Spatial Data Analysis using Python

Tanay Singhal

Problem Statement

Design a code-book for analysing spatial data set using python programming language which can be used to analyse shapefiles that are most commonly done/analysed using the **ArcGis** which is a very heavy software and is **not open source**.

Significance

It’ll be helpful for those people in data science community who cannot afford the ArcGis software and by introducing python, the certain constraint of having a specific system configuration is also removed as we can use jupyter notebooks in cloud.

And in addition to that the information flow will become more consistent and attractive by the addition of Jupyter

Notebooks.

CONTENTS

1. Extracting the Data and Density Plot (LAB 1 and LAB 2)
2. Overlay, Intersection and Projection (LAB 3)
3. Working with CSV data (LAB 4)
4. Working with Rastar Data (LAB 5)
5. Variogram and Krigging (LAB 6 and LAB 7)

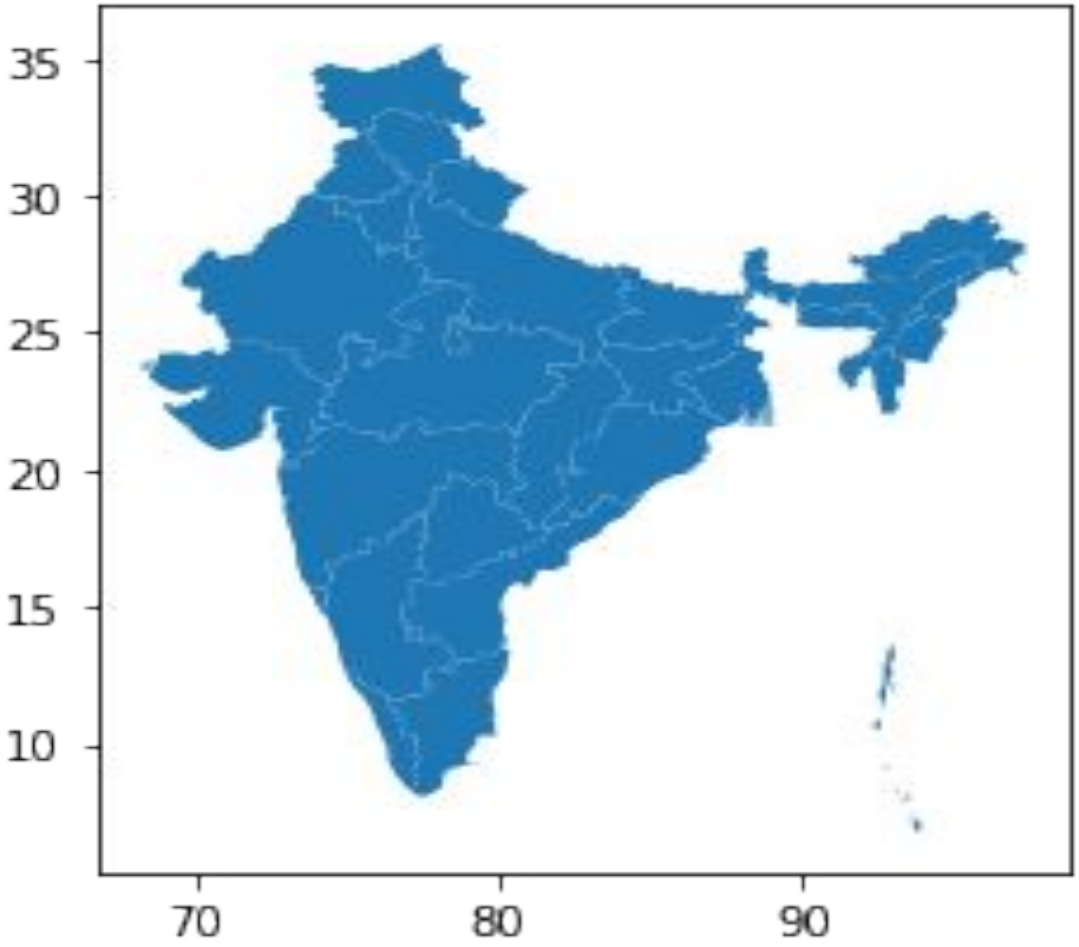
Extracting the Data and Density Plot

Python Implementation of what we did using ArcMap in LAB 1 and LAB 2

```
[11] IND_districts = geopandas.read_file(r"IND_districts_from_DIVA-GIS.shp")
```

```
[12] IND_states = geopandas.read_file(r"IND_states_from_DIVA-GIS.shp")
```

```
[13] IND_national = geopandas.read_file(r"IND_national_from_DIVA-GIS.shp")
     IND_taluks = geopandas.read_file(r"IND_taluks_from_DIVA-GIS.shp")
```


