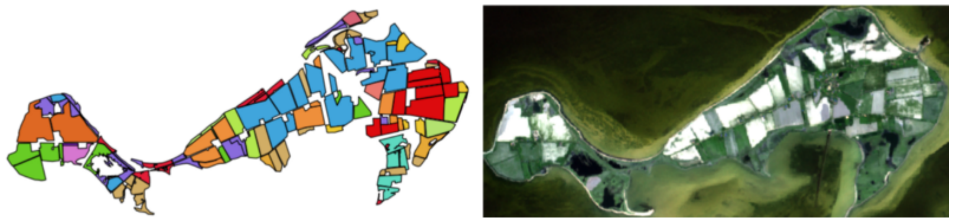
Essential geospatial Python libraries

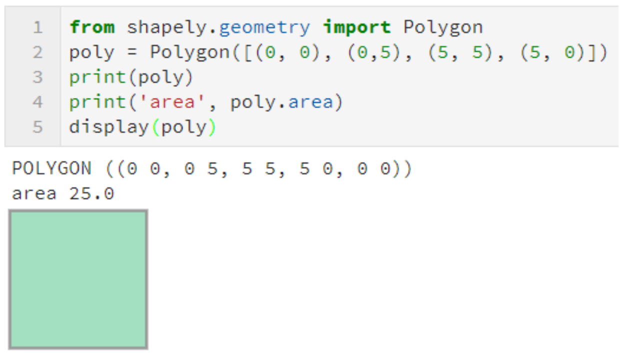
Christoph Rieke
Mar 25, 2018

This is a quick overview of essential python libraries for working with geospatial data. What I think might be valuable for newcomers in this field is some insight on **how these libraries interact and are connected**.



Shapely and Geopandas

When dealing with geometry data, there is just no alternative to the functionality of the combined use of shapely and geopandas. With **shapely**, you can create shapely geometry objects (e.g. Point, Polygon, Multipolygon) and manipulate them, e.g. buffer, calculate the area or an intersection etc.

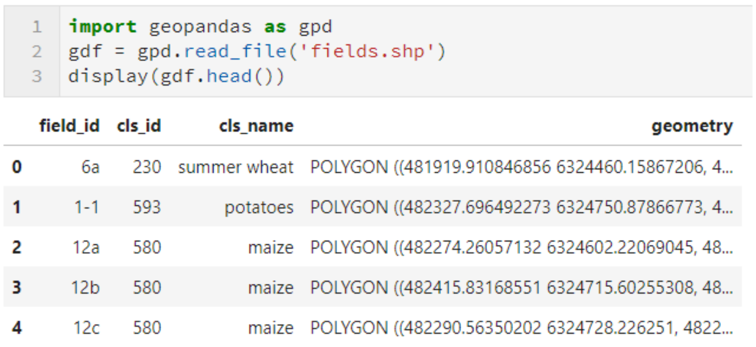


```python
from shapely.geometry import Polygon
poly = Polygon([(0, 0), (0,5), (5, 5), (5, 0)])
print(poly)
print('area', poly.area)
display(poly)
```

```
POLYGON ((0 0, 0 5, 5 5, 5 0, 0 0))
area 25.0
```

Shapely itself does not provide options to read/write vector file formats (e.g. shapefiles or geojson) or handle projection

conversions. This can be handled e.g. with the Fiona library. But there is an even more convenient way:
Geopandas combines the geometry objects of shapely, the read/write/ projection functions of fiona and the powerful dataframe interface of the pandas library in one awesome package. In the spreadsheet-like dataframe, the last column ‘geometry’ stores the shapely geometry objects, all shapely functions can be applied. The pandas mechanics offers super easy ways to manipulate, plot and analyze the data, e.g. dataframe groupby operations etc.

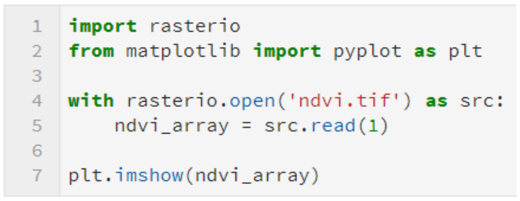


```python
import geopandas as gpd
gdf = gpd.read_file('fields.shp')
display(gdf.head())
```

	field_id	cls_id	cls_name	geometry
0	6a	230	summer wheat	POLYGON ((481919.910846856 6324460.15867206, 4...
1	1-1	593	potatoes	POLYGON ((482327.696492273 6324750.87866773, 4...
2	12a	580	maize	POLYGON ((482274.26057132 6324602.22069045, 48...
3	12b	580	maize	POLYGON ((482415.83168551 6324715.60255308, 48...
4	12c	580	maize	POLYGON ((482290.56350202 6324728.226251, 4822...

Rasterio

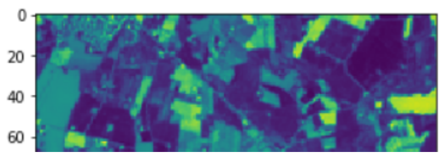
Rasterio is the go-to library for raster data handling. It lets you read/write raster files to/from numpy arrays (the de-facto standard for Python array operations), offers many convenient ways to manipulate these array (e.g. masking, vectorizing etc.) and can handle transformations of coordinate reference systems. Just like any other numpy array, the data can also be easily plotted, e.g. using the matplotlib library.

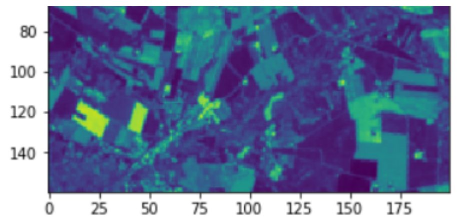


```python
import rasterio
from matplotlib import pyplot as plt

with rasterio.open('ndvi.tif') as src:
    ndvi_array = src.read(1)

plt.imshow(ndvi_array)
```





GDAL

Although I rarely use **GDAL** functions directly and would recommend beginners to concentrate on rasterio and shapely/geopandas, the **G**eospatial **D**ata **A**bstraction **L**ibrary needs to be on this list. Many of the libraries which are described here rely on GDAL, it is the cornerstone for reading, writing and manipulating raster and vector data in many software packages. However, the GDAL Python bindings (GDAL is originally written in C) are not as intuitive as expected from standard Python. The other libraries on this list use modern Python language features and imho offer more convenience and functionality.

rasterstats: For zonal statistics. Extracts statistics from rasters files or numpy arrays based on geometries.

scikit-image: Library for image manipulation, e.g. histogram adjustments, filter, segmentation/edge detection operations, texture feature extraction etc.

scikit-learn: The best and at the same time easy-to-use Python machine learning library. Regression, classification, dimensionality reductions etc.

folium: Lets you visualize spatial data on interactive leaflet maps.

descartes: Enables plotting of shapely geometries as matplotlib paths/ patches. Also a dependency for the geometry plotting functions of geopandas.

pyproj: For transformation of projections. Mostly unnecessary when using the more conveniant geopandas coordinate reference system (crs) functions.

PySAL: The Python Spatial Analysis Library contains a multitude of functions for spatial analysis, statistical modeling and plotting.

xarray: Great for handling extensive image time series stacks, imagine 5 vegetation indices x 24 dates x 256 pixel x 256 pixel. xarray lets you label the dimensions of the multidimensional numpy array and combines this with many functions and the syntax of the pandas library (e.g. groupby, rolling window, plotting). Not essential for beginners, but it is a great addition when working with extensive time series data.

Edit: There have been quite a few recommendations for other geospatial libraries and ressources in the comments, take a look! I also recommend checking out the **"Awesome geospatial"** list.

You can follow me on Twitter **@ chrieke**

Data Science Geospatial Python Remote Sensing GIS

490 claps 7

Christoph Rieke

geo. space. tech. geopolitics.

Follow