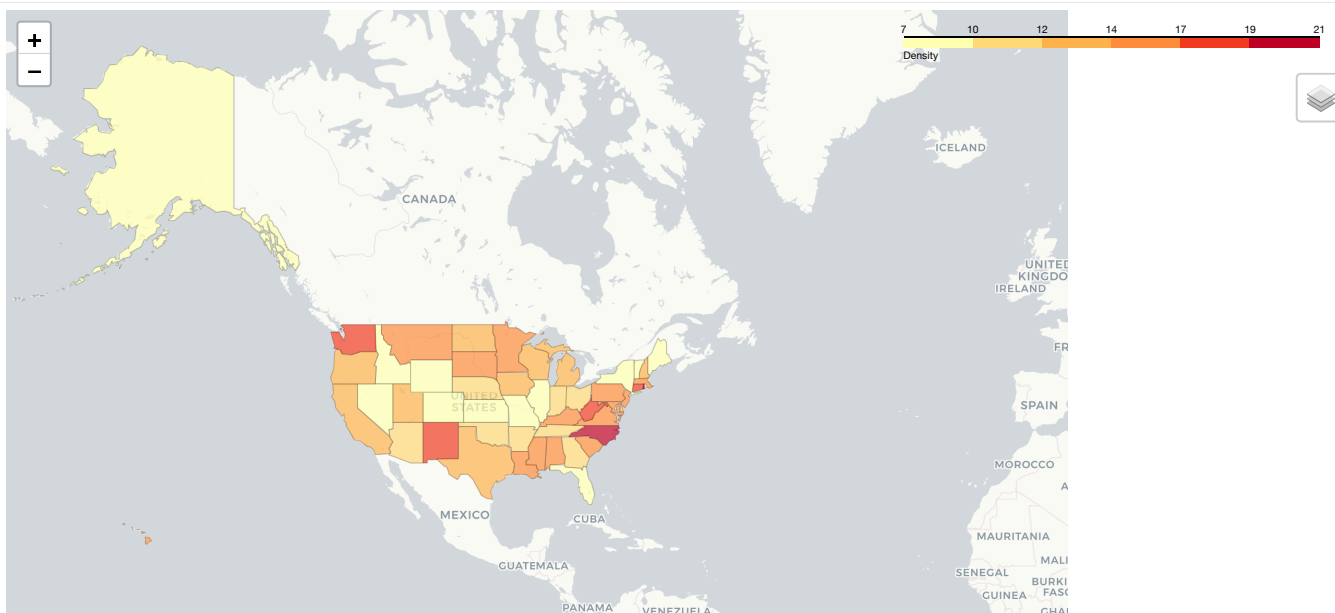


▼ folium Choropleth

Basic Choropleth:

```
1 import folium
2 import geopandas as gpd
3 import json
4 import numpy as np
5
6 # Load data
7 world = gpd.read_file("https://raw.githubusercontent.com/python-visualization/folium/master/examples/data/us-states.json")
8
9 # Create a dummy 'density' column (since it's not in the original data)
10 # For demonstration, assign random density values based on state name length
11 world['density'] = world['name'].apply(lambda x: len(x) + np.random.rand() * 10)
12
13 # Convert to GeoJSON (folium requirement)
14 world_json = world.to_json()
15
16 # Create base map
17 m = folium.Map(location=[20, 0], zoom_start=2, tiles='CartoDB positron')
18
19 # Add choropleth
20 folium.Choropleth(
21     geo_data=world_json,
22     name='Density',
23     data=world,
24     columns=['name', 'density'],
25     key_on='feature.properties.name',
26     fill_color='YlOrRd',
27     fill_opacity=0.7,
28     line_opacity=0.2,
29     legend_name='Density'
30 ).add_to(m)
31
32 folium.LayerControl().add_to(m)
33
34 m.save('world_population_choropleth.html')
35 display(m)
```



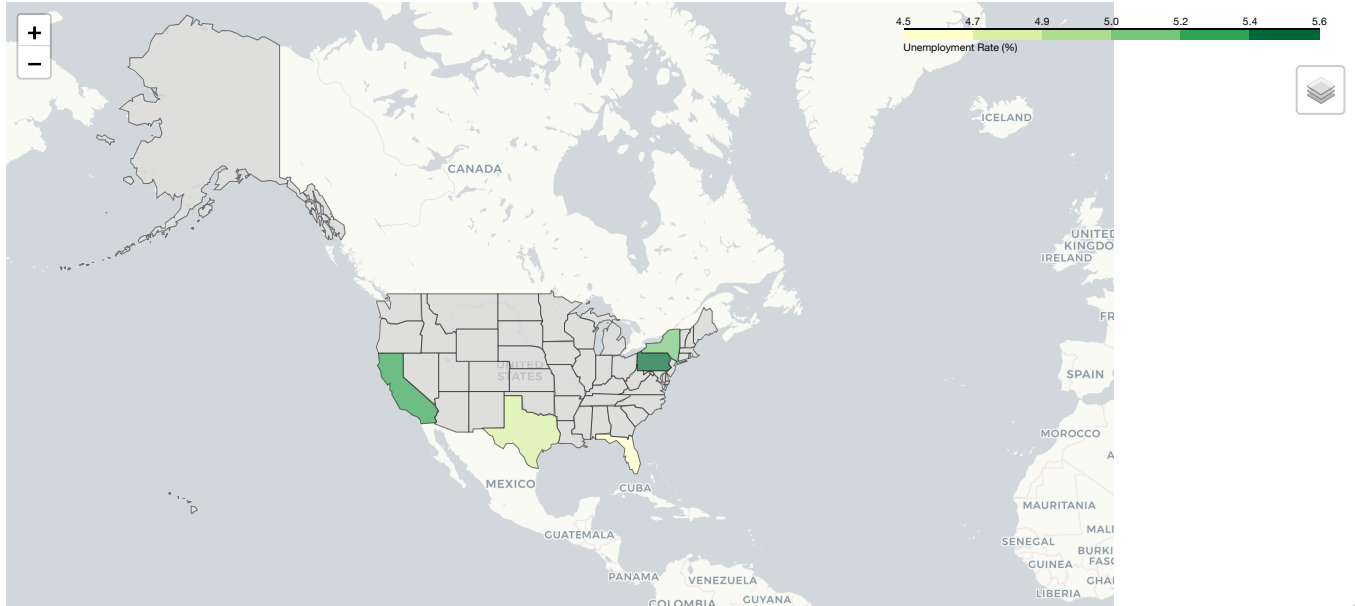
With Pandas DataFrame:

```
1 import folium
2 import pandas as pd
3 import geopandas as gpd
4
5 # Load US states
6 us_states = gpd.read_file('https://raw.githubusercontent.com/python-visualization/folium/master/examples/data/us-states.json')
7
8 # Create sample data
9 state_data = pd.DataFrame({
10     'state': ['California', 'Texas', 'Florida', 'New York', 'Pennsylvania'],
11     'unemployment': [5.3, 4.8, 4.5, 5.1, 5.6]
12 })
13
14 # Create map
15 m = folium.Map(location=[37.8, -96], zoom_start=4, tiles='CartoDB positron')
16
17 folium.Choropleth(
18     geo_data=us_states,
19     name='choropleth',
```

```

20 data=state_data,
21 columns=['state', 'unemployment'],
22 key_on='feature.properties.name',
23 fill_color='YlGn',
24 fill_opacity=0.7,
25 line_opacity=0.5,
26 legend_name='Unemployment Rate (%)',
27 nan_fill_color='lightgray'
28 ).add_to(m)
29
30 folium.LayerControl().add_to(m)
31 m.save('us_unemployment.html')
32 display(m)

```

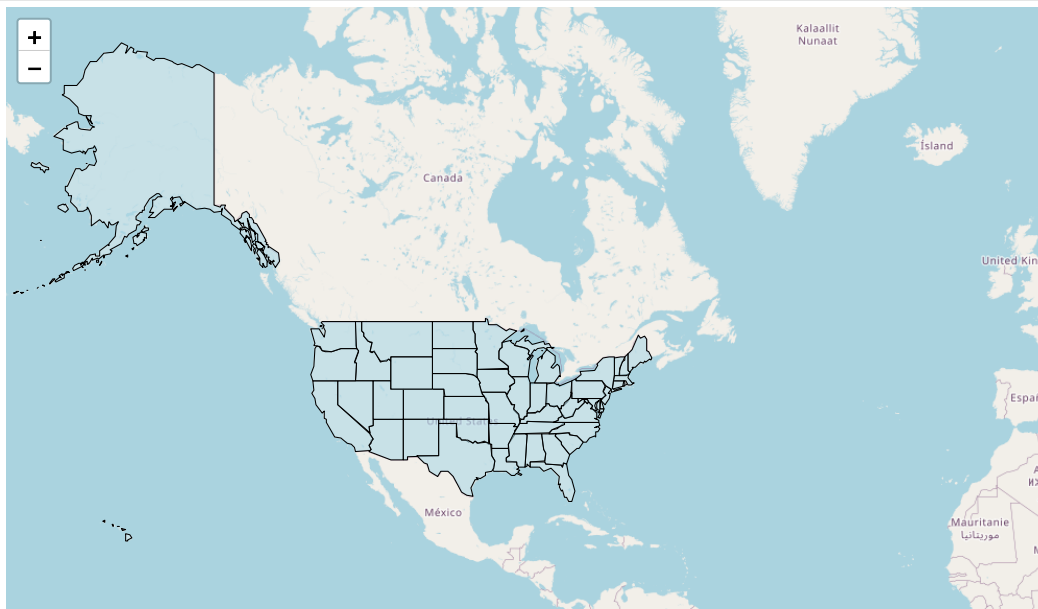


Interactive with Tooltips:

```

1 import folium
2 import geopandas as gpd
3
4 world = gpd.read_file("https://raw.githubusercontent.com/python-visualization/folium/master/examples/data/us-states.json")
5
6 m = folium.Map(location=[20, 0], zoom_start=2)
7
8 # Add GeoJson with tooltip
9 folium.GeoJson(
10     world,
11     name='countries',
12     tooltip=folium.GeoJsonTooltip(
13         fields=['name'],
14         aliases=['State Name'],
15         localize=True
16     ),
17     style_function=lambda x: {
18         'fillColor': 'lightblue',
19         'color': 'black',
20         'weight': 1,
21         'fillOpacity': 0.6
22     },
23     highlight_function=lambda x: {
24         'fillColor': 'darkblue',
25         'fillOpacity': 0.8
26     }
27 ).add_to(m)
28
29 m.save('world_interactive_tooltip.html')
30 display(m)

```

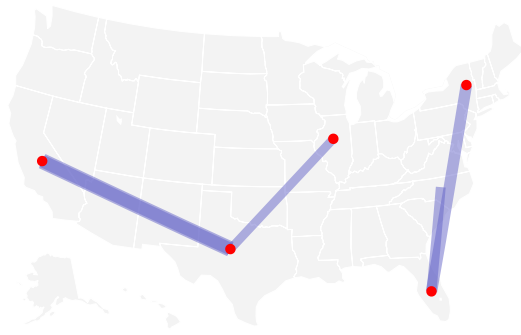


▼ Sankey on Maps

Using plotly for Sankey-Style Flows:

```
1 import plotly.graph_objects as go
2 import pandas as pd
3
4 # Migration data example
5 migration = pd.DataFrame({
6     'origin': ['California', 'Texas', 'Florida', 'New York', 'Illinois'],
7     'destination': ['Texas', 'California', 'North Carolina', 'Florida', 'Texas'],
8     'migrants': [150000, 120000, 95000, 110000, 85000],
9     'origin_lat': [36.7783, 31.9686, 27.6648, 43.2994, 40.6331],
10    'origin_lon': [-119.4179, -99.9018, -81.5158, -74.2179, -89.3985],
11    'dest_lat': [31.9686, 36.7783, 35.7596, 27.6648, 31.9686],
12    'dest_lon': [-99.9018, -119.4179, -79.0193, -81.5158, -99.9018]
13 })
14
15 # Create figure
16 fig = go.Figure()
17
18 # Add flows as lines with width based on volume
19 for idx, row in migration.iterrows():
20     fig.add_trace(go.Scattergeo(
21         lon=[row['origin_lon'], row['dest_lon']],
22         lat=[row['origin_lat'], row['dest_lat']],
23         mode='lines',
24         line=dict(
25             width=row['migrants']/10000,
26             color='rgba(100, 100, 200, 0.5)'
27         ),
28         hoverinfo='text',
29         text=f"{row['origin']} → {row['destination']}<br>{row['migrants']:,} migrants"
30     ))
31
32 # Add origin points
33 fig.add_trace(go.Scattergeo(
34     lon=migration['origin_lon'],
35     lat=migration['origin_lat'],
36     mode='markers',
37     marker=dict(size=10, color='red'),
38     text=migration['origin'],
39     hoverinfo='text'
40 ))
41
42 fig.update_layout(
43     title='Interstate Migration Flows',
44     geo=dict(
45         scope='usa',
46         projection_type='albers usa',
47         showland=True,
48         landcolor='rgb(243, 243, 243)',
49         coastlinecolor='rgb(204, 204, 204)',
50     ),
51     showlegend=False,
52     height=600
53 )
54
55 fig.show()
```

Interstate Migration Flows



True Sankey Diagram (Non-Geographic):

```
1 import plotly.graph_objects as go
2
3 # Sankey for regional trade
4 fig = go.Figure(data=[go.Sankey(
5     node=dict(
6         pad=15,
7         thickness=20,
8         line=dict(color='black', width=0.5),
9         label=['North America', 'Europe', 'Asia',
10              'Manufacturing', 'Services', 'Agriculture',
11              'Exports', 'Domestic'],
12         color=['blue', 'green', 'red', 'orange', 'purple', 'brown', 'pink', 'gray']
13     ),
14     link=dict(
15         source=[0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5],
16         target=[3, 4, 3, 4, 3, 4, 6, 7, 6, 7, 6, 7],
17         value=[100, 80, 90, 70, 150, 60, 200, 180, 120, 90, 40, 30]
18     )
19 )])
20
21 fig.update_layout(title='Regional Trade Flows', font_size=12)
22 fig.show()
```

Regional Trade Flows

