requests.get(geoUrl).text)
```

Working on first map; which shows the details

```
In [12]: folium.GeoJson(
        geoUrl,
        name='geojson'
    ).add_to(pakistan)

    layer_geom = folium.FeatureGroup(name='layer',control=False)
```

Adding markers to the map

```
In [13]: for province,lat,long,total_cases,Death,dratio,cratio in zip(list(final_data['Province']),list(final_data['Latitude']),list(final_data['Longitude']),list(final_data['Total cases']),list(final_data['Deaths']),list(final_data['Death Ratio']),list(final_data['Cases Ratio'])):
        folium.CircleMarker(location = [lat,long],
                            radius = 5,
                            color='red',
                            fill = True,
                            fill_color="red").add_to(pakistan)
        popup_html =str('<strong><b>Province: '+province+'</strong> <br>' +
                        '<strong><b>Total Cases :'+str(total_cases)+'</strong><br>' +
                        '<strong><b>Deaths :'+str(Death)+'</strong><br>' +
                        '<strong><b>Death Ratio :'+str("%0.2f%%" % dratio)+'</strong><br>' +
                        '<strong><b>Cases Ratio :'+str("%0.2f%%" % cratio)+'</strong>')
        folium.Marker(location = [lat,long],
                        popup = folium.Popup(popup_html, max_width=300,min_width=300), icon = folium.Icon(color='red',
icon='map-pin', prefix='fa') ).add_to(pakistan)
```

Adding popus for simple clicks

```

In [14]: provinces = []

for i in range(len(data_geojson_dict["features"])):
    temp_geojson = {
        "features": [data_geojson_dict["features"][i]],
        "type": "FeatureCollection"
    }
    temp_geojson_layer = folium.GeoJson(temp_geojson,
        highlight_function=lambda x: {'weight': 3, 'color': 'red'},
        control=False,
        style_function=lambda feature: {'color': 'green', 'weight': 3})
    popupstr = '<b>' + final_data['Province'][i] + '</b><br>' + '<b>Total Cases :'+str(final_data['Total cases'
][i])+'</b><br>' + '<b>Deaths :'+str(final_data['Deaths'][i])+'</b><br>'

    folium.Popup(popupstr, max_width = 500).add_to(temp_geojson_layer)
    temp_geojson_layer.add_to(layer_geom)

layer_geom.add_to(pakistan)
folium.LayerControl(autoZIndex=False, collapsed=False).add_to(pakistan)

```

Out[14]: <folium.map.LayerControl at 0x25c6c958160>

## Working on second map; which shows cases ratio

Creating a tooltip to add for cases

```

In [15]: tooltip_text = []
for idx in range(len(final_data)):
    tooltip_text.append(final_data["Province"][idx] + "</BR>Cases:" + str(final_data["Total cases"][idx]) + "</BR>Ra
tio:" + str("%0.2f%%" % (final_data["Cases Ratio"][idx])))

```

Append a tooltip column with customised text

```

In [16]: for idx in range(len(tooltip_text)):
    data_geojson_dict['features'][idx]['properties']['cases_tt'] = tooltip_text[idx]

```



## Adding data to the map

```
In [17]: choropleth = folium.Choropleth(
        geo_data=data_geojson_dict,
        name='choropleth',
        data=final_data,
        columns=['IDs', 'Cases Ratio'],
        key_on='feature.id',
        fill_color='OrRd',
        fill_opacity=0.7,
        line_opacity=0.2,
        legend_name='Cases Ratio',
        highlight = True
    ).add_to(cases_map)

    folium.LayerControl().add_to(cases_map)

    choropleth.geojson.add_child(
        folium.features.GeoJsonTooltip(['cases_tt'], labels=False)
    )
```

```
Out[17]: <folium.features.GeoJson at 0x25c697b6a30>
```

## Working on third map; which shows deaths ratio

Creating a tooltip to add for deaths and appending the tooltip column with customised text

```
In [18]: tooltip_text = []
        for idx in range(len(final_data)):
            tooltip_text.append(final_data["Province"][idx] + "</BR>Cases:" + str(final_data["Total cases"][idx]) + "</BR>Deaths:" + str(final_data["Deaths"][idx]) + "</BR>Ratio:" + str("%0.2f%%" % (final_data["Death Ratio"][idx])))
        for idx in range(len(tooltip_text)):
            data_geojson_dict['features'][idx]['properties']['deaths_tt'] = tooltip_text[idx]
```

## Adding data to the map

