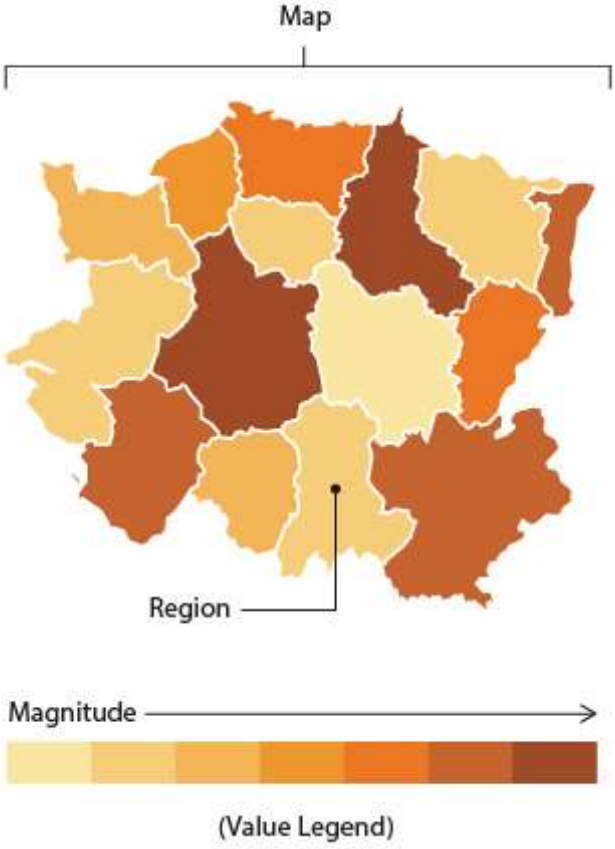
Plotting Choropleth Maps using Python

A choropleth map uses colour coding to indicate quantitative values across geographical areas on a map.



> A choropleth map is a type of thematic map in which areas are shaded or patterned in proportion to a statistical variable that represents an aggregate summary of a geographic characteristic within each area, such as population density or per-capita income.

In [1]:
```
import json
import numpy as np
import pandas as pd
import plotly.express as px
```

```
C:\Users\User\anaconda3\Lib\site-packages\pandas\core\arrays\masked.py:60: U
serWarning: Pandas requires version '1.3.6' or newer of 'bottleneck' (versio
n '1.3.5' currently installed).
  from pandas.core import (
```

In [2]:
```
import plotly.io as pio
pio.renderers.default = 'browser'
```

In [3]:
```
india_states = json.load(open("states_india.geojson", "r"))
```

In [4]:
```
state_id_map = {}
for feature in india_states["features"]:
    feature["id"] = feature["properties"]["state_code"]
    state_id_map[feature["properties"]["st_nm"]] = feature["id"]
```

In [5]:
```
df = pd.read_csv("india_census.csv")
df["Density"] = df["Density[a]"].apply(lambda x: int(x.split("/")[0].replace(
df["id"] = df["State or union territory"].apply(lambda x: state_id_map[x])
```

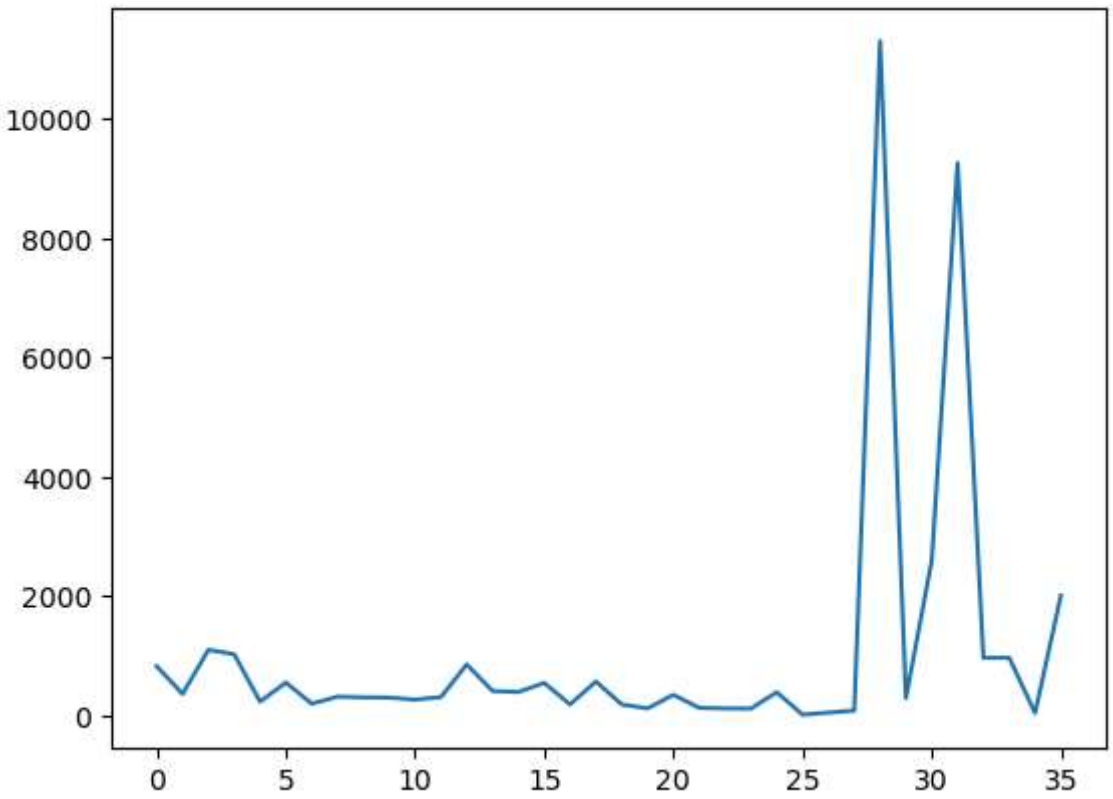
In [6]:
```
df.head()
```

Out[6]:

	Rank	State or union territory	Population	Population (%)	Decadal growth(2001–2011)	Rural population	Percent rural	Urban population	P
0	1.0	Uttar Pradesh	199812341	NaN	20.20%	155317278	NaN	44495063	
1	2.0	Maharashtra	112374333	NaN	20.00%	61556074	NaN	50818259	
2	3.0	Bihar	104099452	NaN	25.40%	92341436	NaN	11758016	
3	4.0	West Bengal	91276115	NaN	13.80%	62183113	NaN	29093002	
4	5.0	Madhya Pradesh	72626809	NaN	16.30%	52557404	NaN	20069405	

In [7]:
```
df["Density"].plot()
```

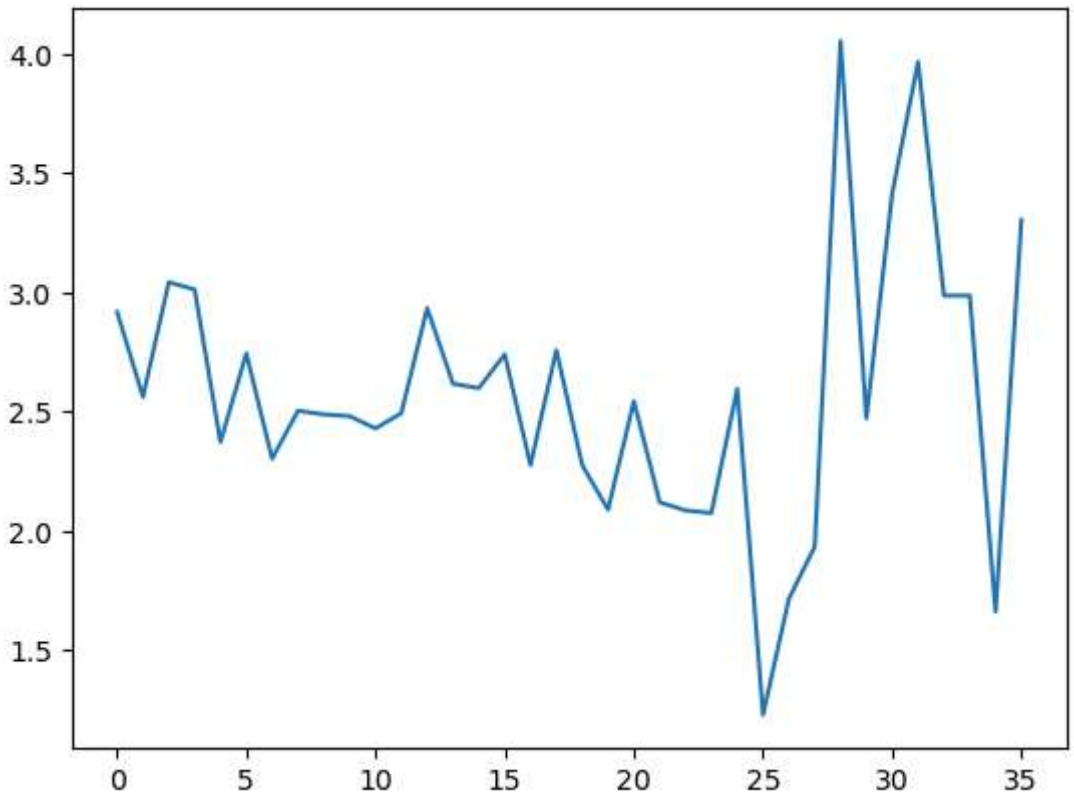
Out[7]: <Axes: >



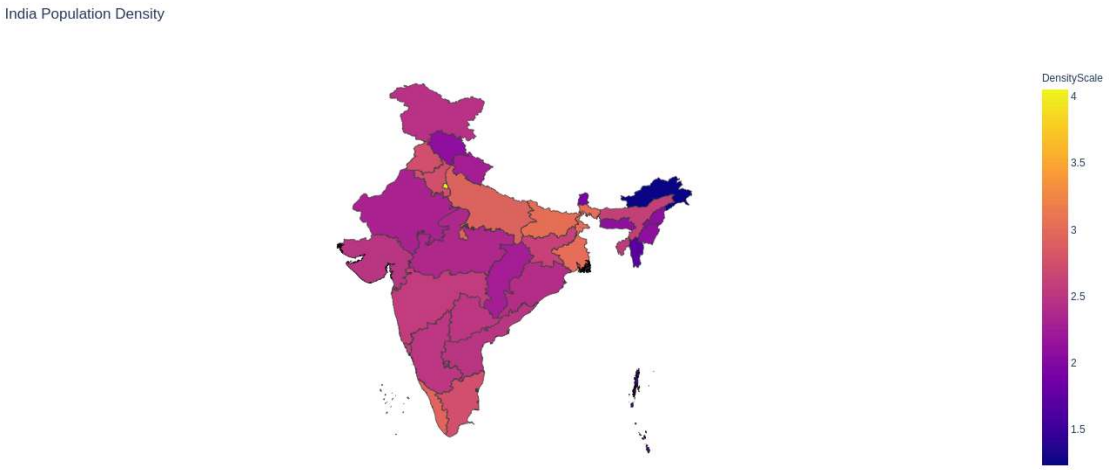
In [8]:
```
df["DensityScale"] = np.log10(df["Density"])
```

In [9]:
```
df["DensityScale"].plot()
```

Out[9]: <Axes: >

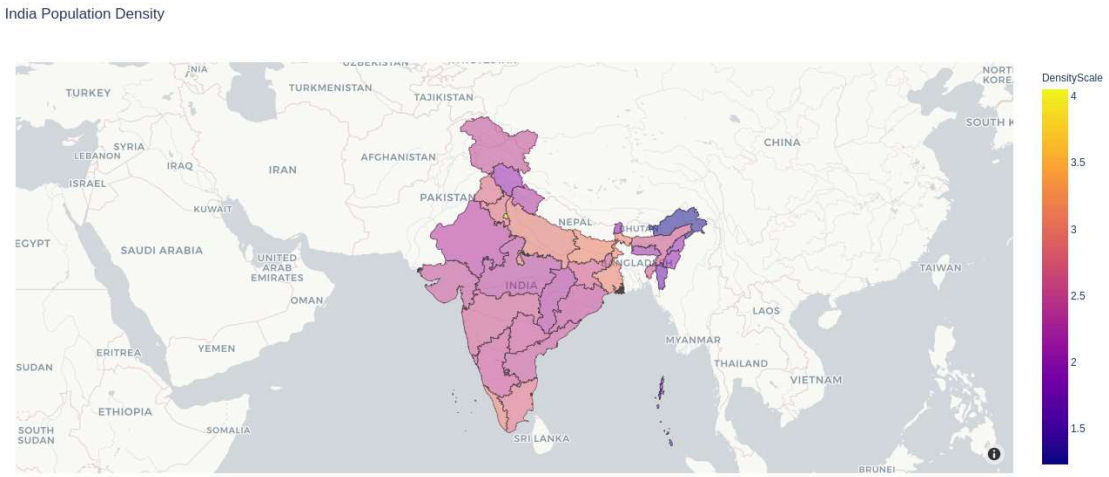


In [10]:
```
fig = px.choropleth(
    df,
    locations="id",
    geojson=india_states,
    color="DensityScale",
    hover_name="State or union territory",
    hover_data=["Density"],
    title="India Population Density",
)
fig.update_geos(fitbounds="locations", visible=False)
fig.show()
```