35
30
25
20
15
10
70
80
90

+ Code + Text

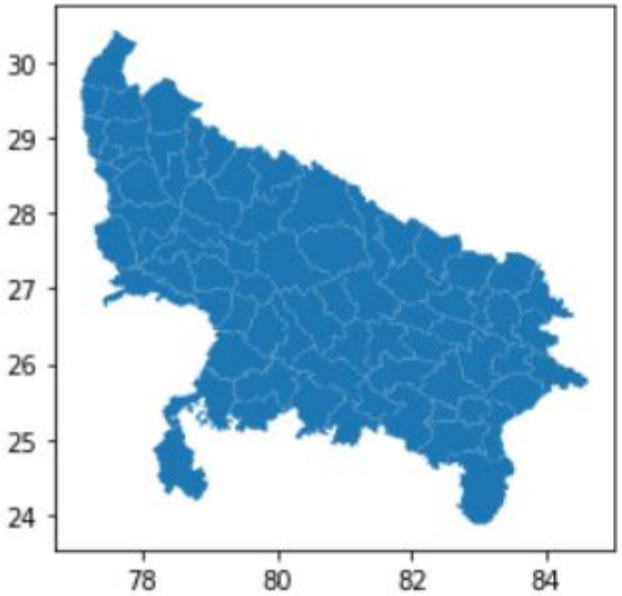
```
[15] IND_districts.head()
```

	ID_0	ISO	NAME_0	ID_1	NAME_1	ID_2	NAME_2	HASC_2	CCN_2	CCA_2	TYPE_2	ENGTYPE_2	NL_NAME_2	VARNAME_2	geometry
0	105	IND	India	1	Andaman and Nicobar	1	Nicobar Islands	IN.AN.NI	0	None	District	District	None	None	MULTIPOLYGON (((92.78778 9.24417, 92.78889 9.2...
1	105	IND	India	1	Andaman and Nicobar	2	North and Middle Andaman	IN.AN.NM	0	None	District	District	None	None	MULTIPOLYGON (((93.64841 14.93487, 93.64917 14...
2	105	IND	India	1	Andaman and Nicobar	3	South Andaman	IN.AN.SA	0	None	District	District	None	None	MULTIPOLYGON (((93.83970 12.32082, 93.85775 12...
3	105	IND	India	2	Andhra Pradesh	4	Anantapur	IN.AD.AN	0	None	District	District	None	Anantpur, Ananthapur	POLYGON ((77.69000 15.17628, 77.69378 15.17347...
4	105	IND	India	2	Andhra Pradesh	5	Chittoor	IN.AD.CH	0	None	District	District	None	Chitoor|Chittor	POLYGON ((78.47611 13.93680, 78.48208 13.93007...

```
[16] IND_UP = IND_districts[IND_districts["NAME_1"] == "Uttar Pradesh"]
```

```
IND_UP.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f09d0ce19d0>
```



```
[55] IND_UP.head()
```

	ID_0	ISO	NAME_0	ID_1	NAME_1	ID_2	NAME_2	HASC_2	CCN_2	CCA_2	TYPE_2	ENGTYPE_2	NL_NAME_2	VARNAME_2	geometry	Perimeter	Area	Density
559	105	IND	India	34	Uttar Pradesh	559	Agra	IN.UP.AG	0	None	District	District	None	None	POLYGON ((77.90040 27.26083, 77.90006 27.25176...	6.359313	0.320664	19.831714
560	105	IND	India	34	Uttar Pradesh	560	Aligarh	IN.UP.AL	0	None	District	District	None	None	POLYGON ((77.83832 28.14633, 77.83477 28.13895...	4.196781	0.354363	11.843183
561	105	IND	India	34	Uttar Pradesh	561	Allahabad	IN.UP.AH	0	None	District	District	None	None	POLYGON ((81.93710 25.74369, 81.94847 25.74222...	5.169219	0.461476	11.201499
562	105	IND	India	34	Uttar Pradesh	562	Ambedkar Nagar	IN.UP.AN	0	None	District	District	None	None	POLYGON ((82.57671 26.60777, 82.58559 26.60561...	3.348613	0.210339	15.920109
563	105	IND	India	34	Uttar Pradesh	563	Amethi	None	0	None	District	District	None	None	POLYGON ((81.60329 26.63856, 81.60937 26.63056...	4.596429	0.298901	15.377744

```
IND_UP['Perimeter'] = IND_UP.length
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: Geometry is in a geographic CRS. Results from 'length' are likely incorrec

  """Entry point for launching an IPython kernel.
/usr/local/lib/python3.7/dist-packages/geopandas/geodataframe.py:1351: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  super().__setitem__(key, value)
```

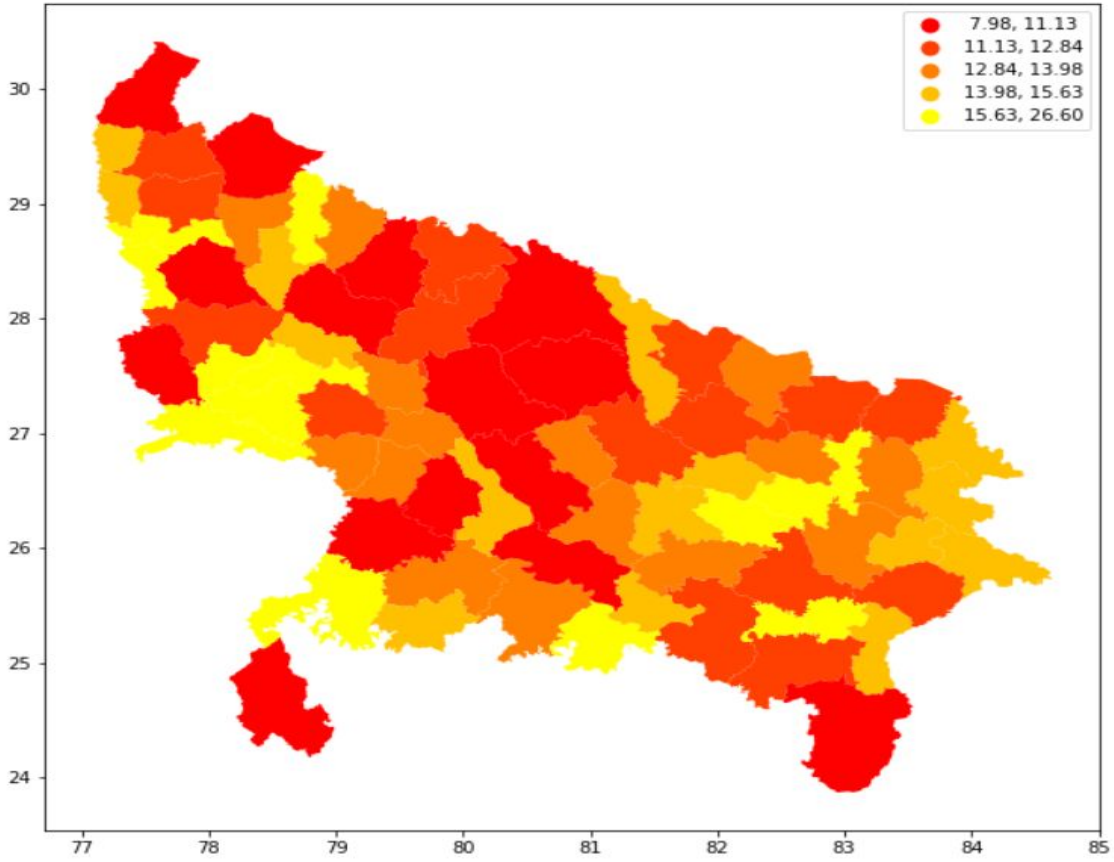
```
[23] IND_UP['Area'] = IND_UP.area
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: Geometry is in a geographic CRS. Results from 'area' are likely incorrect.

  """Entry point for launching an IPython kernel.
/usr/local/lib/python3.7/dist-packages/geopandas/geodataframe.py:1351: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  super().__setitem__(key, value)
```

```
IND_UP['Density'] = IND_UP['Perimeter'] / IND_UP['Area']
```

```
/usr/local/lib/python3.7/dist-packages/geopandas/geodataframe.py:1351: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

Density Plot



Overlay, Intersection and Projection

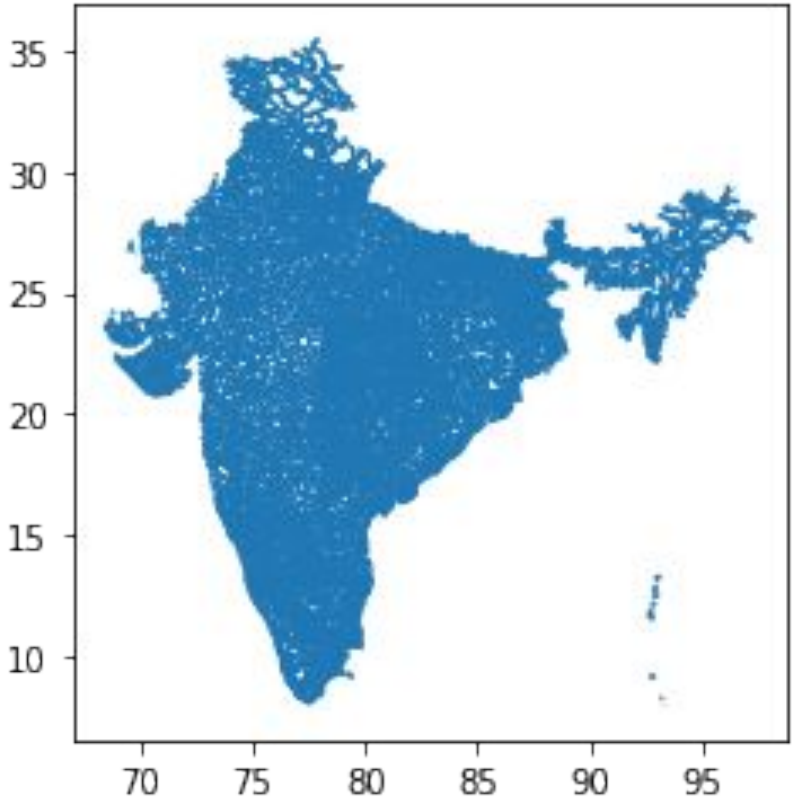
Python Implementation of what we did using ArcMap in LAB 3

```
[ ]  roads = geopandas.read_file(r"IND_rds/IND_roads.shp")
```

```
roads.head()
```

	MED_DESCRI	RTT_DESCRI	F_CODE_DES	ISO	ISOCOUNTRY	geometry
0	Without Median	Secondary Route	Road	IND	INDIA	LINESTRING (77.82774 35.50128, 77.82575 35.49950)
1	Without Median	Secondary Route	Road	IND	INDIA	LINESTRING (77.82373 35.50099, 77.82575 35.49950)
2	Without Median	Secondary Route	Road	IND	INDIA	MULTILINESTRING ((77.82575 35.49950, 77.82975 ...

Roads Plot



Checking Coordinate System for Projection:

```
IND_UP.crs
```

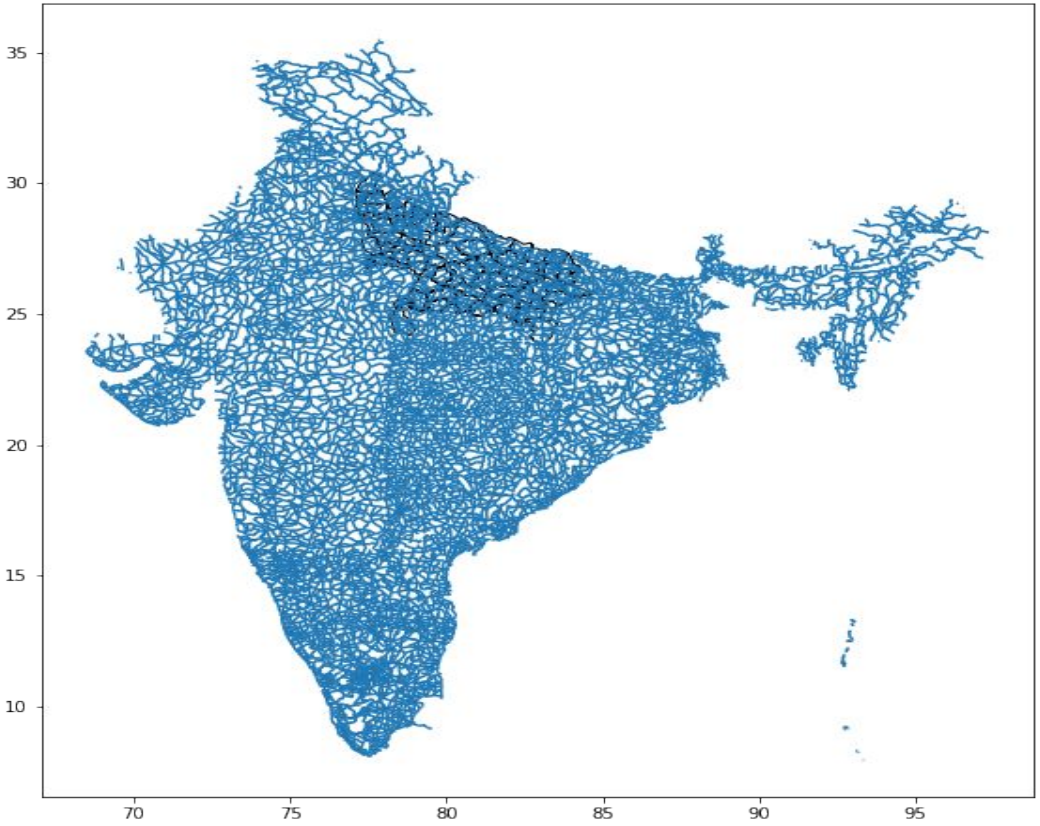
```
<Geographic 2D CRS: EPSG:4326>
Name: WGS 84
Axis Info [ellipsoidal]:
- Lat[north]: Geodetic latitude (degree)
- Lon[east]: Geodetic longitude (degree)
Area of Use:
- name: World.
- bounds: (-180.0, -90.0, 180.0, 90.0)
Datum: World Geodetic System 1984 ensemble
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich
```

For Roads data

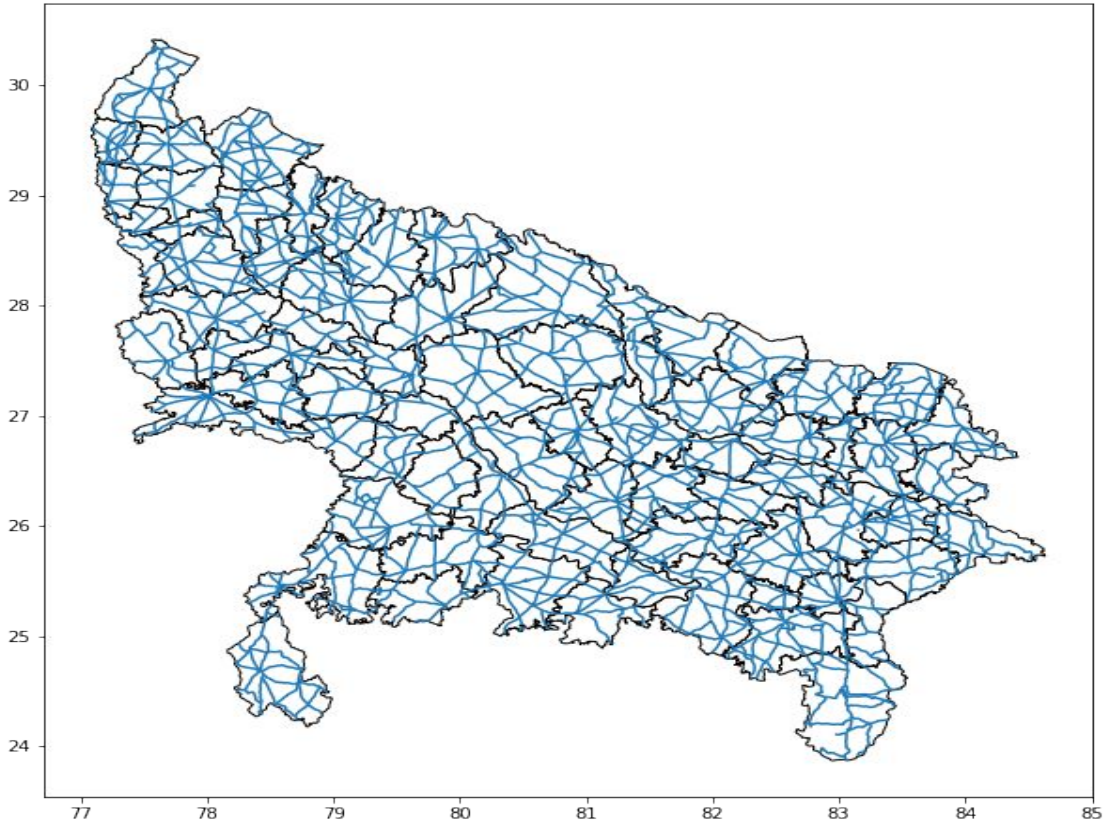
```
roads.crs
```

```
<Geographic 2D CRS: EPSG:4326>
Name: WGS 84
Axis Info [ellipsoidal]:
- Lat[north]: Geodetic latitude (degree)
- Lon[east]: Geodetic longitude (degree)
Area of Use:
- name: World.
- bounds: (-180.0, -90.0, 180.0, 90.0)
Datum: World Geodetic System 1984 ensemble
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich
```

Overlaying the data



Intersection



Working with CSV data

Python Implementation of what we did using ArcMap in LAB 4.

```
[ ] import pandas as pd
```

```
[ ] df = pd.read_csv('Final_Data_excel.csv')
```

```
[ ] df
```

	FID	FID_Final_	OBJECTID_1	FID_fishne	OBJECTID	Id	Shape_Leng	FID_UttarP	ID_0	ISO	...	Shape_Le_1	Shape_Area	FID_Ground	STATE	DISTRICT	STATION	Latitud
0	0	4	5	79	80	0	30341.553632	0	105	IND	...	30341.553632	5.747599e+07	137	UTTAR PRADESH	SONBHADRA	Babhani	23.953
1	1	11	12	179	180	0	30332.888089	0	105	IND	...	30332.888089	5.744411e+07	2734	UTTAR PRADESH	SONBHADRA	Babhani Q	24.016
2	2	9	10	177	178	0	30351.626952	0	105	IND	...	30351.626952	5.751511e+07	422	UTTAR PRADESH	SONBHADRA	Jarha	24.007
3	3	10	11	178	179	0	30342.233579	0	105	IND	...	30342.233579	5.747951e+07	2093	UTTAR	SONBHADRA	Chopan	24.022

Converting the Dataframe into GeoDataFrame

```
gdf = geopandas.GeoDataFrame(
    df, geometry=geopandas.points_from_xy(df.Longtitude, df.Latitude_1)
)
```

```
gdf
```

	DISTRICT	STATION	Latitude_1	Longtitude	Level__m_	Grnd_lvl_2	geometry
	SONBHADRA	Babhani	23.953889	83.079167	11.15	11.15	POINT (83.07917 23.95389)
	SONBHADRA	Babhani Q	24.016111	83.091111	-	0.00	POINT (83.09111 24.01611)
	SONBHADRA	Jarha	24.007778	82.903333	8.45	8.45	POINT (82.90333 24.00778)
	SONBHADRA	Chopan	24.022222	83.026389	9.55	9.55	POINT (83.02639 24.02222)

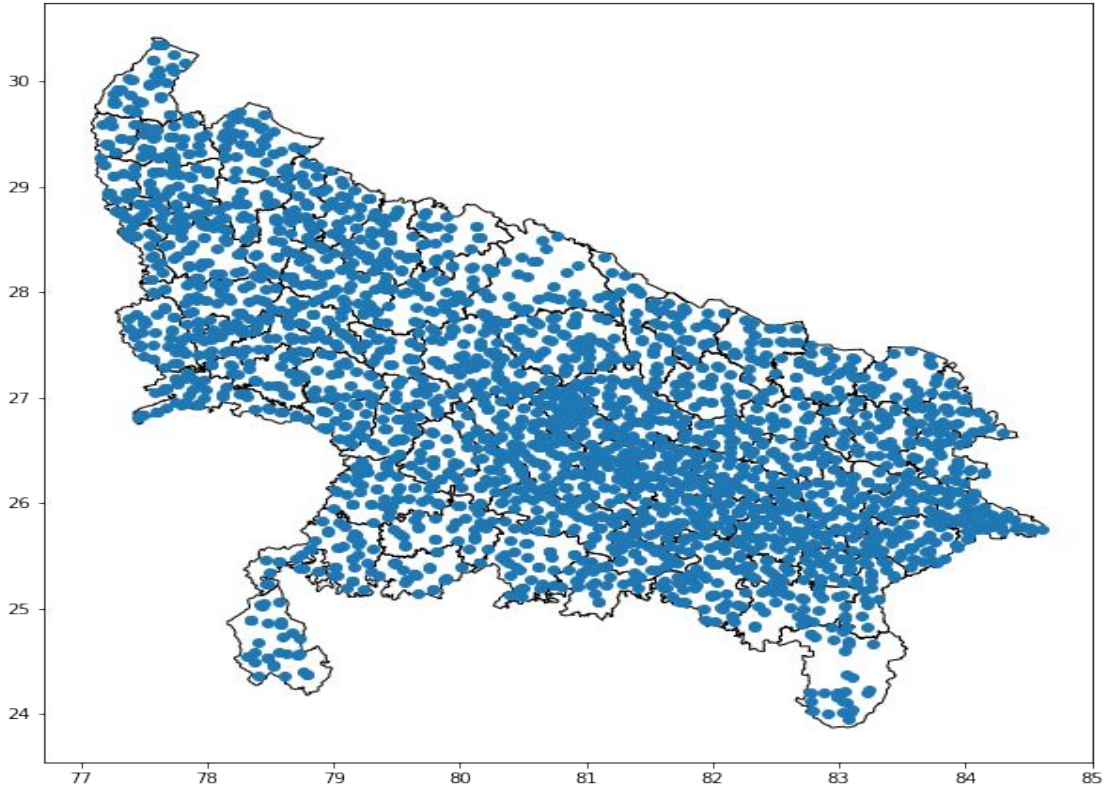
Setting the Projection of Spatial Dataset

```
gdf_2 = gdf.set_crs("EPSG:4326")
```

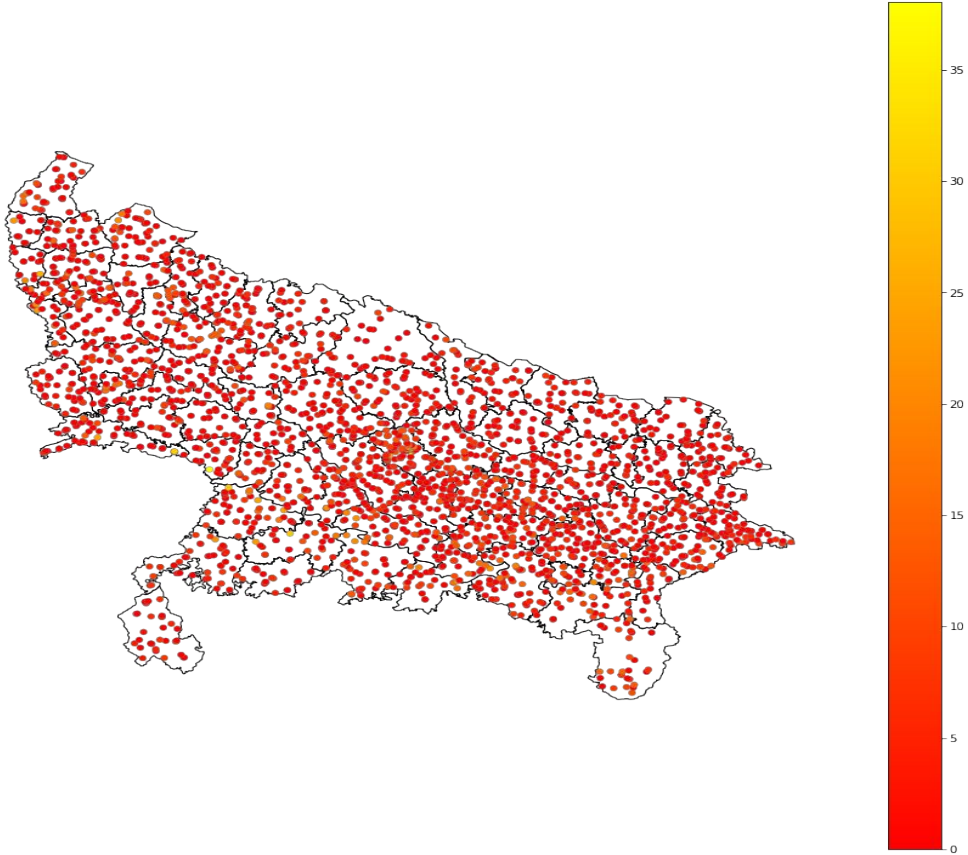
```
gdf_2.crs
```

```
<Geographic 2D CRS: EPSG:4326>
Name: WGS 84
Axis Info [ellipsoidal]:
- Lat[north]: Geodetic latitude (degree)
- Lon[east]: Geodetic longitude (degree)
Area of Use:
- name: World.
- bounds: (-180.0, -90.0, 180.0, 90.0)
Datum: World Geodetic System 1984 ensemble
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich
```

Overlay Plot

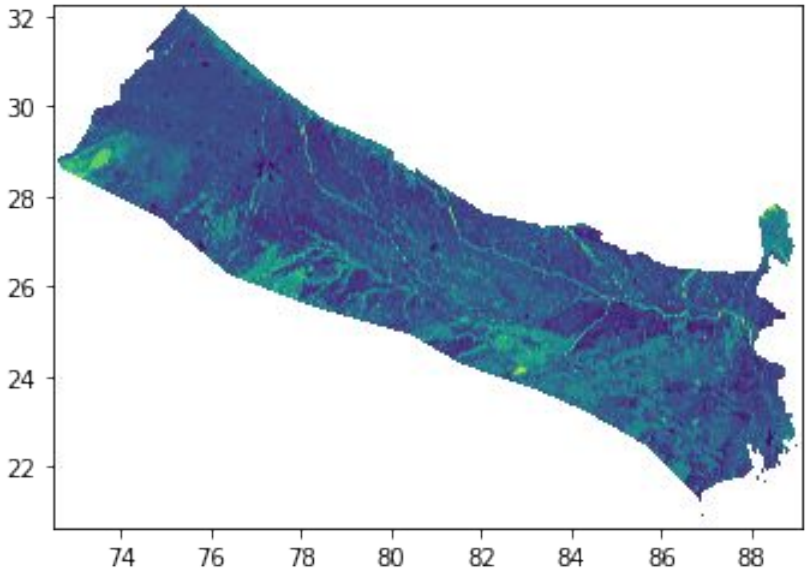


Plot w.r.t to ground water depth

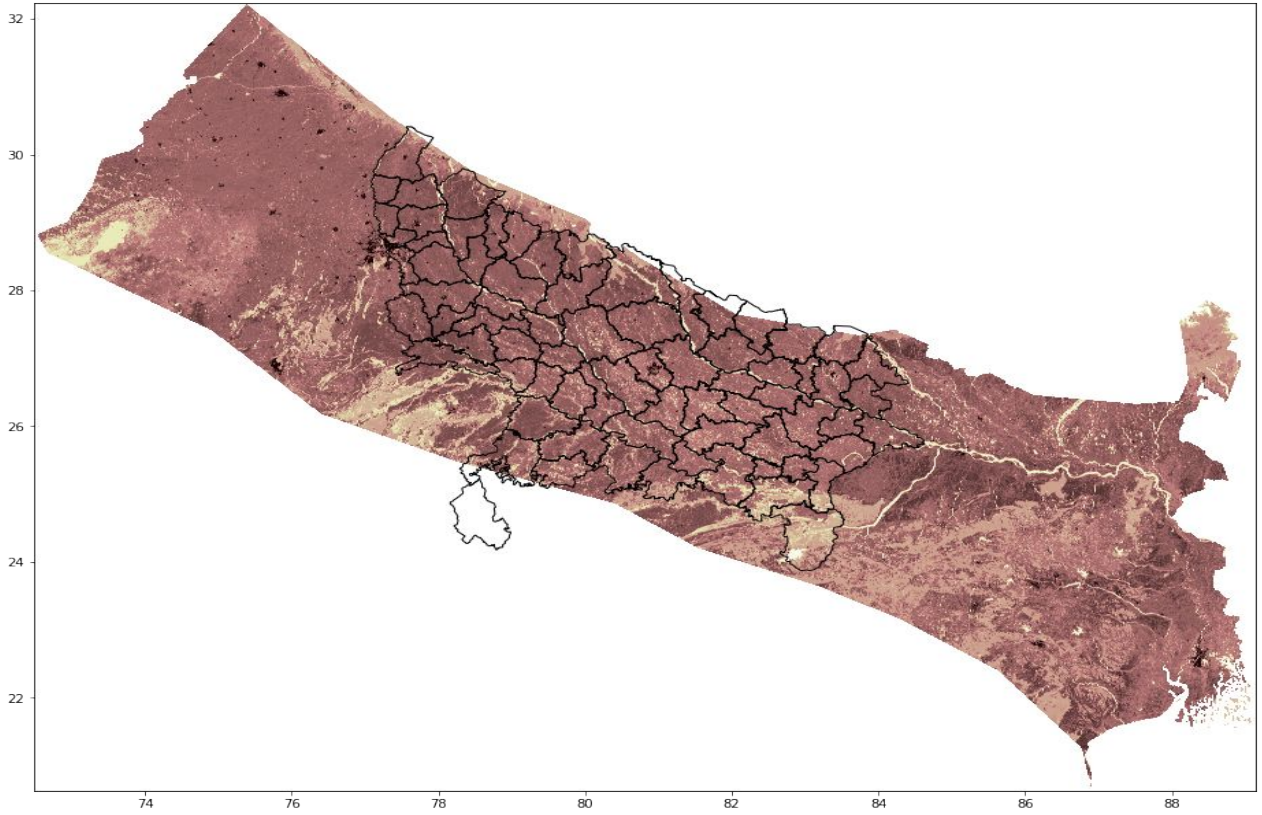


Working with Raster Data

Work done using **raterio** package:



Overlay with Raster Data

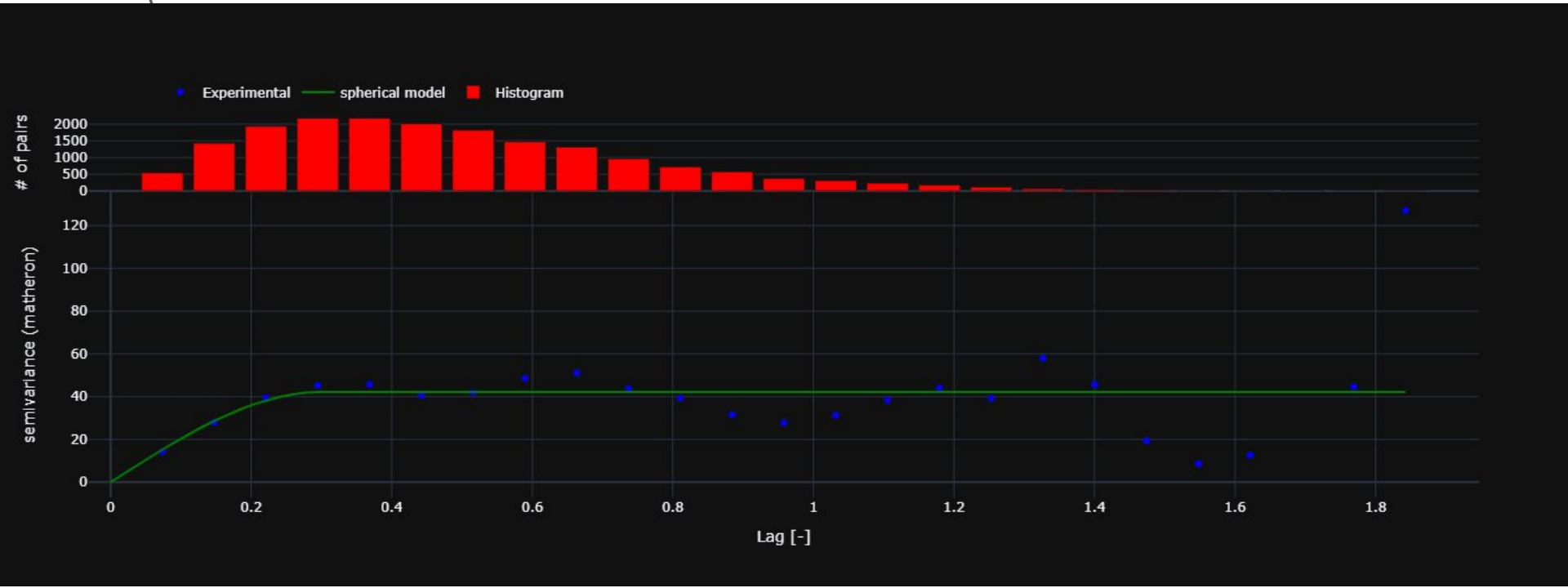


Variogram and Krigging

Work done using **skgstat** package:

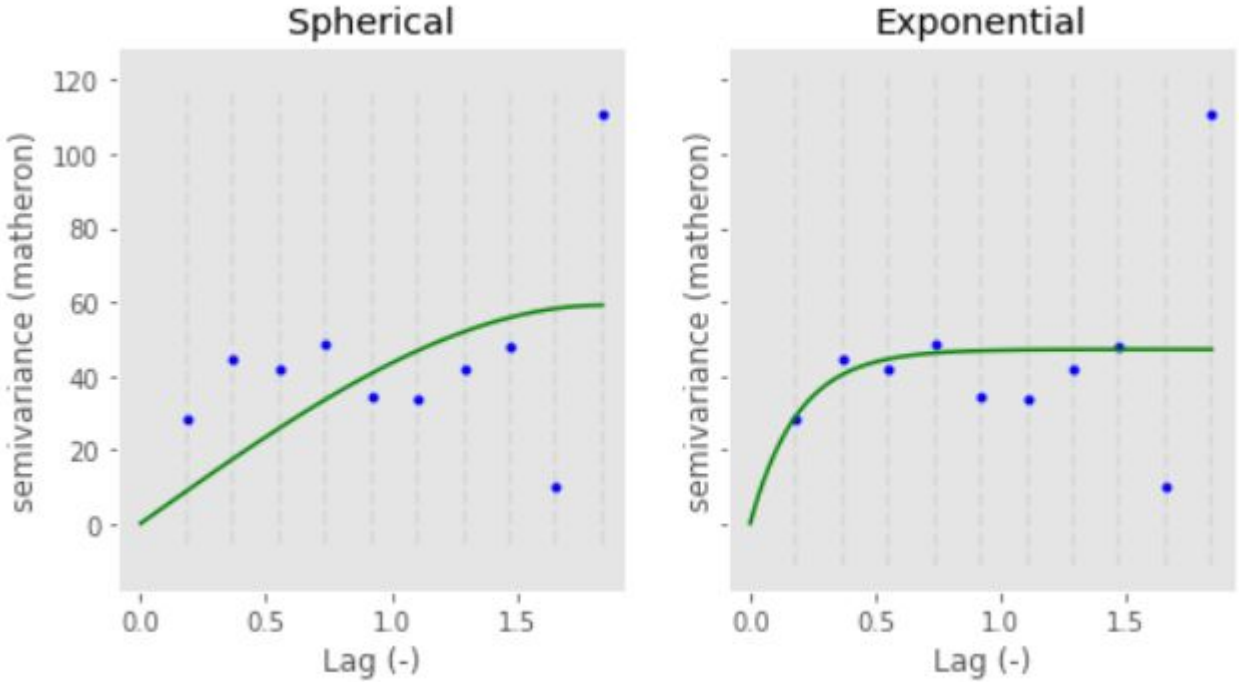
Data set used: “`Final_Data_excel.csv” and "up_state_gwl_v6.csv"`

Variogram Plot

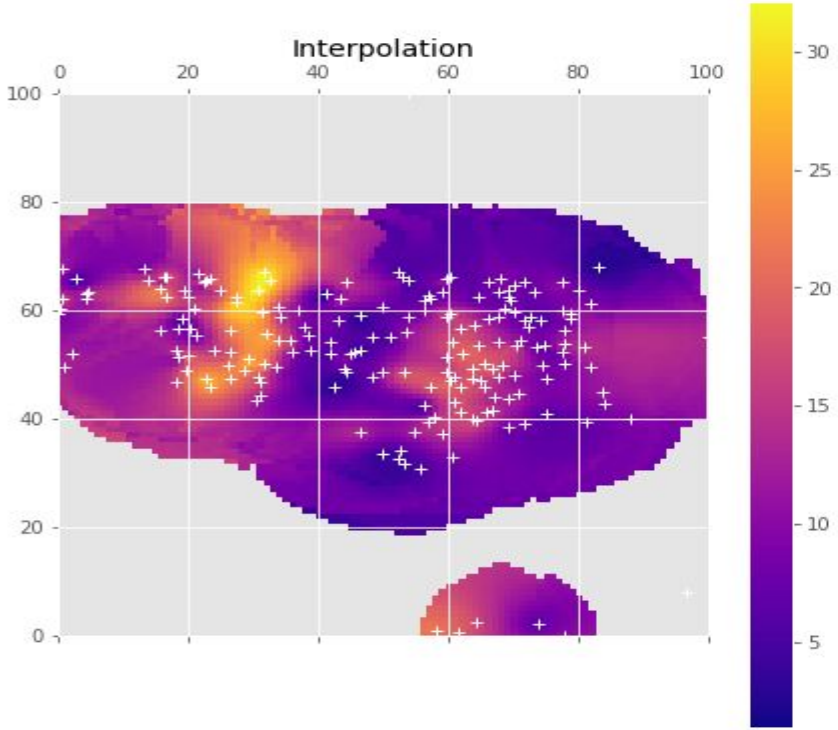


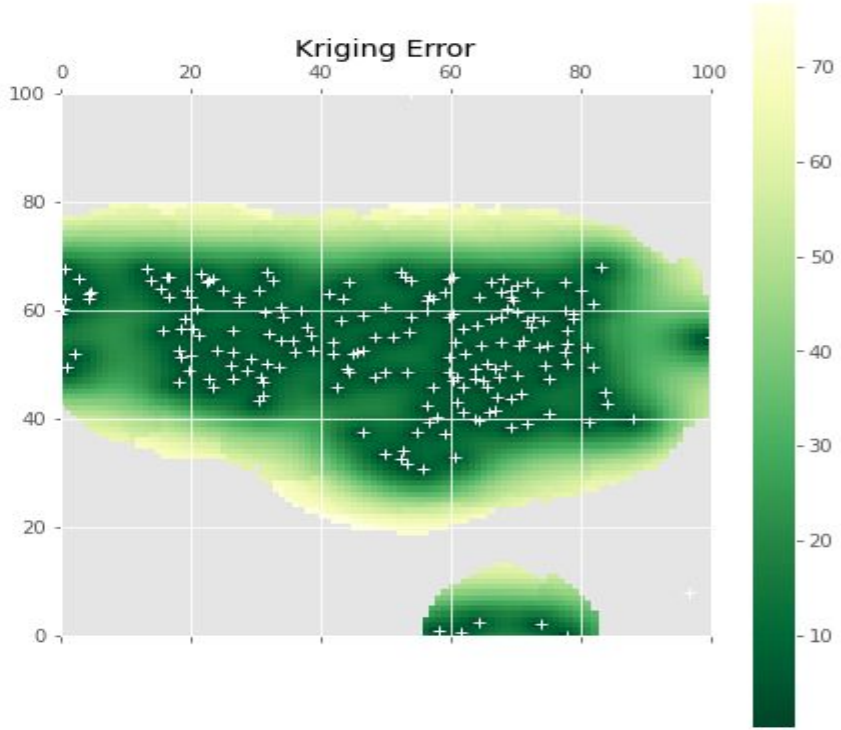
Variogram Plot

\



Krigging Result





Thank You