```
In [19]: choropleth = folium.Choropleth(  
        geo_data=data_geojson_dict,  
        name='choropleth',  
        data=final_data,  
        columns=['IDs', 'Death Ratio'],  
        key_on='feature.id',  
        fill_color='OrRd',  
        fill_opacity=0.7,  
        line_opacity=0.2,  
        legend_name='Cases to Death Ratio',  
        highlight = True  
    ).add_to(deaths_map)  
  
    folium.LayerControl().add_to(deaths_map)  
  
    choropleth.geojson.add_child(  
        folium.features.GeoJsonTooltip(['deaths_tt'], labels=False)  
    )
```

```
Out[19]: <folium.features.GeoJson at 0x25c6c281310>
```

## Saving to file

```
In [20]: pakistan.save('Covid_Statistics.html')  
        cases_map.save('Covid_Cases.html')  
        deaths_map.save('Covid_Deaths.html')
```

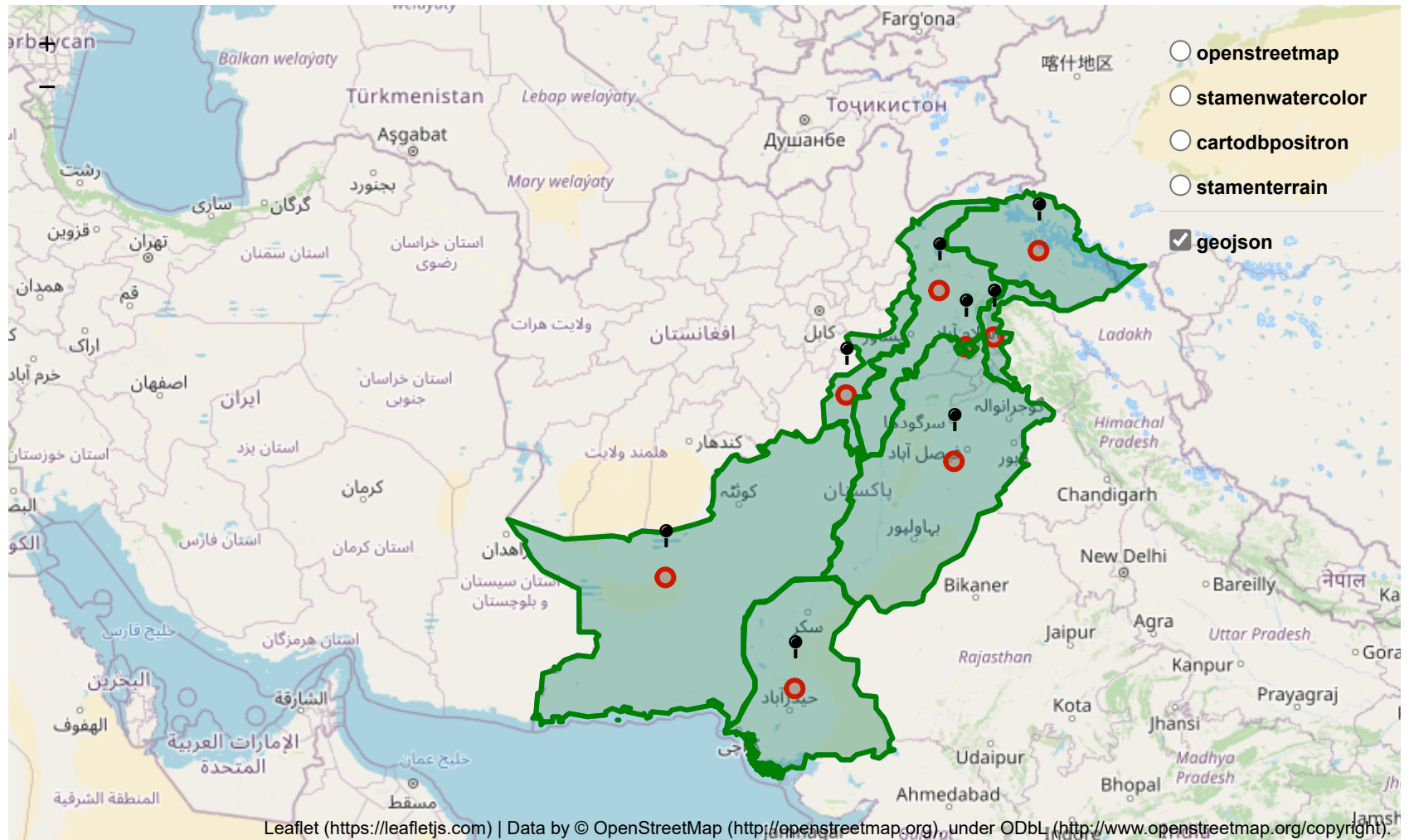
## Displayng the maps

### Pakistan COVID Map

Click the province to reveal stats

In [21]: pakistan

Out[21]:

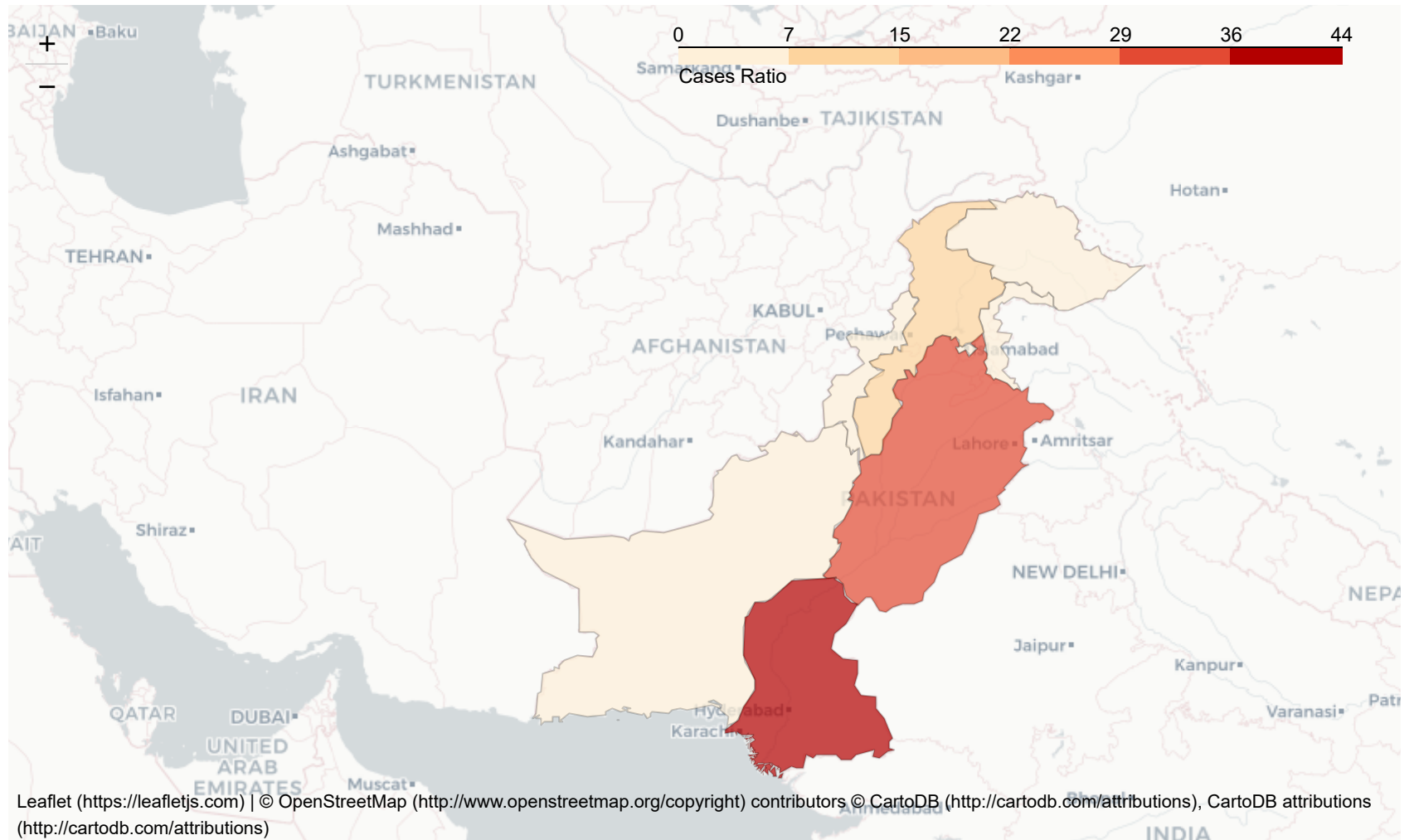


## Pakistan Cases map

Hover the province to reveal stats

In [22]: cases\_map

Out[22]:

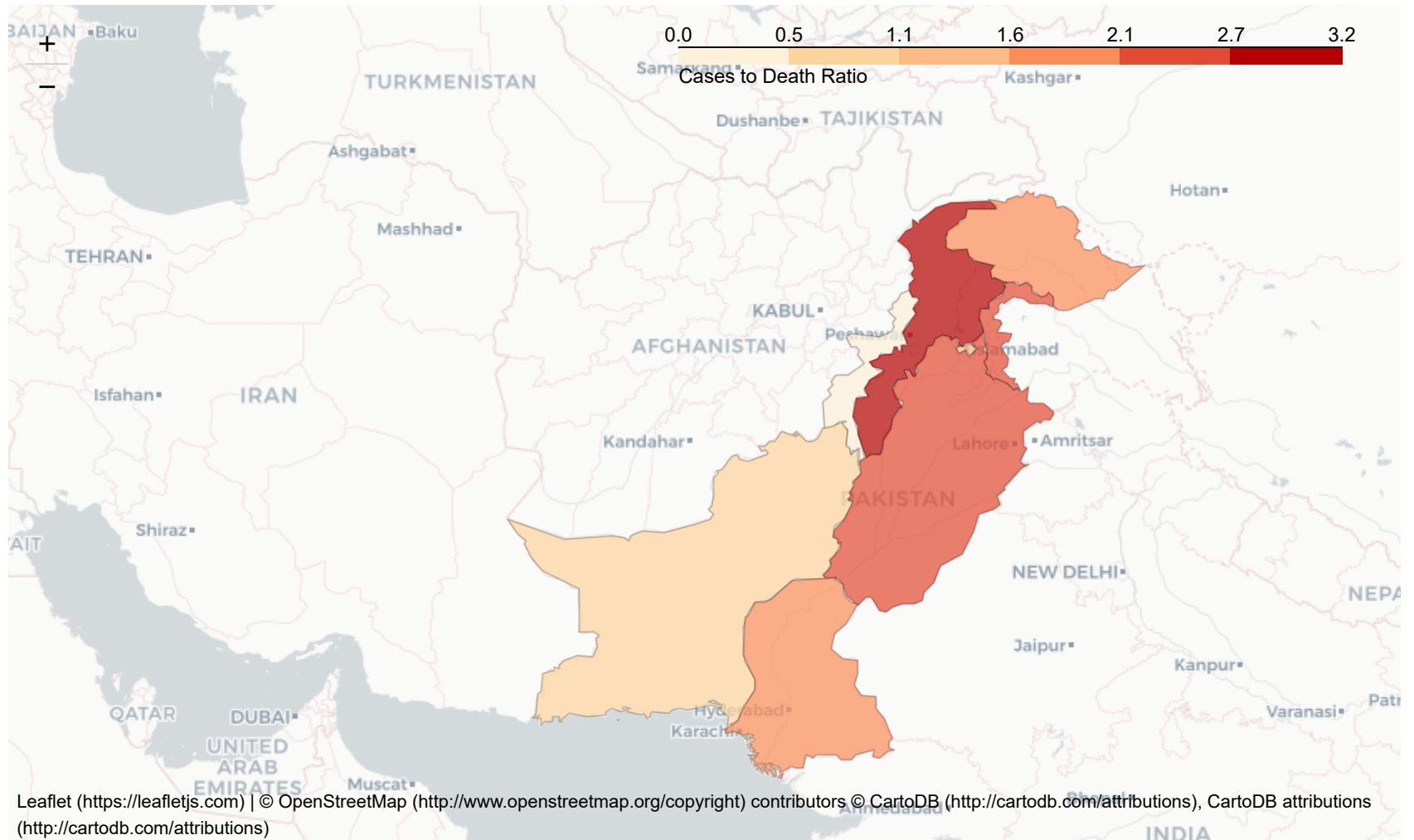


## Pakistan Deaths map

Hover the province to reveal stats

In [23]: deaths\_map

Out[23]:



In [ ]: