In [68]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
!pip install geopandas
import geopandas as gpd
import datetime
```

```
Requirement already satisfied: geopandas in c:\users\my hp\anaconda3\lib\site-pack
ages (0.12.2)
Requirement already satisfied: shapely>=1.7 in c:\users\my hp\anaconda3\lib\site-p
ackages (from geopandas) (2.0.1)
Requirement already satisfied: pyproj>=2.6.1.post1 in c:\users\my hp\anaconda3\lib
\site-packages (from geopandas) (3.5.0)
Requirement already satisfied: packaging in c:\users\my hp\anaconda3\lib\site-pack
ages (from geopandas) (21.3)
Requirement already satisfied: fiona>=1.8 in c:\users\my hp\anaconda3\lib\site-pac
kages (from geopandas) (1.9.3)
Requirement already satisfied: pandas>=1.0.0 in c:\users\my hp\anaconda3\lib\site-
packages (from geopandas) (1.4.4)
Requirement already satisfied: importlib-metadata in c:\users\my hp\anaconda3\lib
\site-packages (from fiona>=1.8->geopandas) (4.11.3)
Requirement already satisfied: munch>=2.3.2 in c:\users\my hp\anaconda3\lib\site-p
ackages (from fiona>=1.8->geopandas) (2.5.0)
Requirement already satisfied: attrs>=19.2.0 in c:\users\my hp\anaconda3\lib\site-
packages (from fiona>=1.8->geopandas) (21.4.0)
Requirement already satisfied: click~=8.0 in c:\users\my hp\anaconda3\lib\site-pac
kages (from fiona>=1.8->geopandas) (8.0.4)
Requirement already satisfied: certifi in c:\users\my hp\anaconda3\lib\site-packag
es (from fiona>=1.8->geopandas) (2022.9.14)
Requirement already satisfied: click-plugins>=1.0 in c:\users\my hp\anaconda3\lib
\site-packages (from fiona>=1.8->geopandas) (1.1.1)
Requirement already satisfied: cligj>=0.5 in c:\users\my hp\anaconda3\lib\site-pac
kages (from fiona>=1.8->geopandas) (0.7.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\my hp\anaconda3\lib\site-p
ackages (from pandas>=1.0.0->geopandas) (2022.1)
Requirement already satisfied: numpy>=1.18.5 in c:\users\my hp\anaconda3\lib\site-
packages (from pandas>=1.0.0->geopandas) (1.24.2)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\my hp\anaconda3
\lib\site-packages (from pandas>=1.0.0->geopandas) (2.8.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\my hp\anaconda
3\lib\site-packages (from packaging->geopandas) (3.0.9)
Requirement already satisfied: colorama in c:\users\my hp\anaconda3\lib\site-packa
ges (from click~=8.0->fiona>=1.8->geopandas) (0.4.6)
Requirement already satisfied: six in c:\users\my hp\anaconda3\lib\site-packages
(from munch>=2.3.2->fiona>=1.8->geopandas) (1.16.0)
Requirement already satisfied: zipp>=0.5 in c:\users\my hp\anaconda3\lib\site-pack
ages (from importlib-metadata->fiona>=1.8->geopandas) (3.8.0)
```

In [193…

```
df = pd.read_csv("Installed_Power_Capacity_State_wise_MW.csv")
df1 = pd.read_csv("Power_Generation.csv")
df2 = pd.read_csv("gdp.csv")
shp_gdf = gpd.read_file("India_State_Boundary.shp")
df3 = pd.read_excel("scapped_data_1.xlsx")
```

In [70]:

```
df.head(2)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[70]:

	id	month	region	state	coal	gas	diesel	thermal_total	nuclear	hydro	res
0	1	Jan-2019	Northern	Chandigarh	0.0	0.0	0.0	0.0	0.0	0.0	32.40
1	2	Jan-2019	Northern	Delhi	135.0	2208.4	0.0	2343.4	0.0	0.0	176.21

In [71]:

```
df1.head(2)
```

Out[71]:

	index	Date	Region	Thermal Generation Actual (in MU)	Thermal Generation Estimated (in MU)	Nuclear Generation Actual (in MU)	Nuclear Generation Estimated (in MU)	Hydro Generation Actual (in MU)	Hyd Generati Estimat (in M
0	0	2017-09-01	Northern	624.23	484.21	30.36	35.57	273.27	320.
1	1	2017-09-01	Western	1,106.89	1,024.33	25.17	3.81	72.00	21.

In [72]:

```
year_2019 = df.iloc[397:432,:]
year_2020 = df.iloc[840:876,:]
year_2021 = df.iloc[1282:1317,:]
```

In [73]:

```
total_grand1 = year_2019["grand_total"].sum()
total_grand2 = year_2020["grand_total"].sum()
total_grand3 = year_2021['grand_total'].sum()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [74]:

```
total_thermal = year_2019["thermal_total"].sum()
total_nuclear = year_2019['nuclear'].sum()
total_hydro = year_2019['hydro'].sum()
total_renewable = year_2019['res'].sum()

total_thermal2 = year_2020["thermal_total"].sum()
total_nuclear2 = year_2020['nuclear'].sum()
total_hydro2 = year_2020['hydro'].sum()
total_renewable2 = year_2020['res'].sum()

total_thermal3 = year_2021["thermal_total"].sum()
total_nuclear3 = year_2021['nuclear'].sum()
total_hydro3 = year_2021['hydro'].sum()
total_renewable3 = year_2021['res'].sum()

total_coal = year_2019["coal"].sum()
total_gas = year_2019['gas'].sum()
total_diesel = year_2019['diesel'].sum()

total_coal1 = year_2020["coal"].sum()
total_gas1 = year_2020['gas'].sum()
total_diesel1 = year_2020['diesel'].sum()

total_coal2 = year_2021["coal"].sum()
total_gas2 = year_2021['gas'].sum()
total_diesel2 = year_2021['diesel'].sum()

total_grand1 = year_2019["grand_total"].sum()
total_grand2 = year_2020["grand_total"].sum()
total_grand3 = year_2021['grand_total'].sum()
```

In [75]:

```
lst_tmp = {"COAL" : [total_coal,total_coal1,total_coal2]
            ,"NUCLEAR":[total_nuclear,total_nuclear2,total_nuclear3]
           ,"GAS":[total_gas,total_gas1,total_gas2]
           ,'DIESEL':[total_diesel,total_diesel1,total_diesel2]
            ,"HYDRO" :[total_hydro,total_hydro2,total_hydro3]
           ,"RENEWABLE":[total_renewable,total_renewable2,total_renewable3]
           ,"YEAR":[2019,2020,2021]}
```

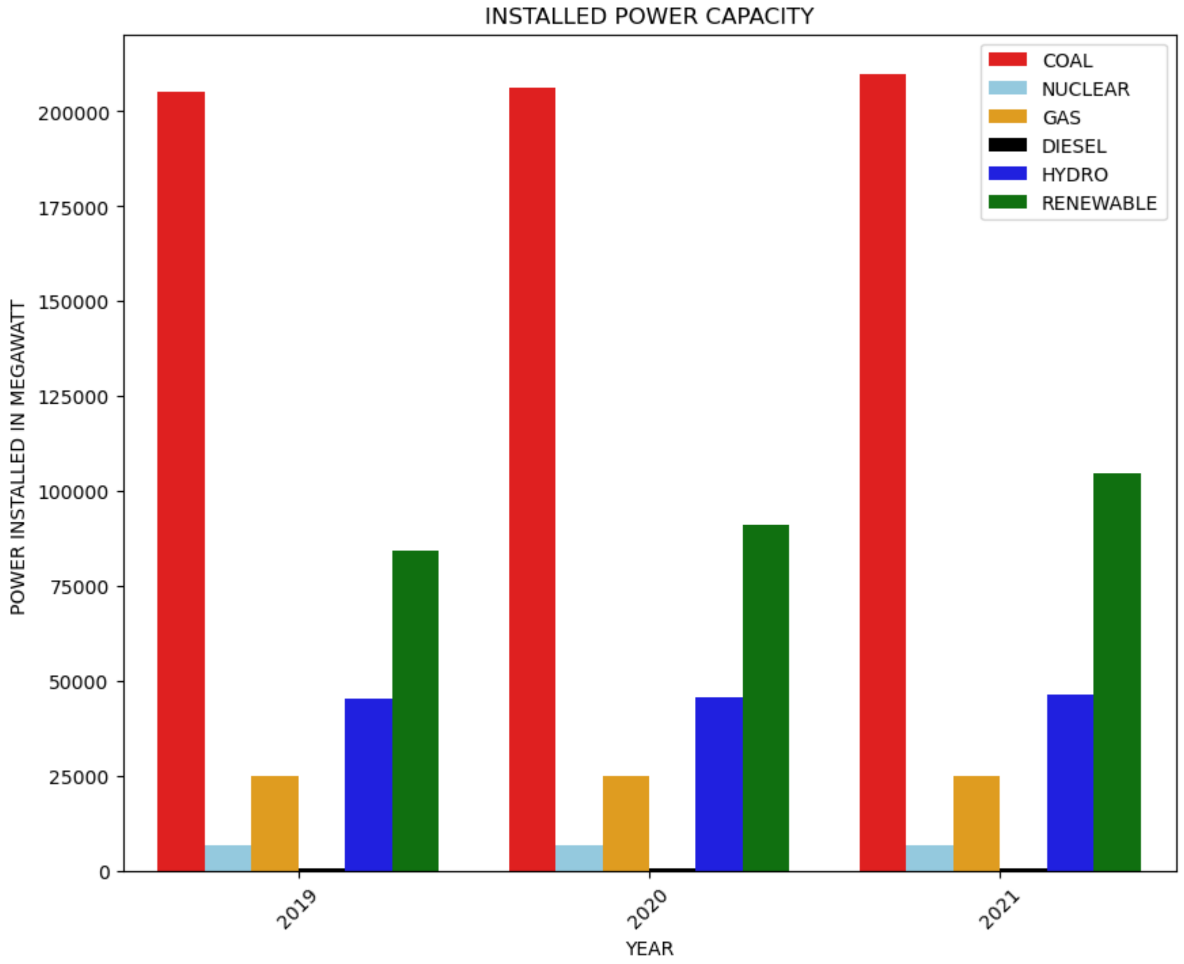
In [76]:

```
df_mod = pd.DataFrame(lst_tmp)
df_melted = pd.melt(df_mod,id_vars='YEAR',var_name='variable',value_name='value')
```

In [77]:

```
plt.figure(figsize=(10,8))
colors = {"COAL":"red","NUCLEAR": "skyblue", "GAS": "orange", "DIESEL": "black","H

sns.barplot(x='YEAR', y='value', hue='variable', data=df_melted,palette=colors)
plt.xticks(rotation=45)
plt.title("INSTALLED POWER CAPACITY")
plt.xlabel("YEAR")
plt.ylabel("POWER INSTALLED IN MEGAWATT ")
plt.legend()
plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [78]:
```
def percentage_change(new,old):
    return (((new - old)/old)*100)
```

In [79]:
```
c1 = percentage_change(total_coal1,total_coal)
c2 = percentage_change(total_coal2,total_coal1)
n1 = percentage_change(total_nuclear2,total_nuclear)
n2 = percentage_change(total_nuclear3,total_nuclear2)
g1 = percentage_change(total_gas1,total_gas)
g2 = percentage_change(total_gas2,total_gas1)
d1 = percentage_change(total_diesel1,total_diesel)
d2 = percentage_change(total_diesel2,total_diesel1)
r1 = percentage_change(total_renewable2,total_renewable)
r2 = percentage_change(total_renewable3,total_renewable2)
h1 = percentage_change(total_hydro2,total_hydro)
h2 = percentage_change(total_hydro3,total_hydro2)
# print(per)
```

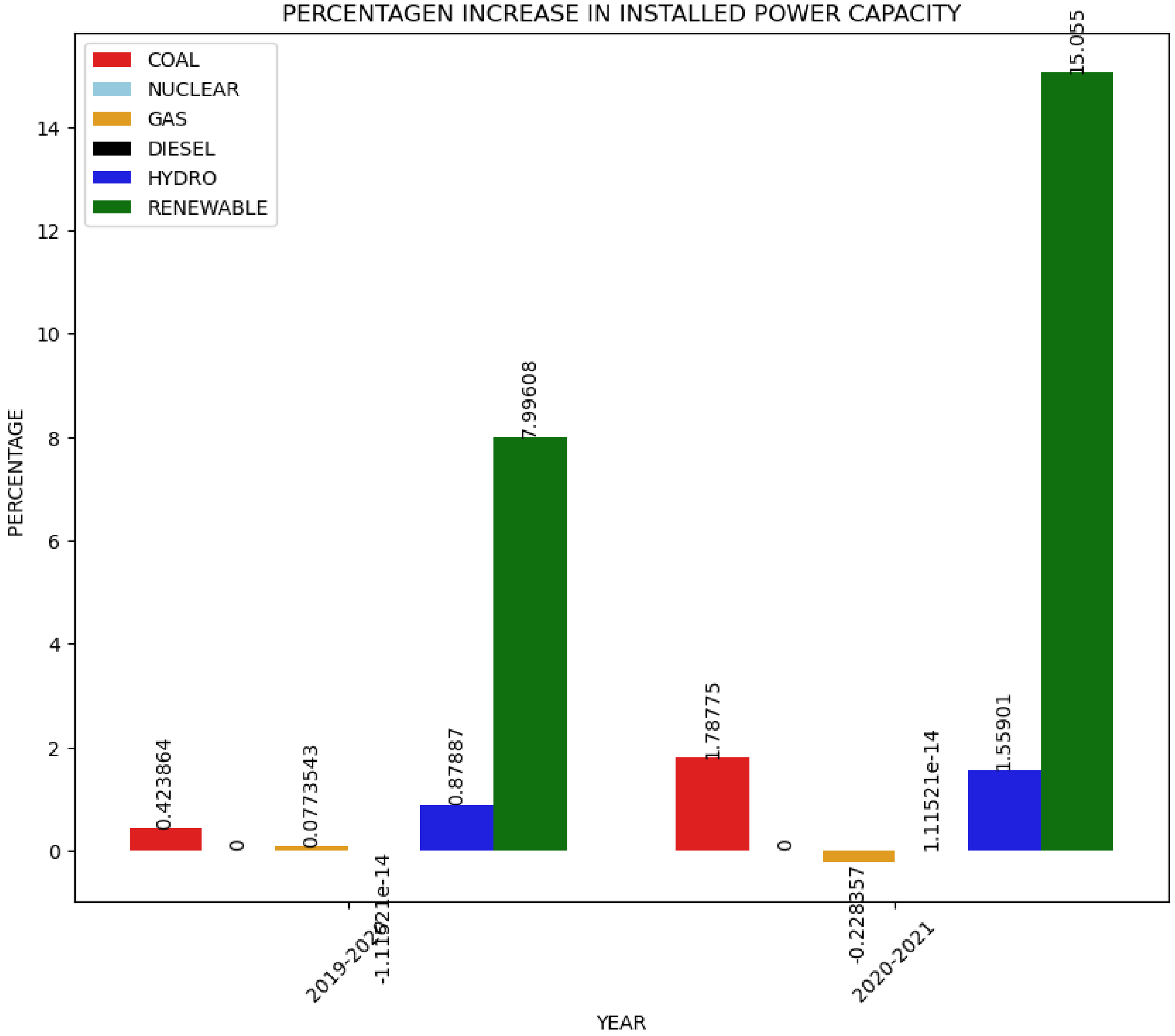
In [80]:
```
lst_tmp2 = {"COAL" : [c1,c2]
           ,"NUCLEAR":[n1,n2]
          ,"GAS":[g1,g2]
          ,'DIESEL':[d1,d2]
           ,"HYDRO" :[h1,h2]
          ,"RENEWABLE":[r1,r2]
          ,"YEAR":["2019-2020","2020-2021"]}
```

In [81]:
```
df_mod2 = pd.DataFrame(lst_tmp2)
```

In [82]:
```
df_melted2 = pd.melt(df_mod2,id_vars='YEAR',var_name='variable',value_name='value'
```

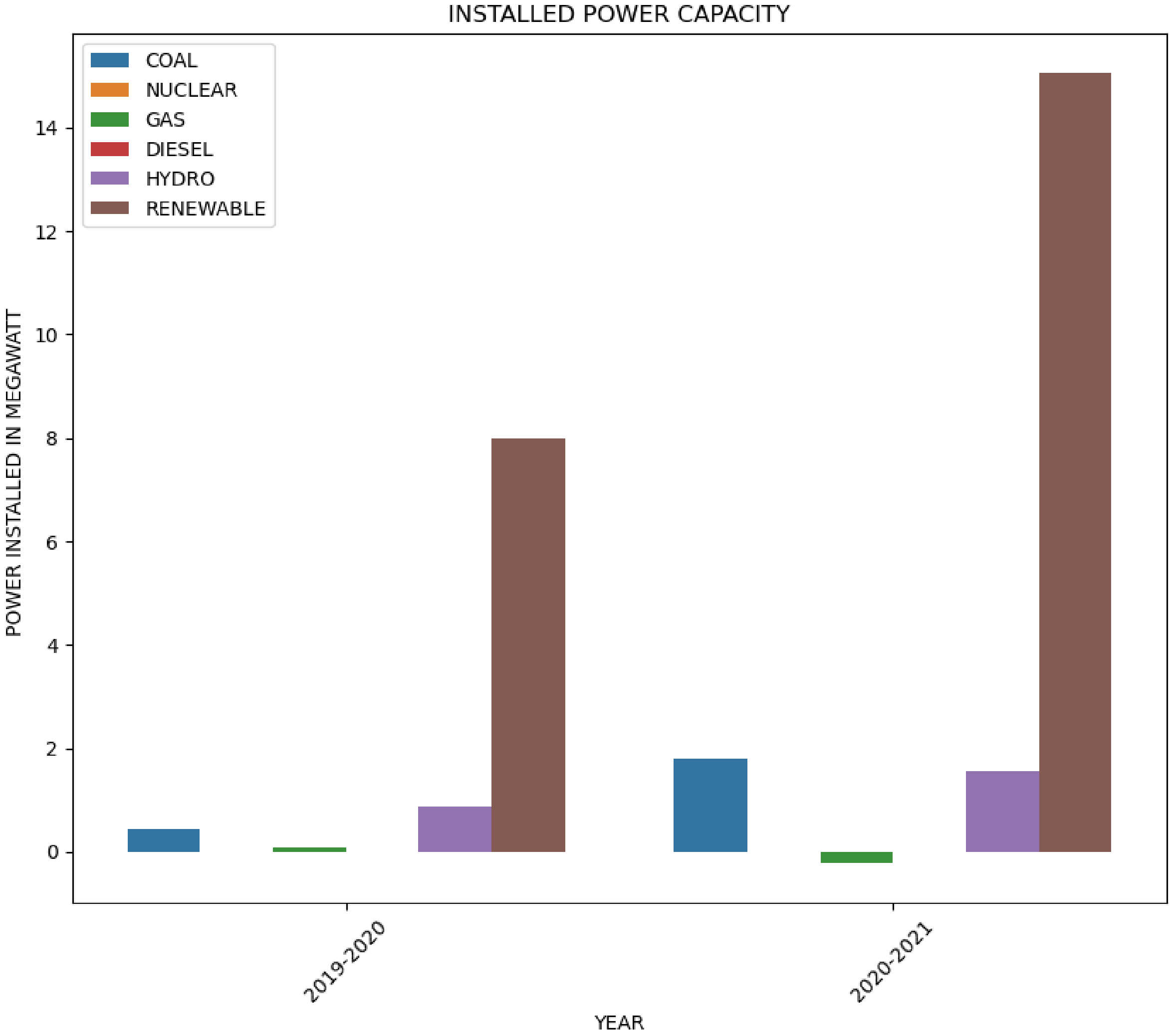
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
# define custom colors for each variable
colors = {"COAL":"red","NUCLEAR": "skyblue", "GAS": "orange", "DIESEL": "black","H

ax = sns.barplot(x='YEAR', y='value', hue='variable', data=df_melted2, palette=col

# to add the values at the top of the bar
for p in ax.containers:
    ax.bar_label(p, label_type="edge",fontsize = 10,rotation=90)
plt.xticks(rotation=45)
plt.title("PERCENTAGEN INCREASE IN INSTALLED POWER CAPACITY")
plt.xlabel("YEAR")
plt.ylabel("PERCENTAGE ")
plt.legend()
plt.show()
```



In [84]:

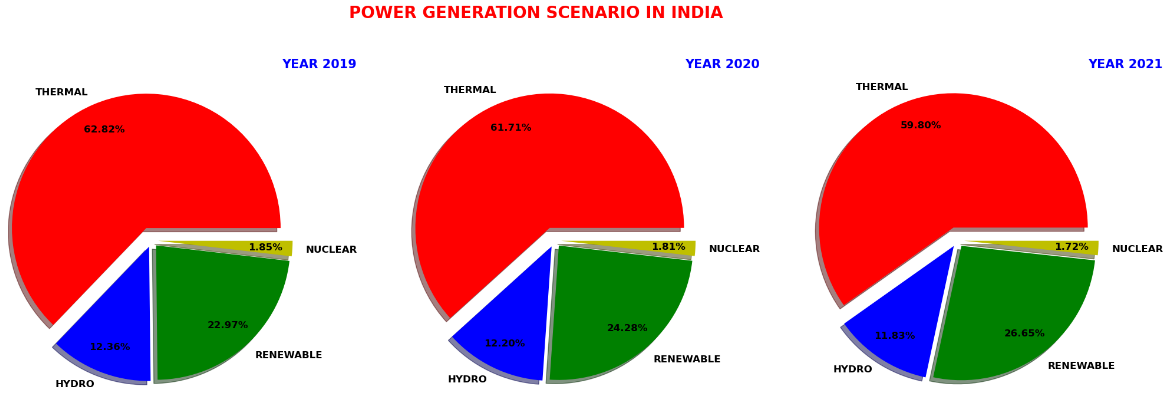
```
c = ["red","yellow",'voilet','black',"blue","green"]
plt.figure(figsize=(10,8))
sns.barplot(x='YEAR', y='value', hue='variable', data=df_melted2)
plt.xticks(rotation=45)
plt.title("INSTALLED POWER CAPACITY")
plt.xlabel("YEAR")
plt.ylabel("POWER INSTALLED IN MEGAWATT ")
plt.legend()
plt.show()
```



```
In [85]: x = [total_thermal,total_hydro,total_renewable,total_nuclear]
         y = ['THERMAL',"HYDRO","RENEWABLE","NUCLEAR"]
         ex = [0.1,0.05,0.05,0.05]
         c = ["r","b","g","y"]
         x2 = [total_thermal2,total_hydro2,total_renewable2,total_nuclear2]
         x3 = [total_thermal3,total_hydro3,total_renewable3,total_nuclear3]
```

```
In [86]: plt.figure(figsize=(25,20))
         plt.subplot(1,3,1)
         plt.pie(x,labels=y,explode=ex,autopct="%0.2f%%",shadow=True,radius=1,labeldistance
                 ,startangle=0,textprops={"fontsize":12,"fontweight":'bold'},pctdistance=0.
         plt.title("YEAR 2019",x=1,y=1,color='blue',fontsize=15,fontweight='bold')
         # plt.show()
         plt.subplot(1,3,2)
         plt.pie(x2,labels=y,explode=ex,autopct="%0.2f%%",shadow=True,radius=1,labeldistanc
                 ,startangle=0,textprops={"fontsize":12,"fontweight":'bold'},pctdistance=0.
         plt.title("YEAR 2020",x=1,y=1,color='blue',fontsize=15,fontweight='bold')
         plt.subplot(1,3,3)
         plt.pie(x3,labels=y,explode=ex,autopct="%0.2f%%",shadow=True,radius=1,labeldistanc
                 ,startangle=0,textprops={"fontsize":12,"fontweight":'bold'},pctdistance=0.
         plt.title("YEAR 2021",x=1,y=1,color='blue',fontsize=15,fontweight='bold')
         plt.suptitle("POWER GENERATION SCENARIO IN INDIA",fontsize=20,color='red',fontweig
         plt.subplots_adjust(top=1.45)
         plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [87]:

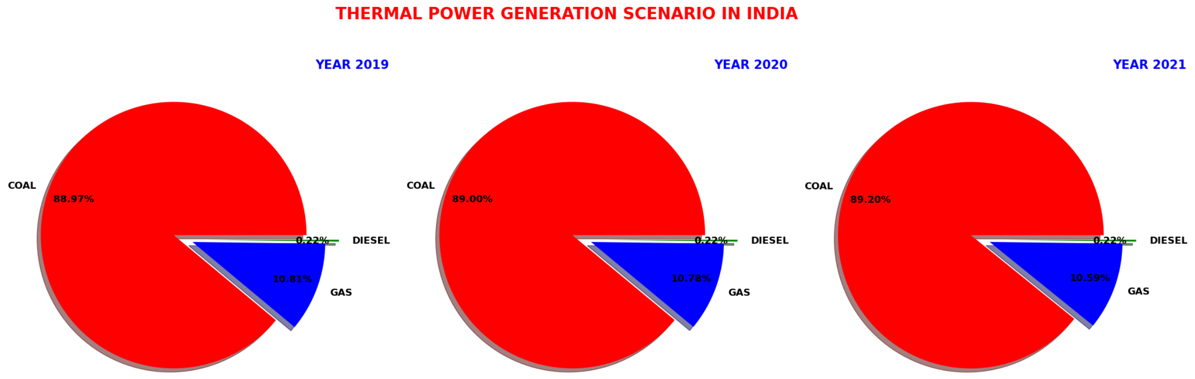
```
total_coal = year_2019["coal"].sum()
total_gas = year_2019['gas'].sum()
total_diesel = year_2019['diesel'].sum()

total_coal1 = year_2020["coal"].sum()
total_gas1 = year_2020['gas'].sum()
total_diesel1 = year_2020['diesel'].sum()

total_coal2 = year_2021["coal"].sum()
total_gas2 = year_2021['gas'].sum()
total_diesel2 = year_2021['diesel'].sum()
```

In [88]:

```
x_1 = [total_coal,total_gas,total_diesel]
x_2 = [total_coal1,total_gas1,total_diesel1]
x_3 = [total_coal2,total_gas2,total_diesel2]
y_1 = ["COAL","GAS","DIESEL"]
ex_1 = [0.1,0.05,0.15]
c_1 = ["r","b","g"]
```

In [89]:

```
plt.figure(figsize=(25,20))
plt.subplot(1,3,1)
plt.pie(x_1,labels=y_1,explode=ex_1,autopct="%0.2f%%",shadow=True,radius=1,labeldi
        ,startangle=0,textprops={"fontsize":12,"fontweight":'bold'},pctdistance=0.
plt.title("YEAR 2019",x=1,y=1,color='blue',fontsize=15,fontweight='bold')
# plt.show()
plt.subplot(1,3,2)
plt.pie(x_2,labels=y_1,explode=ex_1,autopct="%0.2f%%",shadow=True,radius=1,labeldi
        ,startangle=0,textprops={"fontsize":12,"fontweight":'bold'},pctdistance=0.
plt.title("YEAR 2020",x=1,y=1,color='blue',fontsize=15,fontweight='bold')
plt.subplot(1,3,3)
plt.pie(x_3,labels=y_1,explode=ex_1,autopct="%0.2f%%",shadow=True,radius=1,labeldi
        ,startangle=0,textprops={"fontsize":12,"fontweight":'bold'},pctdistance=0.
plt.title("YEAR 2021",x=1,y=1,color='blue',fontsize=15,fontweight='bold')
plt.suptitle("THERMAL POWER GENERATION SCENARIO IN INDIA",fontsize=20,color='red',
plt.subplots_adjust(top=1.45)
plt.show()
```



In [90]:
```
tmp = df[['state' , 'grand_total']]
hydro = df[['state',"hydro"]]
renewable = df[['state','res']]
nuclear = df[['state','nuclear']]
coal = df[['state','coal']]
```

In [91]:
```
tmp1 = tmp.iloc[396:432,:]
tmp2 = tmp.iloc[839:876,:]
tmp3 = tmp.iloc[1281:1317,:]
```

In [92]:
```
shp_gdf.drop(columns='Type',inplace=True)
shp_gdf.iloc[1,0] = 'Andaman & Nicobar Islands'
shp_gdf.iloc[14,0]= 'Chhatisgarh'
shp_gdf.iloc[13,0] = 'Tamil Nadu'
shp_gdf.iloc[15,0] = 'Telangana'
shp_gdf.iloc[34,0] = 'Jammu and Kashmir'
shp_gdf.iloc[16,0] = 'Andhra Pradesh.'
# shp_gdf.head(35)
```

In [93]:
```
shp_gdf.shape
```

Out[93]:
```
(36, 2)
```

In [94]:
```
shp_gdf.iloc[1,0] = 'Andaman & Nicobar Islands'
shp_gdf.iloc[14,0]= 'Chhatisgarh'
shp_gdf.iloc[13,0] = 'Tamil Nadu'
shp_gdf.iloc[15,0] = 'Telangana'
shp_gdf.iloc[34,0] = 'Jammu and Kashmir'
shp_gdf.iloc[16,0] = 'Andhra Pradesh.'
# shp_gdf.head(35)
```

In [95]:
```
tmp1.iloc[16,0] = "Daman and Diu and Dadra and Nagar Haveli"
tmp1.iloc[16,1] = 18.84
```

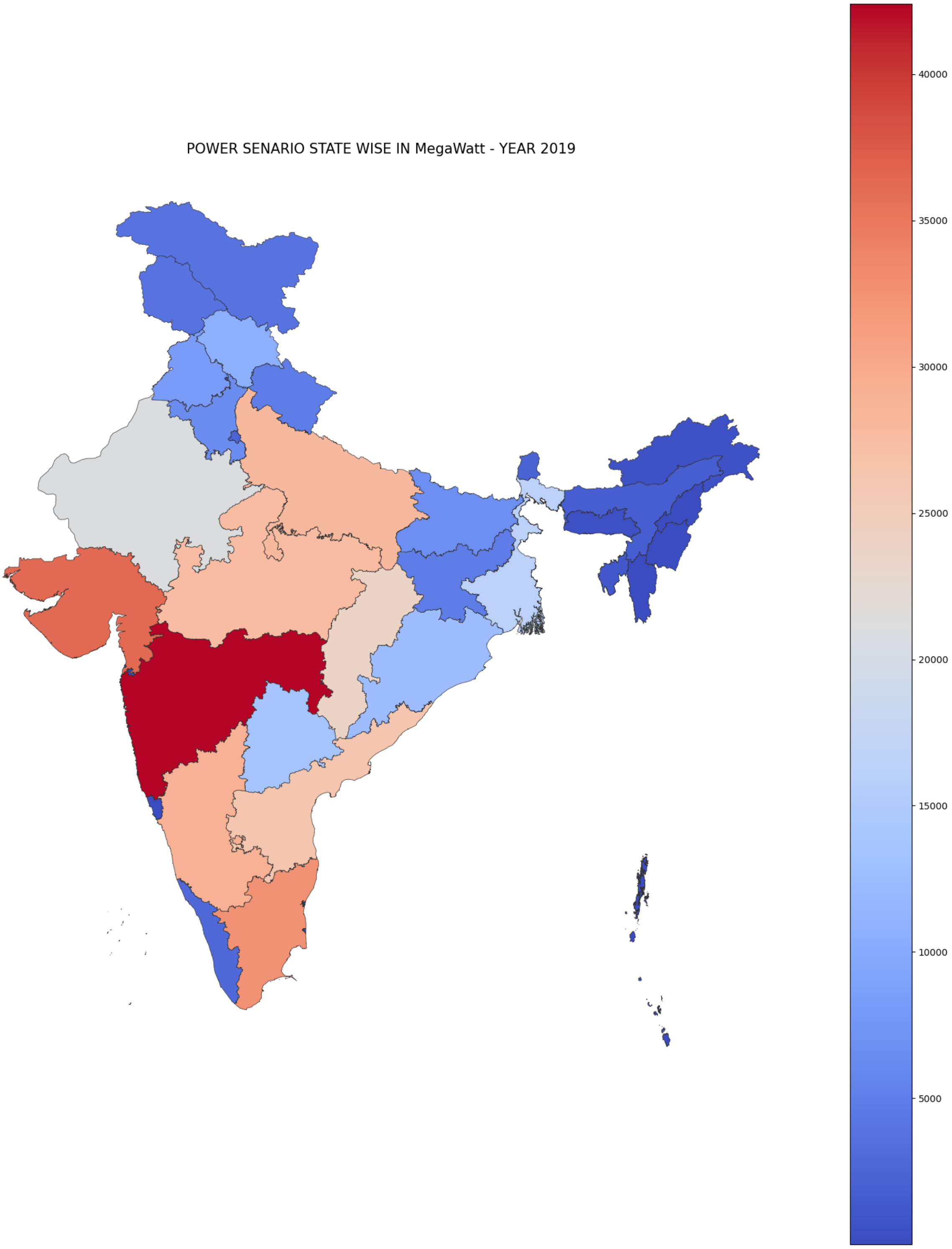
In [96]:
```
merged = shp_gdf.set_index('Name').join(tmp1.set_index('state'))
merged.iloc[33,1] = 3823.76 # MAking Ladake same as J&K since one before 2020
```

In [97]:
```
merged.iloc[33,1]
```

Out[97]:
```
3823.76
```

In [98]:
```
fig ,ax = plt.subplots(1,figsize=(20,24))
ax.axis('off')
ax.set_title("POWER SENARIO STATE WISE IN MegaWatt - YEAR 2019",fontdict={'fontsiz
fig = merged.plot(column='grand_total' , cmap='coolwarm',linewidth=0.5,ax=ax,edgec
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



POWER SENARIO STATE WISE IN MegaWatt - YEAR 2019

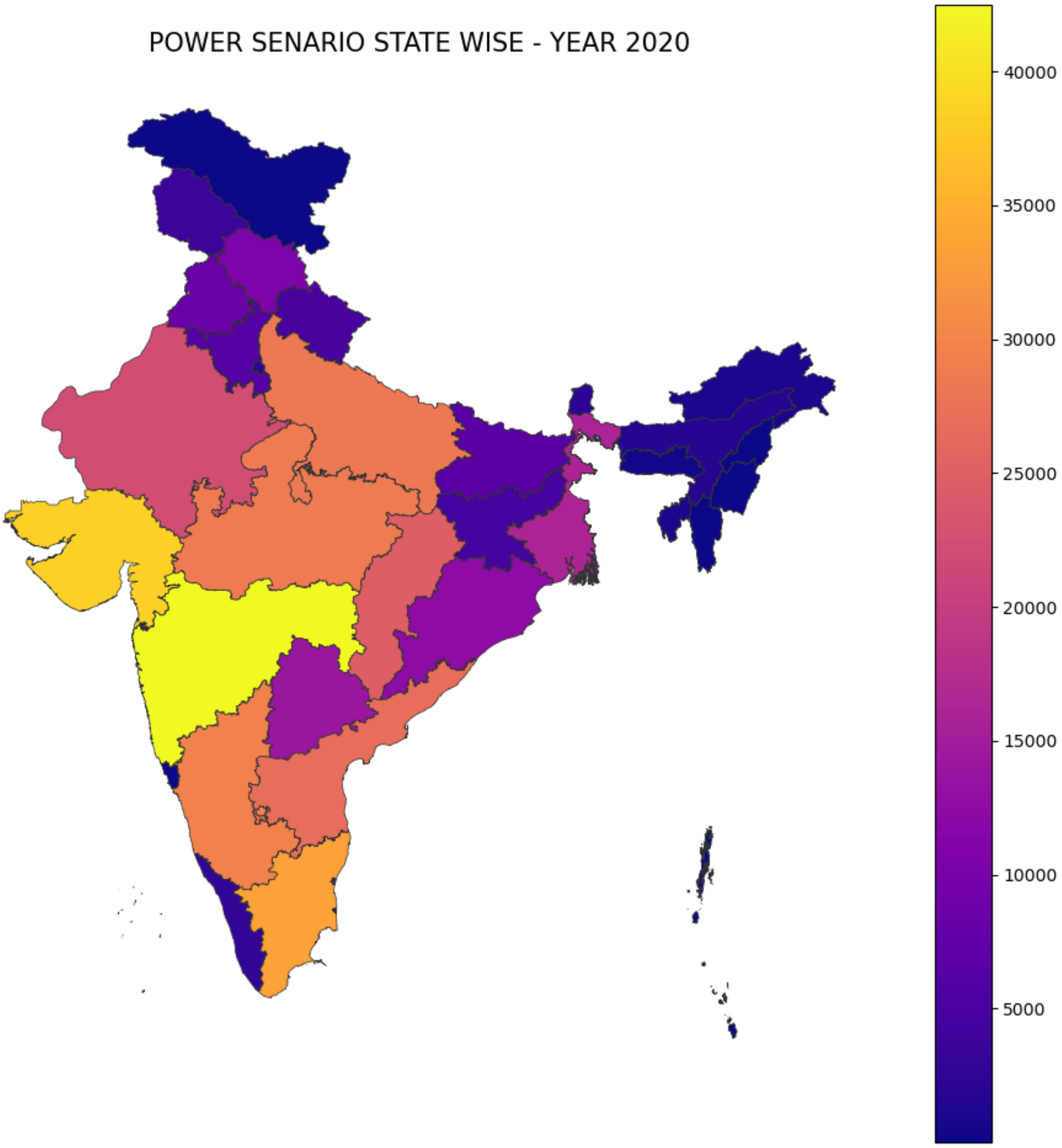
In [99]:

```
shp_gdf.iloc[16,0] = 'Andhra Pradesh'
merged1 = shp_gdf.set_index('Name').join(tmp2.set_index('state'))
merged1.iloc[3,1] = 45.46
```

In [100…

```
fig ,ax = plt.subplots(1,figsize=(12,12))
ax.axis('off')
ax.set_title("POWER SENARIO STATE WISE - YEAR 2020",fontdict={'fontsize':'15','fon
fig = merged1.plot(column='grand_total' , cmap='plasma',linewidth=0.5,ax=ax,edgeco
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [101…

```
merged2 = shp_gdf.set_index('Name').join(tmp3.set_index('state'))
```

In [106…

```
merged2
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[106]:

	geometry	grand_total
Name		
West Bengal	MULTIPOLYGON (((88.01861 21.57278, 88.01889 21...	16203.10
Andaman & Nicobar Islands	MULTIPOLYGON (((92.90124 12.91071, 92.90157 12...	74.79
Chandigarh	POLYGON ((76.77232 30.79420, 76.77286 30.79343...	53.45
Daman and Diu and Dadra and Nagar Haveli	MULTIPOLYGON (((72.96339 20.33227, 72.96390 20...	NaN
Delhi	POLYGON ((77.10591 28.87005, 77.10668 28.86989...	2471.52
Haryana	POLYGON ((76.57526 30.10063, 76.57645 30.10152...	6848.47
Jharkhand	POLYGON ((87.69613 24.16027, 87.69625 24.15974...	4556.42
Karnataka	MULTIPOLYGON (((77.33232 18.45086, 77.33267 18...	29861.60
Kerala	POLYGON ((74.98896 12.79553, 74.98953 12.79554...	3184.45
Lakshadweep	MULTIPOLYGON (((73.08025 8.32651, 73.08029 8.3...	3.27
Madhya Pradesh	POLYGON ((78.37211 26.86406, 78.37423 26.86091...	29606.38
Maharashtra	MULTIPOLYGON (((74.38926 22.03241, 74.38942 22...	42076.27
Puducherry	MULTIPOLYGON (((79.75669 11.00258, 79.75682 11...	44.55
Tamil Nadu	MULTIPOLYGON (((80.30358 13.47307, 80.30577 13...	35412.30
Chhatisgarh	POLYGON ((83.32707 24.10232, 83.32939 24.10130...	24660.58
Telangana	POLYGON ((78.33565 19.88358, 78.33688 19.88300...	14579.84
Andhra Pradesh	POLYGON ((84.67571 19.16721, 84.67725 19.16670...	27325.85
Goa	POLYGON ((73.87042 15.78117, 73.87097 15.78029...	66.88
Himachal Pradesh	POLYGON ((76.79412 33.25569, 76.79482 33.25557...	11095.76
Punjab	POLYGON ((75.83876 32.51269, 75.84094 32.51158...	8542.41
Rajasthan	POLYGON ((73.97266 30.19800, 73.97266 30.19800...	27640.84
Gujarat	POLYGON ((72.26126 21.49999, 72.25911 21.49999...	41520.77

	geometry	grand_total
Name		
Uttarakhand	POLYGON ((79.06793 31.46153, 79.07046 31.46010...	5211.91
Uttar Pradesh	POLYGON ((77.59189 30.40632, 77.59337 30.40621...	30516.86
Sikkim	POLYGON ((88.61635 28.12763, 88.62302 28.12625...	2338.76
Assam	POLYGON ((93.24961 24.97312, 93.25244 24.97428...	1825.03
Arunachal Pradesh	POLYGON ((96.08538 29.45928, 96.09262 29.45613...	1257.34
Nagaland	POLYGON ((95.19272 27.02710, 95.19475 27.01442...	108.71
Manipur	POLYGON ((94.57415 25.69043, 94.57852 25.68761...	158.65
Mizoram	POLYGON ((92.76384 24.52098, 92.76422 24.51967...	104.35
Tripura	POLYGON ((92.16949 24.53175, 92.16961 24.53168...	1130.48
Meghalaya	POLYGON ((91.82617 26.11925, 91.82728 26.11914...	372.46
Bihar	POLYGON ((84.10880 27.52173, 84.10896 27.52131...	8346.95
Ladakh	POLYGON ((76.80933 33.24349, 76.79356 33.25175...	136.44
Jammu and Kashmir	POLYGON ((76.80933 33.24349, 76.80894 33.24367...	3726.55
Odisha	POLYGON ((87.47639 21.64343, 87.47362 21.63205...	12279.29

In [102…

```
tmp3
```

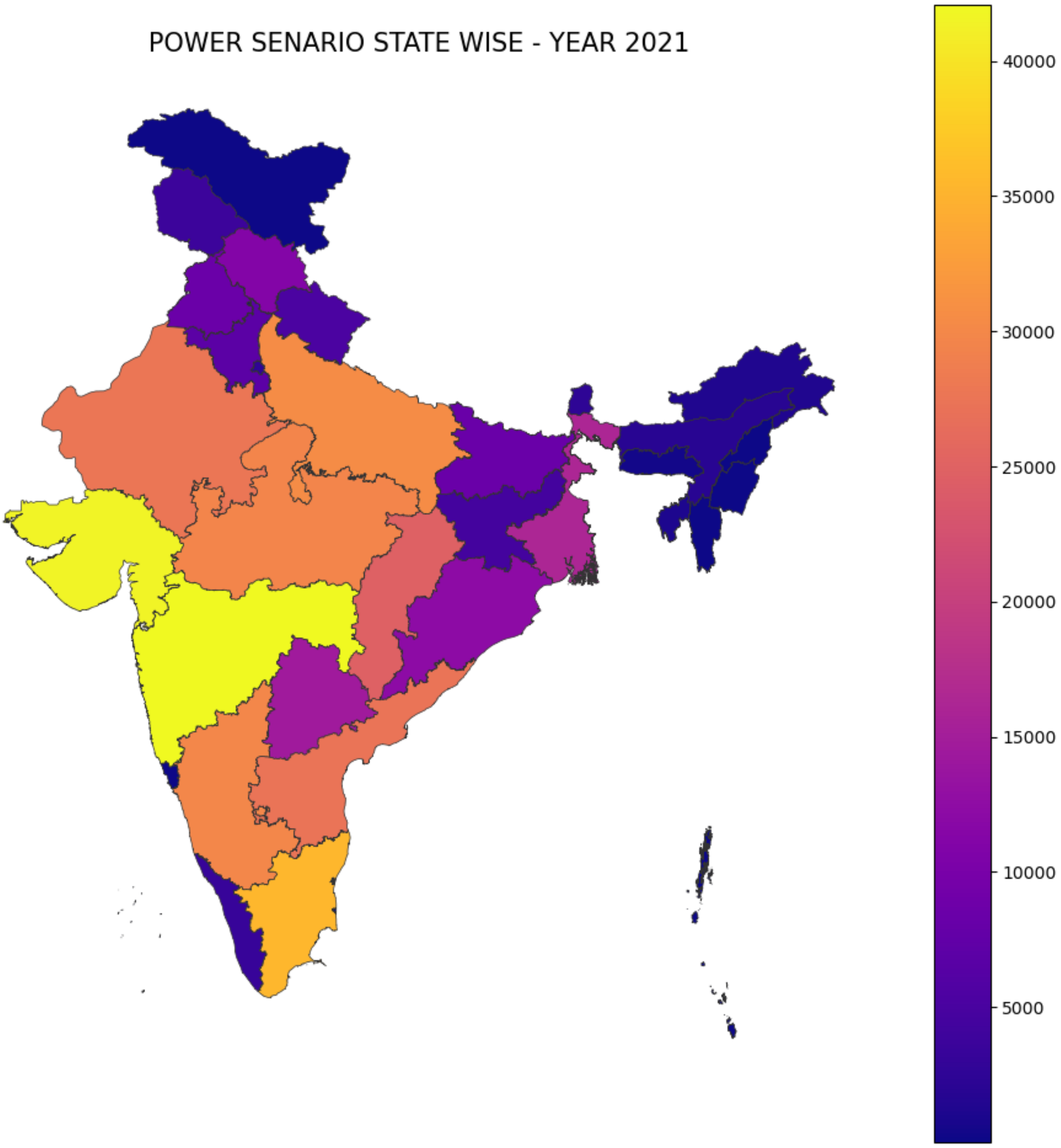
Out[102]:

	state	grand_total
1281	Chandigarh	53.45
1282	Delhi	2471.52
1283	Haryana	6848.47
1284	Himachal Pradesh	11095.76
1285	Jammu and Kashmir	3726.55
1286	Ladakh	136.44
1287	Punjab	8542.41
1288	Rajasthan	27640.84
1289	Uttar Pradesh	30516.86
1290	Uttarakhand	5211.91
1291	Andaman & Nicobar Islands	74.79
1292	Bihar	8346.95
1293	Jharkhand	4556.42
1294	Odisha	12279.29
1295	Sikkim	2338.76
1296	West Bengal	16203.10
1297	Chhatisgarh	24660.58
1298	Dadra and Nagar Haveli and Dam	46.18
1299	Goa	66.88
1300	Gujarat	41520.77
1301	Madhya Pradesh	29606.38
1302	Maharashtra	42076.27
1303	Andhra Pradesh	27325.85
1304	Karnataka	29861.60
1305	Kerala	3184.45
1306	Lakshadweep	3.27
1307	Puducherry	44.55
1308	Tamil Nadu	35412.30
1309	Telangana	14579.84
1310	Arunachal Pradesh	1257.34
1311	Assam	1825.03
1312	Manipur	158.65
1313	Meghalaya	372.46
1314	Mizoram	104.35
1315	Nagaland	108.71
[illegible]	[illegible]	130.48

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [103…

```
fig ,ax = plt.subplots(1,figsize=(12,12))
ax.axis('off')
ax.set_title("POWER SENARIO STATE WISE - YEAR 2021",fontdict={'fontsize':'15','fon
fig = merged2.plot(column='grand_total' , cmap='plasma',linewidth=0.5,ax=ax,edgeco
```



In [107…

```
hydro1 = hydro.iloc[396:432,:]
hydro2 = hydro.iloc[839:876,:]
hydro3 = hydro.iloc[1281:1317,:]
renewable1 = renewable.iloc[396:432,:]
renewable2 = renewable.iloc[839:876,:]
renewable3 = renewable.iloc[1281:1317,:]
nuclear1 = nuclear.iloc[396:432,:]
nuclear2 = nuclear.iloc[839:876,:]
nuclear3 = nuclear.iloc[1281:1317,:]
coal1 = coal.iloc[396:432,:]
coal2 = coal.iloc[839:876,:]
coal3 = coal.iloc[1281:1317,:]
```

In [115…

```
hydro1
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[115]:

	state	hydro
396	Chandigarh	0.00
397	Delhi	0.00
398	Haryana	0.00
399	Himachal Pradesh	9809.02
400	Jammu and Kashmir	3449.00
401	Punjab	1096.30
402	Rajasthan	411.00
403	Uttar Pradesh	501.60
404	Uttarakhand	3756.35
405	Andaman & Nicobar Islands	0.00
406	Bihar	0.00
407	Jharkhand	210.00
408	Odisha	2142.25
409	Sikkim	2169.00
410	West Bengal	1341.20
411	Chhatisgarh	120.00
412	Dadra & Nagar Haveli	0.00
413	Daman & Diu	0.00
414	Goa	0.00
415	Gujarat	1990.00
416	Madhya Pradesh	2235.00
417	Maharashtra	3047.00
418	Andhra Pradesh.	1610.00
419	Karnataka	3644.20
420	Kerala	1856.50
421	Lakshadweep	0.00
422	Puducherry	0.00
423	Tamil Nadu	2178.20
424	Telangana	2405.60
425	Arunachal Pradesh	515.00
426	Assam	350.00
427	Manipur	105.00
428	Meghalaya	322.00
429	Mizoram	60.00
430	Nagaland	75.00

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [111…

```python
merged_hydro = shp_gdf.set_index('Name').join(hydro1.set_index('state'))
merged_hydro.iloc[33,1] = 3449 # MAking Ladake same as J&K since one before 2020
```

In [112…

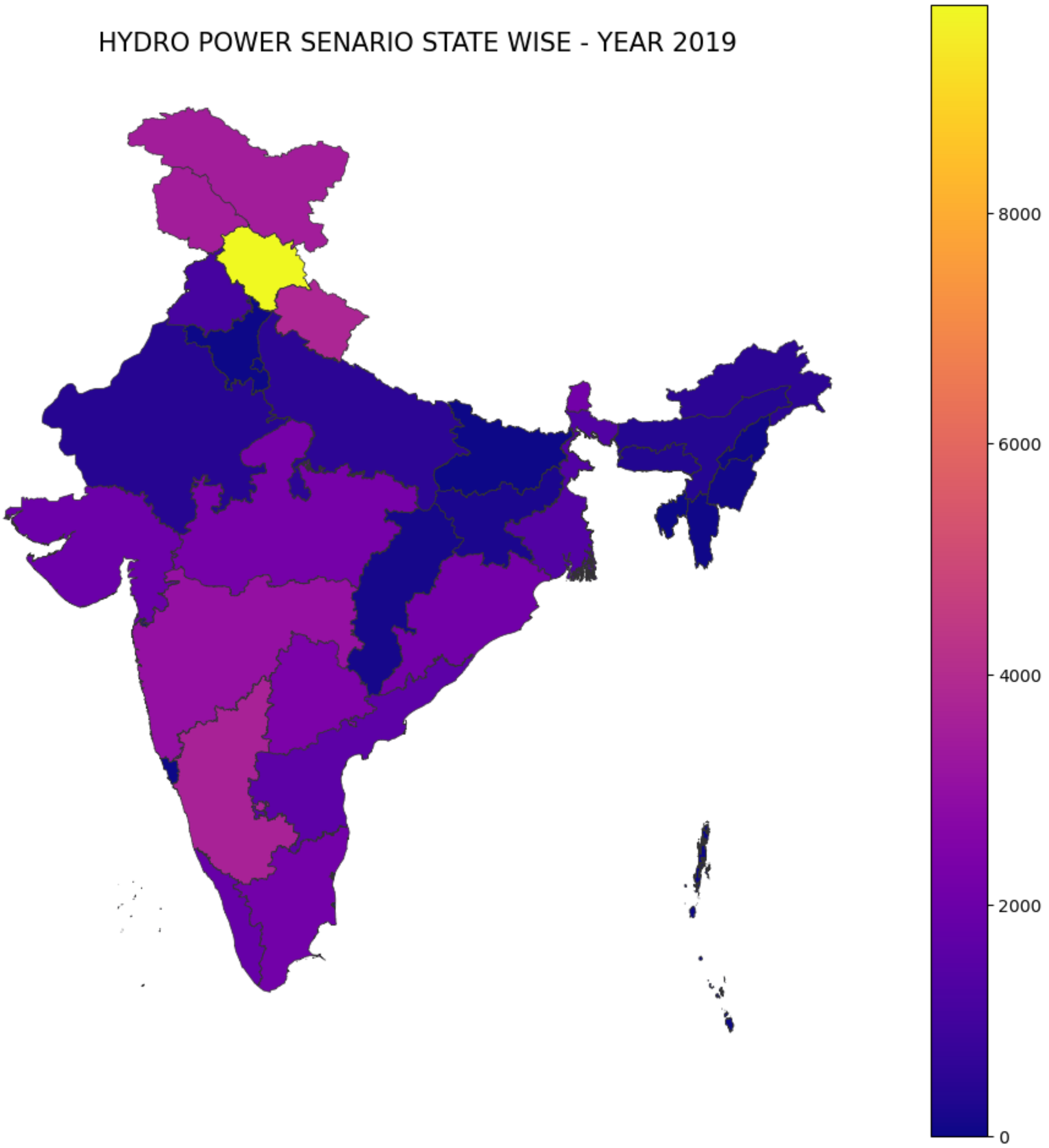
```python
merged_hydro.iloc[3,1] = 0
```

In [116…

```python
merged_hydro.iloc[16,1] = 1610.1
```

In [117…

```python
fig ,ax = plt.subplots(1,figsize=(12,12))
ax.axis('off')
ax.set_title("HYDRO POWER SENARIO STATE WISE - YEAR 2019",fontdict={'fontsize':'15
fig = merged_hydro.plot(column='hydro' , cmap='plasma',linewidth=0.5,ax=ax,edgecol
```



In [118…

```python
merged_hydro_2 = shp_gdf.set_index('Name').join(hydro2.set_index('state'))
# merged_hydro.iloc[33,1] = 3449 # MAking Ladake same as J&K since one before 2020
```

In [119…

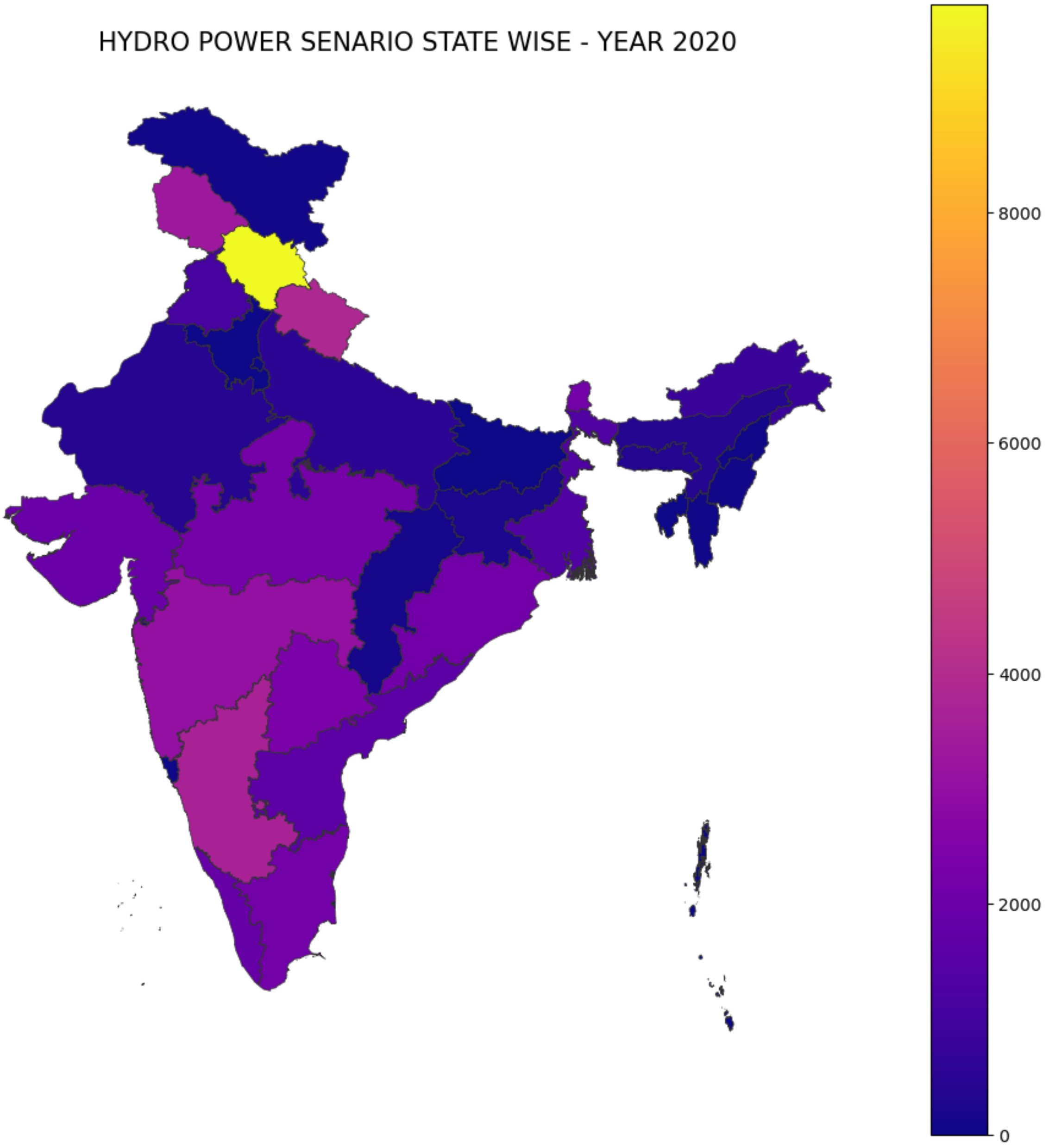
```python
merged_hydro_2.iloc[3,1] = 0
merged_hydro_2.iloc[16,1] = 1610
```

In [120…

```python
fig ,ax = plt.subplots(1,figsize=(12,12))
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
ax.set_title("HYDRO POWER SENARIO STATE WISE - YEAR 2020",fontdict={'fontsize':'15
fig = merged_hydro_2.plot(column='hydro' , cmap='plasma',linewidth=0.5,ax=ax,edgec
```



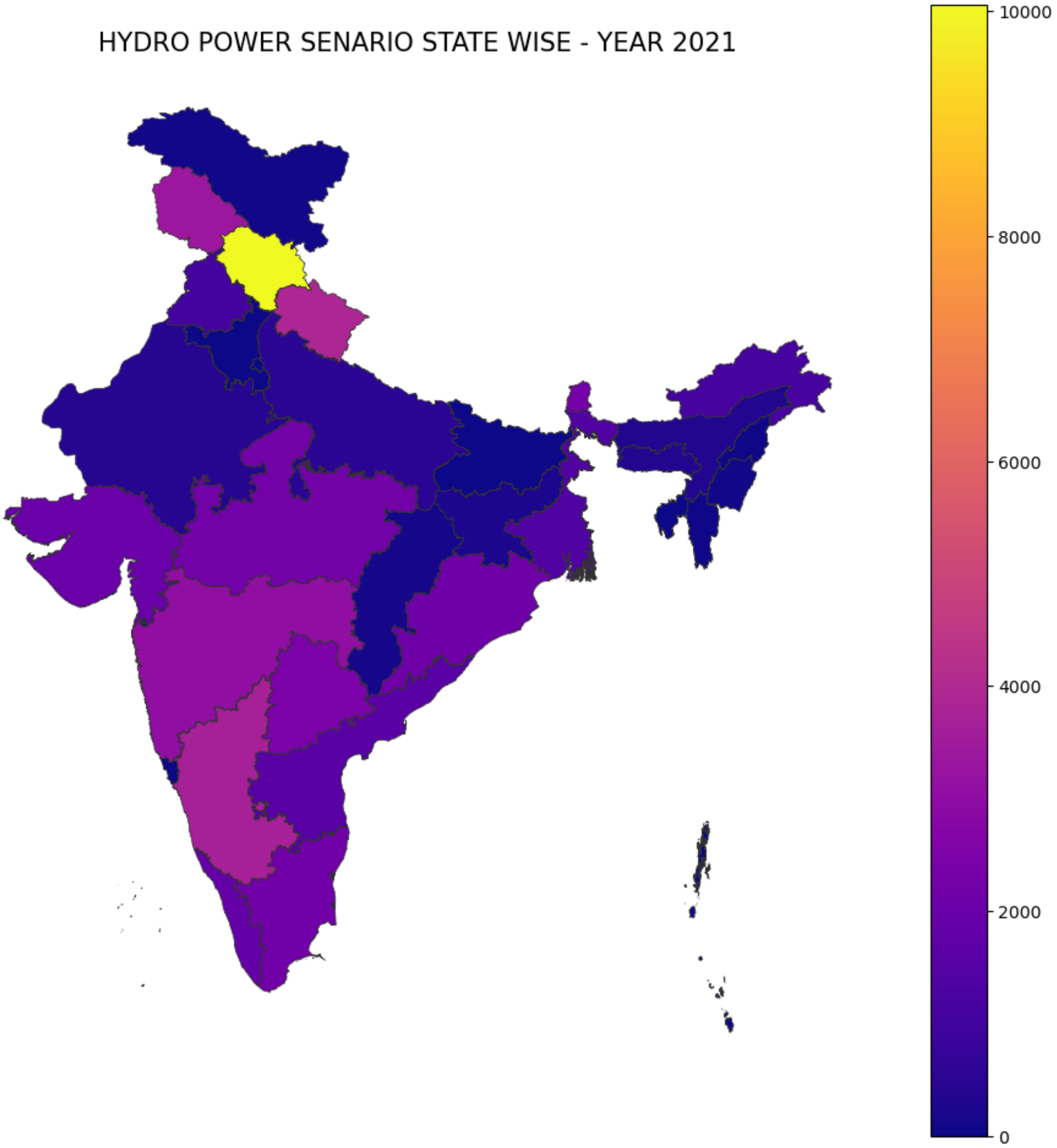
In [121…

```
merged_hydro_3 = shp_gdf.set_index('Name').join(hydro3.set_index('state'))
```

In [122…

```
merged_hydro_3.iloc[3,1] = 0
merged_hydro_3.iloc[16,1] = 1610
fig ,ax = plt.subplots(1,figsize=(12,12))
ax.axis('off')
ax.set_title("HYDRO POWER SENARIO STATE WISE - YEAR 2021",fontdict={'fontsize':'15
fig = merged_hydro_3.plot(column='hydro' , cmap='plasma',linewidth=0.5,ax=ax,edgec
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [123…

```
merged_rene_1 = shp_gdf.set_index('Name').join(renewable1.set_index('state'))
merged_rene_1.iloc[33,1] = 199.76 # MAking Ladake same as J&K since one before 202
merged_rene_1.iloc[3,1] = 16.56
```

In [124…

```
merged_rene_1
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[124]:

	geometry	res
Name		
West Bengal	MULTIPOLYGON (((88.01861 21.57278, 88.01889 21...	527.29
Andaman & Nicobar Islands	MULTIPOLYGON (((92.90124 12.91071, 92.90157 12...	17.44
Chandigarh	POLYGON ((76.77232 30.79420, 76.77286 30.79343...	36.99
Daman and Diu and Dadra and Nagar Haveli	MULTIPOLYGON (((72.96339 20.33227, 72.96390 20...	16.56
Delhi	POLYGON ((77.10591 28.87005, 77.10668 28.86989...	200.41
Haryana	POLYGON ((76.57526 30.10063, 76.57645 30.10152...	527.72
Jharkhand	POLYGON ((87.69613 24.16027, 87.69625 24.15974...	46.75
Karnataka	MULTIPOLYGON (((77.33232 18.45086, 77.33267 18...	15011.86
Kerala	POLYGON ((74.98896 12.79553, 74.98953 12.79554...	426.68
Lakshadweep	MULTIPOLYGON (((73.08025 8.32651, 73.08029 8.3...	0.75
Madhya Pradesh	POLYGON ((78.37211 26.86406, 78.37423 26.86091...	4973.39
Maharashtra	MULTIPOLYGON (((74.38926 22.03241, 74.38942 22...	9368.80
Puducherry	MULTIPOLYGON (((79.75669 11.00258, 79.75682 11...	5.51
Tamil Nadu	MULTIPOLYGON (((80.30358 13.47307, 80.30577 13...	13516.82
Chhatisgarh	POLYGON ((83.32707 24.10232, 83.32939 24.10130...	537.85
Telangana	POLYGON ((78.33565 19.88358, 78.33688 19.88300...	4017.32
Andhra Pradesh	POLYGON ((84.67571 19.16721, 84.67725 19.16670...	NaN
Goa	POLYGON ((73.87042 15.78117, 73.87097 15.78029...	4.83
Himachal Pradesh	POLYGON ((76.79412 33.25569, 76.79482 33.25557...	945.24
Punjab	POLYGON ((75.83876 32.51269, 75.84094 32.51158...	1448.50
Rajasthan	POLYGON ((73.97266 30.19800, 73.97266 30.19800...	9188.08
Gujarat	POLYGON ((72.26126 21.49999, 72.25911 21.49999...	10178.26

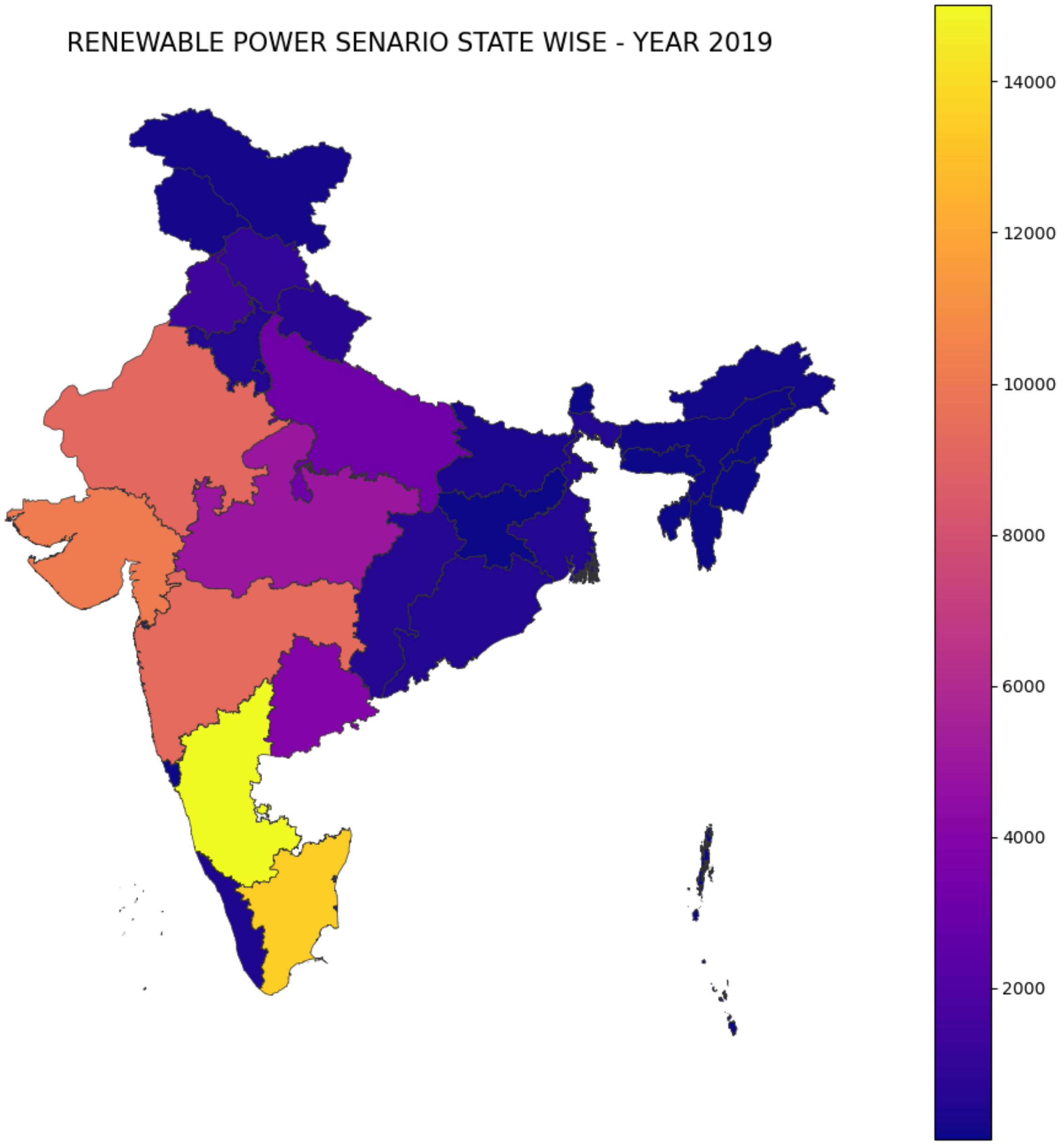
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

	geometry	res
Name		
Uttarakhand	POLYGON ((79.06793 31.46153, 79.07046 31.46010...	660.21
Uttar Pradesh	POLYGON ((77.59189 30.40632, 77.59337 30.40621...	3185.71
Sikkim	POLYGON ((88.61635 28.12763, 88.62302 28.12625...	52.18
Assam	POLYGON ((93.24961 24.97312, 93.25244 24.97428...	75.34
Arunachal Pradesh	POLYGON ((96.08538 29.45928, 96.09262 29.45613...	136.72
Nagaland	POLYGON ((95.19272 27.02710, 95.19475 27.01442...	31.67
Manipur	POLYGON ((94.57415 25.69043, 94.57852 25.68761...	9.84
Mizoram	POLYGON ((92.76384 24.52098, 92.76422 24.51967...	37.97
Tripura	POLYGON ((92.16949 24.53175, 92.16961 24.53168...	25.42
Meghalaya	POLYGON ((91.82617 26.11925, 91.82728 26.11914...	46.45
Bihar	POLYGON ((84.10880 27.52173, 84.10896 27.52131...	341.25
Ladakh	POLYGON ((76.80933 33.24349, 76.79356 33.25175...	199.76
Jammu and Kashmir	POLYGON ((76.80933 33.24349, 76.80894 33.24367...	199.76
Odisha	POLYGON ((87.47639 21.64343, 87.47362 21.63205...	521.21

In [125…

```
fig ,ax = plt.subplots(1,figsize=(12,12))
ax.axis('off')
ax.set_title("RENEWABLE POWER SENARIO STATE WISE - YEAR 2019",fontdict={'fontsize'
fig = merged_rene_1.plot(column='res' , cmap='plasma',linewidth=0.5,ax=ax,edgecolo
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [477…

```
merged_rene_2 = shp_gdf.set_index('Name').join(renewable2.set_index('state'))
# merged_rene_1.iloc[33,1] = 199.76 # MAking Ladake same as J&K since one before 2
merged_rene_2.iloc[3,1] = 5.46
merged_rene_2.iloc[16,1] =8605.56
```

In [478…

```
merged_rene_2
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[478]:

	geometry	res
Name		
West Bengal	MULTIPOLYGON (((88.01861 21.57278, 88.01889 21...	568.26
Andaman & Nicobar Islands	MULTIPOLYGON (((92.90124 12.91071, 92.90157 12...	34.47
Chandigarh	POLYGON ((76.77232 30.79420, 76.77286 30.79343...	45.16
Daman and Diu and Dadra and Nagar Haveli	MULTIPOLYGON (((72.96339 20.33227, 72.96390 20...	5.46
Delhi	POLYGON ((77.10591 28.87005, 77.10668 28.86989...	228.46
Haryana	POLYGON ((76.57526 30.10063, 76.57645 30.10152...	547.78
Jharkhand	POLYGON ((87.69613 24.16027, 87.69625 24.15974...	47.41
Karnataka	MULTIPOLYGON (((77.33232 18.45086, 77.33267 18...	15366.70
Kerala	POLYGON ((74.98896 12.79553, 74.98953 12.79554...	433.71
Lakshadweep	MULTIPOLYGON (((73.08025 8.32651, 73.08029 8.3...	0.75
Madhya Pradesh	POLYGON ((78.37211 26.86406, 78.37423 26.86091...	5182.50
Maharashtra	MULTIPOLYGON (((74.38926 22.03241, 74.38942 22...	9872.72
Puducherry	MULTIPOLYGON (((79.75669 11.00258, 79.75682 11...	7.54
Tamil Nadu	MULTIPOLYGON (((80.30358 13.47307, 80.30577 13...	14890.62
Chhatisgarh	POLYGON ((83.32707 24.10232, 83.32939 24.10130...	560.90
Telangana	POLYGON ((78.33565 19.88358, 78.33688 19.88300...	4361.23
Andhra Pradesh	POLYGON ((84.67571 19.16721, 84.67725 19.16670...	8605.56
Goa	POLYGON ((73.87042 15.78117, 73.87097 15.78029...	5.17
Himachal Pradesh	POLYGON ((76.79412 33.25569, 76.79482 33.25557...	963.44
Punjab	POLYGON ((75.83876 32.51269, 75.84094 32.51158...	1604.85
Rajasthan	POLYGON ((73.97266 30.19800, 73.97266 30.19800...	9861.40
Gujarat	POLYGON ((72.26126 21.49999, 72.25911 21.49999...	12267.24

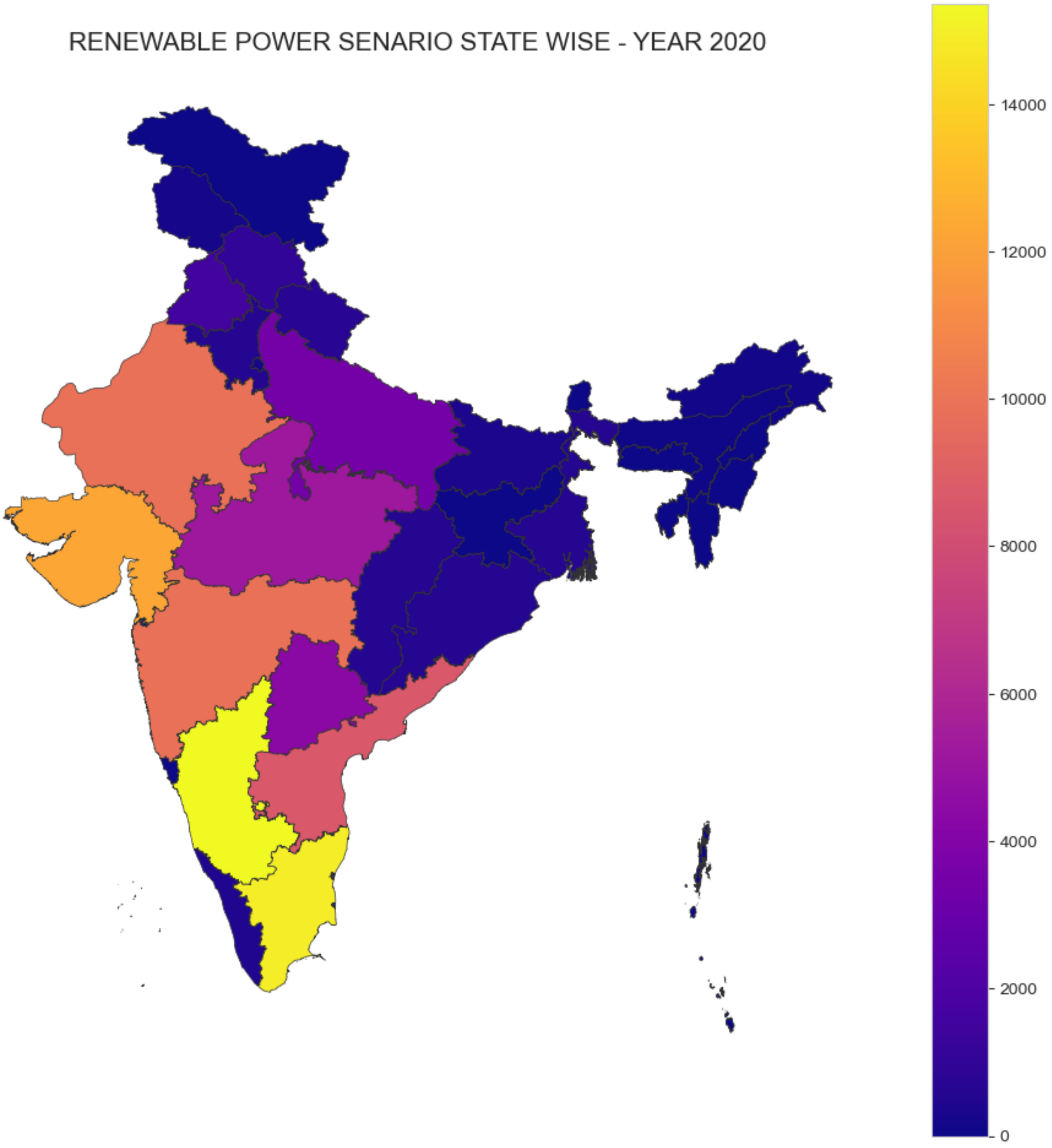
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

	geometry	res
Name		
Uttarakhand	POLYGON ((79.06793 31.46153, 79.07046 31.46010...	664.98
Uttar Pradesh	POLYGON ((77.59189 30.40632, 77.59337 30.40621...	3459.21
Sikkim	POLYGON ((88.61635 28.12763, 88.62302 28.12625...	52.18
Assam	POLYGON ((93.24961 24.97312, 93.25244 24.97428...	79.10
Arunachal Pradesh	POLYGON ((96.08538 29.45928, 96.09262 29.45613...	136.72
Nagaland	POLYGON ((95.19272 27.02710, 95.19475 27.01442...	31.67
Manipur	POLYGON ((94.57415 25.69043, 94.57852 25.68761...	11.81
Mizoram	POLYGON ((92.76384 24.52098, 92.76422 24.51967...	38.00
Tripura	POLYGON ((92.16949 24.53175, 92.16961 24.53168...	25.42
Meghalaya	POLYGON ((91.82617 26.11925, 91.82728 26.11914...	46.45
Bihar	POLYGON ((84.10880 27.52173, 84.10896 27.52131...	352.81
Ladakh	POLYGON ((76.80933 33.24349, 76.79356 33.25175...	0.00
Jammu and Kashmir	POLYGON ((76.80933 33.24349, 76.80894 33.24367...	206.71
Odisha	POLYGON ((87.47639 21.64343, 87.47362 21.63205...	547.45

In [479...

```
fig ,ax = plt.subplots(1,figsize=(12,12))
ax.axis('off')
ax.set_title("RENEWABLE POWER SENARIO STATE WISE - YEAR 2020",fontdict={'fontsize'
fig = merged_rene_2.plot(column='res' , cmap='plasma',linewidth=0.5,ax=ax,edgecolo
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [480…

```
merged_rene_3 = shp_gdf.set_index('Name').join(renewable2.set_index('state'))
# merged_rene_1.iloc[33,1] = 199.76 # MAking Ladake same as J&K since one before 2
merged_rene_3.iloc[3,1] = 46.18
merged_rene_3.iloc[16,1] =9190.51
```

In [481…

```
merged_rene_3
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[481]:

	geometry	res
Name		
West Bengal	MULTIPOLYGON (((88.01861 21.57278, 88.01889 21...	568.26
Andaman & Nicobar Islands	MULTIPOLYGON (((92.90124 12.91071, 92.90157 12...	34.47
Chandigarh	POLYGON ((76.77232 30.79420, 76.77286 30.79343...	45.16
Daman and Diu and Dadra and Nagar Haveli	MULTIPOLYGON (((72.96339 20.33227, 72.96390 20...	46.18
Delhi	POLYGON ((77.10591 28.87005, 77.10668 28.86989...	228.46
Haryana	POLYGON ((76.57526 30.10063, 76.57645 30.10152...	547.78
Jharkhand	POLYGON ((87.69613 24.16027, 87.69625 24.15974...	47.41
Karnataka	MULTIPOLYGON (((77.33232 18.45086, 77.33267 18...	15366.70
Kerala	POLYGON ((74.98896 12.79553, 74.98953 12.79554...	433.71
Lakshadweep	MULTIPOLYGON (((73.08025 8.32651, 73.08029 8.3...	0.75
Madhya Pradesh	POLYGON ((78.37211 26.86406, 78.37423 26.86091...	5182.50
Maharashtra	MULTIPOLYGON (((74.38926 22.03241, 74.38942 22...	9872.72
Puducherry	MULTIPOLYGON (((79.75669 11.00258, 79.75682 11...	7.54
Tamil Nadu	MULTIPOLYGON (((80.30358 13.47307, 80.30577 13...	14890.62
Chhatisgarh	POLYGON ((83.32707 24.10232, 83.32939 24.10130...	560.90
Telangana	POLYGON ((78.33565 19.88358, 78.33688 19.88300...	4361.23
Andhra Pradesh	POLYGON ((84.67571 19.16721, 84.67725 19.16670...	9190.51
Goa	POLYGON ((73.87042 15.78117, 73.87097 15.78029...	5.17
Himachal Pradesh	POLYGON ((76.79412 33.25569, 76.79482 33.25557...	963.44
Punjab	POLYGON ((75.83876 32.51269, 75.84094 32.51158...	1604.85
Rajasthan	POLYGON ((73.97266 30.19800, 73.97266 30.19800...	9861.40
Gujarat	POLYGON ((72.26126 21.49999, 72.25911 21.49999...	12267.24

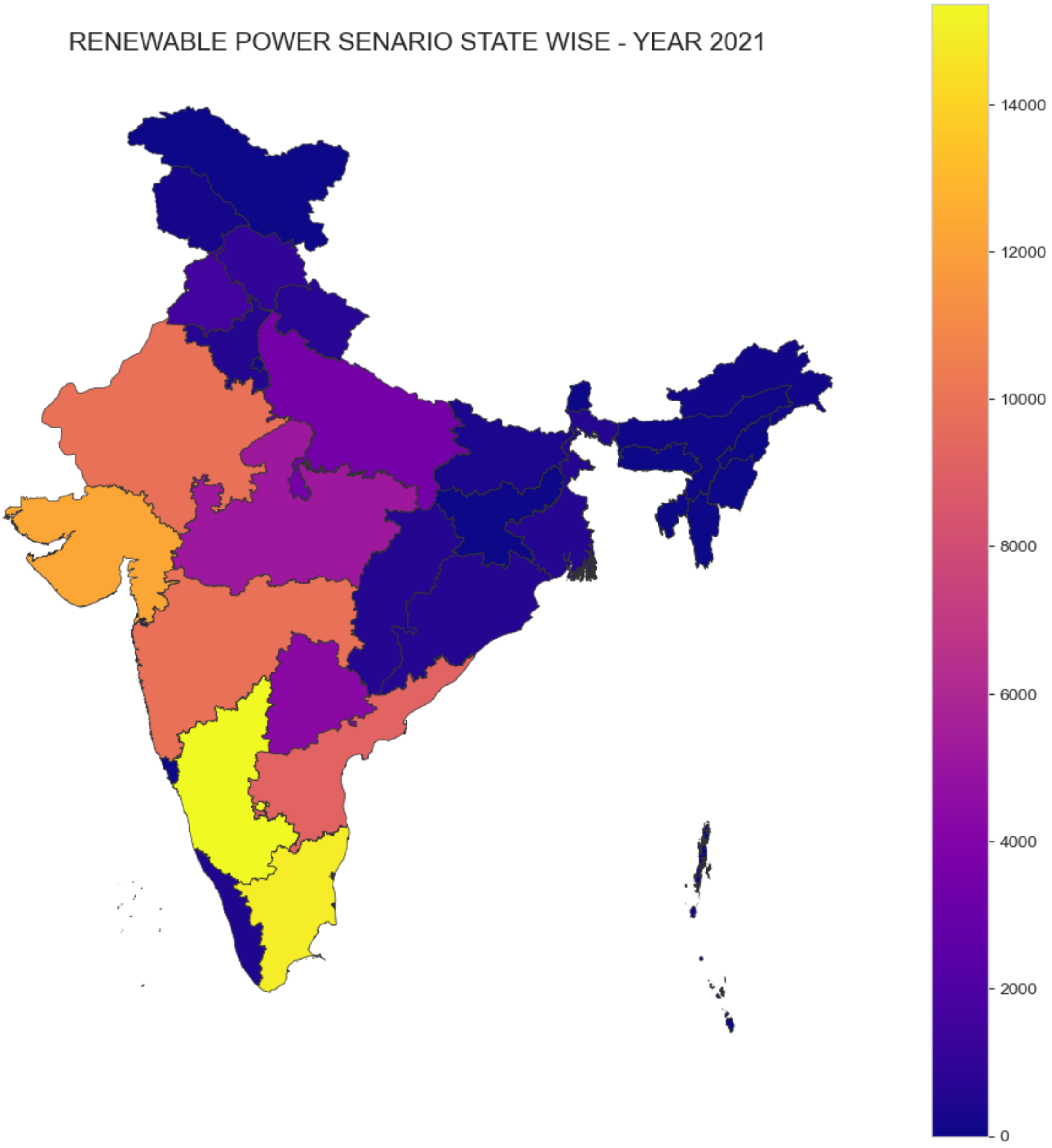
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

	geometry	res
Name		
Uttarakhand	POLYGON ((79.06793 31.46153, 79.07046 31.46010...	664.98
Uttar Pradesh	POLYGON ((77.59189 30.40632, 77.59337 30.40621...	3459.21
Sikkim	POLYGON ((88.61635 28.12763, 88.62302 28.12625...	52.18
Assam	POLYGON ((93.24961 24.97312, 93.25244 24.97428...	79.10
Arunachal Pradesh	POLYGON ((96.08538 29.45928, 96.09262 29.45613...	136.72
Nagaland	POLYGON ((95.19272 27.02710, 95.19475 27.01442...	31.67
Manipur	POLYGON ((94.57415 25.69043, 94.57852 25.68761...	11.81
Mizoram	POLYGON ((92.76384 24.52098, 92.76422 24.51967...	38.00
Tripura	POLYGON ((92.16949 24.53175, 92.16961 24.53168...	25.42
Meghalaya	POLYGON ((91.82617 26.11925, 91.82728 26.11914...	46.45
Bihar	POLYGON ((84.10880 27.52173, 84.10896 27.52131...	352.81
Ladakh	POLYGON ((76.80933 33.24349, 76.79356 33.25175...	0.00
Jammu and Kashmir	POLYGON ((76.80933 33.24349, 76.80894 33.24367...	206.71
Odisha	POLYGON ((87.47639 21.64343, 87.47362 21.63205...	547.45

In [482…

```
fig ,ax = plt.subplots(1,figsize=(12,12))
ax.axis('off')
ax.set_title("RENEWABLE POWER SENARIO STATE WISE - YEAR 2021",fontdict={'fontsize'
fig = merged_rene_3.plot(column='res' , cmap='plasma',linewidth=0.5,ax=ax,edgecolo
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [126…

```
merged_nuclear_1 = shp_gdf.set_index('Name').join(nuclear1.set_index('state'))
merged_nuclear_1.iloc[33,1] = 0 # MAking Ladake same as J&K since one before 2020
merged_nuclear_1.iloc[3,1] = 0
```

In [129…

