Spatial Analysis and Geospatial Analysis   
Data Science in Python



**Anish Mahapatra**

Contents

[Introduction 4](#_Toc116429394)

[What is Geospatial Data? 4](#_Toc116429395)

[How to work with spatial data in python? 5](#_Toc116429396)

[The Adoption of Python in GIS 8](#_Toc116429397)

[Desktop and Interactive computational geospatial programming Applications of Python in GIS 8](#_Toc116429398)

[The Basics 10](#_Toc116429399)

[Text 10](#_Toc116429400)

[Vector 10](#_Toc116429401)

[Raster 11](#_Toc116429402)

[Coordinate Reference System (CRS) 11](#_Toc116429403)

[Map Projections 11](#_Toc116429404)

[Georeferencing 12](#_Toc116429405)

[Geocoding 12](#_Toc116429406)

[Python Geo-spatial libraries 13](#_Toc116429407)

[What can you put into geometry? 13](#_Toc116429408)

[Loading data 13](#_Toc116429409)

[CRS 13](#_Toc116429410)

[Visualization 13](#_Toc116429411)

[Reading in data 13](#_Toc116429412)

[Plotting the outbreak 13](#_Toc116429413)

[How Jupyter Notebook is used in GIS 13](#_Toc116429414)

[Conclusion 14](#_Toc116429415)

[Frequently Asked Questions (FAQs) 14](#_Toc116429416)

[Is Python or R better for spatial analysis? 14](#_Toc116429417)

[How do you do geospatial analysis in Python? 15](#_Toc116429418)

[What are the 5 concepts of spatial analysis? 15](#_Toc116429419)

[What is spatial analysis method? 15](#_Toc116429420)

[What is spatial analysis used for? 15](#_Toc116429421)

[Author 16](#_Toc116429422)

# Introduction

Spatial data is any form of data that helps us directly or indirectly reference a specific location or geographical area either on the surface of the earth or elsewhere. Geographic Information systems, or GIS, in short, is the most common method of processing and analysing spatial data. This includes the entire stack of data management, manipulation, customisation, visualisation and analysis of the spatial data. GIS is a combination of programs working together, aiding users to understand and make sense of spatial data.

For example, if you were to work with GIS data for any project pertaining to spatial data within your geographical area, you would be dealing with different types of data such as vector data (lines - street data), polygons (boundaries of a geographic area) and point locations (buildings, skyscrapers, schools, etc.). These datasets would each exist as a layer of their own in GIS, where the placement of these layers becomes crucial for your understanding and analysis

The applications of GIS field and study extend much further than digital mapping and cartography, consisting of a multitude of categories such as remote sensing, spatial analysis and geo-visualisation. Here, in each of these applications, the spatial data becomes much more complex to use.

With this article, we shall tap into the understanding of spatial data and geospatial data analysis with Python through some examples and also how to perform operations from spatial statistics python libraries. We shall also go through a few basics and prerequisites that will be necessary for understanding spatial data, with how Python for spatial analysis has taken centre stage in today’s world for the application of GIS. Jupyter Notebook’s relevance is also included that allows us to work with two of the most popular software for GIS which is ArcGIS (Online cloud-based mapping and analysis solution) and QGIS (Quantum Geographic Information System, a free, open-source GIS software with many free online resources and maps available to download) for spatial data analysis with Python. This can be learnt in the [data science online courses](https://www.knowledgehut.com/data-science-courses).

# What is Geospatial Data?

Let us first try to understand what geospatial data is and look at a few examples. Geospatial data is information about describing objects, events, and other features with a location on or near the earth’s surface. The geospatial data combines the information about the location, which typically consists of the coordinates of the earth and also the attribute information, which talks about the characteristics, events, or phenomena regarding the objects, along with its temporal information, which is the life span or time at which the attributes and location exist. It typically consists of large datasets of spatial data obtained from multiple sources in different formats, including telephone data, satellite imagery, weather data, etc.

Much of the geospatial data that is available is open source (freely accessible to users) cause it consists of data that can reference roads, localities, water bodies, and public amenities, which are of general interest to a wide range of users and is helpful for a number of purposes to both public and private organisations. This open-source data is mainly made accessible through open standards, which are heavily supported within the geospatial community. This is due to the fact that primarily, a large number of agencies, both locally and globally, are involved in the generation of geospatial data, and secondarily because of the wide range of applications.

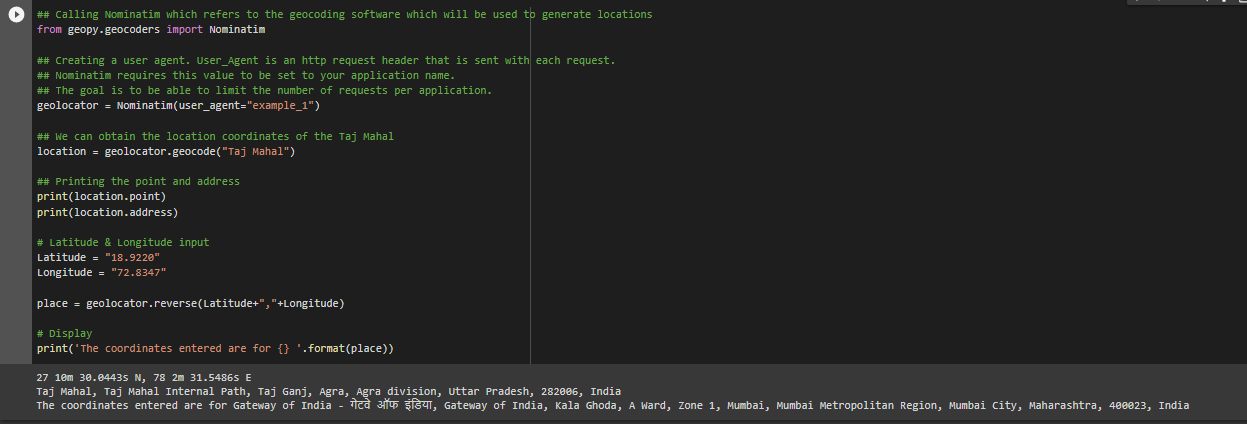
Geospatial analytics is used mainly to add timing and location to traditional data. Maps, graphs, statistics, and cartograms that depict recent and historical developments can be included in these visualisations. This added background enables a fuller understanding of the events. Easy-to-identify visual patterns and graphics are used to convey insights that might be missed in a large spreadsheet.

In the next section, we will be looking at performing geospatial data analysis in Python that employs the python spatial analysis library

# How to work with spatial data in Python?

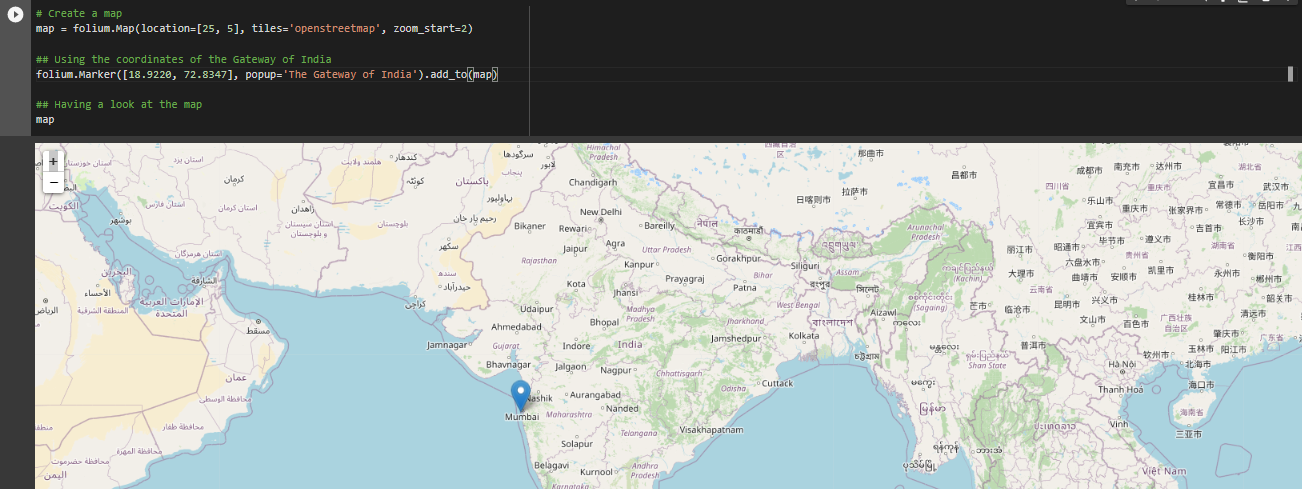
Now that we have understood what spatial/geospatial data looks like, we shall now look into a few exercises which will introduce us to how to use Python for geospatial data analysis. We shall start if with showing a few basic functions within the GeoPy library from Python, which uses third-party geocoders and other data sources to quickly find the coordinates of addresses, cities, nations, and landmarks all around the world. Each geolocation service that we use, such as Google Maps, Bing Maps, etc., has its own class in geopy.geocoders which abstracts the service API.

**Exercise 1**: Let’s simply begin with checking if we can get the coordinates by entering the name of a popular place and vice versa. Here we shall be using the Taj Mahal as a reference for our exercise. This example will serve as an introduction to working with coordinates and locations around the world.



*Image by* [*Author*](https://www.linkedin.com/in/anishmahapatra/)

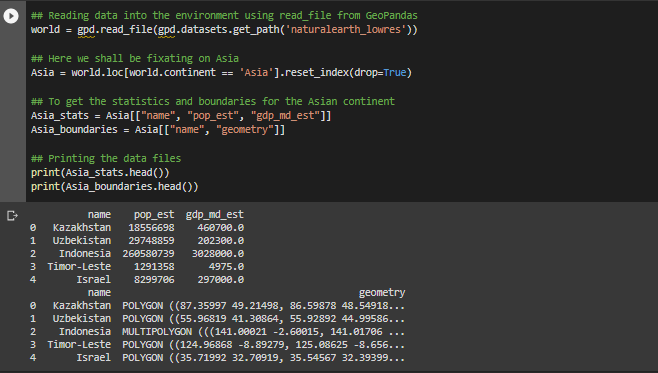
**Exercise 2**: Here, we shall locate the Gateway of India on the map. For doing this, we shall be using the folium. Folium draws on the data manipulation and mapping prowess of the Python ecosystem and the leaflet.js package. The package makes it easier to visualise data that has been manipulated in Python on an interactive leaflet (Leading open-source JS library for mobile-friendly interactive maps) map. It enables both the binding of data to a map for choropleth visualisations as well as passing rich vector/raster/HTML visualisations as markers on the map.



*Image by* [*Author*](https://www.linkedin.com/in/anishmahapatra/)

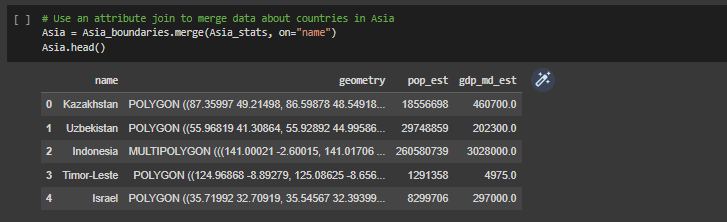
In the next couple of exercises, we will be looking into using the GeoPandas library and how to perform a few operations using this library. GeoPandas is a python library that expands the datatypes that pandas use to include geometric types for spatial operations. Shapely performs geometric operations. GeoPandas also uses matplotlib for charting and Fiona for file access.

**Exercise 3:** Here, we shall look into reading spatial data into the environment. Spatial data is stored as shapely data. As mentioned previously, GeoPandas makes use of Shapely’s geometric objects, which means the geometries are stored in a column called geometry (default column name), as shown below, which are shapely Polygon objects.



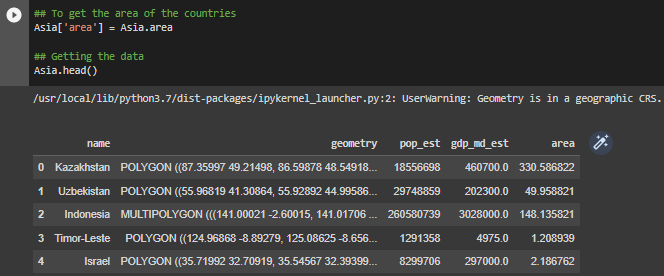
*Image by* [*Author*](https://www.linkedin.com/in/anishmahapatra/)

Once we have read the data into the environment using the read\_file function from GeoPandas and performed a few transformations, we will go ahead and apply joins using a similar function as pandas.DataFrame.join() which is GeoPandas,DataFrame.join()



*Image by* [*Author*](https://www.linkedin.com/in/anishmahapatra/)

**Exercise 4:** In this next exercise, we shall see how we can calculate the area of the polygon that has been listed under the countries in Asia. Here by using the function of area for spatial data, we can have it calculated for us.



*Image by* [*Author*](https://www.linkedin.com/in/anishmahapatra/)

# The Adoption of Python in GIS

In the above section, we looked into spatial analysis in Python. Here we look into what makes Python the go-to language for spatial data and GIS. Python, in recent years, has seen widespread adoption across many domains. The rich and versatile libraries within Python make it well-suited for any sort of project one would want to pick up. This can be majorly attributed to two reasons:

* It supports both structured programming and object orientation which makes it a **multi-paradigm** programming language
* As an interpreted language, Python lends itself to **rapid prototyping and development cycles.**

GIScience (Geographic Information Science) has found a great receptive audience in Python due to the emphasis on readability, support across platforms, and lower start-up costs. Python offers flexibility through various modes of development for geospatial programming. Let’s look into the applications to understand how effective Python for geospatial analysis is.

## Desktop and Interactive computational geospatial programming Applications of Python in GIS

* **ArcGIS** (post version 9.0) has included Python as a core scripting language, where the **ArcPy** package provides a platform for geoprocessing tools, functions, classes, and modules.
* **QGIS** (Open-source GIS package) offers a python console through its GUI, providing an interactive shell to support experimentation with QGIS workshop allowing users to build workflows within existing sessions. Python has also been used to develop a processing framework which is a geoprocessing environment for running native or third-party algorithms within QGIS
* Python has also been used for developing standalone geospatial applications. These Python-based packages contain advanced geospatial capabilities inside a GUI. A Few examples are:
  + **GeoDaSpace**: Spatial regression analysis package
  + **CAST**: Crime Analytics in Space-Time
  + **STARS**: Space-Time Analysis of Regional Systems

We shall talk about the multiple spatial analysis python libraries using a table to talk about a few of the popular or commonly encountered packages from each layer in the stack.

|  |  |  |
| --- | --- | --- |
| **Layer** | **Package** | **Description** |
| Spatial Data Analysis | PySAL | To analyse clean spatial data in an interactive computational environment |
|  | GeoPandas | Pandas and shapely are combined to aid in working with geospatial vector data sets |
|  | GDAL/OGR | Allows working with both vector and raster data |
| Spatial Modelling | spint | SPatial INTeraction Modeling package for a collection of tools for studying spatial interaction data |
|  | mesa | A python framework for agent-based modelling |
|  | clusterpy | It is a library of spatially constrained clustering algorithms |
| geovisualization | cartopy | A package designed for geospatial data processing in order to produce maps and other geospatial data analysis |
|  | folium | For creating visualisations on interactive leaflet maps |
|  | datashader | A data rasterisation pipeline for automating the process of creating meaningful visuals for big data |
| geoprocessing | shapely | A package for manipulation and analysis of planar geometric objects |
|  | rasterstats | A package for summarising raster datasets based on the geometrics of vector |
|  | pyproj | A package for performing cartographic transformations and geodetic computations |

# The Basics

## Text

Adding text to a map that only describes geographic features on a map improves the visualisation of geographic information immensely. The main types of text defined are labels, annotation and graphic test.

* **Label:** A piece of text that is automatically placed and consists of text string based on the feature attributes. Labels offer the easiest and fastest way to add descriptive text to the map. Example: Adding dynamic labelling for all the major cities in a country \*\*\*\*
* **Annotation:** These can be used to describe particular features or add general information to the map that is being created. Annotations provide more flexibility in terms of appearance and placement since we will have the ability to select individual text pieces and edit them \*\*\*\*
* **Graphic Text:** This is useful in adding information on and around the map that exists in page space. Use graphic text if you want to display text on your map page that does not change as you pan and zoom the map

## Vector

The most common type of data loaded into a GIS software program is vector data. It represents geographic data as points, lines, or polygons.

The vector data is split into three types which are:

* **Point data:** It is most frequently used to represent discrete data points and nonadjacent features. Since points have no dimensions, this dataset cannot be used to estimate either length or area. Additionally, point features are utilised to represent abstract points. For example, point locations can be utilised for city names and locations.
* **Line data**: Linear features are represented by line (or arc) data. Streets, pathways, and rivers are typical examples. Since line features only have one dimension, length is the only thing they can be utilised for. The line features consist of a starting and ending point
* **Polygons**: Areas like the boundary of a city (on a large-scale map), a lake, or a forest is represented by polygons. Since polygon features are two-dimensional, they can be used to calculate a geographic feature’s area and perimeter.

## Raster

A raster, in its most basic form, is made up of a matrix of cells (or pixels) arranged into rows and columns (or a grid), each containing a value that represents some type of information. Raster includes digital aerial photos, satellite imagery, digital photos, and even scanned maps.

Data in raster formats represent real-world phenomena:

1. Thematic data, commonly referred to as discrete data, represents elements like soil or land use information.
2. Continuous data depict phenomena like temperature or height or spectral data like satellite images and aerial photos.
3. Maps, drawings, and photographs of buildings are examples of pictures.

## Coordinate Reference System (CRS)

Without coordinate reference system (CRS) information that can be used by geospatial applications to display and manipulate the data correctly, a data structure cannot be considered geospatial. CRS information uses a mathematical model to link data to the earth’s surface. CRS then defines how the two-dimensional, projected map in your GIS relates to real places on the earth.

Components of CRS:

* **Datum:** A representation of the earth’s form. It specifies the starting point (i.e., where is (0, 0)?) and has angular units (i.e., degrees), so the angles refer to a significant location on the planet.
* **Projection:** The angular measurements on the round earth are mathematically transformed to a flat surface. Typically, the units connected to a given projection are linear
* **Additional Parameters:** The purpose of the additional parameters is to establish the complete coordinate reference system; additional factors are often required. A definition of the map’s centre is a typical extra parameter.

## Map Projections

In cartography, one of the numerous techniques used to depict the three-dimensional surface of the globe or another spherical body on a two-dimensional plane is map projection (mapmaking). Usually, but not always, this process is a mathematical procedure (some methods are graphically based).

## Georeferencing

Georeferencing is defining the location of your raster data using map coordination and assigning the coordinate system of the map frame. Raster data can be viewed, queried, and analysed with other geographic data using georeferencing.

There are generally four steps involved in Georeferencing process:

1. Adding the raster data that is to be aligned with the projected data
2. The georeferencing tab can be used to create control points that enable connection to the raster data to the known positions on the map
3. Reviewing the control points and the errors
4. Finally, saving the georeferencing results when the alignment looks satisfactory

## Geocoding

Finding geographic coordinates for place names, street addresses, and codes is a process known as geocoding (e.g., zip codes). Preprocessing and standardising the format of the data you will be geocoding are often steps in the data cleansing process that come before geocoding. *The resulting locations are output as geographic features with attributes which can be used for mapping or spatial analysis.* There are many uses for geocoding, ranging from straightforward data analysis to customer and business management to distribution strategies. With geocoded addresses, you can visualise the locations of the addresses and spot patterns in the data.

# Python Geospatial libraries

In this section, we will go over the two most powerful libraries from Python. When it comes to something like Geospatial analysis, it is important to use the right packages, and in Python, they are shapely and GeoPandas, which is also taught in the [bootcamp data science](https://www.knowledgehut.com/data-science/data-science-bootcamp-training).

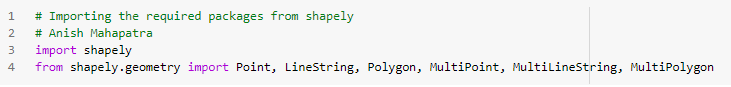
* **GeoPandas**: GeoPandas is a package that enables us to work more efficiently with geospatial data using Python. It leverages pandas as a base library to allow the user to perform spatial analysis on various geometric types. A combination of pandas and shapely help provide a high-level interface to various geometries.
* **Shapely**: Shapely is a popular library that helps with the analysis of objects and helps us manipulate planar geometry effectively.

## What can you put into geometry?

Follow along, the shapely objects are as follows; we have polygons, lines and points. One of the features that helps shapely work at scales is that we can use multiple objects as part of the same object. In addition to this, we also have elements such as multipolygons, multiline and multi-points.

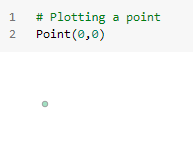
Now, a question arises, where is this feature useful? It is utilised when we define objects that have multiple geometries, such as countries that may have islands and other such physical landforms.

Let’s quickly look over some of the code that we can use to make some of the plots. First, we start off by importing the required packages to be able to plot the different geometries. Here, we import shapely, from where we import the point, linestring, polygon, multi-point and multi-polygon components.



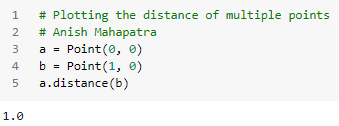
*Image by* [*Author*](https://www.linkedin.com/in/anishmahapatra/)

Next, we plot a point to see what it looks like.



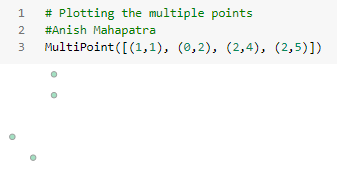
*Image by* [*Author*](https://www.linkedin.com/in/anishmahapatra/)

Post this, we proceed to look at the distance between two points, where the default distance measuring algorithm used is the Euclidian distance.



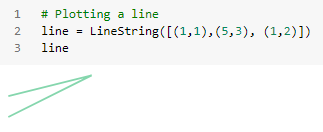
*Image by* [*Author*](https://www.linkedin.com/in/anishmahapatra/)

Next, we plot multiple points.



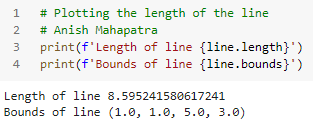
*Image by* [*Author*](https://www.linkedin.com/in/anishmahapatra/)

Now that we have understood well how points are plotted, we proceed to plot a linestring based on the points that we select.



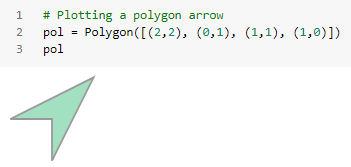
*Image by* [*Author*](https://www.linkedin.com/in/anishmahapatra/)

Post this, we would like to also analyse the distance of the line that has been plotted, and we are also able to get the bounds of the lines that have been plotted. This essentially shows us the boundaries of the plotted points.



*Image by* [*Author*](https://www.linkedin.com/in/anishmahapatra/)

Now that we have understood the bounds of the points, we can proceed and plot a full-fledged polygon. In this case, we will plot the little arrows that we generally see in Google Maps.

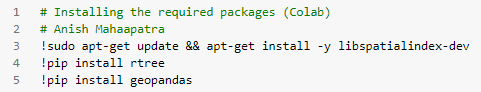


*Image by* [*Author*](https://www.linkedin.com/in/anishmahapatra/)

In the next section, we will look at how we can load the data.

## Loading data

First, we will need to install the packages that are required to be able to leverage GeoPands. Depending on your operating system, you can install the geopandas library. In this case, we have directly run this code on Google Colab. In fact, if you follow along with the code, you can do the same on Google Colab.



*Image by* [*Author*](https://www.linkedin.com/in/anishmahapatra/)

Now, we go ahead and import the relevant libraries to perform geospatial analysis.

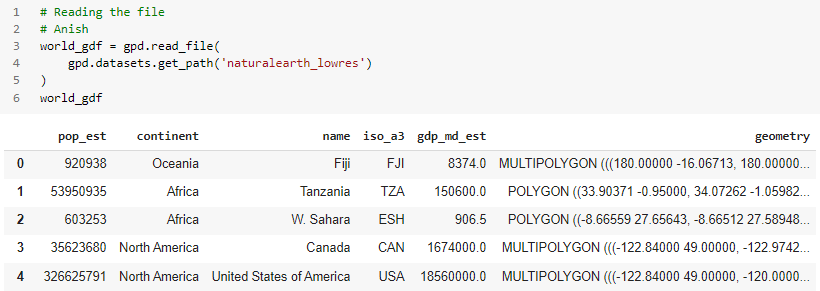


*Image by* [*Author*](https://www.linkedin.com/in/anishmahapatra/)

In the next steps, we will read the data from a region called ‘naturalearth\_lowres’, which contains a low-resolution image of the geometry of all the counties, along with some additional parameters such as GDP (Gross Domestic Population) and Population metrics.

## Reading in data

Next, we will import the relevant dataset from GDP so that we can effectively read the data.

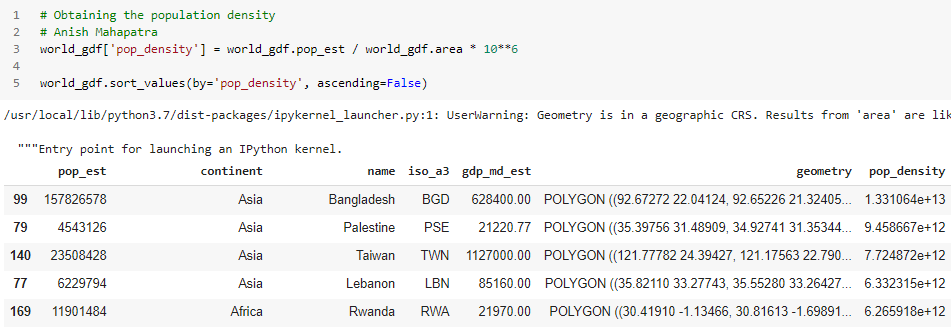


*Image by* [*Author*](https://www.linkedin.com/in/anishmahapatra/)

In the next section, we will leverage a coordinate reference system or CRS, as it is popularly known to obtain more information about the dataset.

## CRS

In this code snippet, we will map the population density of the world map based on the GDP data.



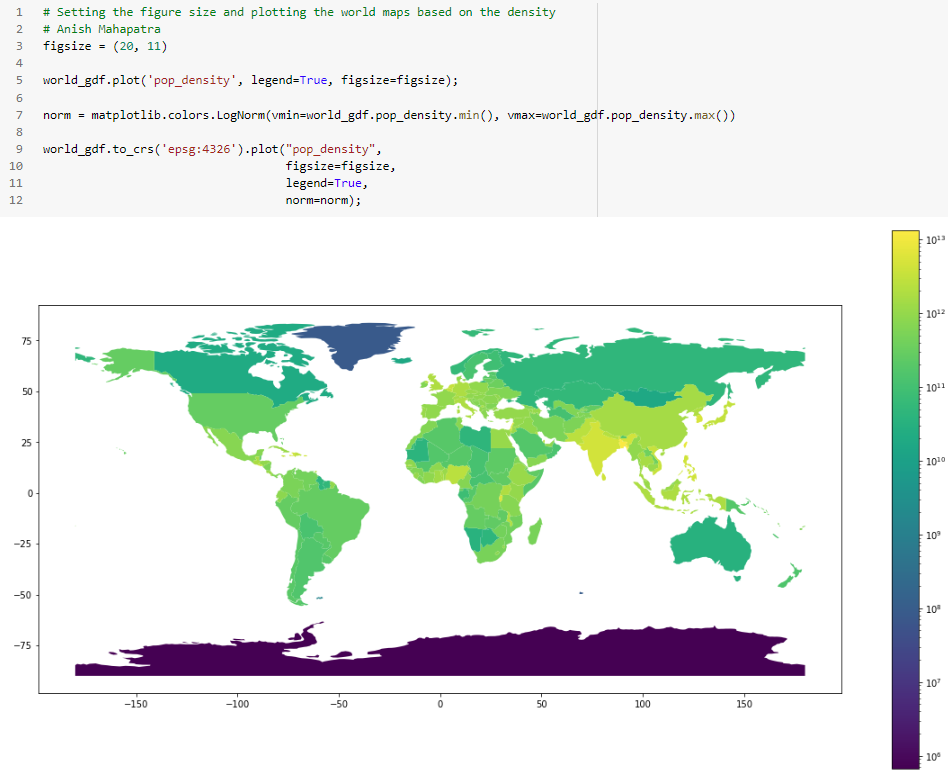
*Image by* [*Author*](https://www.linkedin.com/in/anishmahapatra/)

If we note carefully, we can see the various components of a Coordinate Reference System (CRS):

* **Axes and Units**: We keep track of the latitude and longitude by measuring them in degrees. As a global standard, these are generally measured in meters.
* **Datum**: The Datum is essentially the referencing system, where we measure from an initial point (which is generally the Prime Meridian), and we factor in the shape of the earth, which is an Ellipsoid.
* **Area of use**: Generally, the CRS is optimised based on a particular area that we are interested in. However, the data that we are looking at is optimised for the entire world.

## Visualisation

Finally, in this section, we will see the figure size based on your liking and plot the population based on the density of the population, where light green represents the most highly dense regions, and dark purple denotes the lowest population density.



*Image by* [*Author*](https://www.linkedin.com/in/anishmahapatra/)

# How Jupyter Notebook is used in GIS

Jupyter notebook is a powerful python tool which allows users to create and share documents containing codes, visualisations, explanatory texts and equations. The few main reasons that we can attribute to the growing popularity of Jupyter Notebook could be as follows:

1. **Notebook**: The term notebook is quite applicable to the Jupyter Notebook as the tool allows us to write snippets of executable codes called ‘cells’, comment down or note every procedure and also visualise the data during any step of your analysis
2. **Prototyping of Jupyter Notebook**: These notebooks are extremely useful in situations where we don’t have a final process defined for ourselves. It gives us flexibility in writing code and testing them into independent cells. This allows us to quickly test a code snippet without having to worry about any sequential workflow
3. **Visualising Pandas DataFrame**: You can view these tables anywhere in your notebook when using Jupyter Notebook. This is really helpful since you can view your data’s current state (and the impact of all the operations your code is making on it) as each stage of your logic executes.

Today, Jupyter notebooks have become the go-to tool for GIS analysts who choose to do spatial analysis with Python for a multitude of tasks such as spatial data manipulation, spatial analysis, visualisation, etc. Considering all the challenges that were a part of GIS software for doing geospatial analysis, which includes

1. Data analysis and management of large spatial data
2. One size doesn’t fit all types of tools and analyses within a single application
3. Data format support issues, where not every application allows every format of data for input

The GIS community quickly realised its potential and adopted Python as a tool for GIS analysis; however, Jupyter notebook provided the missing piece of becoming an easy-to-use tool that replaced the code editor as a working environment. Many geospatial Python packages are already available, including everything from geospatial data management to mapping capabilities inside a Jupyter Notebook.

To start utilising the Jupyter Notebook application within a desktop GIS, the ArcGIS Notebook inside ArcGIS Pro comes with a default installation. QGIS users will need to install the IPython QGIS Console plugin. This gives access to the IPython Console inside of QGIS. The IPython Console allows users to execute commands and interact with data inside IPython interpreters, which enables spatial data science python analysis, which can also be learnt in [data science using R syllabus](https://www.knowledgehut.com/data-science-courses).

# Conclusion

Here, with this article, we have covered different aspects of Geospatial analysis. We started off by understanding what geospatial data is, which typically gives information about objects, events, and other features with a location on or near the earth’s surface. Now, with this data, we also looked into how we can get started with working on it using different libraries such as GeoPy and GeoPandas. The base idea was to understand how to see how spatial data looks like and how we can perform simple analysis using python spatial analysis libraries. The adoption of Python shows how Python was accepted by many GIScientists as a go-to source for building desktop applications and standalone geospatial applications. The python ecosystem consists of numerous libraries that can be utilised for tasks across the spectrum to work with geospatial data. We looked into the basic concepts and terminologies of spatial data, which include text, vector and raster forms of data, what a Coordinate reference system is and how it is useful for Map projections, georeferencing and geocoding. Also, we went through to understand the pain points of the GIS and how Jupyter Notebook emerged as one of the leading options for having a single working environment for working with spatial analysis using Python.

# Frequently Asked Questions (FAQs)

## Is Python or R better for spatial analysis?

The fact that Python is easy to learn as a language and the availability of numerous geospatial data analysis python libraries makes it very adaptable to users for Geospatial analysis. Major GIS platforms like ArcGIS and QGIS have adopted Python as the principal scripting, toolmaking and analytical language. Python, according to experts, is also simpler to use than other high-level languages since it allows for a wide range of coding paradigms, including imperative, functional, procedural, and object-oriented ones.

R in comparison to Python, conducting spatial analysis with R can be just as simple. R programming language has seen an increase in the number of packages that are contributed towards GIS. R is also ideal for swiftly developing and visualising vectorised data, which makes it simpler for people who don’t want to put extra time into reading through documentation or creating hundreds of lines of code to accomplish something simple.

It is crucial to realise that both of these programming languages for spatial analysis accomplish different things. Python is excellent for automation when performing tasks like network analysis or cost surface analysis for batches of data. But when working with large datasets, such as when conducting multiple regression analysis, R is considered indispensable. Hence it becomes tricky to choose between both of them and wiser to use them for their own use cases.

## How do you do geospatial analysis in Python?

The availability of a great number of libraries from spatial analysis python packages makes Python a very powerful tool for spatial data analysis. It is simple as a process to load spatial data into Python using libraries such as GeoPandas and perform spatial statistical analysis on the same dataset. We have various other libraries which enable the user to create visualisations suitable to any use case at their disposal.

## What are the 5 concepts of spatial analysis?

This is an interesting question. The five primary components that we need to take care of when dealing with spatial analysis are methods, people, data, software and hardware. Each component is critical to the overall analysis as it feeds into the overall result of the spatial analysis.

## What is the spatial analysis method?

The spatial analysis method consists of the following components:

* **Explore the Area Data**: This is where we look at components such as the mapping and geovisualization, the spatial weights matrix and the global and local measure for spatial autocorrelation
* **Model the Area Data**: Here, we form spatial regression models, perform tests to analyse spatial dependence, essential the spatial regression models and finally perform model parameter interpretation
* **Leverage Models and Methods to perform Spatial interaction Data**: Visualising the spatial interaction data and creating functional specification of OLS regression models, and performing MLE Estimation for the spatial interaction models can help us perform spatial interaction data
* **Leverage Spatial Dependence and Spatial interaction**: Here, we use the Log Normal Spatial Interaction Model along with Spatial Filtering to interpret spatial dependence effectively

## What is the spatial analysis used for?

There are several uses for spatial analytic technology in both the public and business sectors. In addition to addressing global issues, it can be used to enhance local environments or communities. Time-lapsed spatial data analysis can be used to spot patterns and trends as well as anticipate future events quite accurately.

Spatial data analytics uses AI and ML applications to process enormous volumes of data with high levels of precision and efficiency.

# Author

**Anish Mahapatra**   
*Lead Data Scientist*   
LinkedIn: [https://www.linkedin.com/in/anishmahapatra](about:blank) ([LinkedIn](about:blank))   
Email: anishmahapatra01@gmail.com