Geo Spatial Analysis using GeoPandas

GeoPandas is a popular library used for the visualization and manipulation of GeoSpatial Data and maps. It extends on the existing Pandas library.

Advantages of GeoPandas.

* It is free, easy to understand and and highly efficient.
* It is highly flexible to use as it supports data of all formats.
* It is open source and it supports open source.
* It allows for plug ins and third party softwares(GeoDjango,PostGIS etc.)

# Disadvantages of GeoPandas

GIS tools are spread under different python modules and created by different developers.

# What is Spatial Data?

Spatial data refers to any type of data that directly or indirectly refers to a certain geographical area or  location. Spatial data is a numerical representation of a physical item in a geographical coordinate  system, often known as geospatial data or geographic information. Geographic data, on the other hand, is  considerably more than just the geographical component of a map. Users can save geographical  data in a number of formats because it can contain more than just location-specific information. We can learn more about how each variable affects people, communities, and populations by analyzing this data.

There are various types of spatial data, but the two most common are geometric data and geographic data. Let us try two understand these two kinds of data.

## Geometric Data

Geometric data is a sort of spatial data that is represented on a two-dimensional flat surface. For example, geometric data is used in floor plans. Google Maps is a navigation program that uses geometric data to generate precise directions. It is, in fact, one of the most basic examples of geographical data in action.

## Geographic Data

Geographic data is the information that has been plotted around a sphere. Most of the time, the sphere is the planet Earth. Geographic data emphasizes the relationship between latitude and longitude to a given object or area. A Global Positioning System (GPS) is a well-known example of geographic data.

Geospatial data often includes vast sets of spatial data gathered from a variety of sources in various forms and might contain information such as census data, satellite imagery, meteorological data, mobile phone data, drawn images, and social media data. When geospatial data can be discovered, shared, analyzed, and used in conjunction with traditional business data, it is the most useful.

Geospatial analytics adds time and location granularity to standard data sets. Maps, graphs, statistics, and cartograms can all be used to depict historical and current events in various ways. This additional information helps to paint a clearer picture of what transpired. Visual patterns and images that are easy to recognize reveal insights that could otherwise be lost in a huge spreadsheet. Forecasts can be generated more quickly, conveniently, and precisely with this technique.

# Geometric objects and shapely module.

Python has a specific module called Shapely that can be used to create and work with Geometric objects such as points, lines and polygons.

## It can be used for:

- Create a Line or Polygon from a Collection of Point geometries

- Calculate areas/length/bounds etc. of input geometries

- Make geometric operations based on the input geometries such as Union, Difference, Distance etc.

- Make spatial queries between geometries such Intersects, Touches, Crosses, Within etc.

# How to Install?

To install GeoPandas and all its dependencies, we recommend to use the conda package manager. This can be obtained by installing the Anaconda Distribution (a free Python distribution for data science), or through miniconda (minimal distribution only containing Python and the conda package manager). See also the installation docs for more information on how to install Anaconda or miniconda locally.

The advantage of using the conda package manager is that it provides pre-built binaries for all the required and optional dependencies of GeoPandas for all platforms (Windows, Mac, Linux).

To install the latest version of GeoPandas, you can then do:

conda install geopandas Similarly you can install for MAC/linux or if it does not work, please search for python package installer  in your browser and download the packages from the same.

# Different Data Structures in GeoPandas

GeoPandas implements two main data structures, a GeoSeries and a GeoDataFrame. These are subclasses of pandas Series and DataFrame, respectively.

## GeoSeries

A GeoSeries is essentially a vector where each entry in the vector is a set of shapes corresponding to one observation. An entry may consist of only one shape (like a single polygon) or multiple shapes that are meant to be thought of as one observation (like the many polygons that make up the State of Hawaii or a country like Indonesia).

geopandas has three basic classes of geometric objects (which are actually shapely objects):

* Points / Multi-Points
* Lines / Multi-Lines
* Polygons / Multi-Polygons

The GeoSeries class implements nearly all of the attributes and methods of Shapely objects. When  applied to a GeoSeries, they will apply elementwise to all geometries in the series. Binary operations can be applied between two GeoSeries, in which case the operation is carried out elementwise. The two  series will be aligned by matching indices. Binary operations can also be applied to a single geometry, in which case the operation is carried out for each element of the series with that geometry. In either case, a Series or a GeoSeries will be returned, as appropriate.

## Attributes

* area: shape area (units of projection – see projections)
* bounds: tuple of max and min coordinates on each axis for each shape
* total\_bounds: tuple of max and min coordinates on each axis for entire GeoSeries
* geom\_type: type of geometry.
* is\_valid: tests if coordinates make a shape that is reasonable geometric shape

## Methods

* distance(): returns Series with minimum distance from each entry to other
* centroid: returns GeoSeries of centroids
* representative\_point(): returns GeoSeries of points that are guaranteed to be within each  geometry. It does NOT return centroids.
* to\_crs(): change coordinate reference system.
* plot(): plot GeoSeries.

## Tests

* geom\_almost\_equals(): is shape almost the same as other (good when floating point precision  issues  make shapes slightly different)
* contains(): is shape contained within other
* intersects(): does shape intersect other

## GeoDataFrame

A GeoDataFrame is a tabular data structure that contains a GeoSeries.

The most important property of a GeoDataFrame is that it always has one GeoSeries column that holds a special status. This GeoSeries is referred to as the GeoDataFrame’s “geometry”. When a spatial method is applied to a GeoDataFrame (or a spatial attribute like area is called), this commands will always act on the “geometry” column.

The “geometry” column – no matter its name – can be accessed through the geometry attribute (gdf.geometry), and the name of the geometry column can be found by typing gdf.geometry.name.

A GeoDataFrame may also contain other columns with geometrical (shapely) objects, but only one column can be the active geometry at a time. To change which column is the active geometry column, use the GeoDataFrame.set\_geometry() method.

## Attributes

Any of the attributes calls or methods described for a GeoSeries will work on a GeoDataFrame – effectively, they are just applied to the “geometry” GeoSeries.

All the functionalities of Pandas are available directly in Geopandas without the need to call pandas  separately because Geopandas is an extension for Pandas.

For further reference, please visit the GeoPandas documentation:

Link: [GeoPandas 0.10.0+0.g0be92da.dirty — GeoPandas 0.10.0+0.g0be92da.dirty documentation](https://geopandas.readthedocs.io/en/latest/index.html)