```
nuclear1
```

Out[129]:

	state	nuclear
396	Chandigarh	0.0
397	Delhi	0.0
398	Haryana	0.0
399	Himachal Pradesh	0.0
400	Jammu and Kashmir	0.0
401	Punjab	0.0
402	Rajasthan	1180.0
403	Uttar Pradesh	440.0
404	Uttarakhand	0.0
405	Andaman & Nicobar Islands	0.0
406	Bihar	0.0
407	Jharkhand	0.0
408	Odisha	0.0
409	Sikkim	0.0
410	West Bengal	0.0
411	Chhatisgarh	0.0
412	Dadra & Nagar Haveli	0.0
413	Daman & Diu	0.0
414	Goa	0.0
415	Gujarat	440.0
416	Madhya Pradesh	0.0
417	Maharashtra	1400.0
418	Andhra Pradesh.	0.0
419	Karnataka	880.0
420	Kerala	0.0
421	Lakshadweep	0.0
422	Puducherry	0.0
423	Tamil Nadu	2440.0
424	Telangana	0.0
425	Arunachal Pradesh	0.0
426	Assam	0.0
427	Manipur	0.0
428	Meghalaya	0.0
429	Mizoram	0.0
430	Nagaland	0.0

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
In [128…  merged_nuclear_1
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

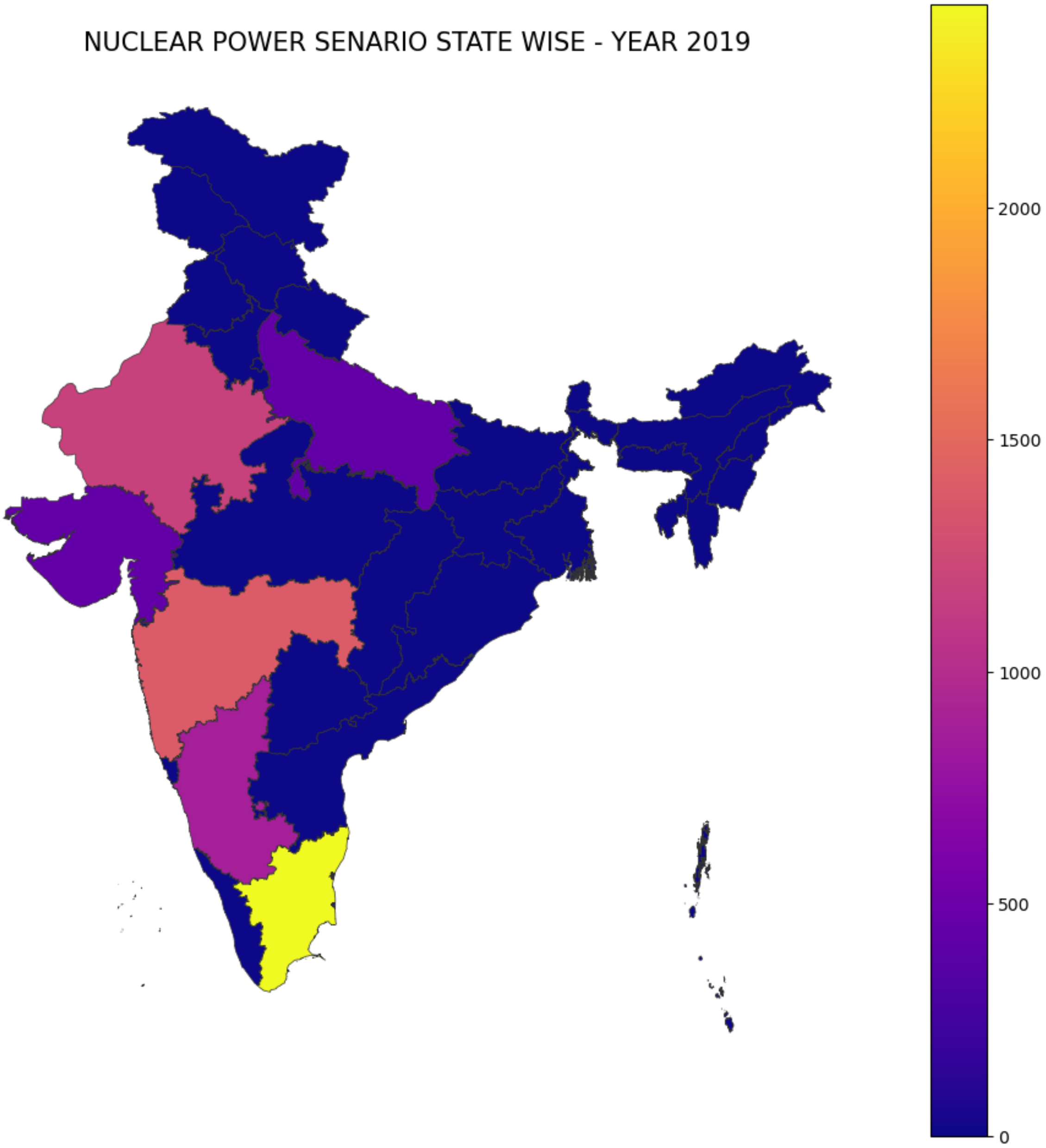
Out[128]:

Name	geometry	nuclear
West Bengal	MULTIPOLYGON (((88.01861 21.57278, 88.01889 21...	0.0
Andaman & Nicobar Islands	MULTIPOLYGON (((92.90124 12.91071, 92.90157 12...	0.0
Chandigarh	POLYGON ((76.77232 30.79420, 76.77286 30.79343...	0.0
Daman and Diu and Dadra and Nagar Haveli	MULTIPOLYGON (((72.96339 20.33227, 72.96390 20...	0.0
Delhi	POLYGON ((77.10591 28.87005, 77.10668 28.86989...	0.0
Haryana	POLYGON ((76.57526 30.10063, 76.57645 30.10152...	0.0
Jharkhand	POLYGON ((87.69613 24.16027, 87.69625 24.15974...	0.0
Karnataka	MULTIPOLYGON (((77.33232 18.45086, 77.33267 18...	880.0
Kerala	POLYGON ((74.98896 12.79553, 74.98953 12.79554...	0.0
Lakshadweep	MULTIPOLYGON (((73.08025 8.32651, 73.08029 8.3...	0.0
Madhya Pradesh	POLYGON ((78.37211 26.86406, 78.37423 26.86091...	0.0
Maharashtra	MULTIPOLYGON (((74.38926 22.03241, 74.38942 22...	1400.0
Puducherry	MULTIPOLYGON (((79.75669 11.00258, 79.75682 11...	0.0
Tamil Nadu	MULTIPOLYGON (((80.30358 13.47307, 80.30577 13...	2440.0
Chhatisgarh	POLYGON ((83.32707 24.10232, 83.32939 24.10130...	0.0
Telangana	POLYGON ((78.33565 19.88358, 78.33688 19.88300...	0.0
Andhra Pradesh	POLYGON ((84.67571 19.16721, 84.67725 19.16670...	NaN
Goa	POLYGON ((73.87042 15.78117, 73.87097 15.78029...	0.0
Himachal Pradesh	POLYGON ((76.79412 33.25569, 76.79482 33.25557...	0.0
Punjab	POLYGON ((75.83876 32.51269, 75.84094 32.51158...	0.0
Rajasthan	POLYGON ((73.97266 30.19800, 73.97266 30.19800...	1180.0
Gujarat	POLYGON ((72.26126 21.49999, 72.25911 21.49999...	440.0

	geometry	nuclear
Name		
Uttarakhand	POLYGON ((79.06793 31.46153, 79.07046 31.46010...	0.0
Uttar Pradesh	POLYGON ((77.59189 30.40632, 77.59337 30.40621...	440.0
Sikkim	POLYGON ((88.61635 28.12763, 88.62302 28.12625...	0.0
Assam	POLYGON ((93.24961 24.97312, 93.25244 24.97428...	0.0
Arunachal Pradesh	POLYGON ((96.08538 29.45928, 96.09262 29.45613...	0.0
Nagaland	POLYGON ((95.19272 27.02710, 95.19475 27.01442...	0.0
Manipur	POLYGON ((94.57415 25.69043, 94.57852 25.68761...	0.0
Mizoram	POLYGON ((92.76384 24.52098, 92.76422 24.51967...	0.0
Tripura	POLYGON ((92.16949 24.53175, 92.16961 24.53168...	0.0
Meghalaya	POLYGON ((91.82617 26.11925, 91.82728 26.11914...	0.0
Bihar	POLYGON ((84.10880 27.52173, 84.10896 27.52131...	0.0
Ladakh	POLYGON ((76.80933 33.24349, 76.79356 33.25175...	0.0
Jammu and Kashmir	POLYGON ((76.80933 33.24349, 76.80894 33.24367...	0.0
Odisha	POLYGON ((87.47639 21.64343, 87.47362 21.63205...	0.0

In [130…
```
merged_nuclear_1.iloc[16,1] = 0.0
```

In [131…
```
fig ,ax = plt.subplots(1,figsize=(12,12))
ax.axis('off')
ax.set_title("NUCLEAR POWER SENARIO STATE WISE - YEAR 2019",fontdict={'fontsize':'
fig = merged_nuclear_1.plot(column='nuclear' , cmap='plasma',linewidth=0.5,ax=ax,e
```



In [132…

```
merged_nuclear_2 = shp_gdf.set_index('Name').join(nuclear2.set_index('state'))
merged_nuclear_2.iloc[33,1] = 0 # MAking Ladake same as J&K since one before 2020
merged_nuclear_2.iloc[3,1] = 0
merged_nuclear_2.iloc[16,1] = 0
merged_nuclear_2
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[132]:

	geometry	nuclear
Name		
West Bengal	MULTIPOLYGON (((88.01861 21.57278, 88.01889 21...	0.0
Andaman & Nicobar Islands	MULTIPOLYGON (((92.90124 12.91071, 92.90157 12...	0.0
Chandigarh	POLYGON ((76.77232 30.79420, 76.77286 30.79343...	0.0
Daman and Diu and Dadra and Nagar Haveli	MULTIPOLYGON (((72.96339 20.33227, 72.96390 20...	0.0
Delhi	POLYGON ((77.10591 28.87005, 77.10668 28.86989...	0.0
Haryana	POLYGON ((76.57526 30.10063, 76.57645 30.10152...	0.0
Jharkhand	POLYGON ((87.69613 24.16027, 87.69625 24.15974...	0.0
Karnataka	MULTIPOLYGON (((77.33232 18.45086, 77.33267 18...	880.0
Kerala	POLYGON ((74.98896 12.79553, 74.98953 12.79554...	0.0
Lakshadweep	MULTIPOLYGON (((73.08025 8.32651, 73.08029 8.3...	0.0
Madhya Pradesh	POLYGON ((78.37211 26.86406, 78.37423 26.86091...	0.0
Maharashtra	MULTIPOLYGON (((74.38926 22.03241, 74.38942 22...	1400.0
Puducherry	MULTIPOLYGON (((79.75669 11.00258, 79.75682 11...	0.0
Tamil Nadu	MULTIPOLYGON (((80.30358 13.47307, 80.30577 13...	2440.0
Chhatisgarh	POLYGON ((83.32707 24.10232, 83.32939 24.10130...	0.0
Telangana	POLYGON ((78.33565 19.88358, 78.33688 19.88300...	0.0
Andhra Pradesh	POLYGON ((84.67571 19.16721, 84.67725 19.16670...	0.0
Goa	POLYGON ((73.87042 15.78117, 73.87097 15.78029...	0.0
Himachal Pradesh	POLYGON ((76.79412 33.25569, 76.79482 33.25557...	0.0
Punjab	POLYGON ((75.83876 32.51269, 75.84094 32.51158...	0.0
Rajasthan	POLYGON ((73.97266 30.19800, 73.97266 30.19800...	1180.0
Gujarat	POLYGON ((72.26126 21.49999, 72.25911 21.49999...	440.0

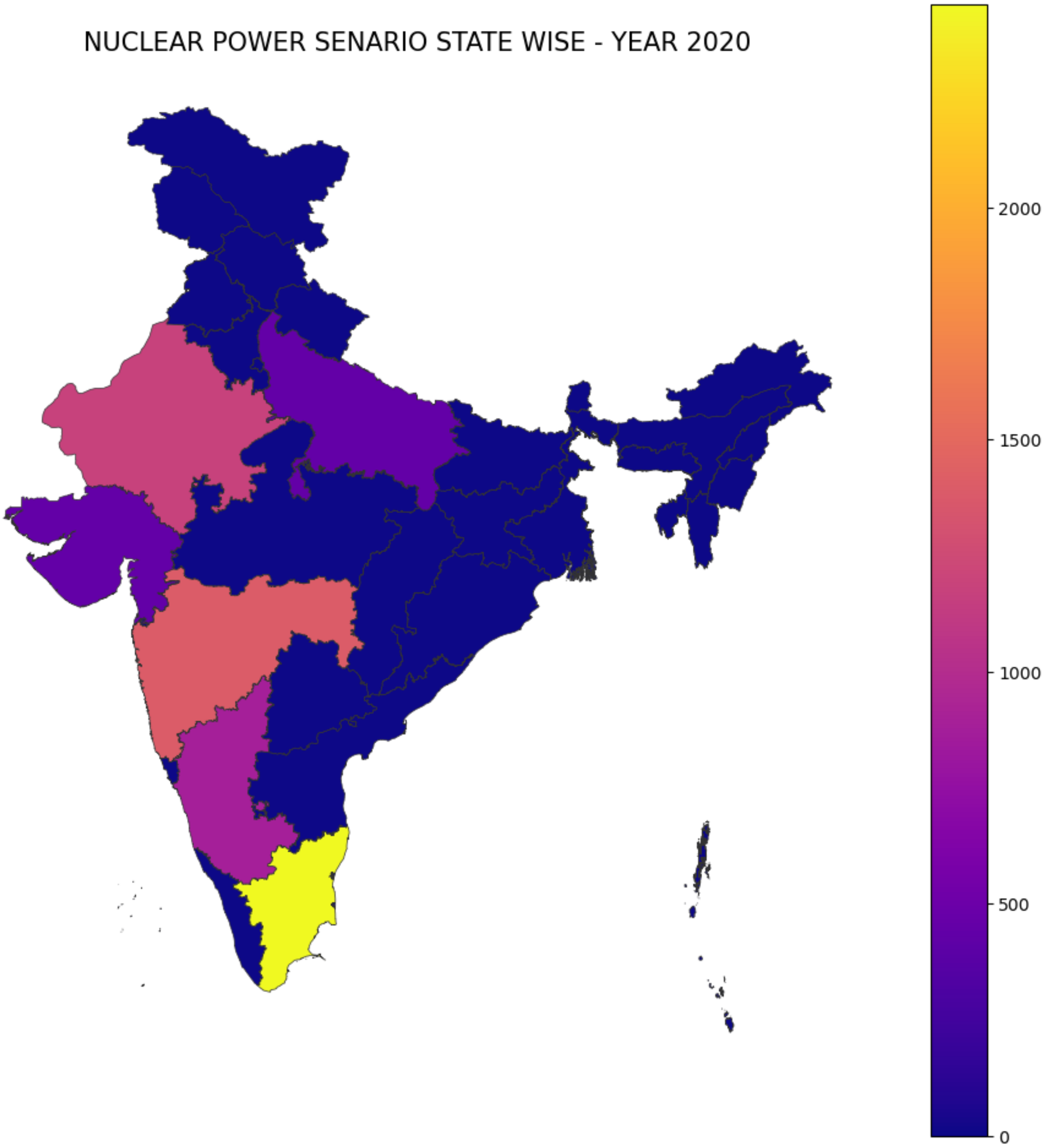
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

	geometry	nuclear
Name		
Uttarakhand	POLYGON ((79.06793 31.46153, 79.07046 31.46010...	0.0
Uttar Pradesh	POLYGON ((77.59189 30.40632, 77.59337 30.40621...	440.0
Sikkim	POLYGON ((88.61635 28.12763, 88.62302 28.12625...	0.0
Assam	POLYGON ((93.24961 24.97312, 93.25244 24.97428...	0.0
Arunachal Pradesh	POLYGON ((96.08538 29.45928, 96.09262 29.45613...	0.0
Nagaland	POLYGON ((95.19272 27.02710, 95.19475 27.01442...	0.0
Manipur	POLYGON ((94.57415 25.69043, 94.57852 25.68761...	0.0
Mizoram	POLYGON ((92.76384 24.52098, 92.76422 24.51967...	0.0
Tripura	POLYGON ((92.16949 24.53175, 92.16961 24.53168...	0.0
Meghalaya	POLYGON ((91.82617 26.11925, 91.82728 26.11914...	0.0
Bihar	POLYGON ((84.10880 27.52173, 84.10896 27.52131...	0.0
Ladakh	POLYGON ((76.80933 33.24349, 76.79356 33.25175...	0.0
Jammu and Kashmir	POLYGON ((76.80933 33.24349, 76.80894 33.24367...	0.0
Odisha	POLYGON ((87.47639 21.64343, 87.47362 21.63205...	0.0

In [133...

```
fig ,ax = plt.subplots(1,figsize=(12,12))
ax.axis('off')
ax.set_title("NUCLEAR POWER SENARIO STATE WISE - YEAR 2020",fontdict={'fontsize':'
fig = merged_nuclear_2.plot(column='nuclear' , cmap='plasma',linewidth=0.5,ax=ax,e
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [135…

```
merged_nuclear_3 = shp_gdf.set_index('Name').join(nuclear3.set_index('state'))
merged_nuclear_3.iloc[33,1] = 0 # MAking Ladake same as J&K since one before 2020
merged_nuclear_3.iloc[3,1] = 0
merged_nuclear_3.iloc[16,1] = 0
merged_nuclear_3
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[135]:

Name	geometry	nuclear
West Bengal	MULTIPOLYGON (((88.01861 21.57278, 88.01889 21...	0.0
Andaman & Nicobar Islands	MULTIPOLYGON (((92.90124 12.91071, 92.90157 12...	0.0
Chandigarh	POLYGON ((76.77232 30.79420, 76.77286 30.79343...	0.0
Daman and Diu and Dadra and Nagar Haveli	MULTIPOLYGON (((72.96339 20.33227, 72.96390 20...	0.0
Delhi	POLYGON ((77.10591 28.87005, 77.10668 28.86989...	0.0
Haryana	POLYGON ((76.57526 30.10063, 76.57645 30.10152...	0.0
Jharkhand	POLYGON ((87.69613 24.16027, 87.69625 24.15974...	0.0
Karnataka	MULTIPOLYGON (((77.33232 18.45086, 77.33267 18...	880.0
Kerala	POLYGON ((74.98896 12.79553, 74.98953 12.79554...	0.0
Lakshadweep	MULTIPOLYGON (((73.08025 8.32651, 73.08029 8.3...	0.0
Madhya Pradesh	POLYGON ((78.37211 26.86406, 78.37423 26.86091...	0.0
Maharashtra	MULTIPOLYGON (((74.38926 22.03241, 74.38942 22...	1400.0
Puducherry	MULTIPOLYGON (((79.75669 11.00258, 79.75682 11...	0.0
Tamil Nadu	MULTIPOLYGON (((80.30358 13.47307, 80.30577 13...	2440.0
Chhatisgarh	POLYGON ((83.32707 24.10232, 83.32939 24.10130...	0.0
Telangana	POLYGON ((78.33565 19.88358, 78.33688 19.88300...	0.0
Andhra Pradesh	POLYGON ((84.67571 19.16721, 84.67725 19.16670...	0.0
Goa	POLYGON ((73.87042 15.78117, 73.87097 15.78029...	0.0
Himachal Pradesh	POLYGON ((76.79412 33.25569, 76.79482 33.25557...	0.0
Punjab	POLYGON ((75.83876 32.51269, 75.84094 32.51158...	0.0
Rajasthan	POLYGON ((73.97266 30.19800, 73.97266 30.19800...	1180.0
Gujarat	POLYGON ((72.26126 21.49999, 72.25911 21.49999...	440.0

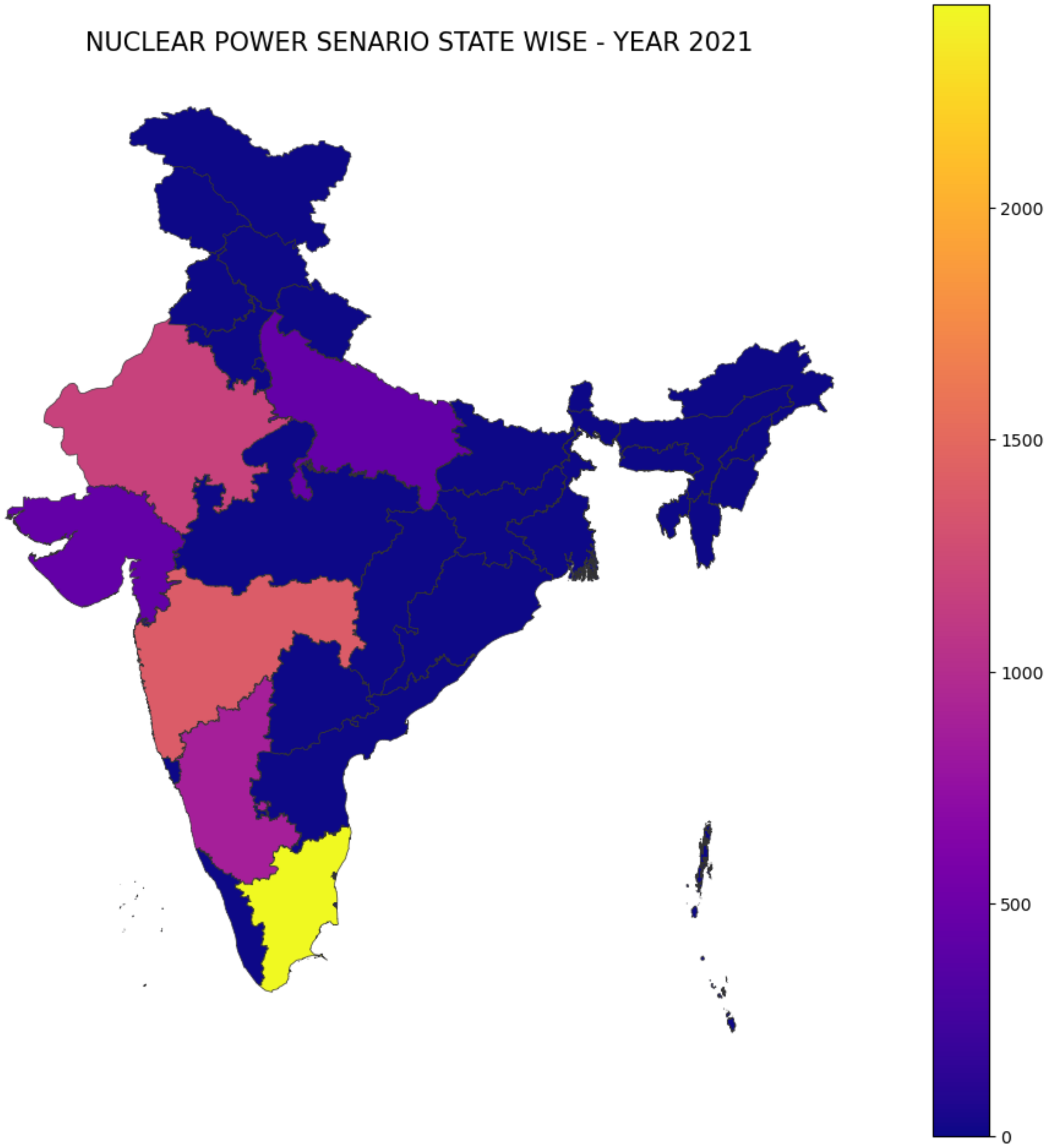
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

	geometry	nuclear
Name		
Uttarakhand	POLYGON ((79.06793 31.46153, 79.07046 31.46010...	0.0
Uttar Pradesh	POLYGON ((77.59189 30.40632, 77.59337 30.40621...	440.0
Sikkim	POLYGON ((88.61635 28.12763, 88.62302 28.12625...	0.0
Assam	POLYGON ((93.24961 24.97312, 93.25244 24.97428...	0.0
Arunachal Pradesh	POLYGON ((96.08538 29.45928, 96.09262 29.45613...	0.0
Nagaland	POLYGON ((95.19272 27.02710, 95.19475 27.01442...	0.0
Manipur	POLYGON ((94.57415 25.69043, 94.57852 25.68761...	0.0
Mizoram	POLYGON ((92.76384 24.52098, 92.76422 24.51967...	0.0
Tripura	POLYGON ((92.16949 24.53175, 92.16961 24.53168...	0.0
Meghalaya	POLYGON ((91.82617 26.11925, 91.82728 26.11914...	0.0
Bihar	POLYGON ((84.10880 27.52173, 84.10896 27.52131...	0.0
Ladakh	POLYGON ((76.80933 33.24349, 76.79356 33.25175...	0.0
Jammu and Kashmir	POLYGON ((76.80933 33.24349, 76.80894 33.24367...	0.0
Odisha	POLYGON ((87.47639 21.64343, 87.47362 21.63205...	0.0

In [136...

```
fig ,ax = plt.subplots(1,figsize=(12,12))
ax.axis('off')
ax.set_title("NUCLEAR POWER SENARIO STATE WISE - YEAR 2021",fontdict={'fontsize':'
fig = merged_nuclear_3.plot(column='nuclear' , cmap='plasma',linewidth=0.5,ax=ax,e
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [137…

```
merged_coal_1 = shp_gdf.set_index('Name').join(coal1.set_index('state'))
merged_coal_1.iloc[33,1] = 0 # MAking Ladake same as J&K since one before 2020
merged_coal_1.iloc[3,1] = 0
```

In [139…

```
coal1
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[139]:

	state	coal
396	Chandigarh	0.0
397	Delhi	0.0
398	Haryana	5540.0
399	Himachal Pradesh	0.0
400	Jammu and Kashmir	0.0
401	Punjab	5680.0
402	Rajasthan	9160.0
403	Uttar Pradesh	22409.0
404	Uttarakhand	0.0
405	Andaman & Nicobar Islands	0.0
406	Bihar	6390.0
407	Jharkhand	4590.0
408	Odisha	9800.0
409	Sikkim	0.0
410	West Bengal	14287.0
411	Chhatisgarh	23128.0
412	Dadra & Nagar Haveli	0.0
413	Daman & Diu	0.0
414	Goa	0.0
415	Gujarat	16232.0
416	Madhya Pradesh	20490.0
417	Maharashtra	25386.0
418	Andhra Pradesh.	11590.0
419	Karnataka	9480.0
420	Kerala	0.0
421	Lakshadweep	0.0
422	Puducherry	0.0
423	Tamil Nadu	13160.0
424	Telangana	7182.5
425	Arunachal Pradesh	0.0
426	Assam	750.0
427	Manipur	0.0
428	Meghalaya	0.0
429	Mizoram	0.0
430	Nagaland	0.0

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [138…

```
merged_coal_1
```

Out[138]:

	geometry	coal
Name		
West Bengal	MULTIPOLYGON (((88.01861 21.57278, 88.01889 21...	14287.0
Andaman & Nicobar Islands	MULTIPOLYGON (((92.90124 12.91071, 92.90157 12...	0.0
Chandigarh	POLYGON ((76.77232 30.79420, 76.77286 30.79343...	0.0
Daman and Diu and Dadra and Nagar Haveli	MULTIPOLYGON (((72.96339 20.33227, 72.96390 20...	0.0
Delhi	POLYGON ((77.10591 28.87005, 77.10668 28.86989...	0.0
Haryana	POLYGON ((76.57526 30.10063, 76.57645 30.10152...	5540.0
Jharkhand	POLYGON ((87.69613 24.16027, 87.69625 24.15974...	4590.0
Karnataka	MULTIPOLYGON (((77.33232 18.45086, 77.33267 18...	9480.0
Kerala	POLYGON ((74.98896 12.79553, 74.98953 12.79554...	0.0
Lakshadweep	MULTIPOLYGON (((73.08025 8.32651, 73.08029 8.3...	0.0
Madhya Pradesh	POLYGON ((78.37211 26.86406, 78.37423 26.86091...	20490.0
Maharashtra	MULTIPOLYGON (((74.38926 22.03241, 74.38942 22...	25386.0
Puducherry	MULTIPOLYGON (((79.75669 11.00258, 79.75682 11...	0.0
Tamil Nadu	MULTIPOLYGON (((80.30358 13.47307, 80.30577 13...	13160.0
Chhatisgarh	POLYGON ((83.32707 24.10232, 83.32939 24.10130...	23128.0
Telangana	POLYGON ((78.33565 19.88358, 78.33688 19.88300...	7182.5
Andhra Pradesh	POLYGON ((84.67571 19.16721, 84.67725 19.16670...	NaN
Goa	POLYGON ((73.87042 15.78117, 73.87097 15.78029...	0.0
Himachal Pradesh	POLYGON ((76.79412 33.25569, 76.79482 33.25557...	0.0
Punjab	POLYGON ((75.83876 32.51269, 75.84094 32.51158...	5680.0
Rajasthan	POLYGON ((73.97266 30.19800, 73.97266 30.19800...	9160.0
Gujarat	POLYGON ((72.26126 21.49999, 72.25911 21.49999...	16232.0

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

	geometry	coal
Name		
Uttarakhand	POLYGON ((79.06793 31.46153, 79.07046 31.46010...	0.0
Uttar Pradesh	POLYGON ((77.59189 30.40632, 77.59337 30.40621...	22409.0
Sikkim	POLYGON ((88.61635 28.12763, 88.62302 28.12625...	0.0
Assam	POLYGON ((93.24961 24.97312, 93.25244 24.97428...	750.0
Arunachal Pradesh	POLYGON ((96.08538 29.45928, 96.09262 29.45613...	0.0
Nagaland	POLYGON ((95.19272 27.02710, 95.19475 27.01442...	0.0
Manipur	POLYGON ((94.57415 25.69043, 94.57852 25.68761...	0.0
Mizoram	POLYGON ((92.76384 24.52098, 92.76422 24.51967...	0.0
Tripura	POLYGON ((92.16949 24.53175, 92.16961 24.53168...	0.0
Meghalaya	POLYGON ((91.82617 26.11925, 91.82728 26.11914...	0.0
Bihar	POLYGON ((84.10880 27.52173, 84.10896 27.52131...	6390.0
Ladakh	POLYGON ((76.80933 33.24349, 76.79356 33.25175...	0.0
Jammu and Kashmir	POLYGON ((76.80933 33.24349, 76.80894 33.24367...	0.0
Odisha	POLYGON ((87.47639 21.64343, 87.47362 21.63205...	9800.0

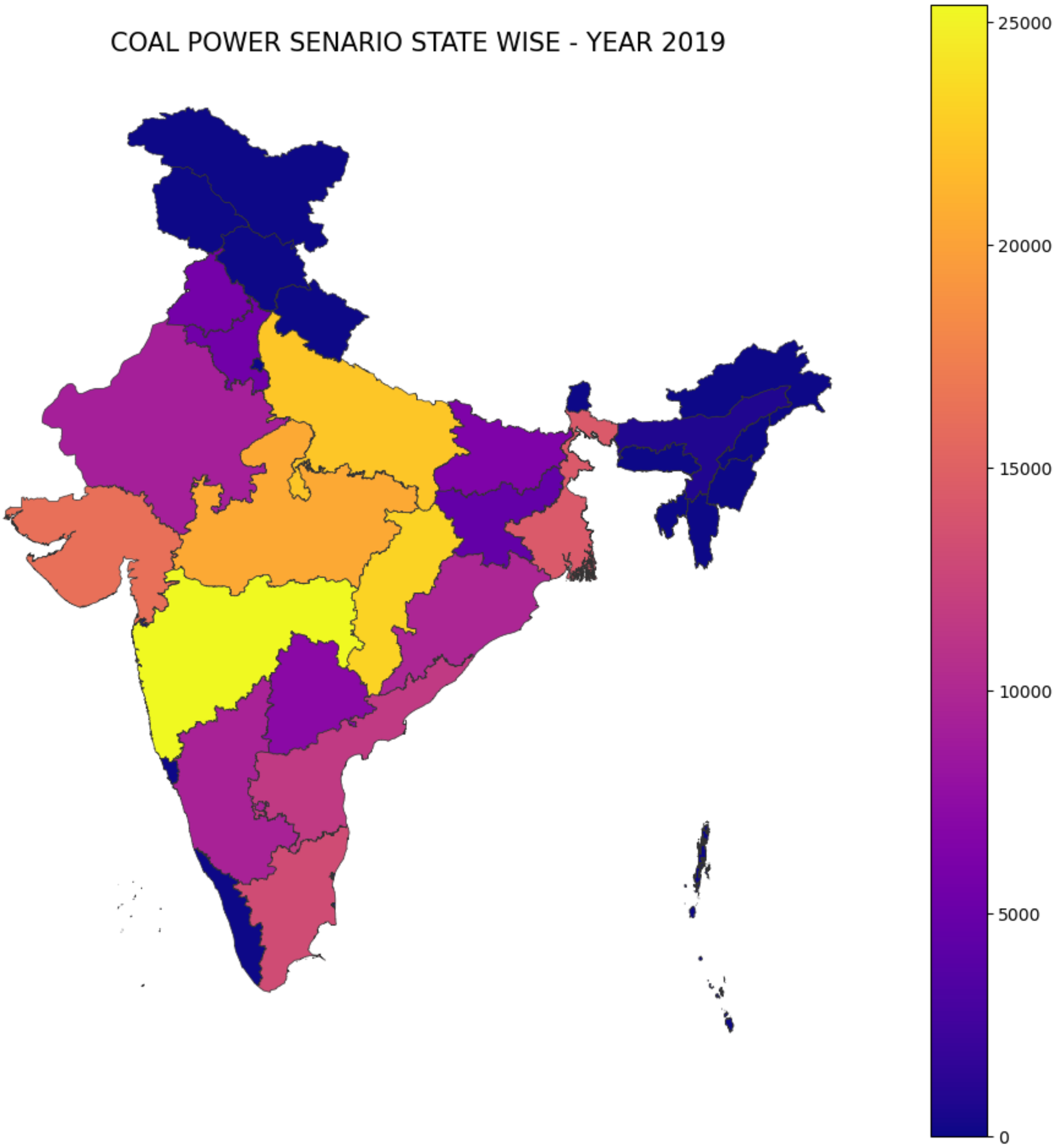
In [141...

```
merged_coal_1.iloc[16,1] = 11590
```

In [142...

```
fig ,ax = plt.subplots(1,figsize=(12,12))
ax.axis('off')
ax.set_title("COAL POWER SENARIO STATE WISE - YEAR 2019",fontdict={'fontsize':'15'
fig = merged_coal_1.plot(column='coal' , cmap='plasma',linewidth=0.5,ax=ax,edgecol
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



COAL POWER SENARIO STATE WISE - YEAR 2019

In [143…

```
merged_coal_2 = shp_gdf.set_index('Name').join(coal2.set_index('state'))
# merged_coal_2.iloc[33,1] = 0 # MAking Ladake same as J&K since one before 2020
merged_coal_2.iloc[3,1] = 0
merged_coal_2.iloc[16,1] = 11590
```

In [144…

```
merged_coal_2
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[144]:

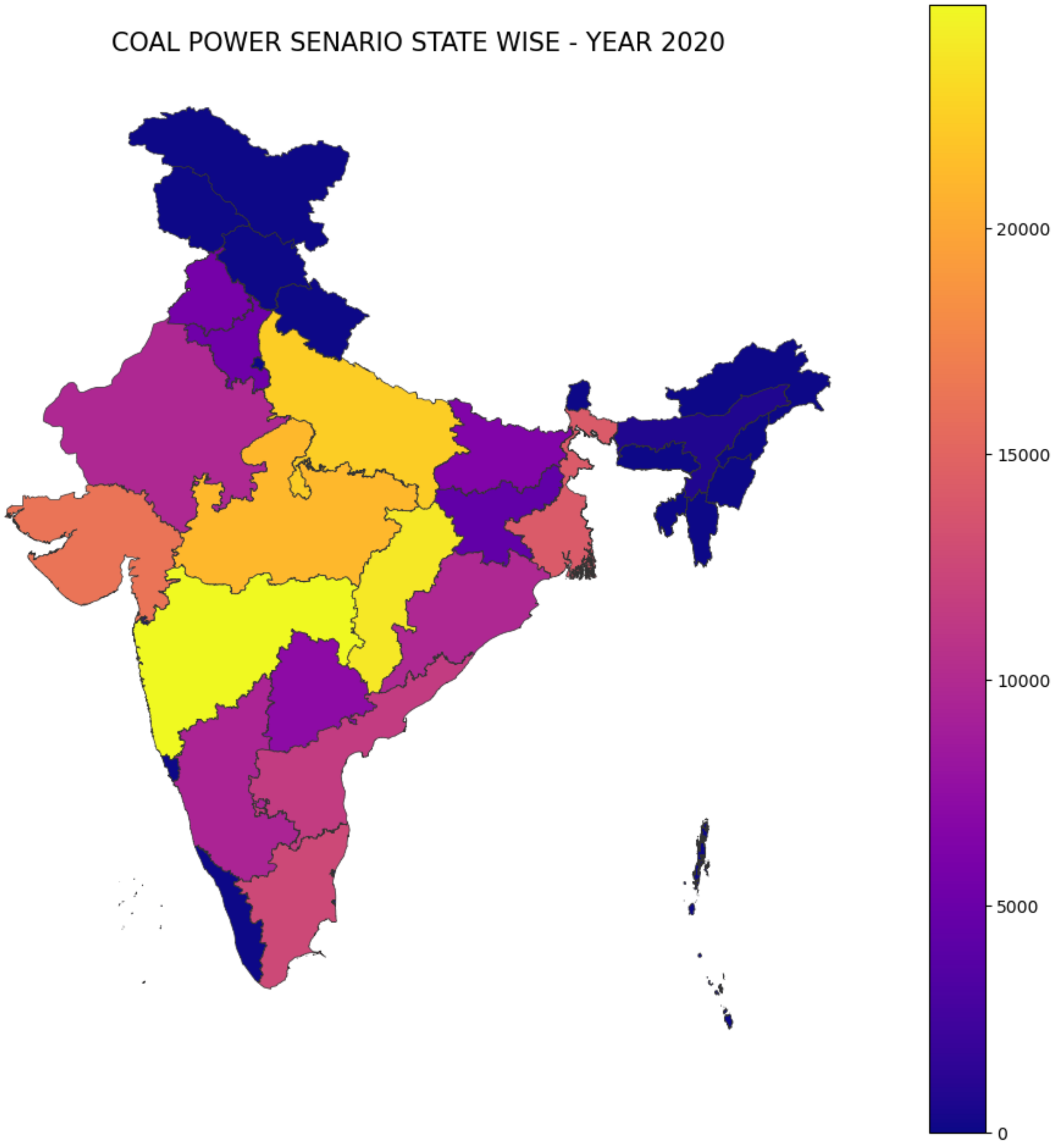
	geometry	coal
Name		
West Bengal	MULTIPOLYGON (((88.01861 21.57278, 88.01889 21...	14177.0
Andaman & Nicobar Islands	MULTIPOLYGON (((92.90124 12.91071, 92.90157 12...	0.0
Chandigarh	POLYGON ((76.77232 30.79420, 76.77286 30.79343...	0.0
Daman and Diu and Dadra and Nagar Haveli	MULTIPOLYGON (((72.96339 20.33227, 72.96390 20...	0.0
Delhi	POLYGON ((77.10591 28.87005, 77.10668 28.86989...	0.0
Haryana	POLYGON ((76.57526 30.10063, 76.57645 30.10152...	5330.0
Jharkhand	POLYGON ((87.69613 24.16027, 87.69625 24.15974...	4460.0
Karnataka	MULTIPOLYGON (((77.33232 18.45086, 77.33267 18...	9480.0
Kerala	POLYGON ((74.98896 12.79553, 74.98953 12.79554...	0.0
Lakshadweep	MULTIPOLYGON (((73.08025 8.32651, 73.08029 8.3...	0.0
Madhya Pradesh	POLYGON ((78.37211 26.86406, 78.37423 26.86091...	21150.0
Maharashtra	MULTIPOLYGON (((74.38926 22.03241, 74.38942 22...	24966.0
Puducherry	MULTIPOLYGON (((79.75669 11.00258, 79.75682 11...	0.0
Tamil Nadu	MULTIPOLYGON (((80.30358 13.47307, 80.30577 13...	12660.0
Chhatisgarh	POLYGON ((83.32707 24.10232, 83.32939 24.10130...	23928.0
Telangana	POLYGON ((78.33565 19.88358, 78.33688 19.88300...	7302.5
Andhra Pradesh	POLYGON ((84.67571 19.16721, 84.67725 19.16670...	11590.0
Goa	POLYGON ((73.87042 15.78117, 73.87097 15.78029...	0.0
Himachal Pradesh	POLYGON ((76.79412 33.25569, 76.79482 33.25557...	0.0
Punjab	POLYGON ((75.83876 32.51269, 75.84094 32.51158...	5680.0
Rajasthan	POLYGON ((73.97266 30.19800, 73.97266 30.19800...	9820.0
Gujarat	POLYGON ((72.26126 21.49999, 72.25911 21.49999...	16232.0

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

	geometry	coal
Name		
Uttarakhand	POLYGON ((79.06793 31.46153, 79.07046 31.46010...	0.0
Uttar Pradesh	POLYGON ((77.59189 30.40632, 77.59337 30.40621...	22409.0
Sikkim	POLYGON ((88.61635 28.12763, 88.62302 28.12625...	0.0
Assam	POLYGON ((93.24961 24.97312, 93.25244 24.97428...	750.0
Arunachal Pradesh	POLYGON ((96.08538 29.45928, 96.09262 29.45613...	0.0
Nagaland	POLYGON ((95.19272 27.02710, 95.19475 27.01442...	0.0
Manipur	POLYGON ((94.57415 25.69043, 94.57852 25.68761...	0.0
Mizoram	POLYGON ((92.76384 24.52098, 92.76422 24.51967...	0.0
Tripura	POLYGON ((92.16949 24.53175, 92.16961 24.53168...	0.0
Meghalaya	POLYGON ((91.82617 26.11925, 91.82728 26.11914...	0.0
Bihar	POLYGON ((84.10880 27.52173, 84.10896 27.52131...	6390.0
Ladakh	POLYGON ((76.80933 33.24349, 76.79356 33.25175...	0.0
Jammu and Kashmir	POLYGON ((76.80933 33.24349, 76.80894 33.24367...	0.0
Odisha	POLYGON ((87.47639 21.64343, 87.47362 21.63205...	9800.0

In [145…

```
fig ,ax = plt.subplots(1,figsize=(12,12))
ax.axis('off')
ax.set_title("COAL POWER SENARIO STATE WISE - YEAR 2020",fontdict={'fontsize':'15'
fig = merged_coal_2.plot(column='coal' , cmap='plasma',linewidth=0.5,ax=ax,edgecol
```



In [146…

```
merged_coal_3 = shp_gdf.set_index('Name').join(coal3.set_index('state'))
# merged_coal_2.iloc[33,1] = 0 # MAking Ladake same as J&K since one before 2020
merged_coal_3.iloc[3,1] = 0
merged_coal_3.iloc[16,1] = 11590
```

In [147…

```
merged_coal_3
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[147]:

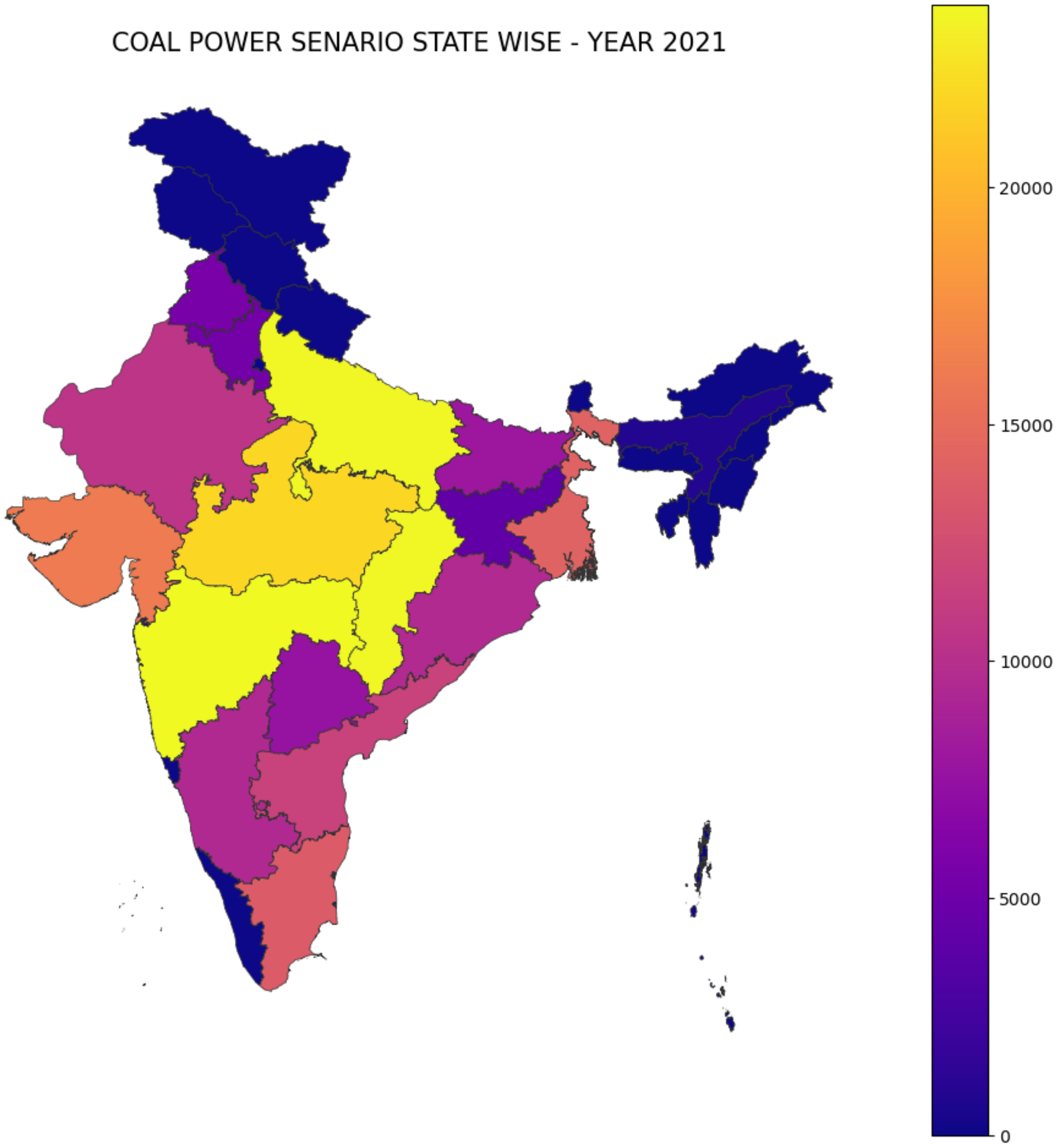
	geometry	coal
Name		
West Bengal	MULTIPOLYGON (((88.01861 21.57278, 88.01889 21...	14177.0
Andaman & Nicobar Islands	MULTIPOLYGON (((92.90124 12.91071, 92.90157 12...	0.0
Chandigarh	POLYGON ((76.77232 30.79420, 76.77286 30.79343...	0.0
Daman and Diu and Dadra and Nagar Haveli	MULTIPOLYGON (((72.96339 20.33227, 72.96390 20...	0.0
Delhi	POLYGON ((77.10591 28.87005, 77.10668 28.86989...	0.0
Haryana	POLYGON ((76.57526 30.10063, 76.57645 30.10152...	5330.0
Jharkhand	POLYGON ((87.69613 24.16027, 87.69625 24.15974...	4250.0
Karnataka	MULTIPOLYGON (((77.33232 18.45086, 77.33267 18...	9480.0
Kerala	POLYGON ((74.98896 12.79553, 74.98953 12.79554...	0.0
Lakshadweep	MULTIPOLYGON (((73.08025 8.32651, 73.08029 8.3...	0.0
Madhya Pradesh	POLYGON ((78.37211 26.86406, 78.37423 26.86091...	21950.0
Maharashtra	MULTIPOLYGON (((74.38926 22.03241, 74.38942 22...	23856.0
Puducherry	MULTIPOLYGON (((79.75669 11.00258, 79.75682 11...	0.0
Tamil Nadu	MULTIPOLYGON (((80.30358 13.47307, 80.30577 13...	13685.0
Chhatisgarh	POLYGON ((83.32707 24.10232, 83.32939 24.10130...	23688.0
Telangana	POLYGON ((78.33565 19.88358, 78.33688 19.88300...	7572.5
Andhra Pradesh	POLYGON ((84.67571 19.16721, 84.67725 19.16670...	11590.0
Goa	POLYGON ((73.87042 15.78117, 73.87097 15.78029...	0.0
Himachal Pradesh	POLYGON ((76.79412 33.25569, 76.79482 33.25557...	0.0
Punjab	POLYGON ((75.83876 32.51269, 75.84094 32.51158...	5680.0
Rajasthan	POLYGON ((73.97266 30.19800, 73.97266 30.19800...	10480.0
Gujarat	POLYGON ((72.26126 21.49999, 72.25911 21.49999...	16092.0

	geometry	coal
Name		
Uttarakhand	POLYGON ((79.06793 31.46153, 79.07046 31.46010...	0.0
Uttar Pradesh	POLYGON ((77.59189 30.40632, 77.59337 30.40621...	23729.0
Sikkim	POLYGON ((88.61635 28.12763, 88.62302 28.12625...	0.0
Assam	POLYGON ((93.24961 24.97312, 93.25244 24.97428...	750.0
Arunachal Pradesh	POLYGON ((96.08538 29.45928, 96.09262 29.45613...	0.0
Nagaland	POLYGON ((95.19272 27.02710, 95.19475 27.01442...	0.0
Manipur	POLYGON ((94.57415 25.69043, 94.57852 25.68761...	0.0
Mizoram	POLYGON ((92.76384 24.52098, 92.76422 24.51967...	0.0
Tripura	POLYGON ((92.16949 24.53175, 92.16961 24.53168...	0.0
Meghalaya	POLYGON ((91.82617 26.11925, 91.82728 26.11914...	0.0
Bihar	POLYGON ((84.10880 27.52173, 84.10896 27.52131...	7960.0
Ladakh	POLYGON ((76.80933 33.24349, 76.79356 33.25175...	0.0
Jammu and Kashmir	POLYGON ((76.80933 33.24349, 76.80894 33.24367...	0.0
Odisha	POLYGON ((87.47639 21.64343, 87.47362 21.63205...	9540.0

In [148…

```
fig ,ax = plt.subplots(1,figsize=(12,12))
ax.axis('off')
ax.set_title("COAL POWER SENARIO STATE WISE - YEAR 2021",fontdict={'fontsize':'15'
fig = merged_coal_3.plot(column='coal' , cmap='plasma',linewidth=0.5,ax=ax,edgecol
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



```
In [149…  df1.head()
```

Out[149]:

	index	Date	Region	Thermal Generation Actual (in MU)	Thermal Generation Estimated (in MU)	Nuclear Generation Actual (in MU)	Nuclear Generation Estimated (in MU)	Hydro Generation Actual (in MU)	Gene Esti (i
0	0	2017-09-01	Northern	624.23	484.21	30.36	35.57	273.27	
1	1	2017-09-01	Western	1,106.89	1,024.33	25.17	3.81	72.00	
2	2	2017-09-01	Southern	576.66	578.55	62.73	49.80	111.57	
3	3	2017-09-01	Eastern	441.02	429.39	NaN	NaN	85.94	
4	4	2017-09-01	NorthEastern	29.11	15.91	NaN	NaN	24.64	

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[151]:
```
str
```

In [152…
```
df1["Thermal Generation Actual (in MU)"] = df1["Thermal Generation Actual (in MU)"
df1["Thermal Generation Actual (in MU)"] = df1["Thermal Generation Actual (in MU)"
df1["Thermal Generation Estimated (in MU)"] = df1["Thermal Generation Estimated (i
df1["Thermal Generation Estimated (in MU)"] = df1["Thermal Generation Estimated (i
```

In [154…
```
input_format = '%Y/%m/%d'
output_format = '%Y/%B/%d'

# Convert date column to datetime format
df1['Date'] = pd.to_datetime(df1['Date'], format=input_format)

# Define lambda function to format dates
format_date = lambda date: datetime.datetime.strftime(date, output_format)

# Apply lambda function to date column and store as new column
df1['Formatted Date'] = df1['Date'].apply(format_date)

# Display DataFrame with formatted dates
df1.fillna(0,inplace=True)
df1["date"] = pd.to_datetime(df1["Formatted Date"])
```

In [155…
```
df_monthly_mean = df1.groupby(pd.Grouper(key='date', freq='M'))['Thermal Generatio
```

In [156…
```
year_2017 = df1.iloc[0:605,:]
year_2018 = df1.iloc[605:2425,:]
year_2019 = df1.iloc[2426:4250,:]
```

In [157…
```
print(type(year_2018.iloc[1,1]))
```

```
<class 'pandas._libs.tslibs.timestamps.Timestamp'>
```

In [158…
```
year_2018.head()
```

Out[158]:

	index	Date	Region	Thermal Generation Actual (in MU)	Thermal Generation Estimated (in MU)	Nuclear Generation Actual (in MU)	Nuclear Generation Estimated (in MU)	Hydro Generation Actual (in MU)	Ge Es
605	605	2018-01-01	Northern	626.60	625.16	30.31	32.09	96.18	
606	606	2018-01-01	Western	1127.88	1152.31	25.19	26.14	41.22	
607	607	2018-01-01	Southern	638.77	556.72	45.38	72.80	70.04	
608	608	2018-01-01	Eastern	478.58	435.94	0.00	0.00	21.76	
609	609	2018-01-01	NorthEastern	31.88	35.76	0.00	0.00	11.28	

In [159…
```
northern_year_2018 = year_2018[year_2018["Region"]=="Northern"]
eastern_year_2018 = year_2018[year_2018["Region"]=="Eastern"]
western_year_2018 = year_2018[year_2018["Region"]=="Western"]
southern_year_2018 = year_2018[year_2018["Region"]=="Southern"]
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js year_2018["Region"]=="NorthEastern"]
```

In [160…

```
type(northern_year_2018.iloc[1,1])
```

Out[160]:

```
pandas._libs.tslibs.timestamps.Timestamp
```

In [161…

```
df_monthly_mean_north = northern_year_2018.groupby(pd.Grouper(key='date', freq='M'
df_monthly_mean_east = eastern_year_2018.groupby(pd.Grouper(key='date', freq='M'))
df_monthly_mean_west = western_year_2018.groupby(pd.Grouper(key='date', freq='M'))
df_monthly_mean_south = southern_year_2018.groupby(pd.Grouper(key='date', freq='M'
df_monthly_mean_north_east = north_eastern_year_2018.groupby(pd.Grouper(key='date'
```

In [162…

```
l = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'Se
```

In [163…

```
df_monthly_mean_east["Month"] =l
df_monthly_mean_north["Month"] = l
df_monthly_mean_south["Month"] =l
df_monthly_mean_west["Month"] = l
df_monthly_mean_north_east["Month"] = l
df_monthly_mean_east["Region"] = ["Eastern"]*12
df_monthly_mean_north["Region"] = ["Northern"]*12
df_monthly_mean_south["Region"] = ["Southern"]*12
df_monthly_mean_west["Region"] = ["Western"]*12
df_monthly_mean_north_east["Region"] = ["NorthEastern"]*12
```

In [164…

```
df_monthly_mean_east
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[164]:

	Thermal Generation Actual (in MU)	Thermal Generation Estimated (in MU)	Nuclear Generation Actual (in MU)	Nuclear Generation Estimated (in MU)	Hydro Generation Actual (in MU)	Hydro Generation Estimated (in MU)	Month	Regi
date								
2018-01-31	485.579032	472.515484	0.0	0.0	21.76	22.392903	January	Easte
2018-02-28	489.607143	510.174286	0.0	0.0	25.10	24.020000	February	Easte
2018-03-31	502.462581	529.025806	0.0	0.0	30.16	27.659032	March	Easte
2018-04-30	464.347333	521.312000	0.0	0.0	39.12	36.087333	April	Easte
2018-05-31	471.855161	504.389677	0.0	0.0	46.26	55.939677	May	Easte
2018-06-30	469.443000	481.531667	0.0	0.0	53.41	67.852000	June	Easte
2018-07-31	448.081613	457.559355	0.0	0.0	69.16	67.558387	July	Easte
2018-08-31	430.125667	438.584333	0.0	0.0	77.99	76.694667	August	Easte
2018-09-30	445.378667	422.011667	0.0	0.0	79.00	90.680667	September	Easte
2018-10-31	484.378387	453.946129	0.0	0.0	55.26	70.809032	October	Easte
2018-11-30	471.405000	469.264667	0.0	0.0	31.20	38.134667	November	Easte
2018-12-31	471.858387	463.579032	0.0	0.0	22.67	27.653548	December	Easte

In [165…

```
merged_pd = pd.concat([df_monthly_mean_east,df_monthly_mean_west,df_monthly_mean_n
```

In [166…

```
merged_pd
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[166]:

	Thermal Generation Actual (in MU)	Thermal Generation Estimated (in MU)	Nuclear Generation Actual (in MU)	Nuclear Generation Estimated (in MU)	Hydro Generation Actual (in MU)	Hydro Generation Estimated (in MU)	Month	R
0	485.579032	472.515484	0.000000	0.000000	21.76	22.392903	January	E
1	489.607143	510.174286	0.000000	0.000000	25.10	24.020000	February	E
2	502.462581	529.025806	0.000000	0.000000	30.16	27.659032	March	E
3	464.347333	521.312000	0.000000	0.000000	39.12	36.087333	April	E
4	471.855161	504.389677	0.000000	0.000000	46.26	55.939677	May	E
5	469.443000	481.531667	0.000000	0.000000	53.41	67.852000	June	E
6	448.081613	457.559355	0.000000	0.000000	69.16	67.558387	July	E
7	430.125667	438.584333	0.000000	0.000000	77.99	76.694667	August	E
8	445.378667	422.011667	0.000000	0.000000	79.00	90.680667	September	E
9	484.378387	453.946129	0.000000	0.000000	55.26	70.809032	October	E
10	471.405000	469.264667	0.000000	0.000000	31.20	38.134667	November	E
11	471.858387	463.579032	0.000000	0.000000	22.67	27.653548	December	E
12	1123.520968	1214.970000	25.190000	26.158387	41.22	22.694194	January	W
13	1162.053929	1193.355357	25.140000	22.900714	40.94	25.429643	February	W
14	1177.012258	1186.429355	25.190000	24.767742	38.10	23.382581	March	W
15	1228.323333	1253.982000	14.070000	32.651333	24.40	26.938667	April	W
16	1218.164194	1273.921935	14.060000	22.951613	23.41	34.360000	May	W
17	1151.192667	1181.464333	25.070000	30.828667	16.99	15.604667	June	W
18	1105.945161	1098.220645	14.060000	32.063871	21.91	17.472258	July	W
19	1134.134667	1159.384333	25.060000	30.083667	43.32	23.140667	August	W
20	1175.713000	1235.914333	14.070000	30.821000	62.03	47.125000	September	W
21	1246.668065	1338.792258	17.120000	31.742581	40.20	33.196452	October	W
22	1238.387667	1262.194667	28.140000	32.557000	40.29	21.204000	November	W
23	1196.792258	1222.675484	17.120000	26.332903	36.84	26.457097	December	W
24	623.482581	641.435484	30.310000	32.129032	96.18	102.880645	January	Nc
25	619.234286	614.679643	30.360000	31.182143	109.45	96.499643	February	Nc
26	627.294516	639.903226	30.320000	29.188710	134.13	101.991935	March	Nc
27	639.649667	612.104333	22.500000	28.579333	154.72	119.649667	April	Nc
28	639.459032	675.096129	26.595484	30.956452	244.89	193.890645	May	Nc
29	661.691333	663.859667	30.830000	32.167333	303.99	287.649667	June	Nc
30	644.303871	593.248710	30.720000	30.308065	314.44	322.609032	July	Nc
31	660.859333	613.143667	30.720000	29.753667	327.74	329.648333	August	Nc
32	664.302000	611.655667	26.330000	28.818000	257.18	300.917000	September	Nc
				29.861613	153.81	188.621935	October	Nc

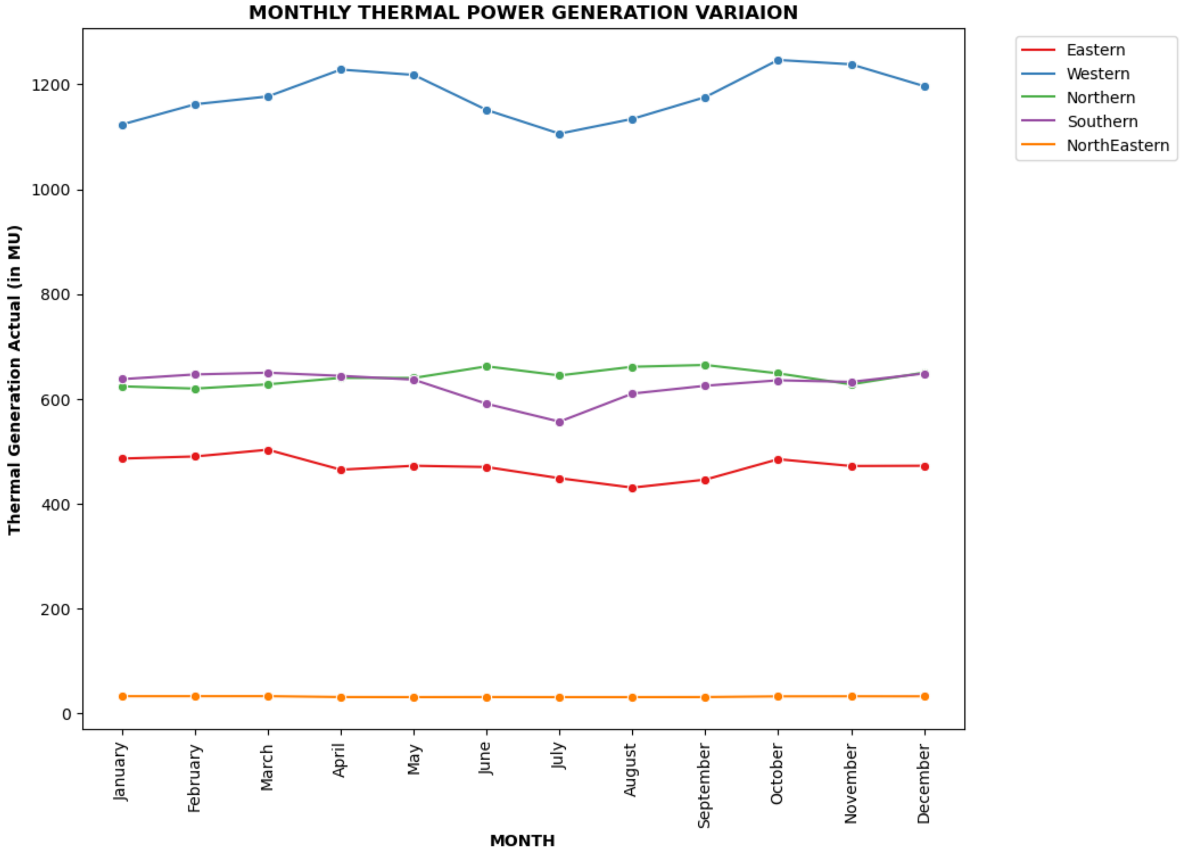
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

	Thermal Generation Actual (in MU)	Thermal Generation Estimated (in MU)	Nuclear Generation Actual (in MU)	Nuclear Generation Estimated (in MU)	Hydro Generation Actual (in MU)	Hydro Generation Estimated (in MU)	Month	R
34	627.259000	613.218333	28.325000	31.055667	115.79	140.842000	November	Nc
35	649.721613	616.280645	26.240000	32.522581	103.54	115.748065	December	Nc
36	637.244516	559.819677	45.380000	68.183226	70.04	57.599032	January	So
37	646.234643	598.123571	56.215000	62.762857	82.80	50.833571	February	So
38	649.520000	676.752903	62.860000	44.912903	88.22	66.717742	March	So
39	643.500667	669.546000	47.030000	45.982000	62.88	61.203333	April	So
40	636.293871	630.227742	46.927097	45.882581	52.96	44.595484	May	So
41	590.188333	507.905333	63.170000	47.378333	38.45	49.362333	June	So
42	556.064839	471.732903	40.090000	49.681613	62.52	77.256774	July	So
43	609.666667	449.226333	44.550000	26.656333	84.41	143.977333	August	So
44	624.611000	608.393000	46.830000	25.724667	95.33	116.524667	September	So
45	635.059355	607.532903	67.322581	36.237742	74.80	97.504516	October	So
46	631.950667	559.034333	62.900000	40.157333	59.93	70.307333	November	So
47	647.974839	559.082581	65.100000	36.414516	56.93	71.060000	December	So
48	31.880000	32.854194	0.000000	0.000000	11.28	7.965806	January	NorthE
49	31.930000	33.323929	0.000000	0.000000	10.24	6.670357	February	NorthE
50	31.960000	32.709032	0.000000	0.000000	12.16	6.376452	March	NorthE
51	30.250000	30.495667	0.000000	0.000000	7.43	6.865333	April	NorthE
52	30.130000	31.796129	0.000000	0.000000	13.75	14.339355	May	NorthE
53	30.250000	30.566667	0.000000	0.000000	21.71	21.344000	June	NorthE
54	30.100000	32.810968	0.000000	0.000000	30.46	24.765806	July	NorthE
55	30.100000	31.371333	0.000000	0.000000	31.81	24.492333	August	NorthE
56	30.220000	28.009333	0.000000	0.000000	31.18	25.458000	September	NorthE
57	31.580000	32.300968	0.000000	0.000000	21.86	17.016129	October	NorthE
58	31.750000	37.037333	0.000000	0.000000	14.37	8.492667	November	NorthE
59	31.650000	35.694839	0.000000	0.000000	11.10	6.983871	December	NorthE

In [167…

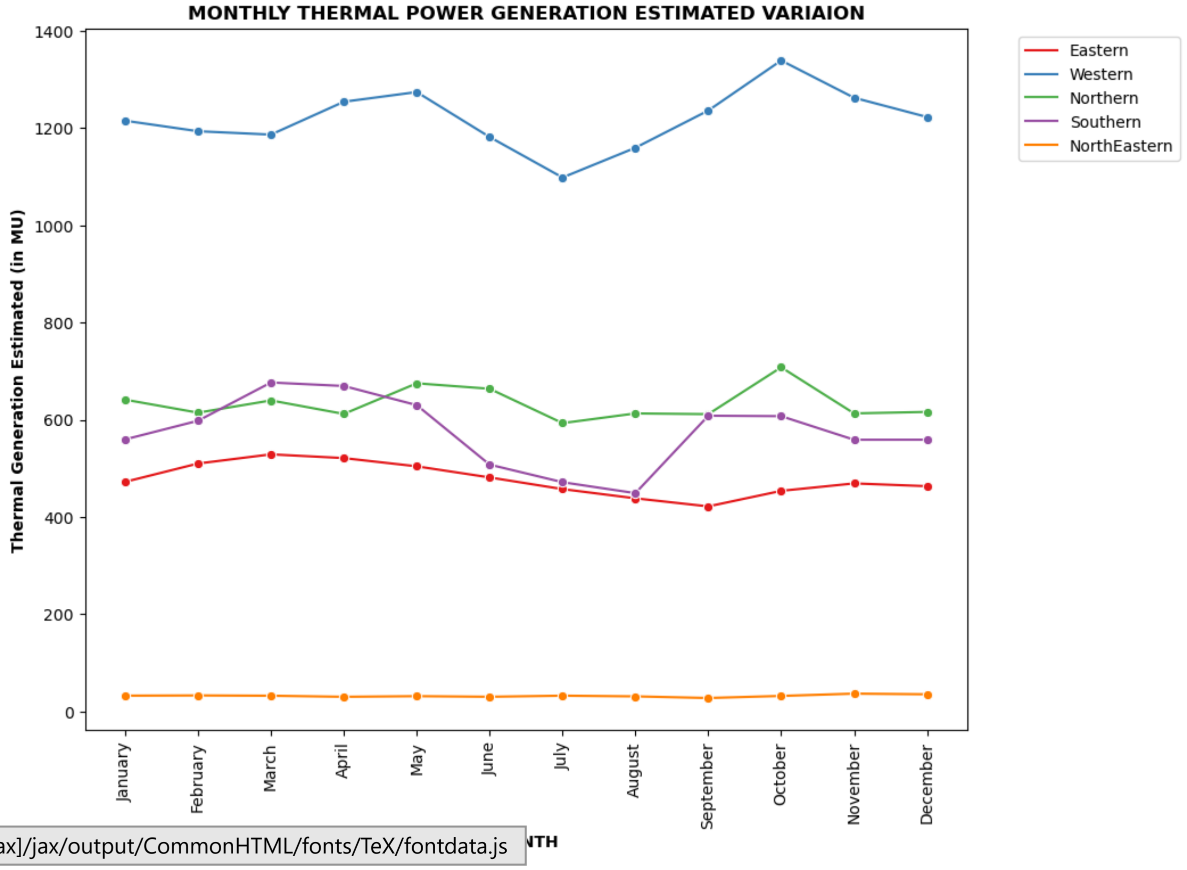
```
plt.figure(figsize=(10,8))
ax = sns.lineplot(x='Month', y='Thermal Generation Actual (in MU)', data=merged_pd
plt.legend(bbox_to_anchor=(1.05,1))
plt.title("MONTHLY THERMAL POWER GENERATION VARIAION",fontweight="bold")
plt.xticks(rotation=90)
plt.xlabel("MONTH",fontweight="bold")
plt.ylabel("Thermal Generation Actual (in MU)",fontweight="bold")

plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [168…

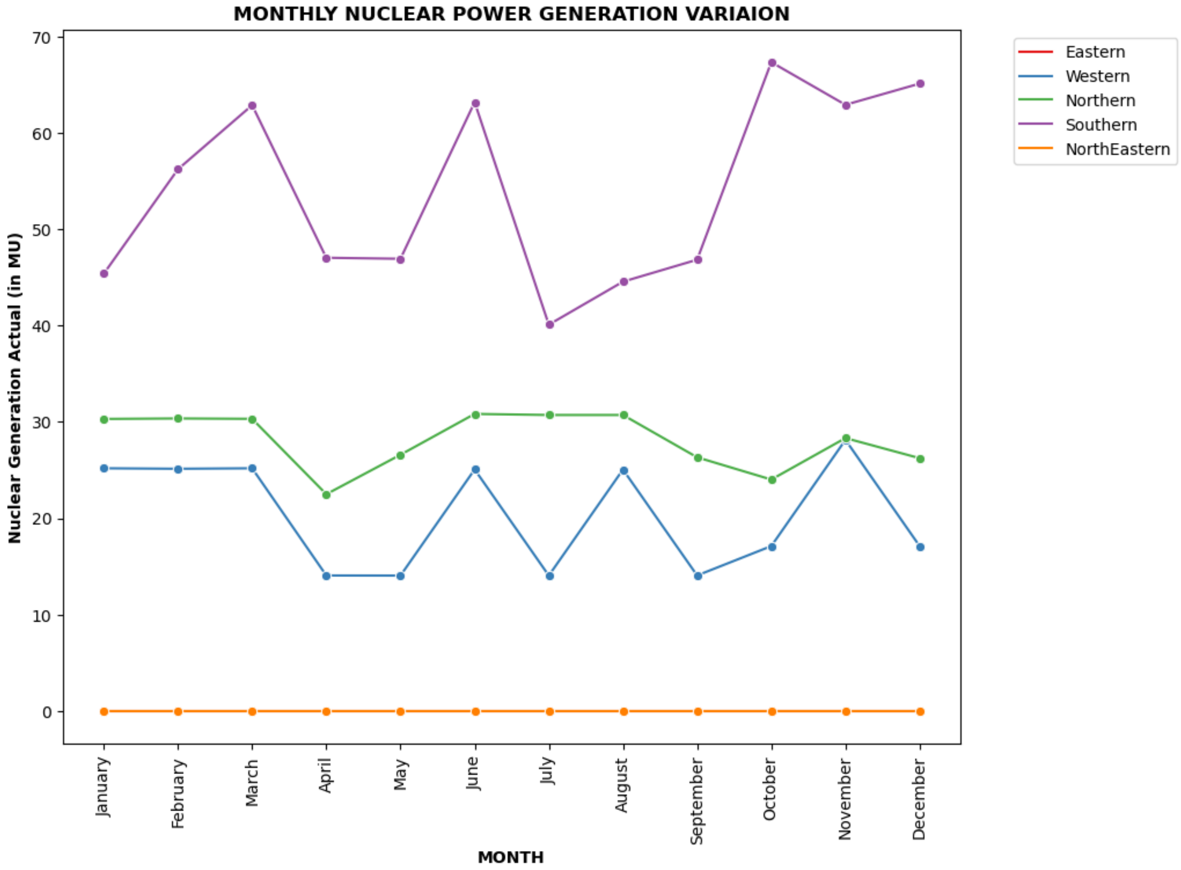
```
plt.figure(figsize=(10,8))
ax = sns.lineplot(x='Month', y='Thermal Generation Estimated (in MU)', data=merged_
plt.legend(bbox_to_anchor=(1.05,1))
plt.title("MONTHLY THERMAL POWER GENERATION ESTIMATED VARIAION",fontweight="bold")
plt.xticks(rotation=90)
plt.xlabel("MONTH",fontweight="bold")
plt.ylabel("Thermal Generation Estimated (in MU)",fontweight="bold")
plt.show()
```



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [169…

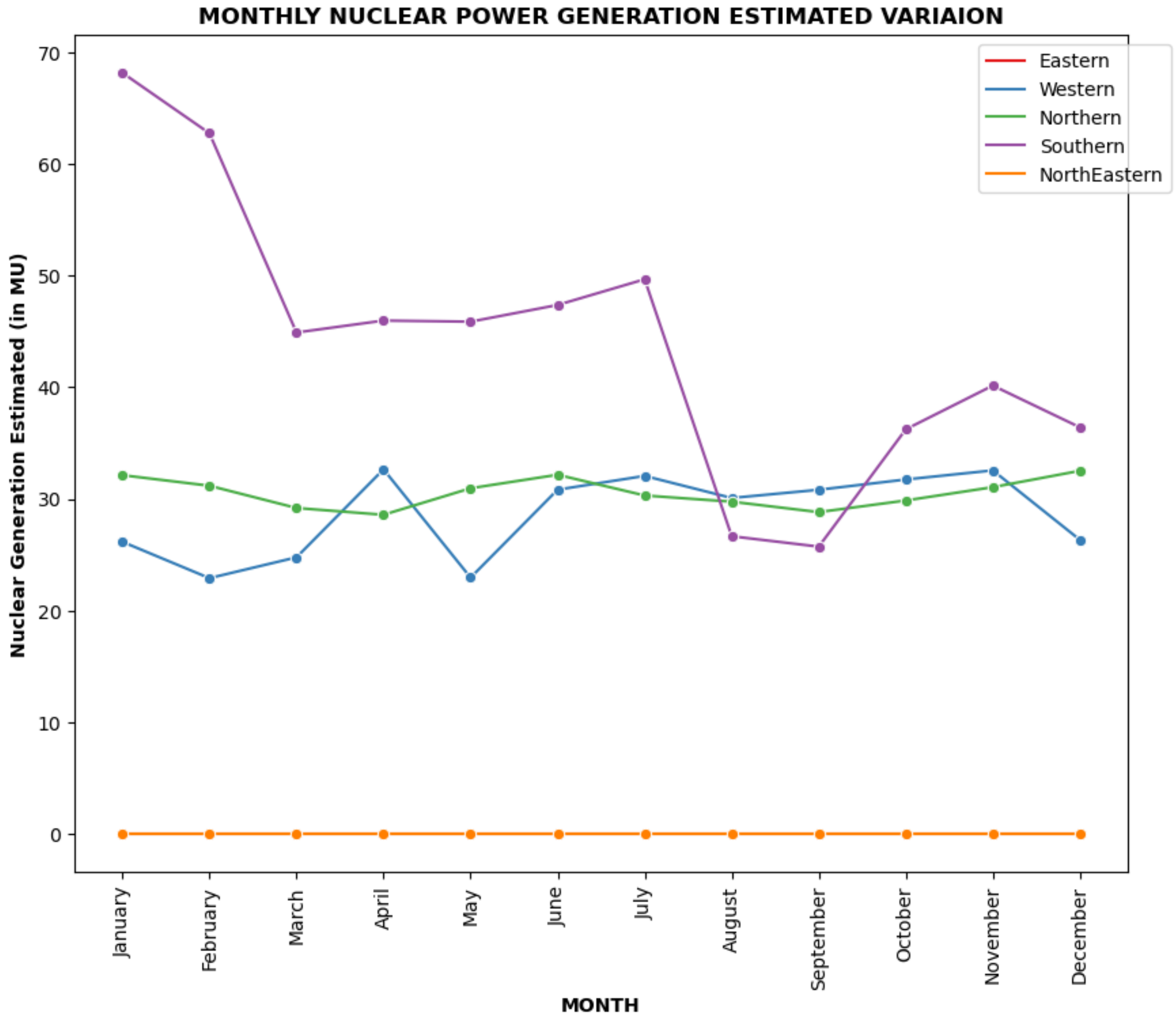
```
plt.figure(figsize=(10,8))
ax = sns.lineplot(x='Month', y='Nuclear Generation Actual (in MU)', data=merged_pd
plt.legend(bbox_to_anchor=(1.05,1))
plt.title("MONTHLY NUCLEAR POWER GENERATION VARIAION",fontweight="bold")
plt.xticks(rotation=90)
plt.xlabel("MONTH",fontweight="bold")
plt.ylabel("Nuclear Generation Actual (in MU)",fontweight="bold")
plt.show()
```



In [170…

```
plt.figure(figsize=(10,8))
ax = sns.lineplot(x='Month', y='Nuclear Generation Estimated (in MU)', data=merged_
plt.legend(bbox_to_anchor=(1.05,1))
plt.title("MONTHLY NUCLEAR POWER GENERATION ESTIMATED VARIAION",fontweight="bold")
plt.xticks(rotation=90)
plt.xlabel("MONTH",fontweight="bold")
plt.ylabel("Nuclear Generation Estimated (in MU)",fontweight="bold")
plt.show()
```

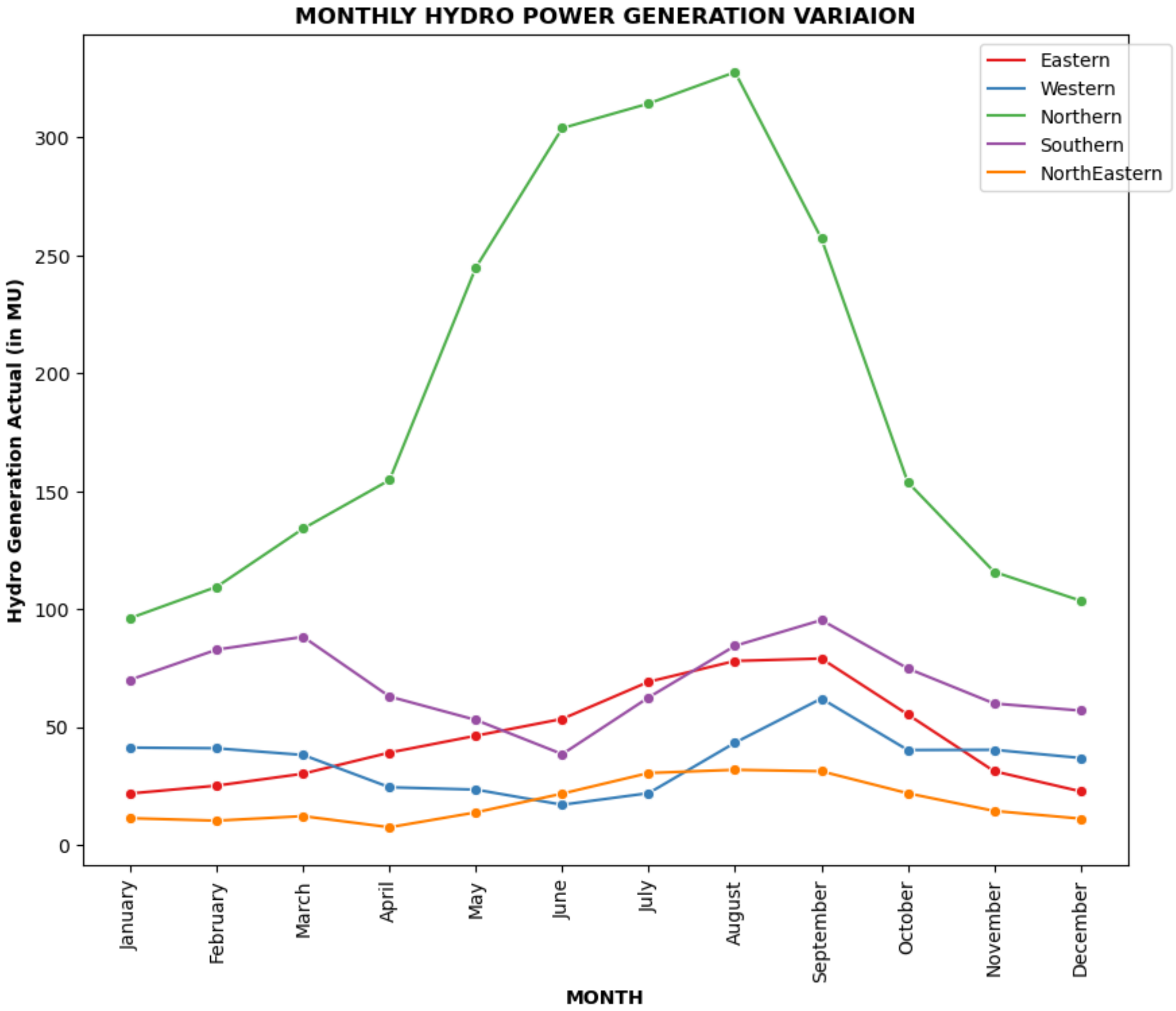
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [171…

```
plt.figure(figsize=(10,8))
ax = sns.lineplot(x='Month', y='Hydro Generation Actual (in MU)', data=merged_pd, 
plt.legend(bbox_to_anchor=(1.05,1))
plt.title("MONTHLY HYDRO POWER GENERATION VARIAION",fontweight="bold")
plt.xticks(rotation=90)
plt.xlabel("MONTH",fontweight="bold")
plt.ylabel("Hydro Generation Actual (in MU)",fontweight="bold")
plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

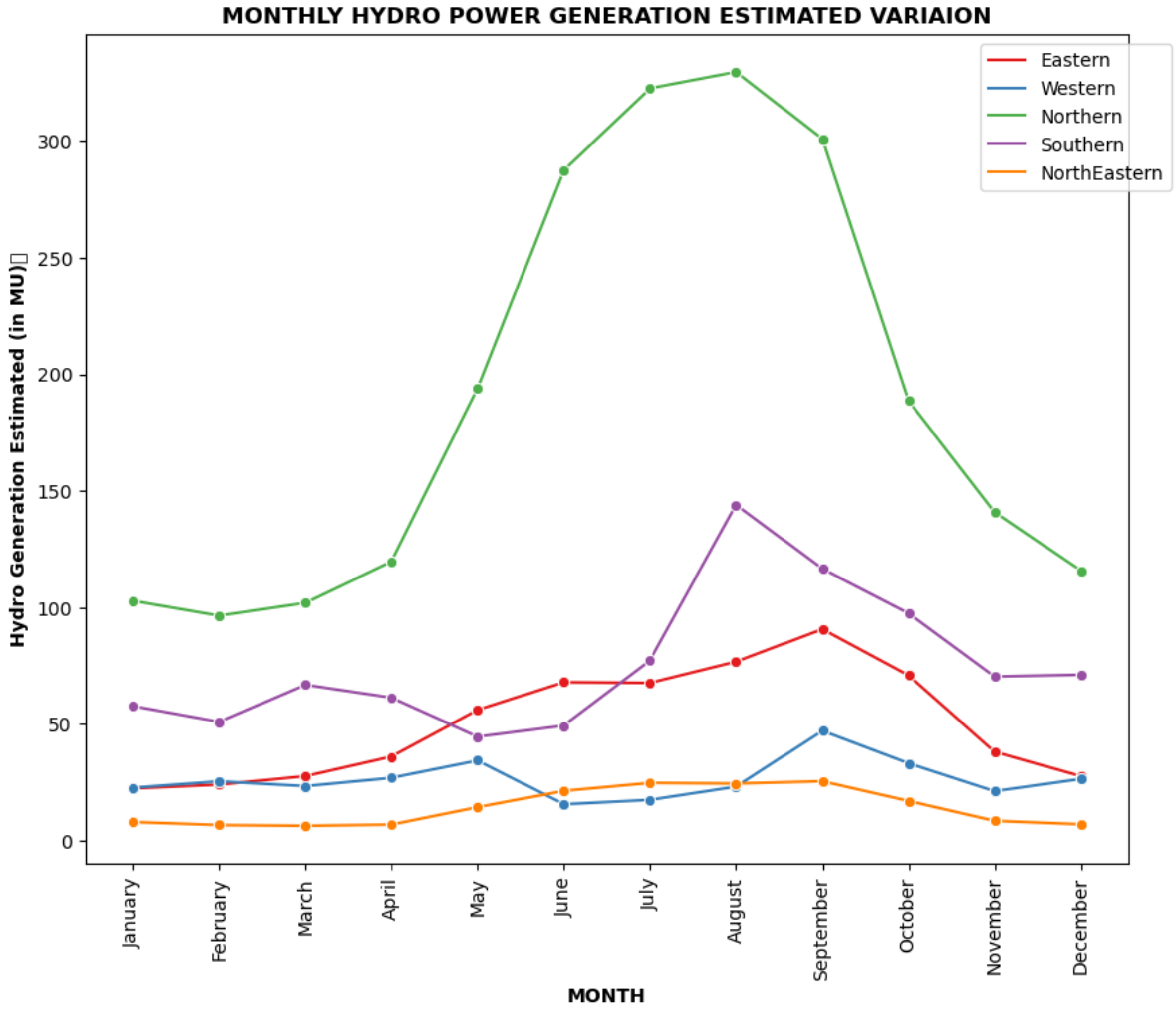


In [172…

```
plt.figure(figsize=(10,8))
ax = sns.lineplot(x='Month', y='Hydro Generation Estimated (in MU)', data=merged_p
plt.legend(bbox_to_anchor=(1.05,1))
plt.title("MONTHLY HYDRO POWER GENERATION ESTIMATED VARIAION",fontweight="bold")
plt.xticks(rotation=90)
plt.xlabel("MONTH",fontweight="bold")
plt.ylabel("Hydro Generation Estimated (in MU)  ",fontweight="bold")
plt.show()
```

```
C:\Users\my hp\anaconda3\lib\site-packages\IPython\core\pylabtools.py:151: UserWar
ning: Glyph 9 ( ) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [175…

```
df2.head(28)
```

Out[175]:

	GROSS STATE DOMESTIC PRODUCT AT CONSTANT (2011-12) PRICES; BASE YEAR 2011-12	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7
0	NaN	As on 15.03.2023	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	GSDP - CONSTANT PRICES (in Crores)	NaN	NaN	NaN	NaN	NaN
3	S. No.	State\UT	2011-12	2012-13	2013-14	2014-15	2015-16	2016-17
4	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
5	1	Andhra Pradesh	379402	380629	407115	444564	498606	540212
6	2	Arunachal Pradesh	11063	11299	12339	14383	14240	14893
7	3	Assam	143175	147342	154525	165212	191109	202081
8	4	Bihar	247144	256851	269650	279482	296488	318797
9	5	Chhattisgarh	158074	165977	182580	185813	190584	213705
10	6	Goa	42367	35850	31568	40116	46091	51249
11	7	Gujarat	615606	682650	734284	811428	894465	981342
12	8	Haryana	297539	320912	347507	370535	413405	456709
13	9	Himachal Pradesh	72720	77384	82847	89060	96274	103055
14	10	Jammu & Kashmir*	78256	80767	85115	82372	97001	100199
15	11	Jharkhand	150918	163250	165816	186534	174881	193174
16	12	Karnataka	606010	643033	704466	748429	831330	941774
17	13	Kerala	364048	387693	402781	419956	451210	485302
18	14	Madhya Pradesh	315562	351683	365134	383944	418736	470669
19	15	Maharashtra	1280369	1357942	1451615	1543165	1654284	1807046
20	16	Manipur	12915	12993	14115	15245	16424	17082
21	17	Meghalaya	19918	20354	20726	20140	20638	21730
22	18	Mizoram	7259	7778	9038	11261	12324	13595
23	19	Nagaland	12177	12868	13793	14399	14660	15650
24	20	Odisha	230987	243363	265892	270665	292229	337348
25	21	Punjab	266628	280823	299450	312125	330052	352721

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

	GROSS STATE DOMESTIC PRODUCT AT CONSTANT (2011-12) PRICES; BASE YEAR 2011-12	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7
26	22	Rajasthan	434837	454564	486230	521509	563340	596746
27	23	Sikkim	11165	11421	12114	13071	14370	15397

In [176…

```
df2 = df2.iloc[3:,:]
```

In [177…

```
df2.head(25)
```

Out[177]:

	GROSS STATE DOMESTIC PRODUCT AT CONSTANT (2011-12) PRICES; BASE YEAR 2011-12	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7
3	S. No.	State\UT	2011-12	2012-13	2013-14	2014-15	2015-16	2016-17
4	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8')
5	1	Andhra Pradesh	379402	380629	407115	444564	498606	540212
6	2	Arunachal Pradesh	11063	11299	12339	14383	14240	14893
7	3	Assam	143175	147342	154525	165212	191109	202081
8	4	Bihar	247144	256851	269650	279482	296488	318797
9	5	Chhattisgarh	158074	165977	182580	185813	190584	213705
10	6	Goa	42367	35850	31568	40116	46091	51249
11	7	Gujarat	615606	682650	734284	811428	894465	981342
12	8	Haryana	297539	320912	347507	370535	413405	456709
13	9	Himachal Pradesh	72720	77384	82847	89060	96274	103055
14	10	Jammu & Kashmir*	78256	80767	85115	82372	97001	100199
15	11	Jharkhand	150918	163250	165816	186534	174881	193174
16	12	Karnataka	606010	643033	704466	748429	831330	941774
17	13	Kerala	364048	387693	402781	419956	451210	485302
18	14	Madhya Pradesh	315562	351683	365134	383944	418736	470669
19	15	Maharashtra	1280369	1357942	1451615	1543165	1654284	1807046
20	16	Manipur	12915	12993	14115	15245	16424	17082
21	17	Meghalaya	19918	20354	20726	20140	20638	21730
22	18	Mizoram	7259	7778	9038	11261	12324	13595
23	19	Nagaland	12177	12868	13793	14399	14660	15650
24	20	Odisha	230987	243363	265892	270665	292229	337348
25	21	Punjab	266628	280823	299450	312125	330052	352721
26	22	Rajasthan	434837	454564	486230	521509	563340	596746
27	23	Sikkim	11165	11421	12114	13071	14370	15397

25 rows × 25 columns

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [178…

```
df_req = df2.iloc[:,9:12]
df_req1 = df2.iloc[:,1]
```

In [179…

```
df_re = pd.concat([df_req,df_req1],axis=1,ignore_index=True)
```

In [180…

```
df_re.fillna(0)
```

Out[180]:

	0	1	2	3
3	2018-19	2019-20	2020-21	State\UT
4	(10)	(11)	(12)	(2)
5	626614	649810	633720	Andhra Pradesh
6	16668	19137	18592	Arunachal Pradesh
7	231040	240707	242946	Assam
8	381383	398283	385728	Bihar
9	244579	251325	246804	Chhattisgarh
10	53063	54812	53360	Goa
11	1183020	1265277	1248189	Gujarat
12	532996	544275	510306	Haryana
13	116411	121168	114814	Himachal Pradesh
14	115062	0	0	Jammu & Kashmir*
15	229274	231755	218962	Jharkhand
16	1085101	1148806	1108212	Karnataka
17	554228	559194	512076	Kerala
18	543272	567525	543935	Madhya Pradesh
19	1957381	2004663	1858370	Maharashtra
20	18262	19187	19053	Manipur
21	23719	24923	23751	Meghalaya
22	16100	17884	16689	Mizoram
23	16868	18477	18621	Nagaland
24	386733	397786	378075	Odisha
25	397019	413295	398343	Punjab
26	643278	676785	663515	Rajasthan
27	18625	19492	19040	Sikkim
28	1204667	1243836	1245595	Tamil Nadu
29	608401	640968	597206	Telangana
30	36754	38063	37244	Tripura
31	1097353	1140712	1077534	Uttar Pradesh
32	186083	189740	178764	Uttarakhand
33	738920	761901	711235	West Bengal
34	6867	7266	6978	Andaman & Nicobar Islands
35	29866	31093	27460	Chandigarh
36	565327	586168	547682	Delhi
37	0	113943	112628	Jammu & Kashmir-UT
[illegible]	[illegible]	[illegible]	[illegible]	Puducherry

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

	0	1	2	3
39	0	0	0	0
40	0	0	0	Source: Directorate of Economics & Statistics...
41	0	0	0	* relates to Jammu and Kashmir and Ladakh

In [181…

```
df_re = df_re.iloc[2:35,:]
```

In [182…

```
df_re.columns
```

Out[182]:

```
RangeIndex(start=0, stop=4, step=1)
```

In [183…

```
df_re.columns = ["2019","2020","2021","state"]
```

In [184…

```
df_re["2019"] = df_re["2019"].astype(float)
df_re["2020"] = df_re["2020"].astype(float)
df_re["2021"] = df_re["2021"].astype(float)
```

In [185…

```
shp_gdf = gpd.read_file("C:\\Users\\my hp\\Desktop\\Data_Visulization\\Power\\Powe
shp_gdf.drop(columns='Type',inplace=True)
shp_gdf.iloc[1,0] = 'Andaman & Nicobar Islands'
shp_gdf.iloc[14,0]= 'Chhatisgarh'
shp_gdf.iloc[13,0] = 'Tamil Nadu'
shp_gdf.iloc[15,0] = 'Telangana'
shp_gdf.iloc[34,0] = 'Jammu and Kashmir'
shp_gdf.iloc[16,0] = 'Andhra Pradesh'
```

In [186…

```
merged_gdp = shp_gdf.set_index('Name').join(df_re.set_index('state'))
```

In [187…

```
merged_gdp.iloc[3,1:] =0
merged_gdp.iloc[9,1:]=0
merged_gdp.iloc[12,1:]=0
merged_gdp.iloc[14,1] = 244579
merged_gdp.iloc[14,2] = 251325
merged_gdp.iloc[14,3] = 246804
merged_gdp.iloc[16,1] = 626614
merged_gdp.iloc[16,2] = 649810
merged_gdp.iloc[16,3] = 633720
merged_gdp.iloc[33,1] = 115062
merged_gdp.iloc[33,2] = 113943
merged_gdp.iloc[33,3] = 112628
merged_gdp.iloc[34,1] = 115062
merged_gdp.iloc[34,2] = 113943
merged_gdp.iloc[34,3] = 112628
```

In [188…

```
merged_gdp
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[188]:

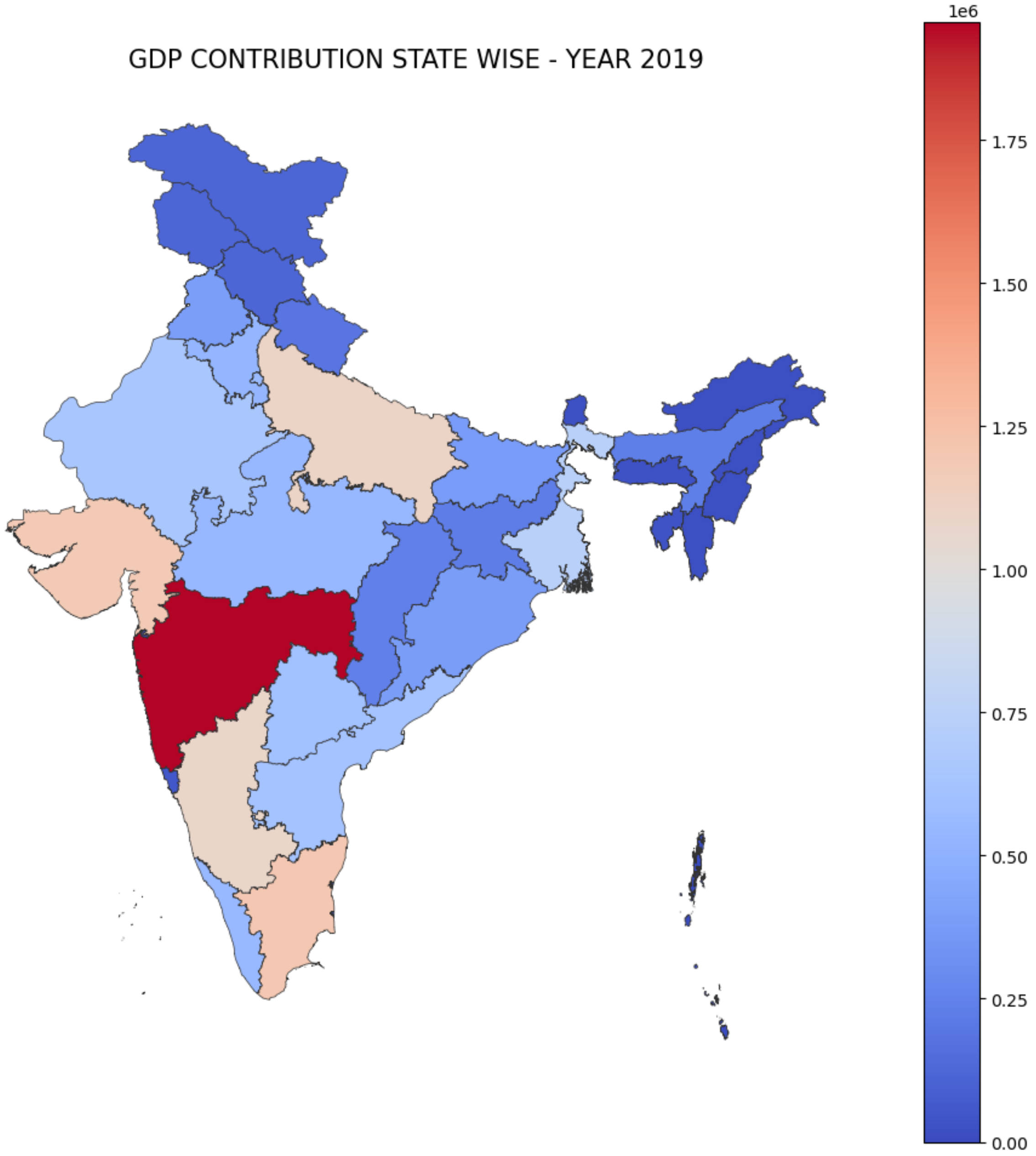
	geometry	2019	2020	2021
Name				
West Bengal	MULTIPOLYGON (((88.01861 21.57278, 88.01889 21...	738920.0	761901.0	711235.0
Andaman & Nicobar Islands	MULTIPOLYGON (((92.90124 12.91071, 92.90157 12...	6867.0	7266.0	6978.0
Chandigarh	POLYGON ((76.77232 30.79420, 76.77286 30.79343...	29866.0	31093.0	27460.0
Daman and Diu and Dadra and Nagar Haveli	MULTIPOLYGON (((72.96339 20.33227, 72.96390 20...	0.0	0.0	0.0
Delhi	POLYGON ((77.10591 28.87005, 77.10668 28.86989...	565327.0	586168.0	547682.0
Haryana	POLYGON ((76.57526 30.10063, 76.57645 30.10152...	532996.0	544275.0	510306.0
Jharkhand	POLYGON ((87.69613 24.16027, 87.69625 24.15974...	229274.0	231755.0	218962.0
Karnataka	MULTIPOLYGON (((77.33232 18.45086, 77.33267 18...	1085101.0	1148806.0	1108212.0
Kerala	POLYGON ((74.98896 12.79553, 74.98953 12.79554...	554228.0	559194.0	512076.0
Lakshadweep	MULTIPOLYGON (((73.08025 8.32651, 73.08029 8.3...	0.0	0.0	0.0
Madhya Pradesh	POLYGON ((78.37211 26.86406, 78.37423 26.86091...	543272.0	567525.0	543935.0
Maharashtra	MULTIPOLYGON (((74.38926 22.03241, 74.38942 22...	1957381.0	2004663.0	1858370.0
Puducherry	MULTIPOLYGON (((79.75669 11.00258, 79.75682 11...	0.0	0.0	0.0
Tamil Nadu	MULTIPOLYGON (((80.30358 13.47307, 80.30577 13...	1204667.0	1243836.0	1245595.0
Chhatisgarh	POLYGON ((83.32707 24.10232, 83.32939 24.10130...	244579.0	251325.0	246804.0
Telangana	POLYGON ((78.33565 19.88358, 78.33688 19.88300...	608401.0	640968.0	597206.0
Andhra Pradesh	POLYGON ((84.67571 19.16721, 84.67725 19.16670...	626614.0	649810.0	633720.0
Goa	POLYGON ((73.87042 15.78117, 73.87097 15.78029...	53063.0	54812.0	53360.0
Himachal Pradesh	POLYGON ((76.79412 33.25569, 76.79482 33.25557...	116411.0	121168.0	114814.0
Punjab	POLYGON ((75.83876 32.51269, 75.84094 32.51158...	397019.0	413295.0	398343.0
Rajasthan	POLYGON ((73.97266 30.19800, 73.97266 30.19800...	643278.0	676785.0	663515.0
Gujarat	POLYGON ((72.26126 21.49999, 72.25911 21.49999...	1183020.0	1265277.0	1248189.0

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Name	geometry	2019	2020	2021
Uttarakhand	POLYGON ((79.06793 31.46153, 79.07046 31.46010...	186083.0	189740.0	178764.0
Uttar Pradesh	POLYGON ((77.59189 30.40632, 77.59337 30.40621...	1097353.0	1140712.0	1077534.0
Sikkim	POLYGON ((88.61635 28.12763, 88.62302 28.12625...	18625.0	19492.0	19040.0
Assam	POLYGON ((93.24961 24.97312, 93.25244 24.97428...	231040.0	240707.0	242946.0
Arunachal Pradesh	POLYGON ((96.08538 29.45928, 96.09262 29.45613...	16668.0	19137.0	18592.0
Nagaland	POLYGON ((95.19272 27.02710, 95.19475 27.01442...	16868.0	18477.0	18621.0
Manipur	POLYGON ((94.57415 25.69043, 94.57852 25.68761...	18262.0	19187.0	19053.0
Mizoram	POLYGON ((92.76384 24.52098, 92.76422 24.51967...	16100.0	17884.0	16689.0
Tripura	POLYGON ((92.16949 24.53175, 92.16961 24.53168...	36754.0	38063.0	37244.0
Meghalaya	POLYGON ((91.82617 26.11925, 91.82728 26.11914...	23719.0	24923.0	23751.0
Bihar	POLYGON ((84.10880 27.52173, 84.10896 27.52131...	381383.0	398283.0	385728.0
Ladakh	POLYGON ((76.80933 33.24349, 76.79356 33.25175...	115062.0	113943.0	112628.0
Jammu and Kashmir	POLYGON ((76.80933 33.24349, 76.80894 33.24367...	115062.0	113943.0	112628.0
Odisha	POLYGON ((87.47639 21.64343, 87.47362 21.63205...	386733.0	397786.0	378075.0

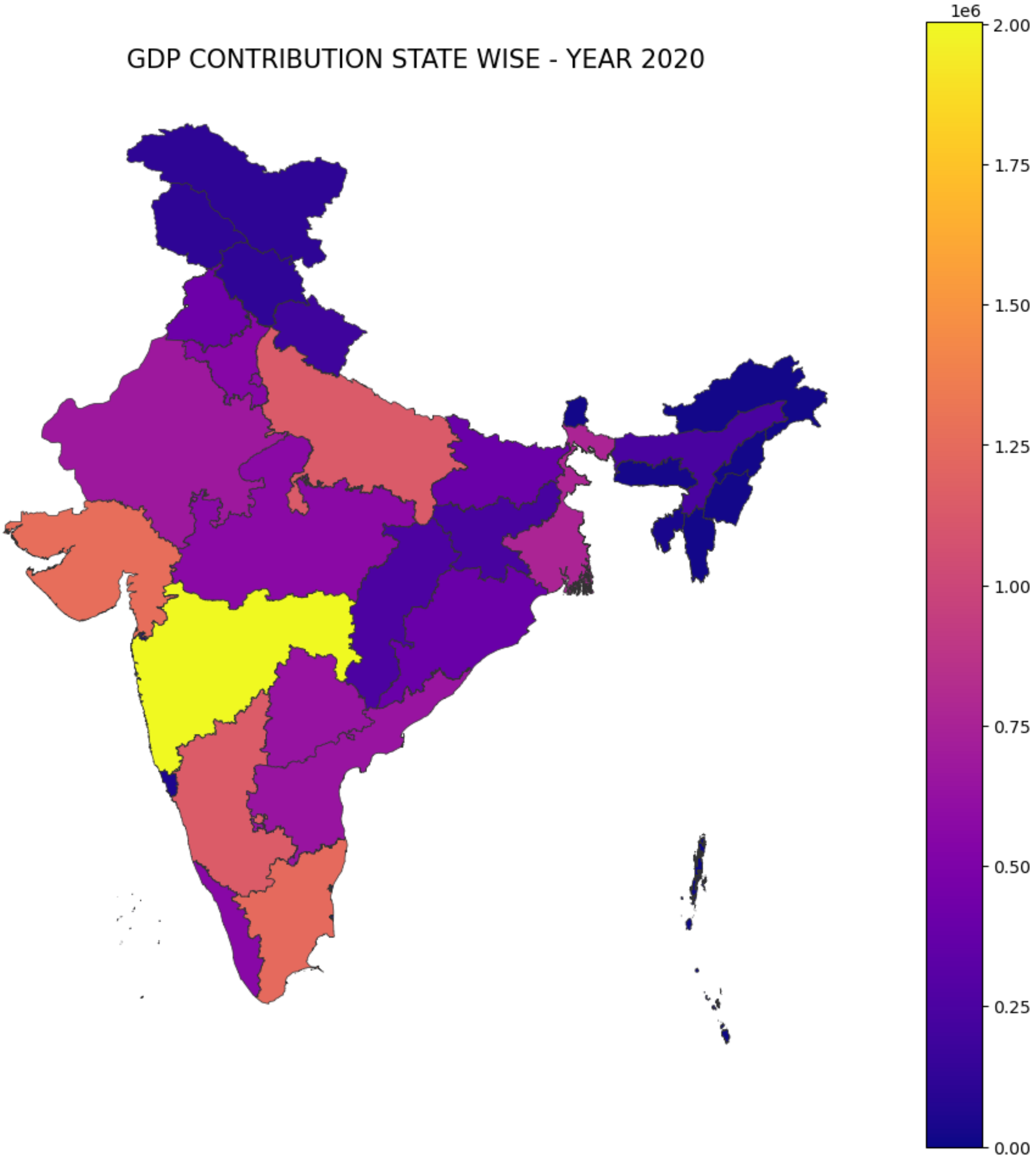
In [190…

```
fig ,ax = plt.subplots(1,figsize=(12,12))
ax.axis('off')
ax.set_title("GDP CONTRIBUTION STATE WISE - YEAR 2019",fontdict={'fontsize':'15','
fig = merged_gdp.plot(column='2019' , cmap='coolwarm',linewidth=0.5,ax=ax,edgecolo
```



```
In [191…
fig ,ax = plt.subplots(1,figsize=(12,12))
ax.axis('off')
ax.set_title("GDP CONTRIBUTION STATE WISE - YEAR 2020",fontdict={'fontsize':'15','
fig = merged_gdp.plot(column='2020' , cmap='plasma',linewidth=0.5,ax=ax,edgecolor=
```

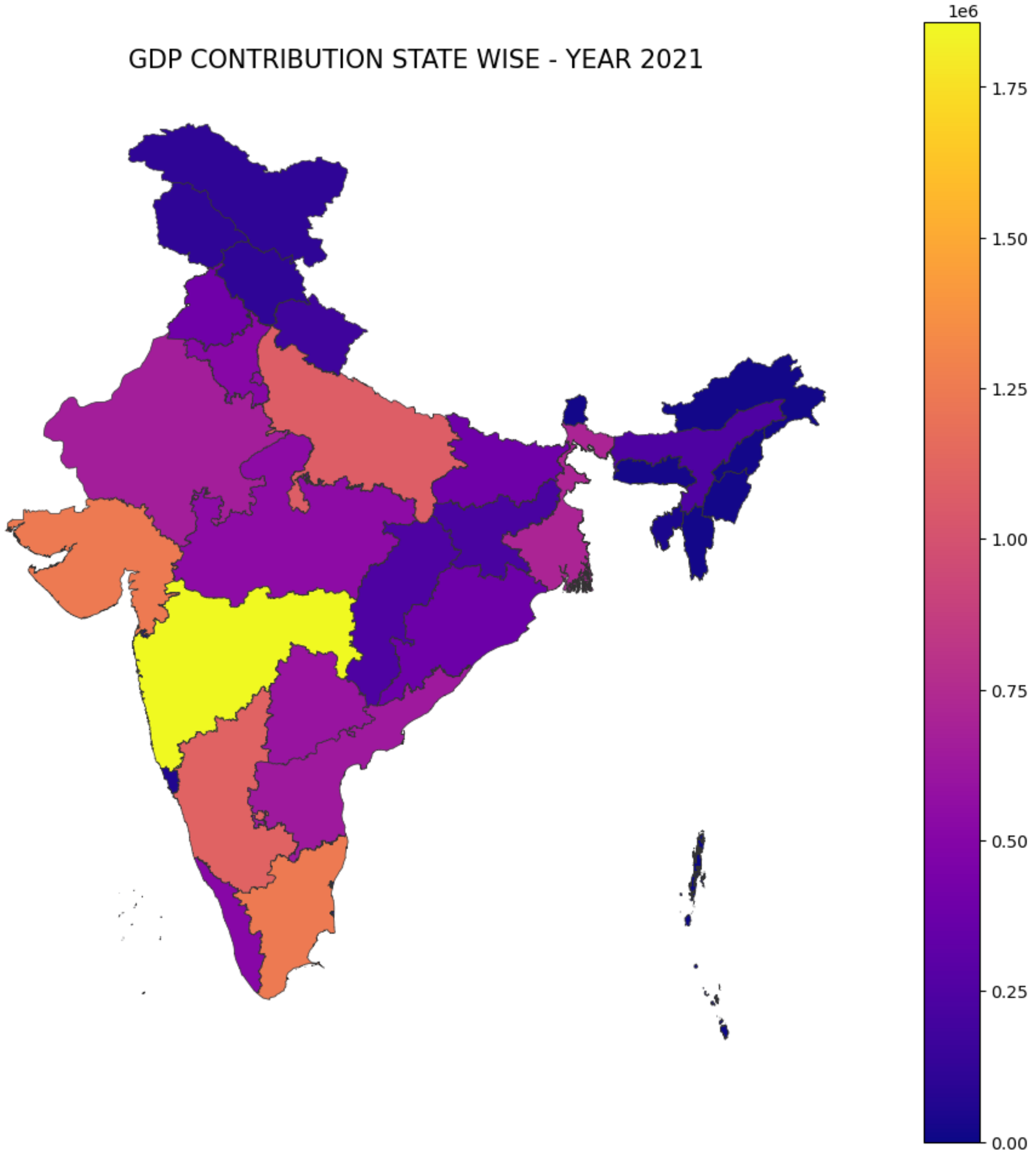
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [192…

```
fig ,ax = plt.subplots(1,figsize=(12,12))
ax.axis('off')
ax.set_title("GDP CONTRIBUTION STATE WISE - YEAR 2021",fontdict={'fontsize':'15','
fig = merged_gdp.plot(column='2021' , cmap='plasma',linewidth=0.5,ax=ax,edgecolor=
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [656…

```
northern_year_2019 = year_2019[year_2019["Region"]=="Northern"]
eastern_year_2019 = year_2019[year_2019["Region"]=="Eastern"]
western_year_2019 = year_2019[year_2019["Region"]=="Western"]
southern_year_2019 = year_2019[year_2019["Region"]=="Southern"]
north_eastern_year_2019 = year_2019[year_2019["Region"]=="NorthEastern"]
```

In [657…

```
df_monthly_mean_north = northern_year_2019.groupby(pd.Grouper(key='date', freq='M'
df_monthly_mean_east = eastern_year_2019.groupby(pd.Grouper(key='date', freq='M'))
df_monthly_mean_west = western_year_2019.groupby(pd.Grouper(key='date', freq='M'))
df_monthly_mean_south = southern_year_2019.groupby(pd.Grouper(key='date', freq='M'
df_monthly_mean_north_east = north_eastern_year_2019.groupby(pd.Grouper(key='date'
```

In [658…

```
l = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'Se
```

In [659…

```
df_monthly_mean_east["Month"] =l
df_monthly_mean_north["Month"] = l
df_monthly_mean_south["Month"] =l
df_monthly_mean_west["Month"] = l
df_monthly_mean_north_east["Month"] = l
df_monthly_mean_east["Region"] = ["Eastern"]*12
                                  Northern"]*12
df_monthly_mean_south["Region"] = ["Southern"]*12
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
df_monthly_mean_west["Region"] = ["Western"]*12
df_monthly_mean_north_east["Region"] = ["NorthEastern"]*12
```

In [660…

```
merged_pd = pd.concat([df_monthly_mean_east,df_monthly_mean_west,df_monthly_mean_n
```

In [661…

```
merged_pd
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[661]:

	Thermal Generation Actual (in MU)	Thermal Generation Estimated (in MU)	Nuclear Generation Actual (in MU)	Nuclear Generation Estimated (in MU)	Hydro Generation Actual (in MU)	Hydro Generation Estimated (in MU)	Month	R
0	496.471613	480.496774	0.000000	0.000000	22.84	24.701935	January	E
1	498.781071	493.595000	0.000000	0.000000	24.00	26.837143	February	E
2	504.179677	509.859032	0.000000	0.000000	29.22	34.010968	March	E
3	514.986667	523.762333	0.000000	0.000000	44.26	48.791667	April	E
4	530.804194	516.730645	0.000000	0.000000	53.28	67.368065	May	E
5	506.476000	506.506333	0.000000	0.000000	66.73	71.599000	June	E
6	492.922903	462.512581	0.000000	0.000000	83.26	76.237419	July	E
7	494.572903	404.119032	0.000000	0.000000	90.02	88.382903	August	E
8	505.050000	438.342667	0.000000	0.000000	86.01	92.731000	September	E
9	528.701290	417.800645	0.000000	0.000000	64.81	76.698065	October	E
10	477.465667	465.361667	0.000000	0.000000	37.92	41.334000	November	E
11	483.697742	477.267419	0.000000	0.000000	25.63	29.118065	December	E
12	1238.349355	1209.854194	17.120000	37.570645	36.29	24.753226	January	W
13	1233.450000	1220.089643	32.680000	39.420714	35.58	23.899643	February	W
14	1263.091613	1283.608710	32.600000	39.250000	34.47	30.604516	March	W
15	1387.648000	1358.071000	32.640000	37.806667	32.58	34.086667	April	W
16	1389.600645	1354.393226	32.600000	39.509677	25.19	45.382581	May	W
17	1299.122667	1298.534667	32.640000	40.688000	26.09	13.012000	June	W
18	1141.740323	1176.279677	37.080000	43.091935	29.87	19.800645	July	W
19	1165.321613	1070.416774	37.080000	44.164516	42.17	76.686452	August	W
20	1261.367667	1041.806000	37.140000	40.710000	45.76	109.946333	September	W
21	1340.025161	1092.060000	34.020000	43.083226	34.47	79.865161	October	W
22	1288.902333	1170.507667	23.070000	28.559667	36.40	43.963333	November	W
23	1277.647742	1244.557097	28.697419	30.053871	33.35	46.311613	December	W
24	675.526667	633.991667	24.530667	27.614667	96.87	104.583000	January	Nc
25	663.731786	555.635357	26.910000	24.656786	105.09	129.408929	February	Nc
26	666.431613	542.901935	26.880000	30.905161	121.72	145.375484	March	Nc
27	692.206000	576.402333	27.030000	31.563000	159.92	212.895000	April	Nc
28	725.139355	659.610968	26.920000	31.420323	260.86	260.141290	May	Nc
29	717.745667	725.495667	27.030000	34.422667	299.27	337.499333	June	Nc
30	645.733871	623.861935	26.880000	34.430323	319.37	349.875161	July	Nc
31	669.701290	617.795484	26.880000	33.118065	341.97	346.271613	August	Nc
32	703.970667	631.873333	27.000000	33.997667	257.05	319.623667	September	Nc
				34.037419	157.27	177.417419	October	Nc

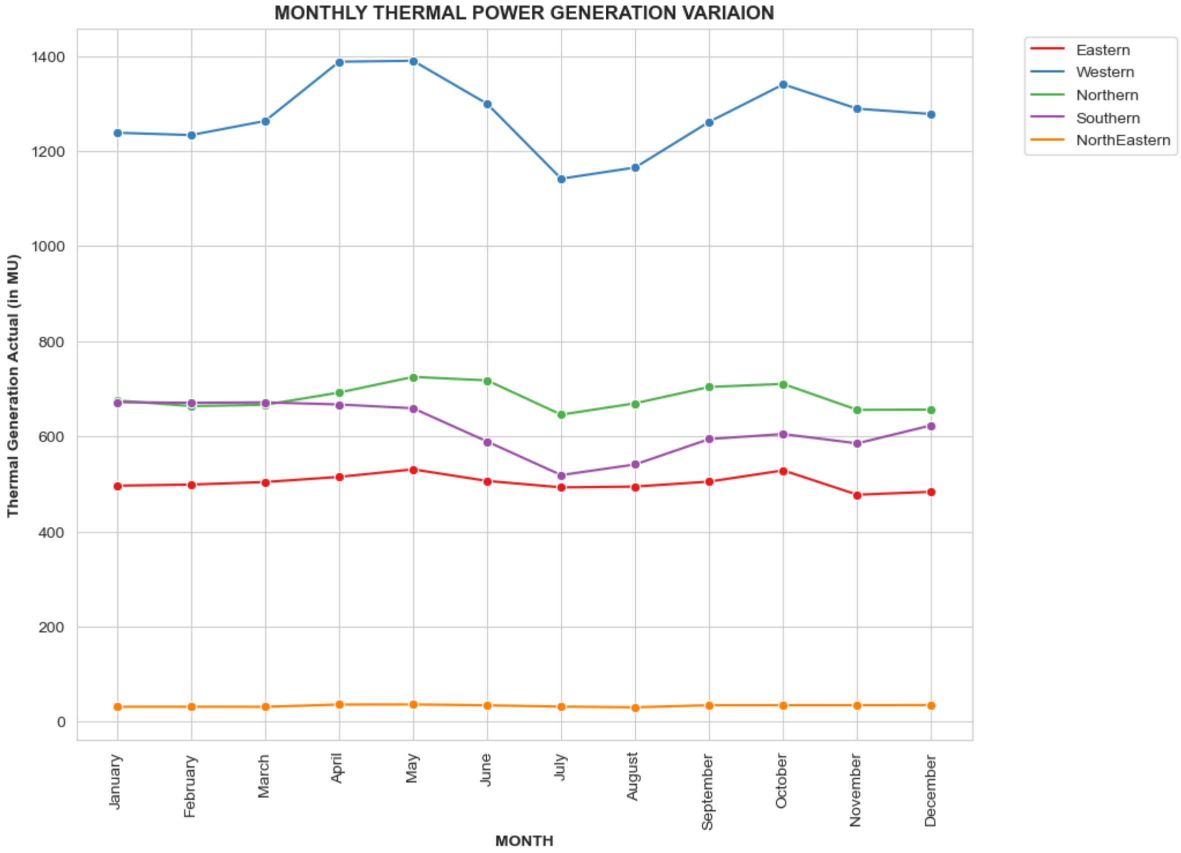
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

	Thermal Generation Actual (in MU)	Thermal Generation Estimated (in MU)	Nuclear Generation Actual (in MU)	Nuclear Generation Estimated (in MU)	Hydro Generation Actual (in MU)	Hydro Generation Estimated (in MU)	Month	R
34	655.826333	504.915667	30.830000	35.506000	117.75	132.291667	November	Nc
35	656.445484	521.208710	30.720000	33.361935	104.80	122.894194	December	Nc
36	671.952903	549.615161	67.320000	31.854516	60.30	67.629677	January	So
37	670.810714	596.149286	67.510000	35.307500	66.95	71.153929	February	So
38	671.547742	678.189355	67.320000	34.663226	73.41	71.763548	March	So
39	667.220000	650.952333	62.760000	36.552000	76.25	69.625667	April	So
40	659.125161	611.806774	54.483548	46.961290	65.75	64.331290	May	So
41	589.232667	569.449667	46.860000	54.949667	57.72	40.540000	June	So
42	518.735161	516.238710	57.288710	58.815806	68.42	38.512258	July	So
43	541.120323	471.003226	67.160000	56.980323	96.14	106.946452	August	So
44	594.482333	473.952000	46.830000	60.387333	100.04	147.438333	September	So
45	605.010323	445.714516	46.770000	57.351613	84.88	137.791613	October	So
46	585.374667	471.636667	67.230000	57.177667	64.27	109.353667	November	So
47	623.274839	500.158387	67.160000	51.257419	59.11	80.465161	December	So
48	31.730000	35.584516	0.000000	0.000000	9.40	6.447419	January	NorthE
49	31.810000	36.086071	0.000000	0.000000	8.62	3.477857	February	NorthE
50	31.700000	36.934194	0.000000	0.000000	9.21	3.205484	March	NorthE
51	36.460000	38.014333	0.000000	0.000000	4.92	4.347333	April	NorthE
52	36.630000	40.079677	0.000000	0.000000	13.71	16.944194	May	NorthE
53	34.878000	39.628667	0.000000	0.000000	22.80	17.181667	June	NorthE
54	32.124516	39.286774	0.000000	0.000000	27.73	26.040968	July	NorthE
55	30.631935	33.159677	0.000000	0.000000	30.44	21.854516	August	NorthE
56	35.016667	31.849333	0.000000	0.000000	33.99	22.036667	September	NorthE
57	35.114516	33.453548	0.000000	0.000000	26.91	15.838710	October	NorthE
58	35.083000	35.250667	0.000000	0.000000	18.37	10.634000	November	NorthE
59	35.406452	34.842903	0.000000	0.000000	14.60	7.786774	December	NorthE

In [662…

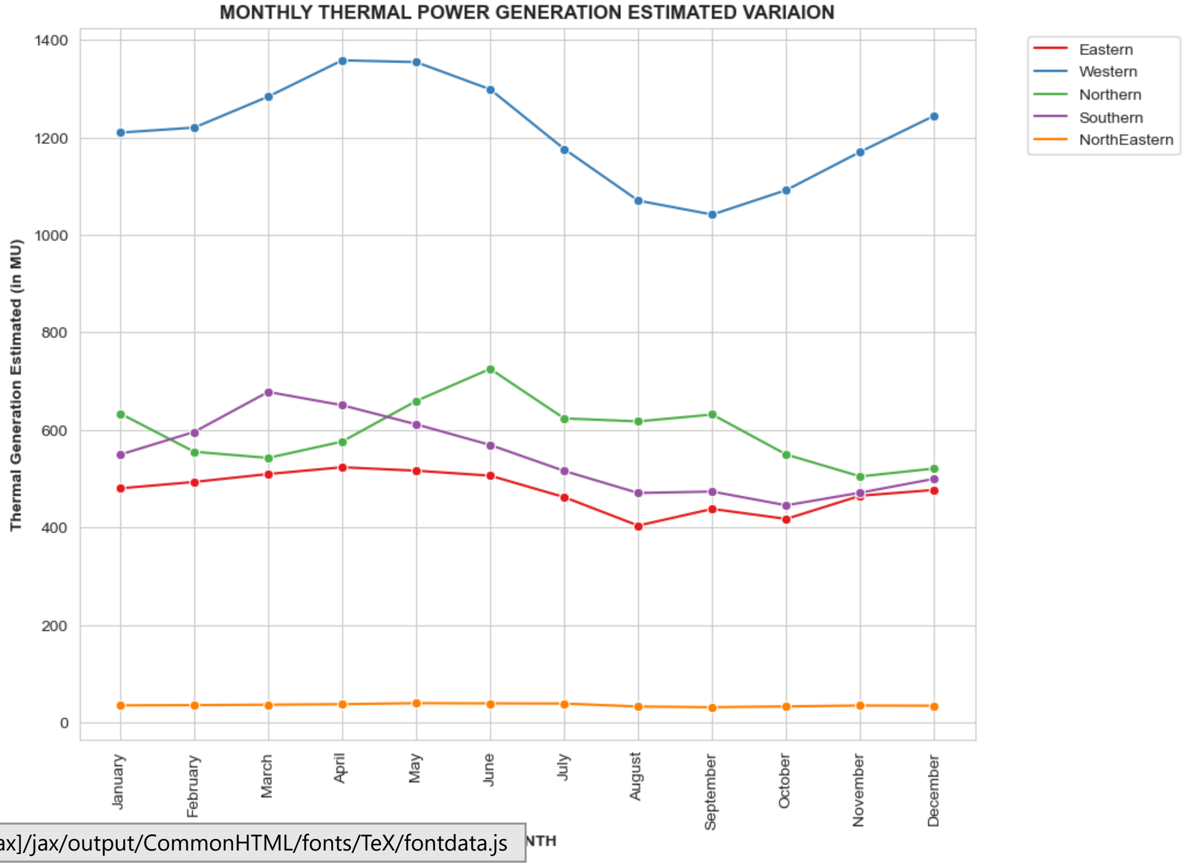
```
plt.figure(figsize=(10,8))
ax = sns.lineplot(x='Month', y='Thermal Generation Actual (in MU)', data=merged_pd
plt.legend(bbox_to_anchor=(1.05,1))
plt.title("MONTHLY THERMAL POWER GENERATION VARIAION",fontweight="bold")
plt.xticks(rotation=90)
plt.xlabel("MONTH",fontweight="bold")
plt.ylabel("Thermal Generation Actual (in MU)",fontweight="bold")

plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [663…

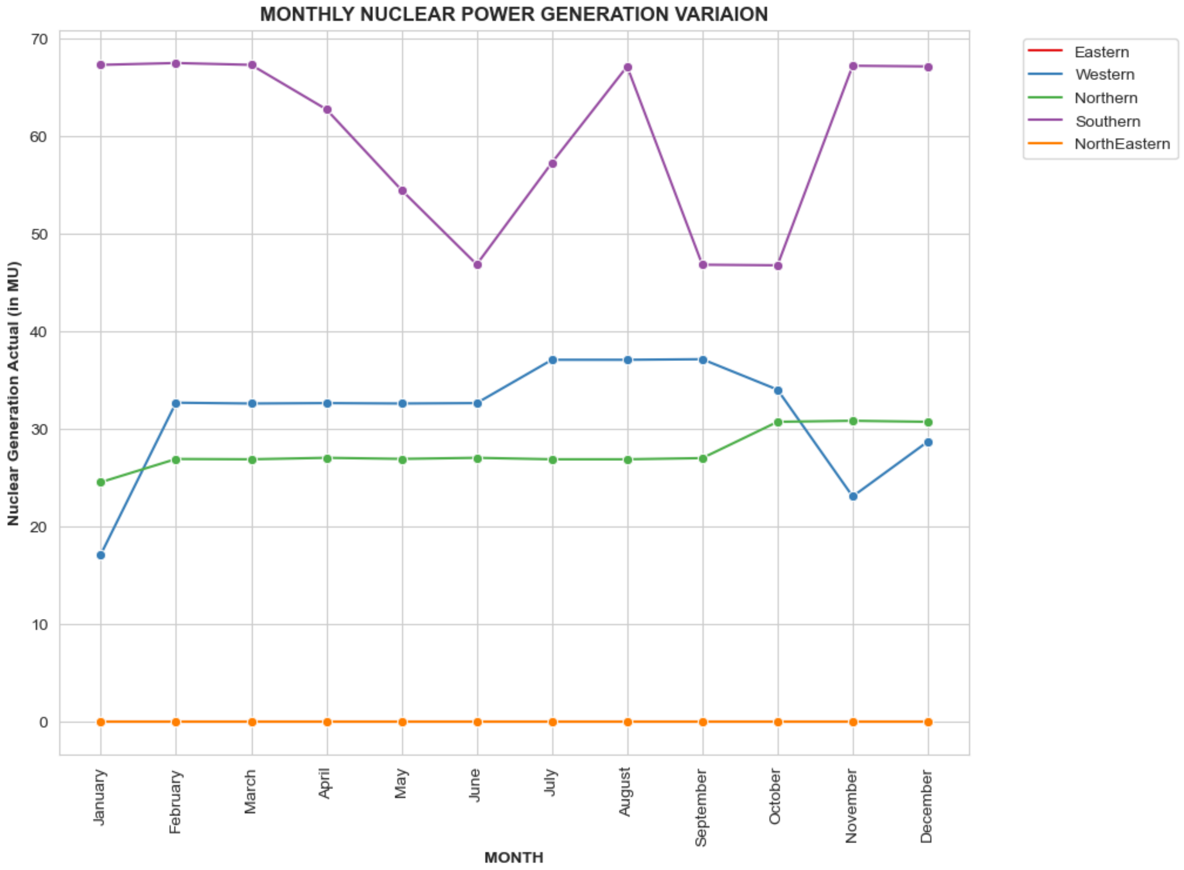
```
plt.figure(figsize=(10,8))
ax = sns.lineplot(x='Month', y='Thermal Generation Estimated (in MU)', data=merged_
plt.legend(bbox_to_anchor=(1.05,1))
plt.title("MONTHLY THERMAL POWER GENERATION ESTIMATED VARIAION",fontweight="bold")
plt.xticks(rotation=90)
plt.xlabel("MONTH",fontweight="bold")
plt.ylabel("Thermal Generation Estimated (in MU)",fontweight="bold")
plt.show()
```



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [664…

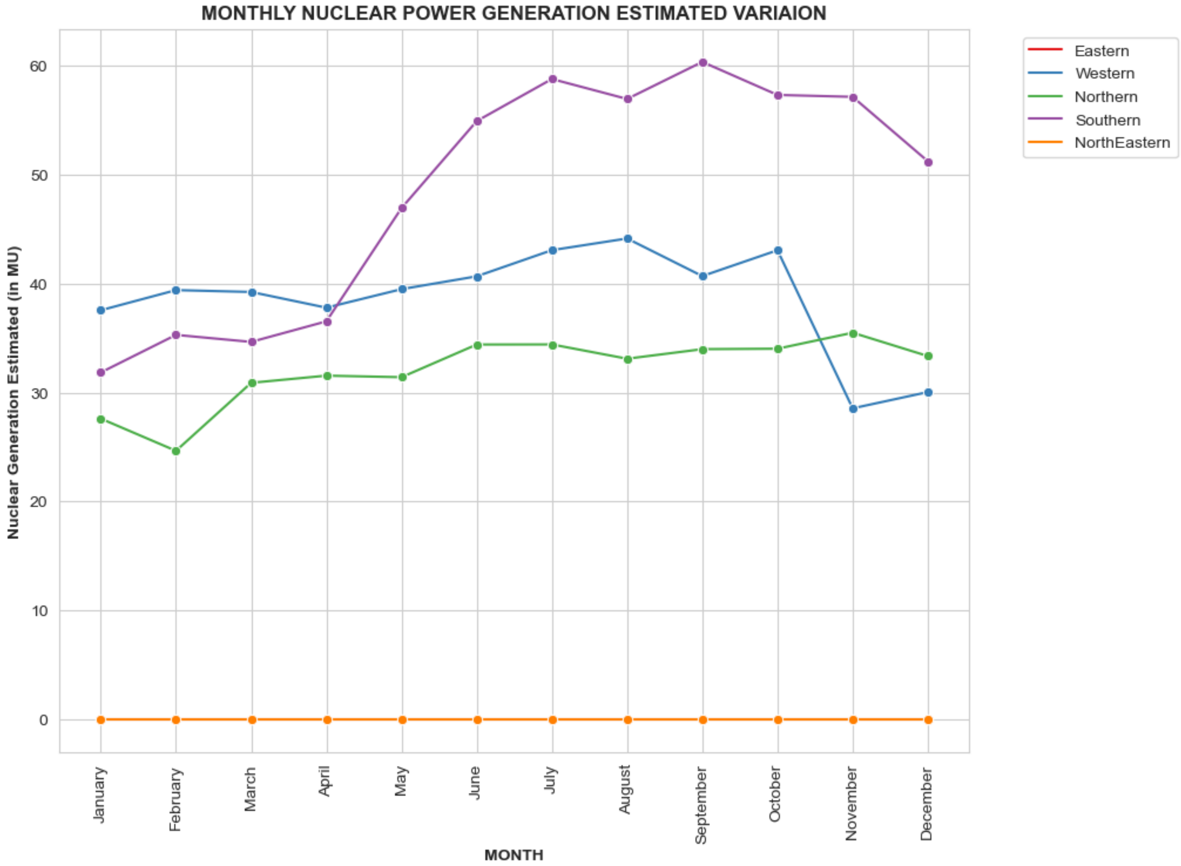
```
plt.figure(figsize=(10,8))
ax = sns.lineplot(x='Month', y='Nuclear Generation Actual (in MU)', data=merged_pd
plt.legend(bbox_to_anchor=(1.05,1))
plt.title("MONTHLY NUCLEAR POWER GENERATION VARIAION",fontweight="bold")
plt.xticks(rotation=90)
plt.xlabel("MONTH",fontweight="bold")
plt.ylabel("Nuclear Generation Actual (in MU)",fontweight="bold")
plt.show()
```



In [665…

```
plt.figure(figsize=(10,8))
ax = sns.lineplot(x='Month', y='Nuclear Generation Estimated (in MU)', data=merged_
plt.legend(bbox_to_anchor=(1.05,1))
plt.title("MONTHLY NUCLEAR POWER GENERATION ESTIMATED VARIAION",fontweight="bold")
plt.xticks(rotation=90)
plt.xlabel("MONTH",fontweight="bold")
plt.ylabel("Nuclear Generation Estimated (in MU)",fontweight="bold")
plt.show()
```

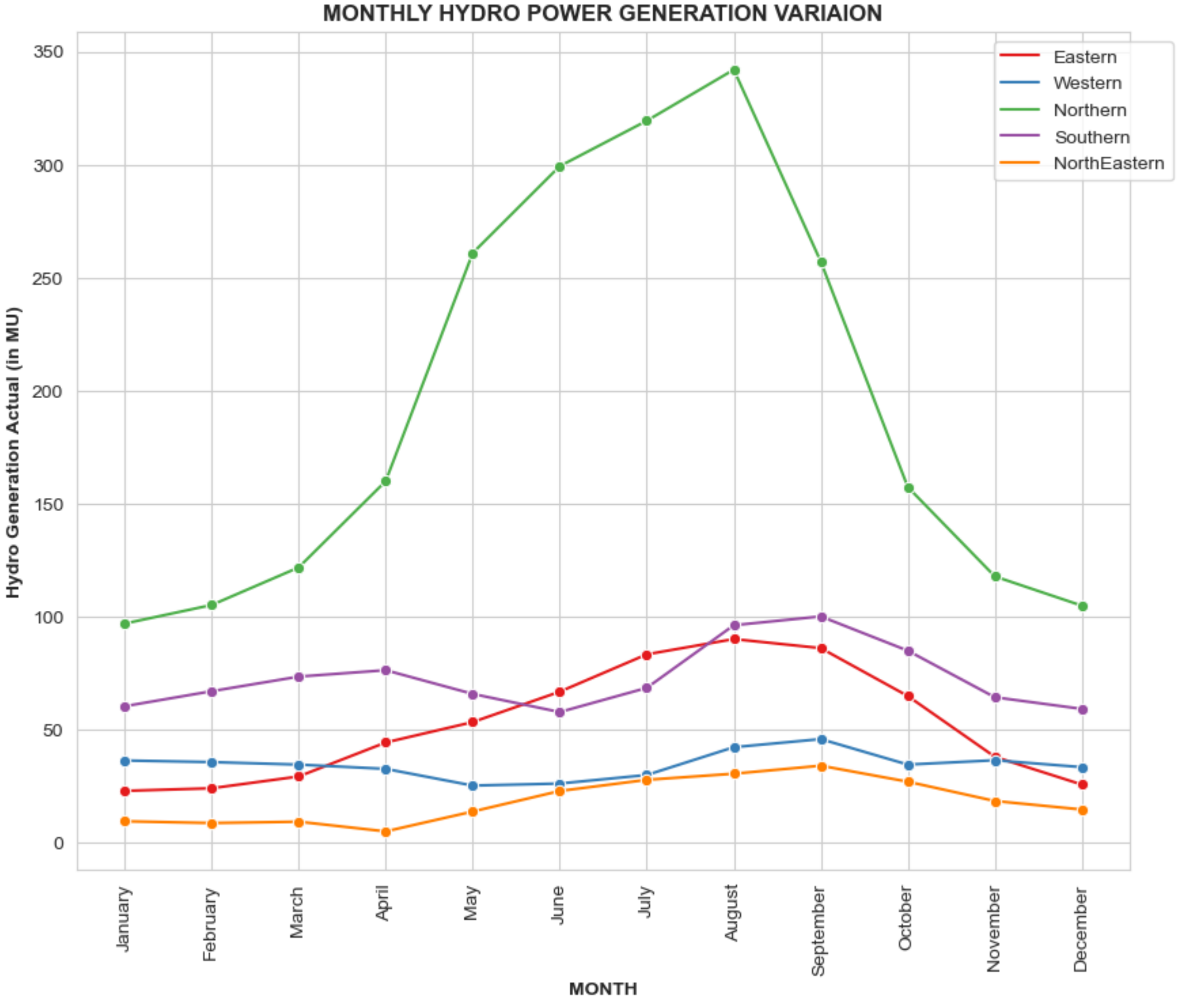
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [666…

```
plt.figure(figsize=(10,8))
ax = sns.lineplot(x='Month', y='Hydro Generation Actual (in MU)', data=merged_pd, 
plt.legend(bbox_to_anchor=(1.05,1))
plt.title("MONTHLY HYDRO POWER GENERATION VARIAION",fontweight="bold")
plt.xticks(rotation=90)
plt.xlabel("MONTH",fontweight="bold")
plt.ylabel("Hydro Generation Actual (in MU)",fontweight="bold")
plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

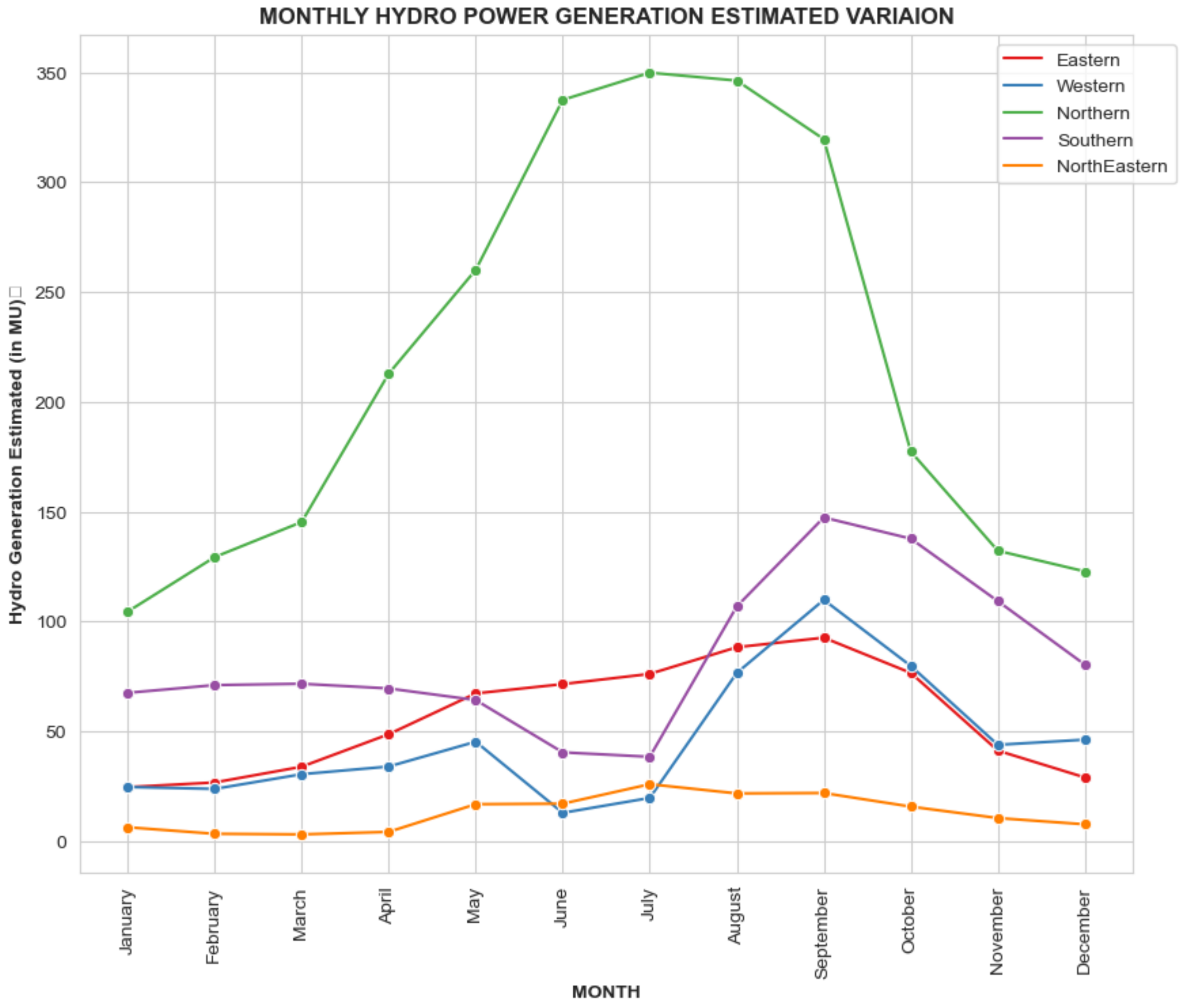


In [667…

```
plt.figure(figsize=(10,8))
ax = sns.lineplot(x='Month', y='Hydro Generation Estimated (in MU)', data=merged_p
plt.legend(bbox_to_anchor=(1.05,1))
plt.title("MONTHLY HYDRO POWER GENERATION ESTIMATED VARIAION",fontweight="bold")
plt.xticks(rotation=90)
plt.xlabel("MONTH",fontweight="bold")
plt.ylabel("Hydro Generation Estimated (in MU)  ",fontweight="bold")
plt.show()
```

```
C:\Users\my hp\anaconda3\lib\site-packages\IPython\core\pylabtools.py:151: UserWar
ning: Glyph 9 ( ) missing from current font.
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



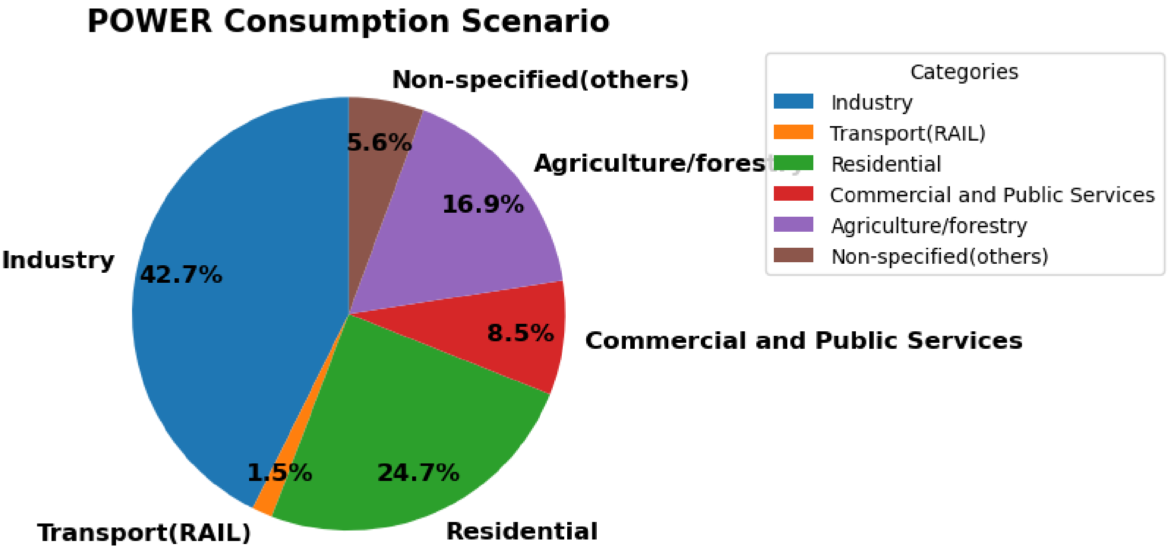
In [673…
```
temp = df3.iloc[7,:]
```

In [674…
```
temp
```

Out[674]:
```
YEAR                                                    2019
Total Electricity Generations                         107335
Electricity Consumption Industry                       45822
Electricity Consumption Transport(RAIL)                 1647
Electricity Consumption Residential                    26552
Electricity Consumption Commercial and Public Services  9120
Electricity Consumption Agriculture/forestry           18171
Electricity Consumption Non-specified(others)           6023
Name: 7, dtype: int64
```

In [194…
```
# Data
labels = ['Industry', 'Transport(RAIL)', 'Residential', 'Commercial and Public Serv
sizes = [45822, 1647, 26552, 9120, 18171, 6023]
total = sum(sizes)

# Create a pie chart
plt.pie(sizes, labels=labels, autopct=lambda pct: "{:.1f}%".format(pct), startangl
       textprops={"fontsize":12,"fontweight":'bold'})

# Add a title
plt.title(f"POWER Consumption Scenario",fontsize=15,fontweight='bold')

# Add a legend
plt.legend(title="Categories", bbox_to_anchor=(1.25,1))

# Show the chart
plt.show()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [196…

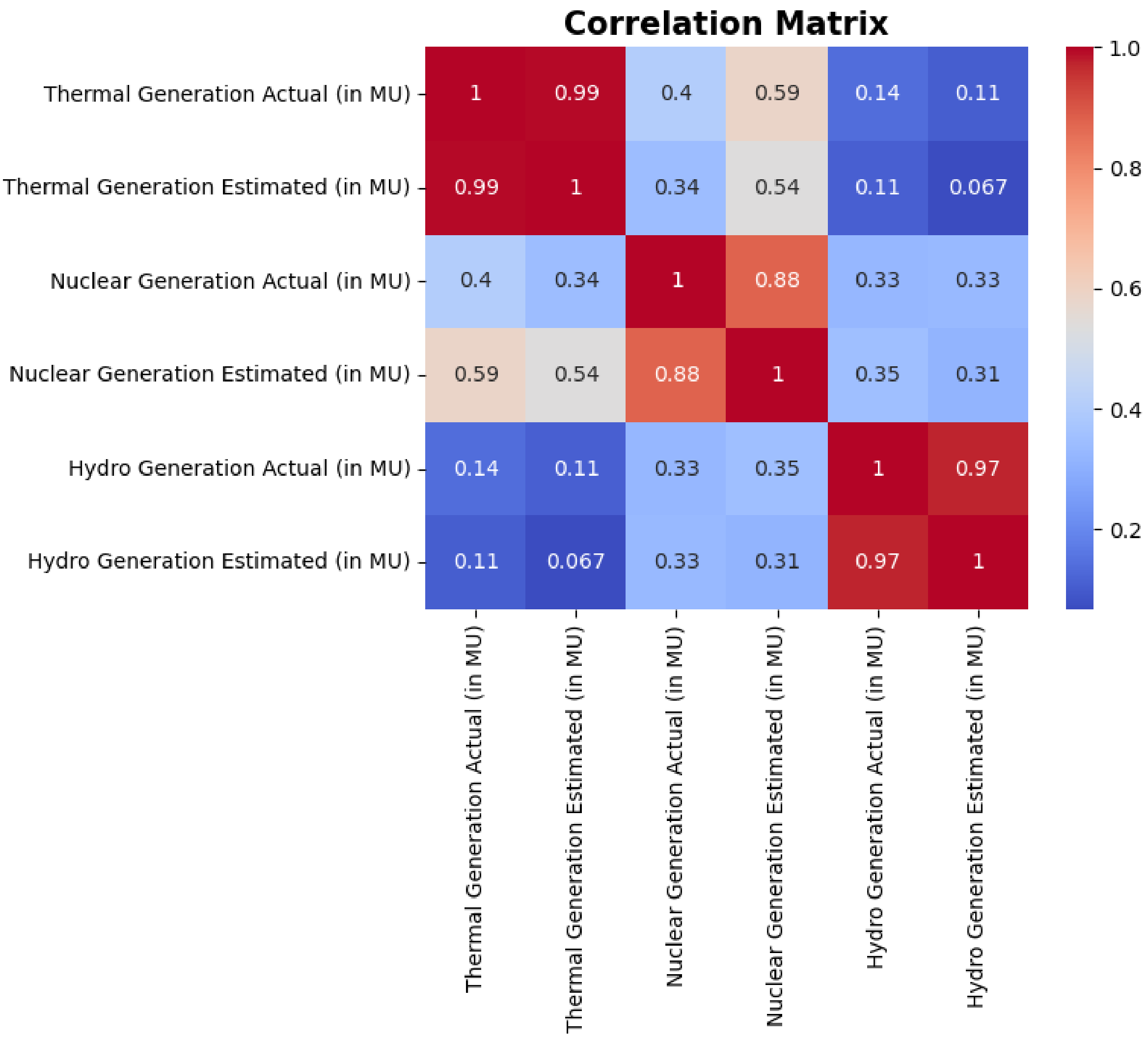
```
req = merged_pd.iloc[:,0:6]
```

In [197…

```
corr = req.corr()
```

In [198…

```
sns.heatmap(corr,annot=True,cmap="coolwarm")
plt.title("Correlation Matrix",fontweight="bold",fontsize=15)
plt.show()
```



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [ ]: