

India Population: Visualization through Geo Spatial Data

In this project, I want to explore the population and other attributes for the country of India. I want to visualize these attributes through geospatial data and plot them on interactive maps.

For this project, the most important attributes are the population of each state and their geospatial geometry.

1) Dataset containing Population: [Population of India](#): This dataset contains the population distribution by state, gender, sex & region

2) Dataset containing the Geospatial coordinates of each State and Union Territory: [India GIS Data](#): This dataset contains the geospatial data of each state of India

Importing the relevant modules required

```
In [1]: import geopandas as gpd  
import pandas as pd
```

I imported the geospatial coordinates of each state of India into a variable 'india_map'

```
In [2]: # Read the shapefile or GeoJSON file of India  
india_map = gpd.read_file('C:/Users/karan/Downloads/archive (3)/India States/Indian_states.shp')
```

```
In [3]: india_map
```

Out[3]:

	st_nm	geometry
0	Andaman & Nicobar Island	MULTIPOLYGON (((93.71976 7.20707, 93.71909 7.2...
1	Arunanchal Pradesh	POLYGON ((96.16261 29.38078, 96.16860 29.37432...
2	Assam	MULTIPOLYGON (((89.74323 26.30362, 89.74290 26...
3	Bihar	MULTIPOLYGON (((84.50720 24.26323, 84.50355 24...
4	Chandigarh	POLYGON ((76.84147 30.75996, 76.83599 30.73623...
5	Chhattisgarh	POLYGON ((83.33532 24.09885, 83.35346 24.09627...
6	Dadara & Nagar Havelli	POLYGON ((73.20657 20.12216, 73.20797 20.10650...
7	Daman & Diu	MULTIPOLYGON (((72.89335 20.44539, 72.89281 20...
8	Goa	MULTIPOLYGON (((74.11918 14.75344, 74.11350 14...
9	Gujarat	MULTIPOLYGON (((71.70375 20.99958, 71.70375 20...
10	Haryana	POLYGON ((76.85065 30.87512, 76.86594 30.86691...
11	Himachal Pradesh	POLYGON ((76.79634 33.25490, 76.80351 33.25275...
12	Jammu & Kashmir	POLYGON ((74.73451 37.02068, 74.73647 37.01937...
13	Jharkhand	POLYGON ((87.60582 25.31512, 87.61279 25.31184...
14	Karnataka	MULTIPOLYGON (((74.69694 13.32782, 74.69562 13...
15	Kerala	POLYGON ((74.99575 12.79227, 75.00006 12.78777...
16	Lakshadweep	MULTIPOLYGON (((74.10131 11.20431, 74.09908 11...
17	Madhya Pradesh	MULTIPOLYGON (((74.17932 22.39059, 74.17776 22...
18	Maharashtra	MULTIPOLYGON (((73.46270 16.03710, 73.46178 16...
19	Manipur	POLYGON ((94.57602 25.64221, 94.57487 25.63858...
20	Meghalaya	POLYGON ((91.85632 26.10353, 91.86717 26.09906...
21	Mizoram	POLYGON ((92.80022 24.41630, 92.80310 24.41603...
22	Nagaland	POLYGON ((95.21458 26.93095, 95.21719 26.92818...
23	NCT of Delhi	POLYGON ((77.09361 28.86940, 77.10973 28.86722...
24	Puducherry	MULTIPOLYGON (((79.76528 10.99681, 79.76963 10...
25	Punjab	POLYGON ((75.88469 32.48841, 75.90303 32.47169...
26	Rajasthan	POLYGON ((73.89849 29.97896, 73.90021 29.97140...
27	Sikkim	POLYGON ((88.65381 28.09883, 88.66265 28.08956...
28	Tamil Nadu	MULTIPOLYGON (((78.19188 8.72730, 78.18996 8.7...
29	Telangana	POLYGON ((81.05614 17.79097, 81.05834 17.75464...
30	Tripura	POLYGON ((92.22108 24.50131, 92.22771 24.49862...
31	Uttar Pradesh	MULTIPOLYGON (((80.44802 24.99631, 80.44080 24...
32	Uttarakhand	POLYGON ((79.21047 31.34846, 79.21386 31.34680...
33	West Bengal	MULTIPOLYGON (((88.01861 21.57278, 88.01889 21...
34	Odisha	MULTIPOLYGON (((86.38937 19.96351, 86.38840 19...
35	Andhra Pradesh	MULTIPOLYGON (((81.10380 17.82269, 81.10610 17...

After looking through the dataset, I noticed a some different format of State names and a few spelling mistakes in the names of states. Since I need to combine the file in india_map to the other dataset with the state population on the basis of state name, I need to make sure that both datasets have the same state names.

In [4]:

#Manually changing the names of the states

```
india_map.st_nm[0]='Andaman & Nicobar Islands'
india_map.st_nm[1]='Arunachal Pradesh'
india_map.st_nm[6]='Dadra and Nagar Haveli'
india_map.st_nm[23]='Delhi'
```

```
C:\Users\karan\AppData\Local\Temp\ipykernel_30880\3782257765.py:3: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy  
india_map.st_nm[0]='Andaman & Nicobar Islands'
```

```
C:\Users\karan\AppData\Local\Temp\ipykernel_30880\3782257765.py:4: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy  
india_map.st_nm[1]='Arunachal Pradesh'
```

```
C:\Users\karan\AppData\Local\Temp\ipykernel_30880\3782257765.py:5: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy  
india_map.st_nm[6]='Dadra and Nagar Haveli'
```

```
C:\Users\karan\AppData\Local\Temp\ipykernel_30880\3782257765.py:6: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy  
india_map.st_nm[23]='Delhi'
```

```
In [5]: india_map.st_nm
```

```
Out[5]: 0      Andaman & Nicobar Islands  
1          Arunachal Pradesh  
2            Assam  
3            Bihar  
4        Chandigarh  
5        Chhattisgarh  
6    Dadra and Nagar Haveli  
7        Daman & Diu  
8            Goa  
9        Gujarat  
10       Haryana  
11      Himachal Pradesh  
12      Jammu & Kashmir  
13       Jharkhand  
14      Karnataka  
15       Kerala  
16      Lakshadweep  
17      Madhya Pradesh  
18      Maharashtra  
19       Manipur  
20      Meghalaya  
21       Mizoram  
22       Nagaland  
23       Delhi  
24      Puducherry  
25       Punjab  
26      Rajasthan  
27       Sikkim  
28      Tamil Nadu  
29       Telangana  
30       Tripura  
31      Uttar Pradesh  
32      Uttarakhand  
33      West Bengal  
34       Odisha  
35      Andhra Pradesh  
Name: st_nm, dtype: object
```

Now importing the file with the population and other attributes of India and storing it in the variable 'population_data'

```
In [6]: population_data = pd.read_csv("C:/Users/karan/Downloads/archive (4)/Population of India.csv")
```

```
In [7]: population_data
```

Out[7] :

Sl No		State/UT	Population[50]	Percent (%)	Male	Female	Difference between male and female	Sex ratio	Rural[51]	Urban[51]	Area[52] (km2)	Density (per km2)
0	1	Uttar Pradesh	199812341	16.50	104480510	95331831	9148679	930	155111022	44470455	240928	828
1	2	Maharashtra	112374333	9.28	58243056	54131277	4111779	929	61545441	50827531	307713	365
2	3	Bihar	104099452	8.60	54278157	49821295	4456862	918	92075028	11729609	94163	1102
3	4	West Bengal	91276115	7.54	46809027	44467088	2341939	950	62213676	29134060	88752	1030
4	5	Madhya Pradesh	72626809	6.00	37612306	35014503	2597803	931	52537899	20059666	308245	236
5	6	Tamil Nadu	72147030	5.96	36137975	36009055	128920	996	37189229	34949729	130058	555
6	7	Rajasthan	68548437	5.66	35550997	32997440	2553557	928	51540236	17080776	342239	201
7	8	Karnataka	61095297	5.05	30966657	30128640	838017	973	37552529	23578175	191791	319
8	9	Gujarat	60439692	4.99	31491260	28948432	2542828	919	34670817	25712811	196024	308
9	10	Andhra Pradesh	49386799	4.08	24738068	24648731	89337	996	34776389	14610410	160205	308
10	11	Odisha	41974218	3.47	21212136	20762082	450054	979	34951234	6996124	155707	269
11	12	Telangana	35193978	2.91	17704078	17489900	214178	988	21585313	13608665	114840	307
12	13	Kerala	33406061	2.76	16027412	17378649	-1,351,237	1084	17445506	15932171	38863	859
13	14	Jharkhand	32988134	2.72	16930315	16057819	872496	948	25036946	7929292	79714	414
14	15	Assam	31205576	2.58	15939443	15266133	673310	958	26780526	4388756	78438	397
15	16	Punjab	27743338	2.29	14639465	13103873	1535592	895	17316800	10387436	50362	550
16	17	Chhattisgarh	25545198	2.11	12832895	12712303	120592	991	19603658	5936538	135191	189
17	18	Haryana	25351462	2.09	13494734	11856728	1638006	879	16531493	8821588	44212	573
18	19	Delhi (UT)	16787941	1.39	8887326	7800615	1086711	868	944727	12905780	1484	11297
19	20	Jammu and Kashmir	12541302	1.04	6640662	5900640	740022	889	9134820	3414106	222236	56
20	21	Uttarakhand	10086292	0.83	5137773	4948519	189254	963	7025583	3091169	53483	189
21	22	Himachal Pradesh	6864602	0.57	3481873	3382729	99144	972	6167805	688704	55673	123
22	23	Tripura	3673917	0.30	1874376	1799541	74835	960	2710051	960981	10486	350
23	24	Meghalaya	2966889	0.25	1491832	1475057	16775	989	2368971	595036	22429	132
24	25	Manipur	2855794	0.24	1438687	1417107	21580	985	1899624	822132	22327	128
25	26	Nagaland	1978502	0.16	1024649	953853	70796	931	1406861	573741	16579	119
26	27	Goa	1458545	0.12	739140	719405	19735	973	551414	906309	3702	394
27	28	Arunachal Pradesh	1383727	0.11	713912	669815	44097	938	1069165	313446	83743	17
28	29	Puducherry (UT)	1247953	0.10	612511	635442	-22,931	1037	394341	850123	479	2598
29	30	Mizoram	1097206	0.09	555339	541867	13472	976	529037	561997	21081	52
30	31	Chandigarh (UT)	1055450	0.09	580663	474787	105876	818	29004	1025682	114	9252
31	32	Sikkim	610577	0.05	323070	287507	35563	890	455962	151726	7096	86
32	33	Andaman and Nicobar Islands (UT)	380581	0.03	202871	177710	25161	876	244411	135533	8249	46
33	34	Dadra and Nagar Haveli (UT)	343709	0.03	193760	149949	43811	774	183024	159829	491	698
34	35	Daman and Diu (UT)	243247	0.02	150301	92946	57355	618	60331	182580	112	2169
35	36	Lakshadweep (UT)	64473	0.01	33123	31350	1773	946	14121	50308	32	2013
36	-	Total (India)	1210854977	100.00	623724248	586469174	35585741	943	833087662	377105760	3287240	382

On the first look, the column names have unnecessary brackets or formats. Hence, I renamed them for better standardization

In [8]: population_data.rename(columns = {'State/UT': 'State', 'Population[50]': 'Population', 'Rural[51]': 'Rural', 'Urban[51]': 'Urban'})

In [9]: population_data

Out[9]:

	Sl No	State	Population	Percent (%)	Male	Female	Difference between male and female	Sex ratio	Rural	Urban	Area(km2)	Density (per km2)
0	1	Uttar Pradesh	199812341	16.50	104480510	95331831	9148679	930	155111022	44470455	240928	828
1	2	Maharashtra	112374333	9.28	58243056	54131277	4111779	929	61545441	50827531	307713	365
2	3	Bihar	104099452	8.60	54278157	49821295	4456862	918	92075028	11729609	94163	1102
3	4	West Bengal	91276115	7.54	46809027	44467088	2341939	950	62213676	29134060	88752	1030
4	5	Madhya Pradesh	72626809	6.00	37612306	35014503	2597803	931	52537899	20059666	308245	236
5	6	Tamil Nadu	72147030	5.96	36137975	36009055	128920	996	37189229	34949729	130058	555
6	7	Rajasthan	68548437	5.66	35550997	32997440	2553557	928	51540236	17080776	342239	201
7	8	Karnataka	61095297	5.05	30966657	30128640	838017	973	37552529	23578175	191791	319
8	9	Gujarat	60439692	4.99	31491260	28948432	2542828	919	34670817	25712811	196024	308
9	10	Andhra Pradesh	49386799	4.08	24738068	24648731	89337	996	34776389	14610410	160205	308
10	11	Odisha	41974218	3.47	21212136	20762082	450054	979	34951234	6996124	155707	269
11	12	Telangana	35193978	2.91	17704078	17489900	214178	988	21585313	13608665	114840	307
12	13	Kerala	33406061	2.76	16027412	17378649	-1,351,237	1084	17445506	15932171	38863	859
13	14	Jharkhand	32988134	2.72	16930315	16057819	872496	948	25036946	7929292	79714	414
14	15	Assam	31205576	2.58	15939443	15266133	673310	958	26780526	4388756	78438	397
15	16	Punjab	27743338	2.29	14639465	13103873	1535592	895	17316800	10387436	50362	550
16	17	Chhattisgarh	25545198	2.11	12832895	12712303	120592	991	19603658	5936538	135191	189
17	18	Haryana	25351462	2.09	13494734	11856728	1638006	879	16531493	8821588	44212	573
18	19	Delhi (UT)	16787941	1.39	8887326	7800615	1086711	868	944727	12905780	1484	11297
19	20	Jammu and Kashmir	12541302	1.04	6640662	5900640	740022	889	9134820	3414106	222236	56
20	21	Uttarakhand	10086292	0.83	5137773	4948519	189254	963	7025583	3091169	53483	189
21	22	Himachal Pradesh	6864602	0.57	3481873	3382729	99144	972	6167805	688704	55673	123
22	23	Tripura	3673917	0.30	1874376	1799541	74835	960	2710051	960981	10486	350
23	24	Meghalaya	2966889	0.25	1491832	1475057	16775	989	2368971	595036	22429	132
24	25	Manipur	2855794	0.24	1438687	1417107	21580	985	1899624	822132	22327	128
25	26	Nagaland	1978502	0.16	1024649	953853	70796	931	1406861	573741	16579	119
26	27	Goa	1458545	0.12	739140	719405	19735	973	551414	906309	3702	394
27	28	Arunachal Pradesh	1383727	0.11	713912	669815	44097	938	1069165	313446	83743	17
28	29	Puducherry (UT)	1247953	0.10	612511	635442	-22,931	1037	394341	850123	479	2598
29	30	Mizoram	1097206	0.09	555339	541867	13472	976	529037	561997	21081	52
30	31	Chandigarh (UT)	1055450	0.09	580663	474787	105876	818	29004	1025682	114	9252
31	32	Sikkim	610577	0.05	323070	287507	35563	890	455962	151726	7096	86
32	33	Andaman and Nicobar Islands (UT)	380581	0.03	202871	177710	25161	876	244411	135533	8249	46
33	34	Dadra and Nagar Haveli (UT)	343709	0.03	193760	149949	43811	774	183024	159829	491	698
34	35	Daman and Diu (UT)	243247	0.02	150301	92946	57355	618	60331	182580	112	2169
35	36	Lakshadweep (UT)	64473	0.01	33123	31350	1773	946	14121	50308	32	2013
36	-	Total (India)	1210854977	100.00	623724248	586469174	35585741	943	833087662	377105760	3287240	382

Also, the union territories have a '(UT)' placed after their name. Eg:'Lakshadweep (UT)' I want to get rid of this

```
In [10]: population_data['State'] = population_data['State'].str.replace(r'\(\w+\)', ' ', regex=True).str.strip()
```

```
In [11]: population_data['State']
```

```
Out[11]: 0          Uttar Pradesh
1          Maharashtra
2          Bihar
3          West Bengal
4          Madhya Pradesh
5          Tamil Nadu
6          Rajasthan
7          Karnataka
8          Gujarat
9          Andhra Pradesh
10         Odisha
11         Telangana
12         Kerala
13         Jharkhand
14         Assam
15         Punjab
16         Chhattisgarh
17         Haryana
18         Delhi
19         Jammu and Kashmir
20         Uttarakhand
21         Himachal Pradesh
22         Tripura
23         Meghalaya
24         Manipur
25         Nagaland
26         Goa
27         Arunachal Pradesh
28         Puducherry
29         Mizoram
30         Chandigarh
31         Sikkim
32         Andaman and Nicobar Islands
33         Dadra and Nagar Haveli
34         Daman and Diu
35         Lakshadweep
36         Total
```

Name: State, dtype: object

After performing further data analysis, I realized that the first dataset used '&' inside the state's full names and the second dataset used 'and'. This will lead to merging issues because the merge function won't recognize them as one. Therefore, I decided to replace the 'and' with '&' in the state names.

```
In [12]: population_data['State'] = population_data['State'].apply(lambda x: x.replace(' and ', ' & '))
```

I chose to replace ' and ' with spaces instead of 'and' because many of the states (eg. Uttarakhand) have 'and' in their name spelling. I don't want to replace that with the '&'. I only want to replace the 'and' with '&' in the names of the states where it is being used in spaces (eg. Jammu and Kashmir)

```
In [13]: population_data['State']
```

```
Out[13]: 0          Uttar Pradesh
1          Maharashtra
2          Bihar
3          West Bengal
4          Madhya Pradesh
5          Tamil Nadu
6          Rajasthan
7          Karnataka
8          Gujarat
9          Andhra Pradesh
10         Odisha
11         Telangana
12         Kerala
13         Jharkhand
14         Assam
15         Punjab
16         Chhattisgarh
17         Haryana
18         Delhi
19         Jammu & Kashmir
20         Uttarakhand
21         Himachal Pradesh
22         Tripura
23         Meghalaya
24         Manipur
25         Nagaland
26         Goa
27         Arunachal Pradesh
28         Puducherry
29         Mizoram
30         Chandigarh
31         Sikkim
32    Andaman & Nicobar Islands
33    Dadra & Nagar Haveli
34    Daman & Diu
35    Lakshadweep
36        Total
Name: State, dtype: object
```

```
In [14]: population_data['Difference between male and female'] = population_data['Difference between male and female'].r
```

```
In [15]: population_data['Difference between male and female']
```

```
Out[15]: 0      9148679
1      4111779
2      4456862
3      2341939
4      2597803
5      128920
6      2553557
7      838017
8      2542828
9      89337
10     450054
11     214178
12     1351237
13     872496
14     673310
15     1535592
16     120592
17     1638006
18     1086711
19     740022
20     189254
21     99144
22     74835
23     16775
24     21580
25     70796
26     19735
27     44097
28     22931
29     13472
30     105876
31     35563
32     25161
33     43811
34     57355
35     1773
36    35585741
Name: Difference between male and female, dtype: object
```

```
In [16]: population_data.dtypes
```

```

Out[16]: Sl No          object
State           object
Population      int64
Percent (%)    float64
Male            int64
Female          int64
Difference between male and female  object
Sex ratio       int64
Rural           int64
Urban            int64
Area(km2)       int64
Density (per km2) int64
dtype: object

```

```
In [17]: population_data
```

	Sl No	State	Population	Percent (%)	Male	Female	Difference between male and female	Sex ratio	Rural	Urban	Area(km2)	Density (per km2)
0	1	Uttar Pradesh	199812341	16.50	104480510	95331831	9148679	930	155111022	44470455	240928	828
1	2	Maharashtra	112374333	9.28	58243056	54131277	4111779	929	61545441	50827531	307713	365
2	3	Bihar	104099452	8.60	54278157	49821295	4456862	918	92075028	11729609	94163	1102
3	4	West Bengal	91276115	7.54	46809027	44467088	2341939	950	62213676	29134060	88752	1030
4	5	Madhya Pradesh	72626809	6.00	37612306	35014503	2597803	931	52537899	20059666	308245	236
5	6	Tamil Nadu	72147030	5.96	36137975	36009055	128920	996	37189229	34949729	130058	555
6	7	Rajasthan	68548437	5.66	35550997	32997440	2553557	928	51540236	17080776	342239	201
7	8	Karnataka	61095297	5.05	30966657	30128640	838017	973	37552529	23578175	191791	319
8	9	Gujarat	60439692	4.99	31491260	28948432	2542828	919	34670817	25712811	196024	308
9	10	Andhra Pradesh	49386799	4.08	24738068	24648731	89337	996	34776389	14610410	160205	308
10	11	Odisha	41974218	3.47	21212136	20762082	450054	979	34951234	6996124	155707	269
11	12	Telangana	35193978	2.91	17704078	17489900	214178	988	21585313	13608665	114840	307
12	13	Kerala	33406061	2.76	16027412	17378649	1351237	1084	17445506	15932171	38863	859
13	14	Jharkhand	32988134	2.72	16930315	16057819	872496	948	25036946	7929292	79714	414
14	15	Assam	31205576	2.58	15939443	15266133	673310	958	26780526	4388756	78438	397
15	16	Punjab	27743338	2.29	14639465	13103873	1535592	895	17316800	10387436	50362	550
16	17	Chhattisgarh	25545198	2.11	12832895	12712303	120592	991	19603658	5936538	135191	189
17	18	Haryana	25351462	2.09	13494734	11856728	1638006	879	16531493	8821588	44212	573
18	19	Delhi	16787941	1.39	8887326	7800615	1086711	868	944727	12905780	1484	11297
19	20	Jammu & Kashmir	12541302	1.04	6640662	5900640	740022	889	9134820	3414106	222236	56
20	21	Uttarakhand	10086292	0.83	5137773	4948519	189254	963	7025583	3091169	53483	189
21	22	Himachal Pradesh	6864602	0.57	3481873	3382729	99144	972	6167805	688704	55673	123
22	23	Tripura	3673917	0.30	1874376	1799541	74835	960	2710051	960981	10486	350
23	24	Meghalaya	2966889	0.25	1491832	1475057	16775	989	2368971	595036	22429	132
24	25	Manipur	2855794	0.24	1438687	1417107	21580	985	1899624	822132	22327	128
25	26	Nagaland	1978502	0.16	1024649	953853	70796	931	1406861	573741	16579	119
26	27	Goa	1458545	0.12	739140	719405	19735	973	551414	906309	3702	394
27	28	Arunachal Pradesh	1383727	0.11	713912	669815	44097	938	1069165	313446	83743	17
28	29	Puducherry	1247953	0.10	612511	635442	22931	1037	394341	850123	479	2598
29	30	Mizoram	1097206	0.09	555339	541867	13472	976	529037	561997	21081	52
30	31	Chandigarh	1055450	0.09	580663	474787	105876	818	29004	1025682	114	9252
31	32	Sikkim	610577	0.05	323070	287507	35563	890	455962	151726	7096	86
32	33	Andaman & Nicobar Islands	380581	0.03	202871	177710	25161	876	244411	135533	8249	46
33	34	Dadra & Nagar Haveli	343709	0.03	193760	149949	43811	774	183024	159829	491	698
34	35	Daman & Diu	243247	0.02	150301	92946	57355	618	60331	182580	112	2169
35	36	Lakshadweep	64473	0.01	33123	31350	1773	946	14121	50308	32	2013
36	-	Total	1210854977	100.00	623724248	586469174	35585741	943	833087662	377105760	3287240	382

```
In [18]: merged_data = india_map.merge(population_data, how='inner', left_on='st_nm', right_on='State')
```

```
In [19]: merged_data
```

Out[19]:

	st_nm	geometry	SI No	State	Population	Percent (%)	Male	Female	Difference between male and female	Sex ratio	Rural	Urban	Area
0	Andaman & Nicobar Islands	MULTIPOLYGON (((93.71976 7.20707, 93.71909 7.2...))	33	Andaman & Nicobar Islands	380581	0.03	202871	177710	25161	876	244411	135533	
1	Arunachal Pradesh	POLYGON (((96.16261 29.38078, 96.16860 29.37432...))	28	Arunachal Pradesh	1383727	0.11	713912	669815	44097	938	1069165	313446	
2	Assam	MULTIPOLYGON (((89.74323 26.30362, 89.74290 26...))	15	Assam	31205576	2.58	15939443	15266133	673310	958	26780526	4388756	
3	Bihar	MULTIPOLYGON (((84.50720 24.26323, 84.50355 24...))	3	Bihar	104099452	8.60	54278157	49821295	4456862	918	92075028	11729609	
4	Chandigarh	POLYGON (((76.84147 30.75996, 76.83599 30.73623...))	31	Chandigarh	1055450	0.09	580663	474787	105876	818	29004	1025682	
5	Chhattisgarh	POLYGON (((83.33532 24.09885, 83.35346 24.09627...))	17	Chhattisgarh	25545198	2.11	12832895	12712303	120592	991	19603658	5936538	1
6	Daman & Diu	MULTIPOLYGON (((72.89335 20.44539, 72.89281 20...))	35	Daman & Diu	243247	0.02	150301	92946	57355	618	60331	182580	
7	Goa	MULTIPOLYGON (((74.11918 14.75344, 74.11350 14...))	27	Goa	1458545	0.12	739140	719405	19735	973	551414	906309	
8	Gujarat	MULTIPOLYGON (((71.70375 20.99958, 71.70375 20...))	9	Gujarat	60439692	4.99	31491260	28948432	2542828	919	34670817	25712811	1
9	Haryana	POLYGON (((76.85065 30.87512, 76.86594 30.86691...))	18	Haryana	25351462	2.09	13494734	11856728	1638006	879	16531493	8821588	
10	Himachal Pradesh	POLYGON (((76.79634 33.25490, 76.80351 33.25275...))	22	Himachal Pradesh	6864602	0.57	3481873	3382729	99144	972	6167805	688704	
11	Jammu & Kashmir	POLYGON (((74.73451 37.02068, 74.73647 37.01937...))	20	Jammu & Kashmir	12541302	1.04	6640662	5900640	740022	889	9134820	3414106	2
12	Jharkhand	POLYGON (((87.60582 25.31512, 87.61279 25.31184...))	14	Jharkhand	32988134	2.72	16930315	16057819	872496	948	25036946	7929292	
13	Karnataka	MULTIPOLYGON (((74.69694 13.32782, 74.69562 13...))	8	Karnataka	61095297	5.05	30966657	30128640	838017	973	37552529	23578175	1
14	Kerala	POLYGON (((74.99575 12.79227, 75.00006 12.78777...))	13	Kerala	33406061	2.76	16027412	17378649	1351237	1084	17445506	15932171	
15	Lakshadweep	MULTIPOLYGON (((74.10131 11.20431, 74.09908 11...))	36	Lakshadweep	64473	0.01	33123	31350	1773	946	14121	50308	
16	Madhya Pradesh	MULTIPOLYGON (((74.17932 22.39059, 74.17776 22...))	5	Madhya Pradesh	72626809	6.00	37612306	35014503	2597803	931	52537899	20059666	3
17	Maharashtra	MULTIPOLYGON (((73.46270 16.03710, 73.46178 16...))	2	Maharashtra	112374333	9.28	58243056	54131277	4111779	929	61545441	50827531	3

18	Manipur	POLYGON ((94.57602 25.64221, 94.57487 25.63858...)	25	Manipur	2855794	0.24	1438687	1417107	21580	985	1899624	822132
19	Meghalaya	POLYGON ((91.85632 26.10353, 91.86717 26.09906...)	24	Meghalaya	2966889	0.25	1491832	1475057	16775	989	2368971	595036
20	Mizoram	POLYGON ((92.80022 24.41630, 92.80310 24.41603...)	30	Mizoram	1097206	0.09	555339	541867	13472	976	529037	561997
21	Nagaland	POLYGON ((95.21458 26.93095, 95.21719 26.92818...)	26	Nagaland	1978502	0.16	1024649	953853	70796	931	1406861	573741
22	Delhi	POLYGON ((77.09361 28.86940, 77.10973 28.86722...)	19	Delhi	16787941	1.39	8887326	7800615	1086711	868	944727	12905780
23	Puducherry	MULTIPOLYGON (((79.76528 10.99681, 79.76963 10...)	29	Puducherry	1247953	0.10	612511	635442	22931	1037	394341	850123
24	Punjab	POLYGON ((75.88469 32.48841, 75.90303 32.47169...)	16	Punjab	27743338	2.29	14639465	13103873	1535592	895	17316800	10387436
25	Rajasthan	POLYGON ((73.89849 29.97896, 73.90021 29.97140...)	7	Rajasthan	68548437	5.66	35550997	32997440	2553557	928	51540236	17080776
26	Sikkim	POLYGON ((88.65381 28.09883, 88.66265 28.08956...)	32	Sikkim	610577	0.05	323070	287507	35563	890	455962	151726
27	Tamil Nadu	MULTIPOLYGON (((78.19188 8.72730, 78.18996 8.7...)	6	Tamil Nadu	72147030	5.96	36137975	36009055	128920	996	37189229	34949729
28	Telangana	POLYGON ((81.05614 17.79097, 81.05834 17.75464...)	12	Telangana	35193978	2.91	17704078	17489900	214178	988	21585313	13608665
29	Tripura	POLYGON ((92.22108 24.50131, 92.22771 24.49862...)	23	Tripura	3673917	0.30	1874376	1799541	74835	960	2710051	960981
30	Uttar Pradesh	MULTIPOLYGON (((80.44802 24.99631, 80.44080 24...)	1	Uttar Pradesh	199812341	16.50	104480510	95331831	9148679	930	155111022	44470455
31	Uttarakhand	POLYGON ((79.21047 31.34846, 79.21386 31.34680...)	21	Uttarakhand	10086292	0.83	5137773	4948519	189254	963	7025583	3091169
32	West Bengal	MULTIPOLYGON (((88.01861 21.57278, 88.01889 21...)	4	West Bengal	91276115	7.54	46809027	44467088	2341939	950	62213676	29134060
33	Odisha	MULTIPOLYGON (((86.38937 19.96351, 86.38840 19...)	11	Odisha	41974218	3.47	21212136	20762082	450054	979	34951234	6996124
34	Andhra Pradesh	MULTIPOLYGON (((81.10380 17.82269, 81.10610 17...)	10	Andhra Pradesh	49386799	4.08	24738068	24648731	89337	996	34776389	14610410

In [20]: merged_data.columns

```
Out[20]: Index(['st_nm', 'geometry', 'Sl No', 'State', 'Population', 'Percent (%)',  
                 'Male', 'Female', 'Difference between male and female', 'Sex ratio',  
                 'Rural', 'Urban', 'Area(km2)', 'Density (per km2)'),  
                dtype='object')
```

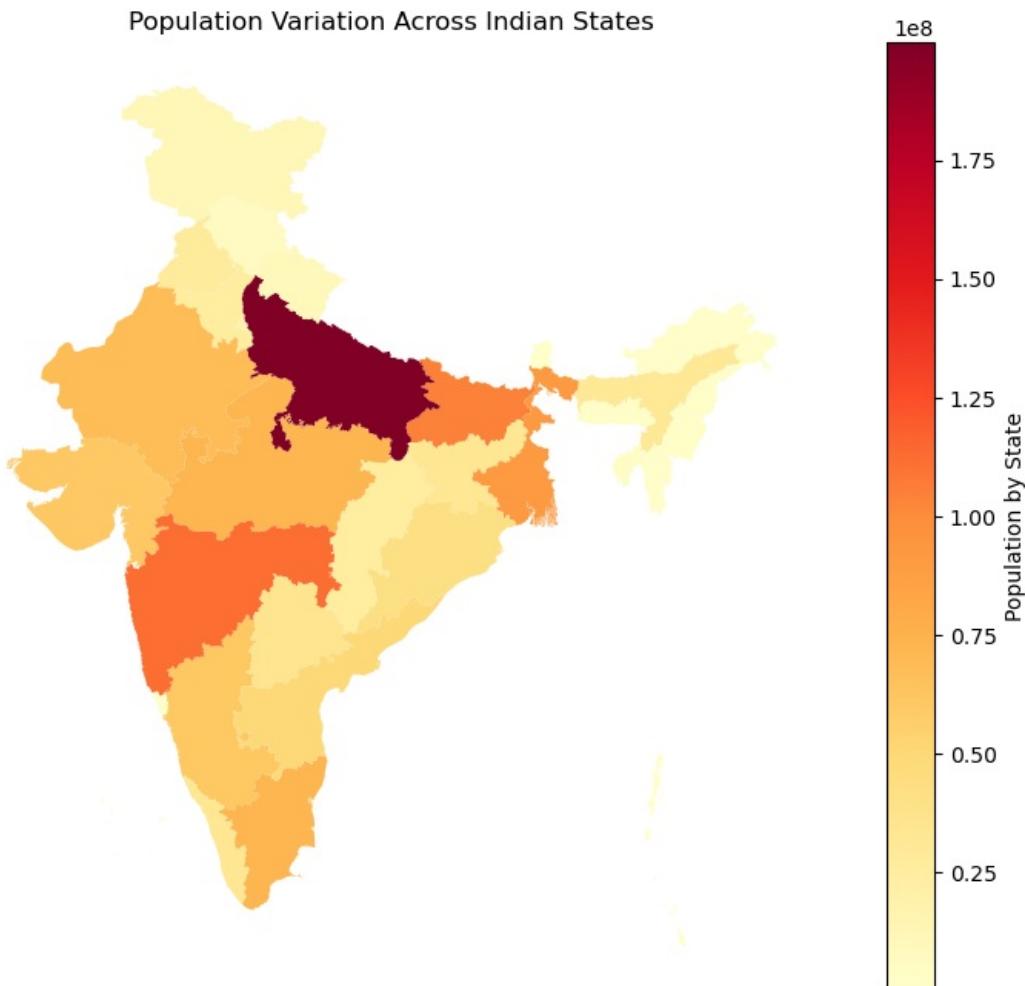
```
In [21]: merged_data= merged_data.drop(columns=['Sl No'])
```

```
In [22]: merged_data.columns
```

```
Out[22]: Index(['st_nm', 'geometry', 'State', 'Population', 'Percent (%)', 'Male',  
                 'Female', 'Difference between male and female', 'Sex ratio', 'Rural',  
                 'Urban', 'Area(km2)', 'Density (per km2)'),  
                dtype='object')
```

We finally have a variable 'merged_data' with all the information required to start plotting geographical visualizations. Firstly, I want to plot a choropleth map showcasing the total population of each state

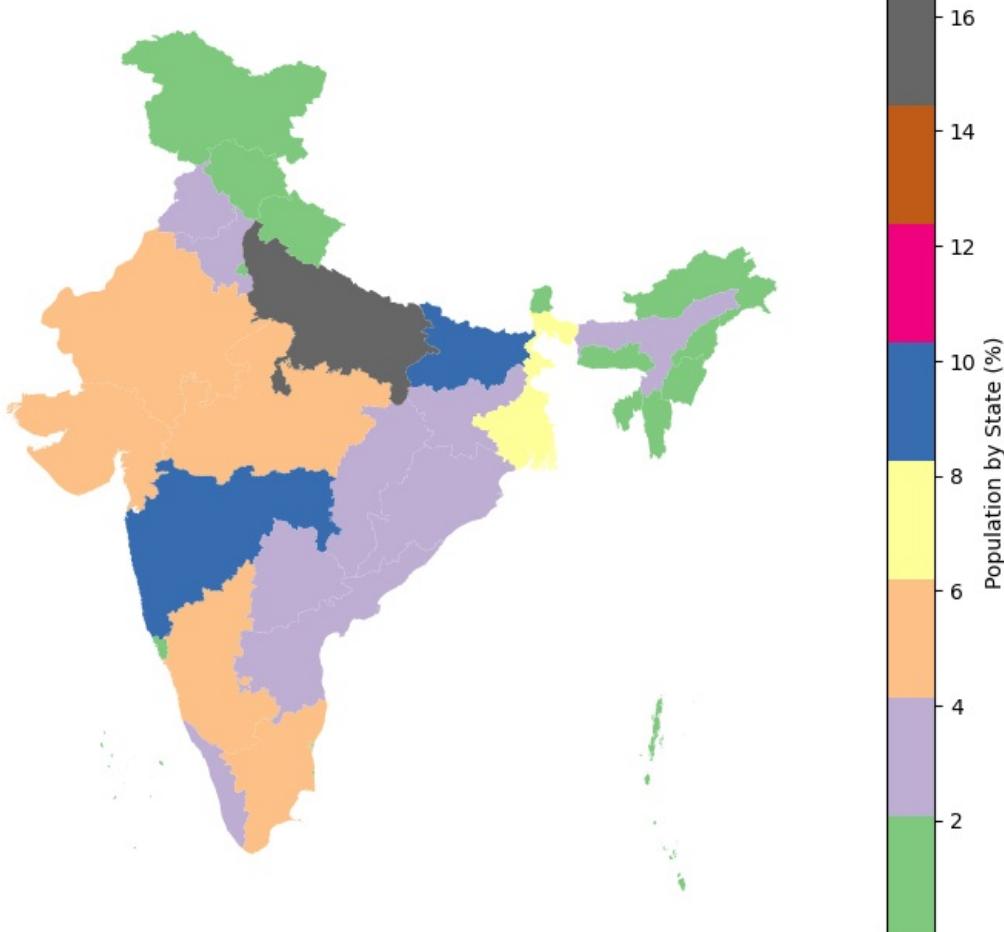
```
In [23]: import matplotlib.pyplot as plt  
  
# Plotting the choropleth map  
fig, ax = plt.subplots(1, 1, figsize=(12, 8))  
  
# Plot the choropleth map showing population variation  
merged_data.plot(column='Population', cmap='YlOrRd', ax=ax, legend=True,  
                  legend_kwds={'label': "Population by State", 'orientation': "vertical"})  
  
plt.title("Population Variation Across Indian States")  
leg = ax.get_legend()  
ax.set_axis_off()  
plt.show()
```



```
In [24]: fig, ax = plt.subplots(1, 1, figsize=(12, 8))
```

```
# Plot the choropleth map showing population variation  
merged_data.plot(column='Percent (%)', cmap='Accent', ax=ax, legend=True,  
                  legend_kwds={'label': "Population by State (%)", 'orientation': "vertical"})  
  
plt.title("Population Percentage Variation Across Indian States")  
leg = ax.get_legend()  
ax.set_axis_off()  
plt.show()
```

Population Percentage Variation Across Indian States



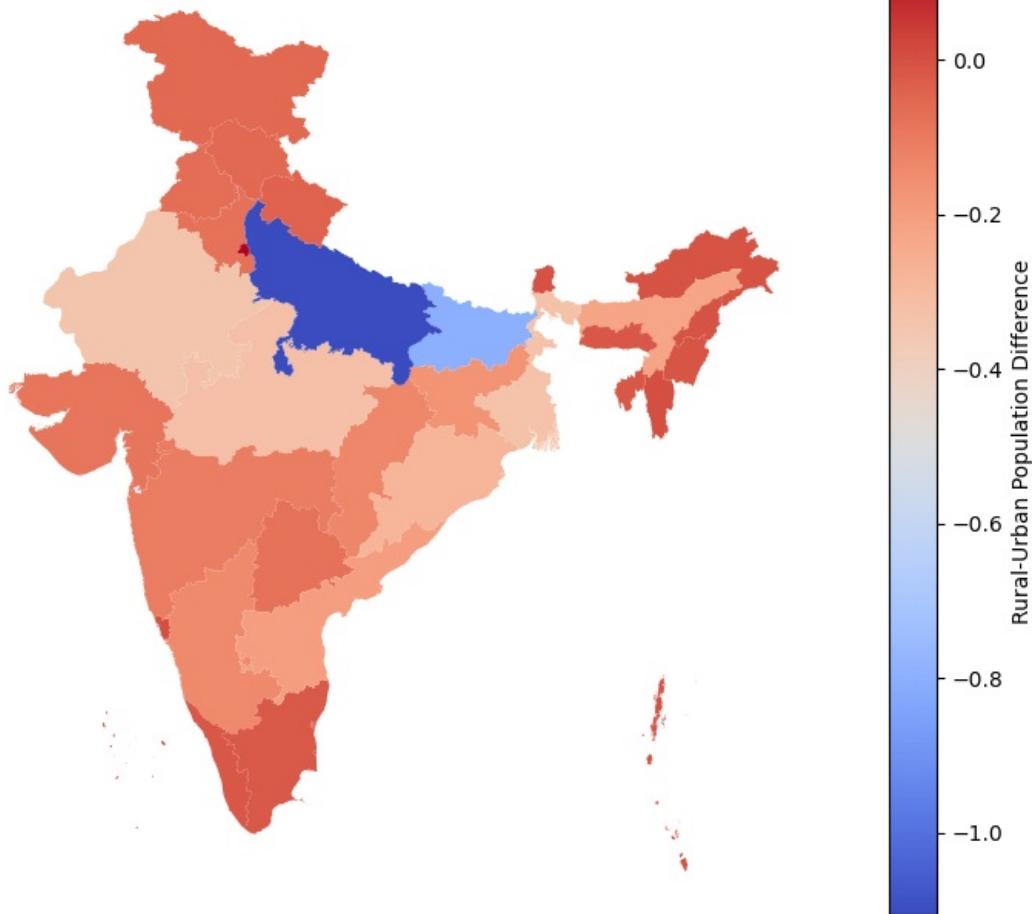
```
In [25]: merged_data['Rural_Urban_Difference'] = merged_data['Urban'] - merged_data['Rural']

# Plotting the choropleth map illustrating the distribution of rural and urban populations
fig, ax = plt.subplots(1, 1, figsize=(12, 8))

# Plot the choropleth map showing the difference between rural and urban populations
merged_data.plot(column='Rural_Urban_Difference', cmap='coolwarm', ax=ax, legend=True,
                  legend_kwds={'label': "Rural-Urban Population Difference", 'orientation': "vertical"})

ax.set_title("Distribution of Rural and Urban Populations Across States")
ax.set_axis_off()
plt.show()
```

Distribution of Rural and Urban Populations Across States



Now I want to generate the same map, except with the sex ratio of each state

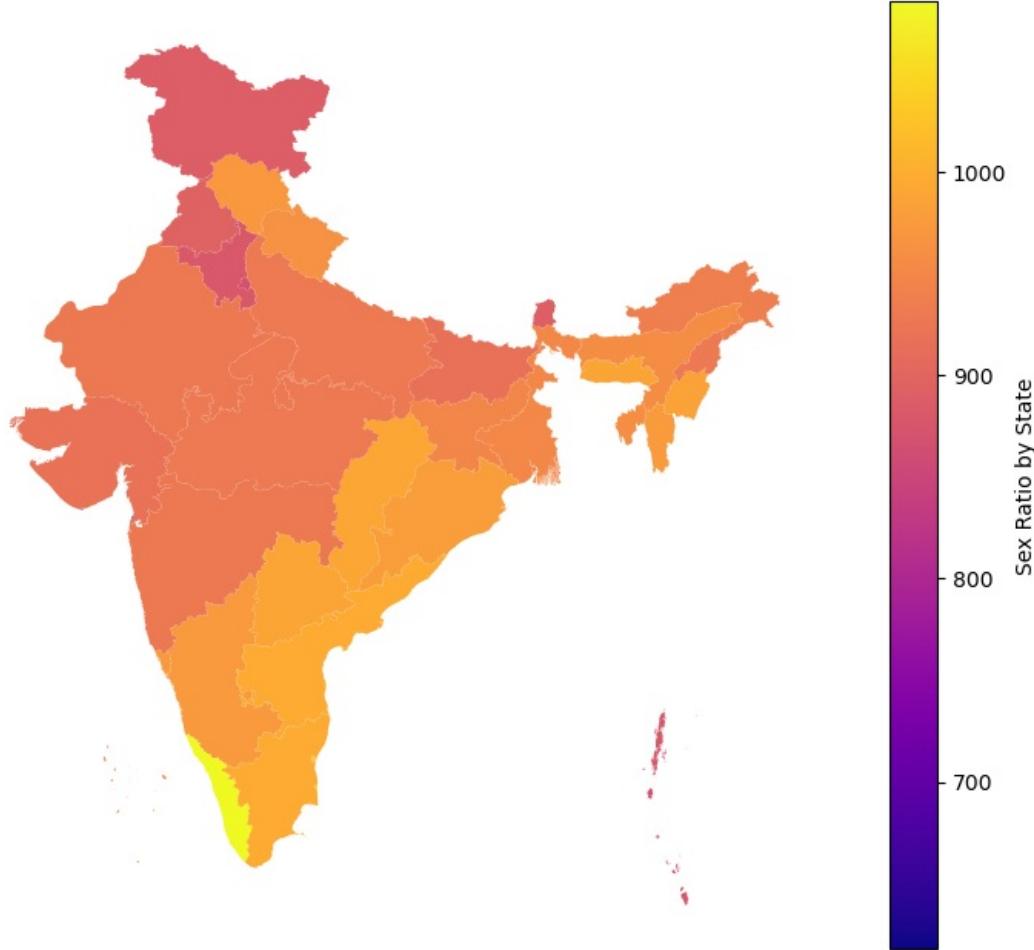
```
In [26]: import matplotlib.pyplot as plt

# Plotting the choropleth map
fig, ax = plt.subplots(1, 1, figsize=(12, 8))

# Plot the choropleth map showing population variation
merged_data.plot(column='Sex ratio', cmap='plasma', ax=ax, legend=True,
                  legend_kwds={'label': "Sex Ratio by State", 'orientation': "vertical"})

plt.title("Sex Ratio Across Indian States")
color_continuous_scale='Viridis_r'
leg = ax.get_legend()
ax.set_axis_off()
plt.show()
```

Sex Ratio Across Indian States



Lastly, a map showcasing the density of each state per kilometer squared.

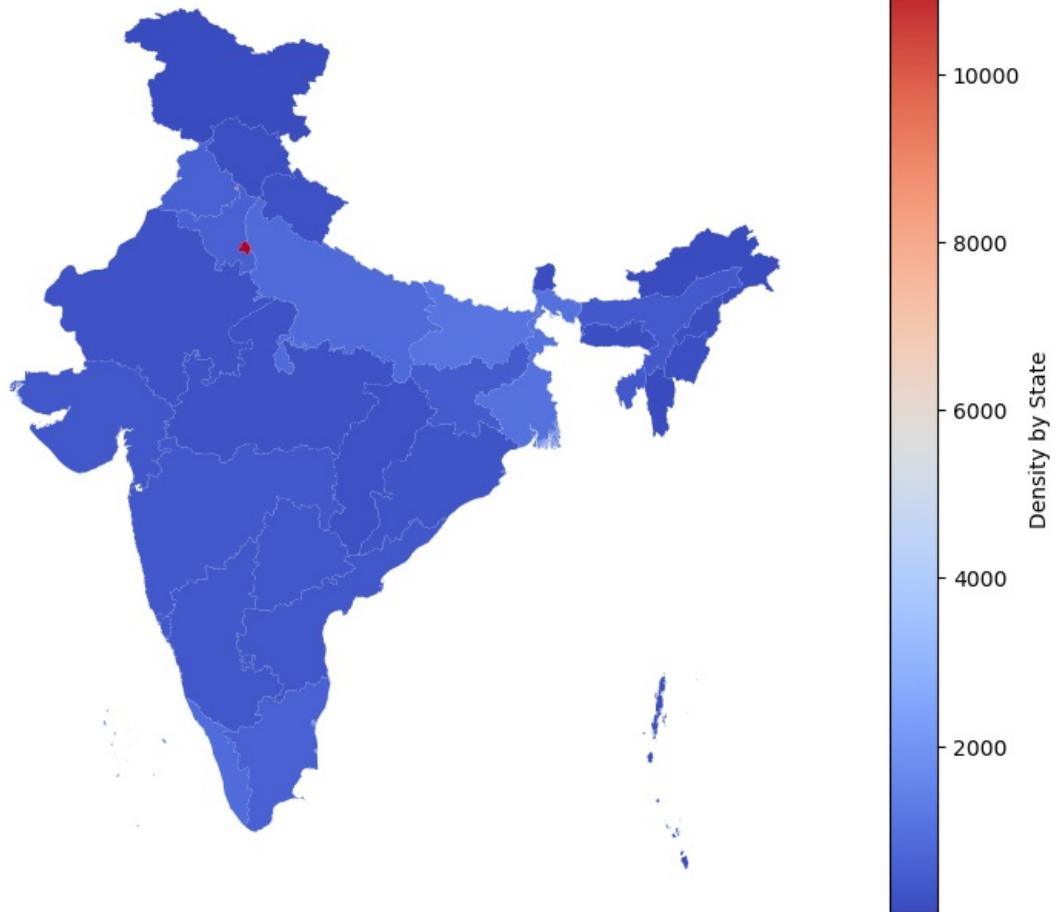
```
In [27]: fig, ax = plt.subplots(1, 1, figsize=(12, 8))

color_scale_range = [merged_data['Density (per km2)').min(), merged_data['Density (per km2)').max()]

# Plot the choropleth map showing population variation with adjusted color scale range
merged_data.plot(column='Density (per km2)', cmap='coolwarm', ax=ax, legend=True,
                  legend_kwds={'label': "Density by State", 'orientation': "vertical"},
                  vmin=color_scale_range[0], vmax=color_scale_range[1])

plt.title("Population Density Across Indian States")
leg = ax.get_legend()
ax.set_axis_off()
plt.show()
```

Population Density Across Indian States



Map showcasing the male, female and total population of India's choropleth map side to side

```
In [28]: fig, axes = plt.subplots(1, 3, figsize=(18, 6)) # Adjust figsize as needed for better visualization

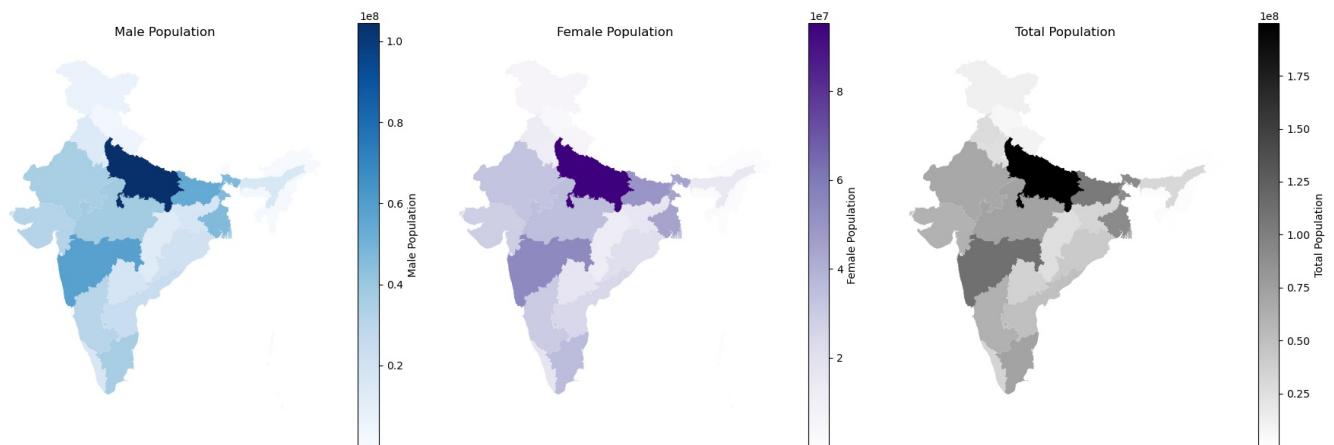
# Parameters to compare: Population, Male population, Female population
parameters = ['Male', 'Female', 'Population']
titles = ['Male Population', 'Female Population', 'Total Population']
cmaps = ['Blues', 'Purples', 'binary'] # Colormaps for better differentiation

for i, param in enumerate(parameters):
    ax = axes[i]

    # Plot choropleth map for each parameter
    merged_data.plot(column=param, cmap=cmaps[i], legend=True,
                      legend_kwds={'label': titles[i], 'orientation': "vertical"}, ax=ax)

    ax.set_title(titles[i])
    ax.set_axis_off()

plt.tight_layout()
plt.show()
```



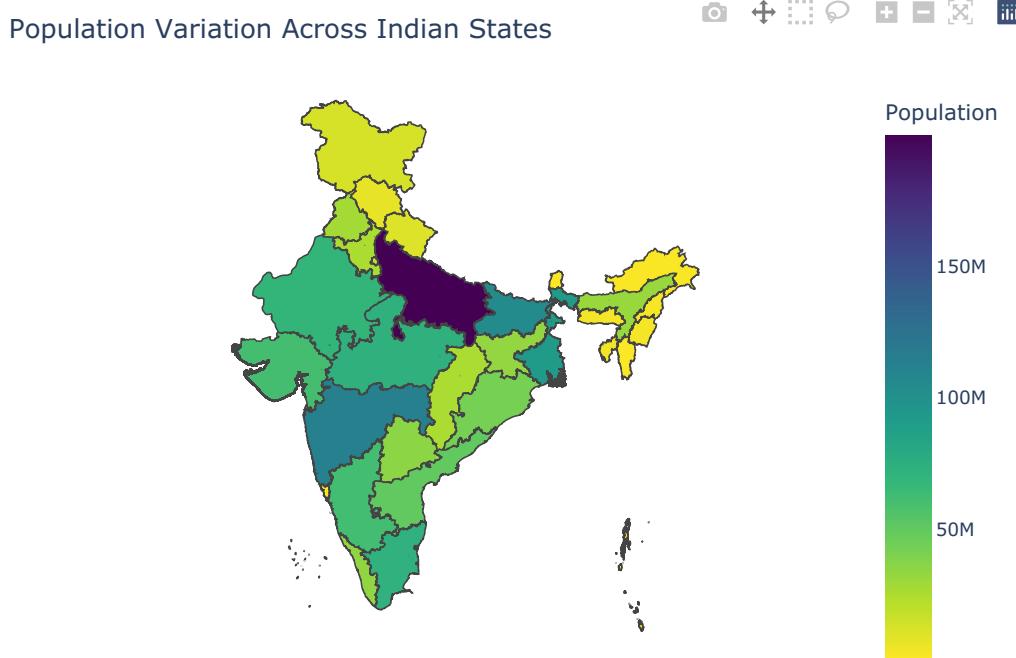
The maps generated above give a great visualization of the different attributes relating to each state of India. However, I want to create a more interactive map. I am proceeding to create a choropleth map where we can hover our mouse over the map to get specific data.

Firstly, I have plotted a geographic map which shows the population and Area of the state when you hover your mouse over it

```
In [29]: import plotly.express as px

# Plotting the interactive choropleth map with hover information
fig = px.choropleth(
    merged_data,
    geojson=merged_data.geometry,
    locations=merged_data.index,
    color='Population',
    projection="mercator",
    hover_name='st_nm', # State name on hover
    hover_data={'Population': True, 'Area(km2)': True},
    color_continuous_scale='Viridis_r', # Additional data on hover
)

fig.update_geos(fitbounds="locations", visible=False)
fig.update_layout(title='Population Variation Across Indian States')
fig.show()
```



Now, I have generated an interactive map with the Sex Ratio of each state

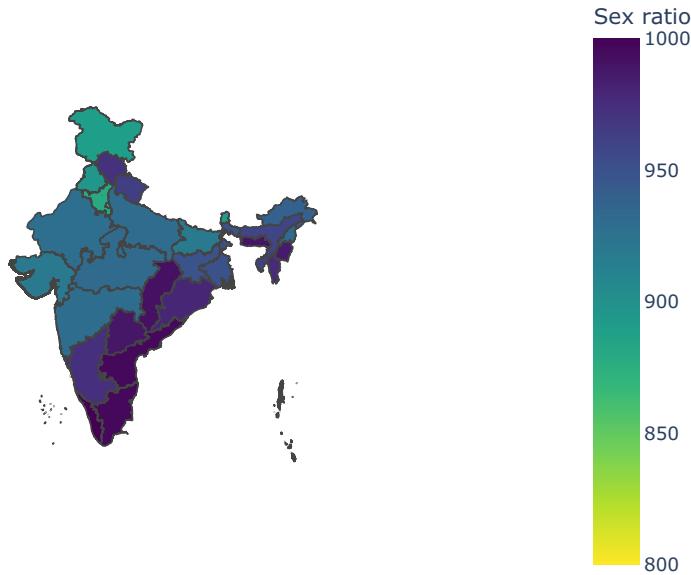
```
In [30]: # Define the range for the color scale (modify this based on your data range)
color_scale_range = [800, 1000] # Example range for sex ratio values

# Plotting the interactive choropleth map with hover information
fig = px.choropleth(
    merged_data,
    geojson=merged_data.geometry,
    locations=merged_data.index,
    color='Sex ratio',
    projection="sinusoidal",
    hover_name='st_nm', # State name on hover
    hover_data={'Male': True, 'Female': True, 'Difference between male and female': True, 'Sex ratio': True},
    color_continuous_scale='Viridis_r',
    range_color=color_scale_range,# Set the custom range for the color scale
)

fig.update_geos(fitbounds="locations", visible=False)
fig.update_layout(title='Sex Ratio Variation each State')
fig.show()
```

Sex Ratio Variation each State

camera full-screen zoom-in zoom-out refresh



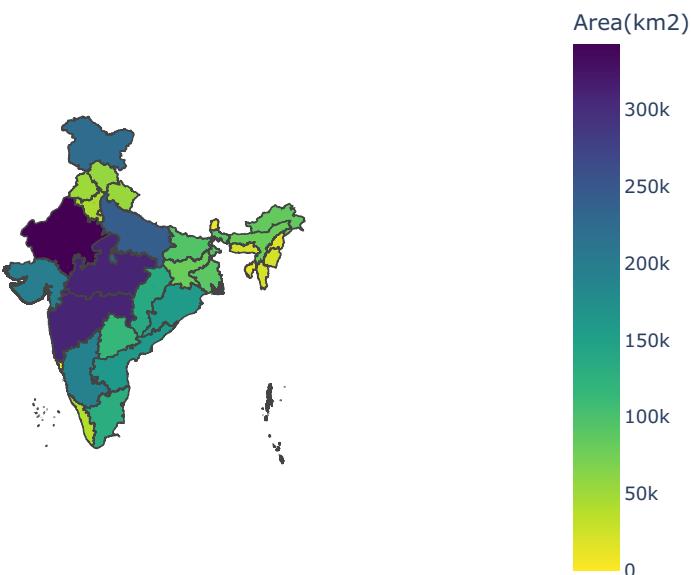
Plotting area of each state over the interactive map

In [31]: *# Plotting the interactive choropleth map with hover information*

```
fig = px.choropleth(  
    merged_data,  
    geojson=merged_data.geometry,  
    locations=merged_data.index,  
    color='Area(km2)',  
    projection="sinusoidal",  
    hover_name='State', # State name on hover  
    hover_data={'Area(km2)', 'Density (per km2)'},  
    color_continuous_scale='Viridis_r',# Set the custom range for the color scale  
)  
  
fig.update_geos(fitbounds="locations", visible=False)  
fig.update_layout(title='Total Area of each State Across Indian States')  
fig.show()
```

camera full-screen zoom-in zoom-out refresh

Total Area of each State Across Indian States



Generating map of population density of each map

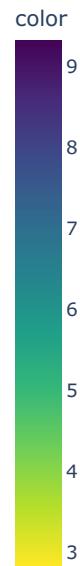
```
In [33]: import numpy as np

# Define a logarithmic color scale range
color_scale_range_log = [np.log(merged_data['Density (per km2)'].min()), np.log(merged_data['Density (per km2)'].max())]

# Plotting with logarithmic color scale
fig = px.choropleth(
    merged_data,
    geojson=merged_data.geometry,
    locations=merged_data.index,
    color=np.log(merged_data['Density (per km2)']),
    projection="sinusoidal",
    hover_name='State',
    hover_data={'Area(km2)', 'Density (per km2)'},
    color_continuous_scale='Viridis_r',
    range_color=color_scale_range_log,
    color_continuous_midpoint=np.mean(color_scale_range_log),
)

# Update geos and layout, then display
fig.update_geos(fitbounds="locations", visible=False)
fig.update_layout(title='Population Density variation across India')
fig.show()
```

Population Density variation across India



In []: