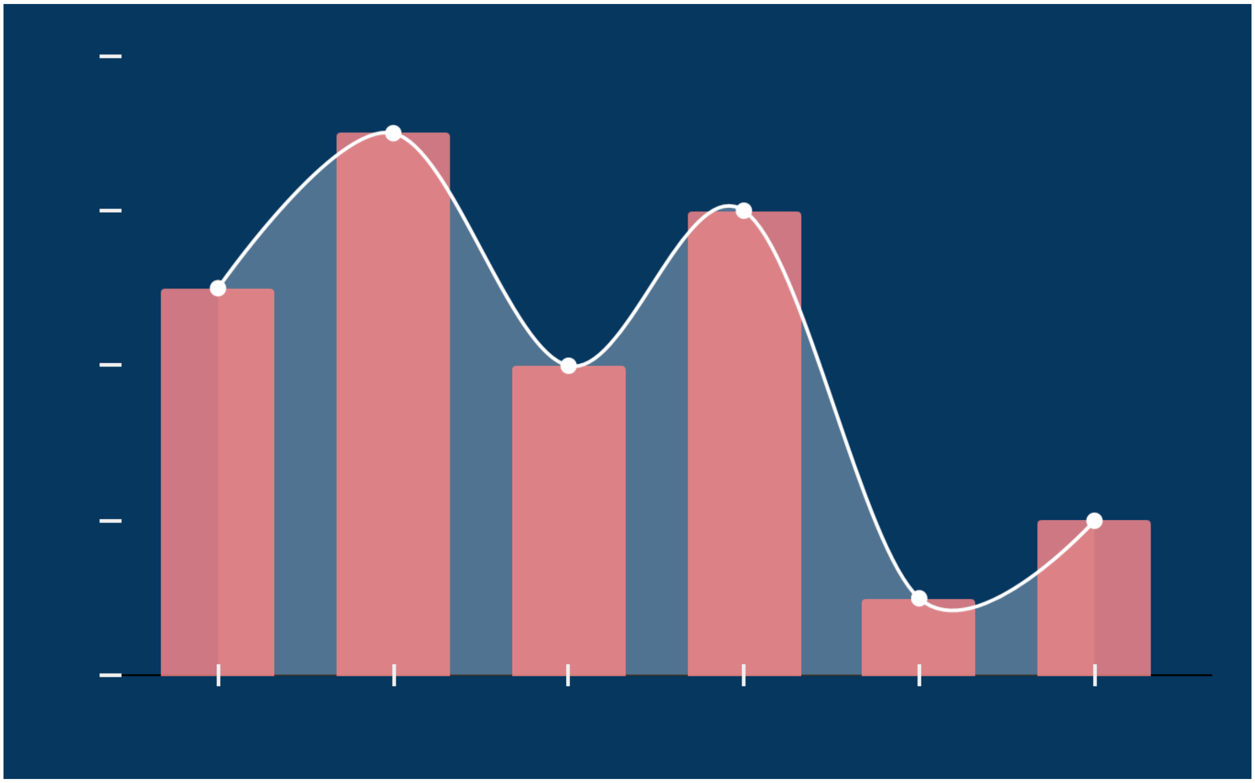
GIS & Bathymetry and sediment distribution in the Hudson River off Manhattan

GIS Analysis in Python



Kya Allen

05.21.2021
301-919-7902

Arendelle Department of Environmental Safety

Abstract

GIS, short for Geographic Information System, is a framework for processing, analyzing, and visualizing spatial or geographic data. In this paper, we will cover several basic topics in the GIS paradigm, and present some basic analysis using GIS in python to make inferences about the Hudson river. Topics covered include: File types for spatial and geographic data, coordinate systems and projections, maps, Bathymetry & Sediment distribution in the Hudson river, and the concept of Spatial Autocorrelation.

Overview of GIS

Geographic Information Systems, rather than just referring to a single software package like ArcGIS, refers to a general paradigm for processing, analyzing, and visualizing spatial data. Unlike some forms of data which can be easily formatted in simple file types like a CSV, or described fully in simple tables, spatial/geographic data tends to be more complicated. Geographic Information Systems allow us to take spatial data stored in file types that retain much of the multidimensional data required to describe their positions in the real world, and with that we can visualize the data in the form of maps. We can combine spatial data. For example you could Visually overlay NASA data on future solar eclipses, overlaid on top of data containing spatial information for world cities, overlaid on top of a world map. Then for more precise inference, you could perform a “Join” on the Eclipse coordinate data and the city coordinate data, to see which cities will be in the path of a future solar eclipse.

Coordinate Systems and Projections

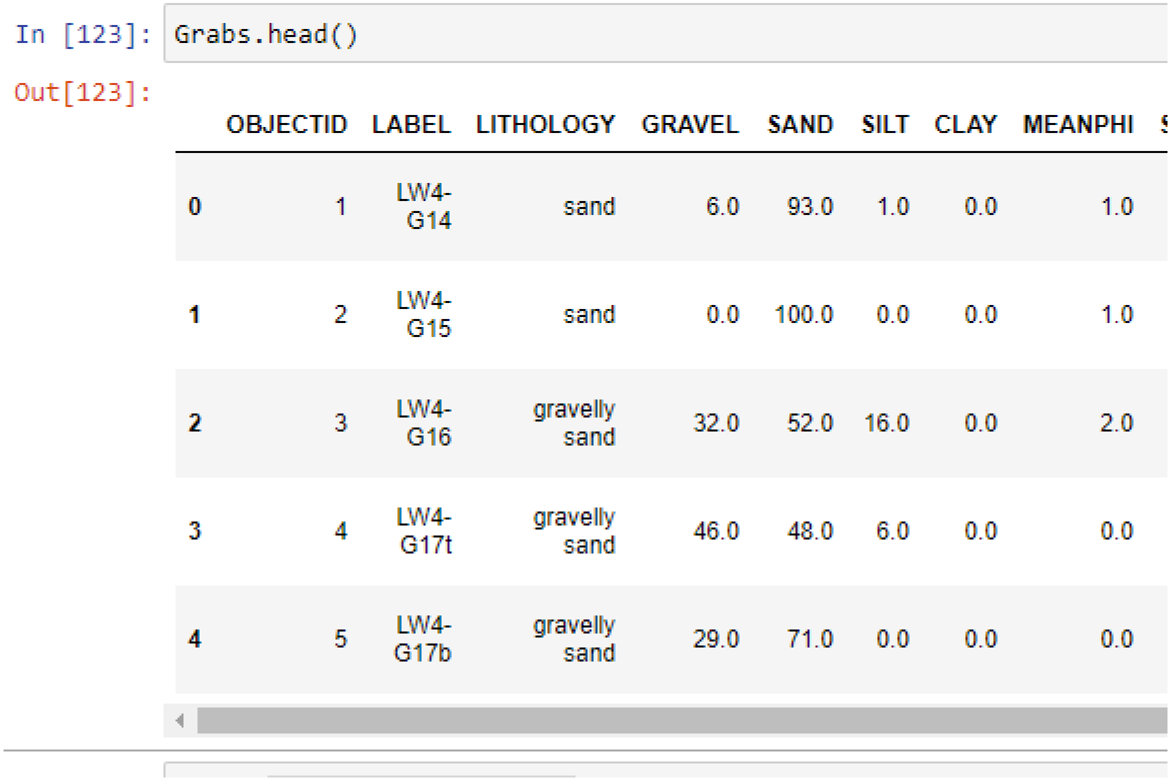
Coordinate systems/projections are a way for us to map data onto a 2 dimensional surface. The earth is, of course, an ellipsoid, so the 2 dimensional maps we look at are never an accurate depiction of the world. In fact Carl Friedrich Gauss proved mathematically that such a surface could never be reconstructed as a flat rectangular 2d surface like a map, without some amount of distortion. Today we have a variety of coordinate systems that project that surface of the earth onto a flat surface in different ways. It’s important to be aware of these different coordinate systems because your data analysis would be invalid if you combine datasets with different coordinate systems, without making some adjustment to their projections beforehand.

Spatial Data Formats

There are a large variety of standard GIS data types for storing spatial data. Broadly speaking, the two main types are Raster and Vector. The main difference between the two is that Raster files are usually based on pixels. It's basically just an image. Like a jpeg. Meanwhile Vector file types store information in the form of "vectors" or "paths" that make up anything from points to shapes. One common GIS file type is the Shapefile (.shp). Like other vector file types, a shapefile is characterized primarily by three elements necessary for storing geographic information. These are Points, Lines and Polygons. Points are zero-dimensional elements, which can be used to denote information you would model as having one point in space, such as the coordinate location of a car crash. Lines are one dimensional elements that can model stuff like roads. Polygons are two dimensional elements you might use to model anything where a full area is relevant, such as a lake, or in some cases this might be the preferred way to deal with roads, if the length and orientation aren't the only important qualities for the purpose of your analysis. Other popular vector based file types are GeoJSON , Vector Product Format (VPF), or a simple cartesian coordinate system (xyz) file.

Working with attribute tables in GIS

There isn't necessarily much to say about this. In specific GIS systems like Python's GeoPandas library, attribute tables are really nothing more than DataFrames with an extra column at the end for the geographical information associated with the observation, which is taken from the shapefile. Each column contains some information about the observation, such as a proportion of sand in a sediment sample from a specific point in a river. Here's an example:



```
In [123]: Grabs.head()
```

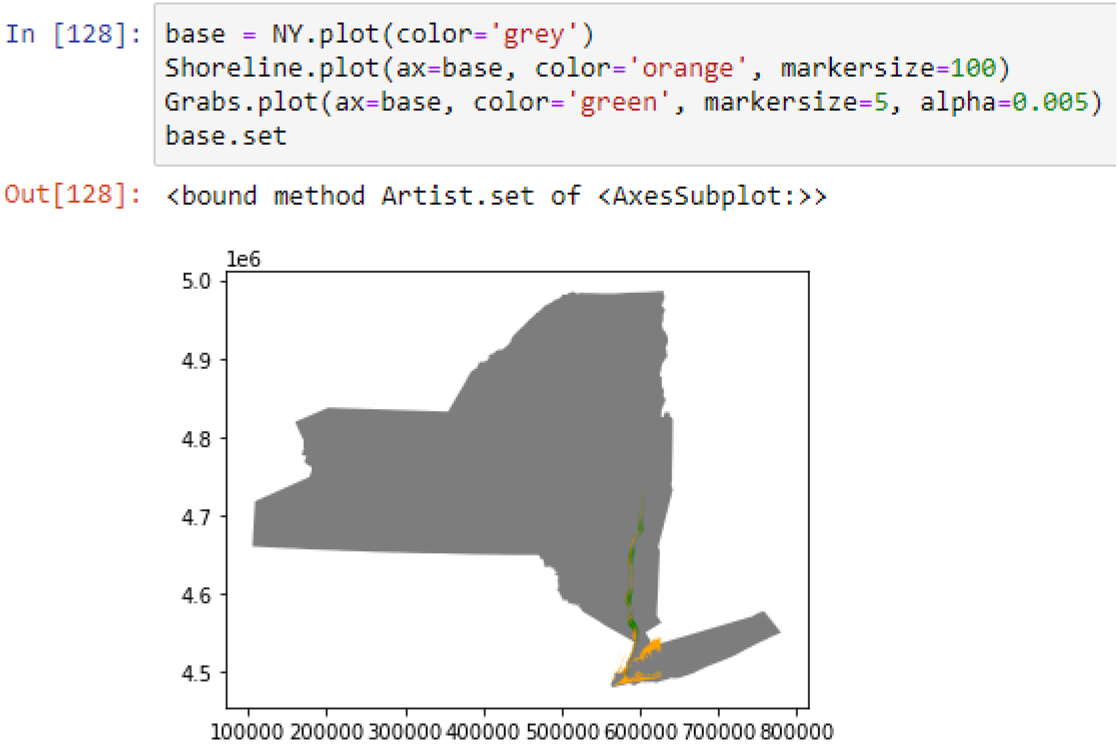
Out[123]:

	OBJECTID	LABEL	LITHOLOGY	GRAVEL	SAND	SILT	CLAY	MEANPHI	S
0	1	LW4-G14	sand	6.0	93.0	1.0	0.0	1.0	
1	2	LW4-G15	sand	0.0	100.0	0.0	0.0	1.0	
2	3	LW4-G16	gravelly sand	32.0	52.0	16.0	0.0	2.0	
3	4	LW4-G17t	gravelly sand	46.0	48.0	6.0	0.0	0.0	
4	5	LW4-G17b	gravelly sand	29.0	71.0	0.0	0.0	0.0	

This is part of an attribute table of data about the Hudson river taken from GIS.NY.GOV

Making Maps in GIS (GeoPandas)

I don't have ArcGIS so I did my GIS work in python. Making maps is pretty simple. You read in a shapefile with data = gpd.read_file("shapefile.shp"). Take for example a shapefile of New York State. Then you go data.plot(). There you go. But then again, it might be more useful to combine it with information from other shapefiles, like maybe a file that contains the geographic information for the shoreline of a river. I don't wanna actually explain all the code to do that, but I will link my code, and you should consider my python notebook when you realize that this isn't a 10 page paper. Anyway, here's a map.



```
In [128]: base = NY.plot(color='grey')
          Shoreline.plot(ax=base, color='orange', markersize=100)
          Grabs.plot(ax=base, color='green', markersize=5, alpha=0.005)
          base.set
```

```
Out[128]: <bound method Artist.set of <AxesSubplot:>>
```

Above, is a map of New York State. The Little orange sliver is the shoreline of the Hudson river. The little green sliver blocking most of the orange sliver represents the coordinates of the grab samples for sediment analysis.

Bathymetry and Sediment analysis - spatial analysis - GIS - Spatial Autocorrelation

So I collected several shapefiles from the GIS.NY.GOV website where they host the results of a project that involved gathering data about the Hudson river. I had absolutely no idea what kind of analysis I could do, So I settled for two ideas. One was to just take some descriptive statistics about stuff like, the average level of sediment types in the samples, or the most common sediment types. Most common depth from Bathymetry. Etc. But also I thought it might be cool to see if there is any autocorrelation with respect to the type of sediment. So basically that idea that samples characterized primarily by x sediment, are more likely to cluster together. My plan for this was to use the euclidean distance metric. There's a common method to test for autocorrelation where a square spatial sample is taken, and the points in that sample are compared to each other via euclidean distance (L2-norm). Then the average distance of the nearest neighbor from each point is compared to what we would expect that same average to be, it the spatial data points had been generated by a homogeneous poisson process. If it's similar to that, then it's less evidence of some structure in the spatial data, so we wouldn’t say there’s spatial autocorrelation. The problem I had initially was that I didn’t know how I could get a standardized square sample within the river. Obviously If I were to just run the test without a reference frame like that, it would appear that there is some autocorrelation because in the broader context, yes all the sediment samples are constrained by the shape of the river. So then instead I thought it would be a good idea to just take the average nearest neighbor distance for all the points as a baseline, then compare that to the within-sediment-type average nearest neighbor distances for each sediment type. I figured if there was some autocorrelation with respect to sediment type, then we’d see greater clustering within them compared to between them. But then I was pretty sure this wouldn’t work because I’d essentially be comparing very differently sized samples. Like a larger sample of points within the same amount of physical space will probably have a smaller average nearest neighbor distance just by virtue of it being a more crowded situation. So it’s not a good comparison. So you can see where I started to do all that in the ipython notebook, but I have nothing to report on the matter.

Results

- The most common sediment type was Sand
- The least common sediment type was gravel
- Clay never made up more than 54 percent of a sample
- The average depth of the river was -5.03
- The greatest depth of the river was -22