In [ ]:

In [11]:
```
fig = px.choropleth_mapbox(
    df,
    locations="id",
    geojson=india_states,
    color="DensityScale",
    hover_name="State or union territory",
    hover_data=["Density"],
    title="India Population Density",
    mapbox_style="carto-positron",
    center={"lat": 24, "lon": 78},
    zoom=3,
    opacity=0.5,
)
fig.show()
```

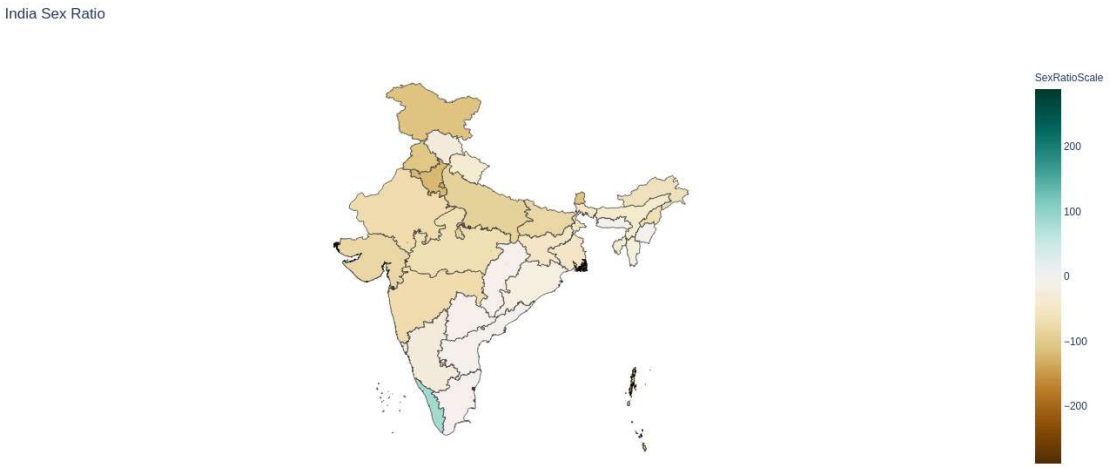


In [ ]:

In [12]:
```
df["SexRatioScale"] = df["Sex ratio"] - 1000
```

In [18]:
```
fig = px.choropleth(
    df,
    locations="id",
    geojson=india_states,
    color="SexRatioScale",
    hover_name="State or union territory",
    hover_data=["Sex ratio"],
    title="India Sex Ratio",
    color_continuous_scale=px.colors.diverging.BrBG,
    color_continuous_midpoint=0,
)
fig.update_geos(fitbounds="locations", visible=False)
fig.show()
```

```
<IPython.core.display.Javascript object>
```



In [14]:
```
import numpy as np

# Generate random latitude and longitude within India's approximate bounds
np.random.seed(0)  # For reproducibility
df['lat'] = np.random.uniform(low=8.0, high=37.0, size=len(df))  # India's la
df['lon'] = np.random.uniform(low=68.0, high=97.0, size=len(df))  # India's l

# Example population values (replace with actual data if available)
df['Population'] = np.random.randint(1000000, 50000000, size=len(df))  # Exam

# Plot the Proportional Symbol Map
fig = px.scatter_geo(
    df,
    lat='lat', lon='lon', size='Population',  # Use 'Population' column for s
    hover_name="State or union territory",  # Name to display on hover
    geojson=india_states,  # Indian states boundaries
    locations="id", locationmode="geojson-id",
    title="Proportional Symbol Map of Indian States by Population"
)

# Customize map appearance
fig.update_geos(fitbounds="locations", visible=False)
fig.update_layout(margin={"r":0,"t":30,"l":0,"b":0})

# Show the map
fig.show()
```

In [ ]:
```
```