Intro to Spatial Data Analysis in Python

Jenny Palomino (UC Berkeley)

FOSS4G NA 2015 Conference

March 10, 2015

Quick Start: Spatial Data Analysis with Python

Spatial data types/formats

Python: what is it, how do you use it, and how can you get started?

Useful Open Source Python Spatial Packages

PySAL Example: Spatial Autocorrelation in Crime Data

Rasterio Example: Raster Calculation of Vegetation Index

Comparison of Python options to other geoprocessing options

Open source web options for publishing spatial data (including Python-based)

Online (Free!) Resources for Learning Python

What is Spatial Data?

<u>Primary Data Types</u>

vector: point, line, polygon

raster: continuous (e.g. elevation) or discrete surfaces (e.g. land use)

<u>Common Data Formats</u>

vector: shapefile, database geometry, tables (.dbf, .xlsx), KML, GeoJSON

raster: ASCII, GeoTIFF, JPEG2000, MrSID, database BLOB, HDF5

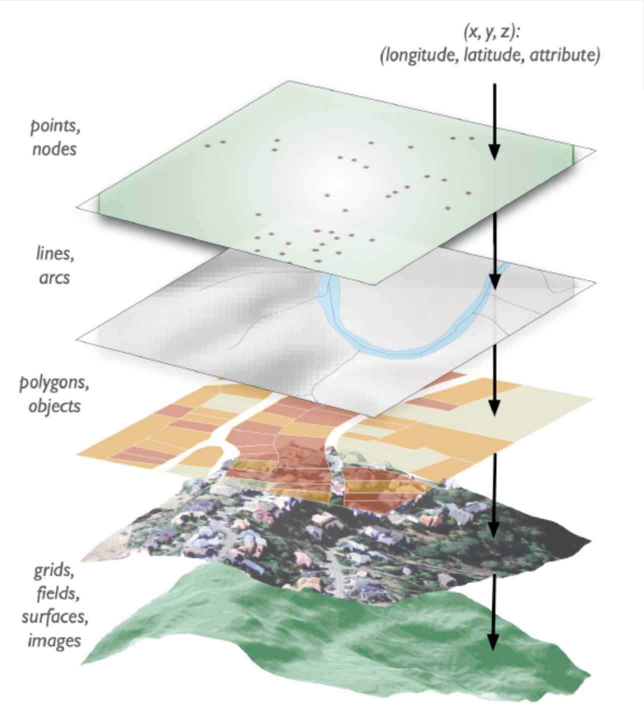


Image: Maggi Kelly, UC Berkeley

What is Spatial Data?

<u>File Formats (data on hard drive)</u>

shapefile, GeoTIFF, Table (.dbf, .xlsx), KML, HDF5

<u>Data Structures (data accessed via syntax)</u>

GeoJSON, URL to dataset (HTTP to render WMS)

<u>Databases (data stored as geometry, BLOB, etc)</u>

Open Source: PostgreSQL, MongoDB, SpatiaLite

Proprietary: ESRI personal and file geodatabases, ESRI ArcSDE, Oracle Spatial, Microsoft SQL Server

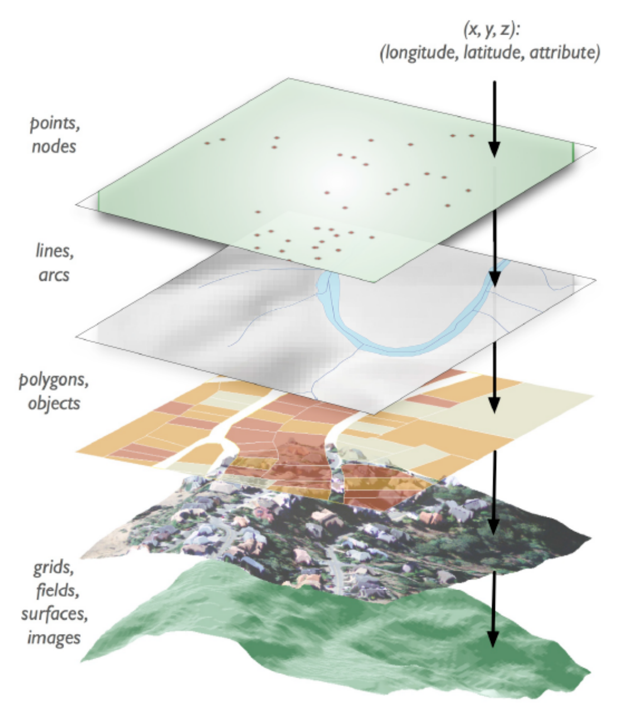
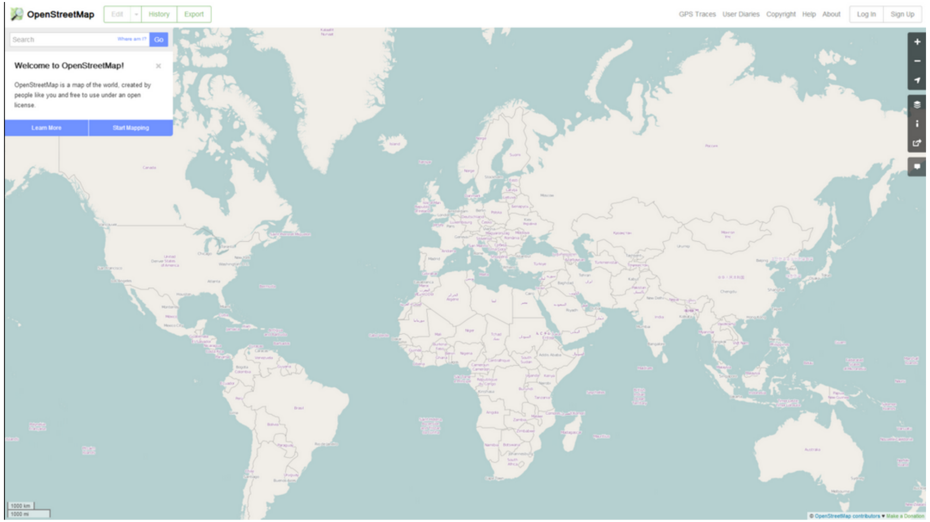


Image: Maggi Kelly, UC Berkeley

Map Projections: Translating 3D to 2D

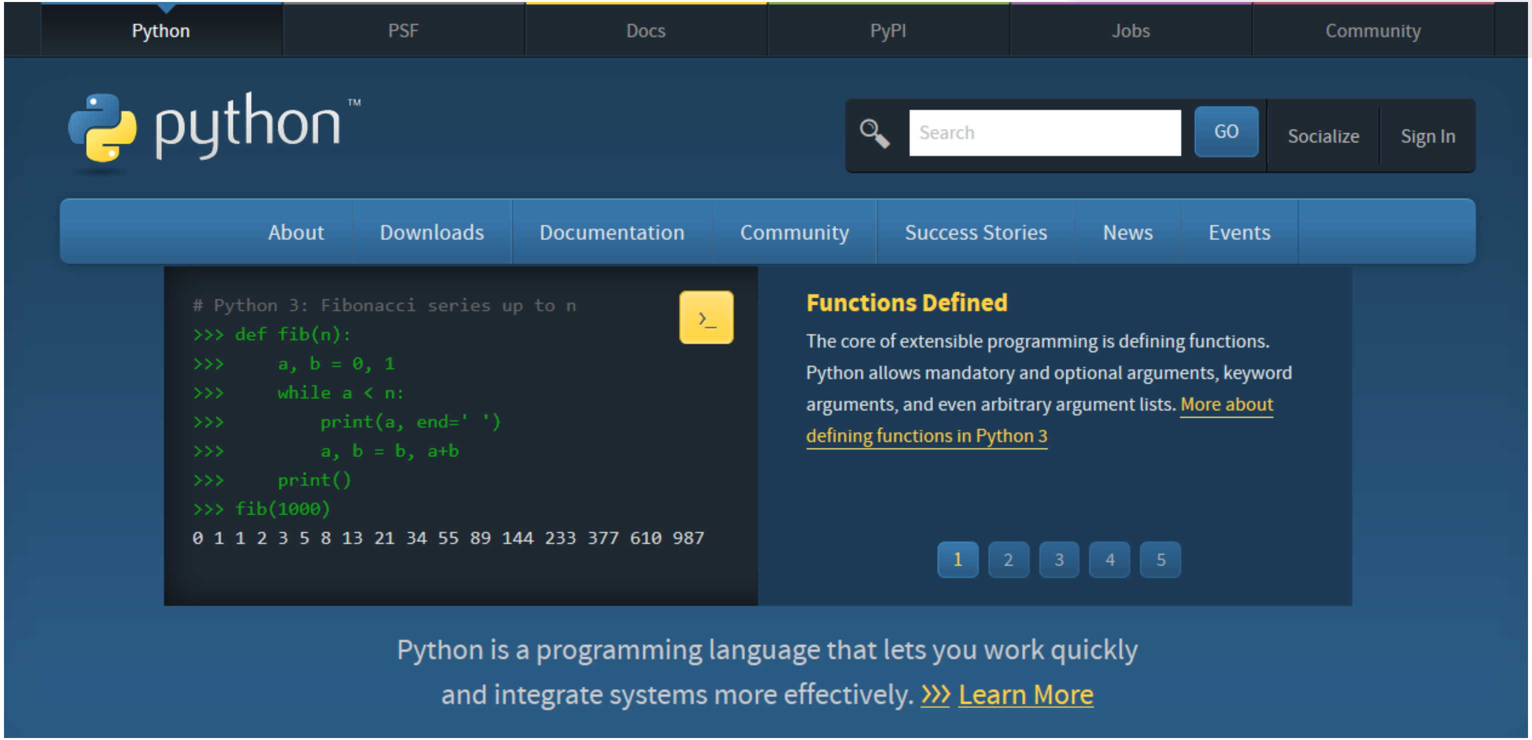
Visual Representation of UTM Zone 13



Web Mercator used in OpenStreetMap

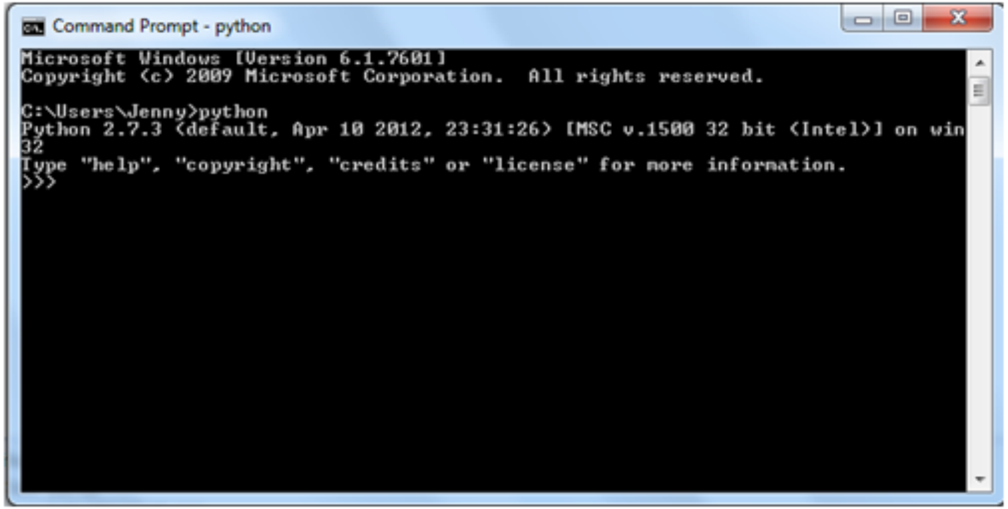
http://www.progonos.com/furuti/MapProj/Normal/TOC/cartTOC.html
http://en.wikipedia.org/wiki/Web_Mercator

What is Python?

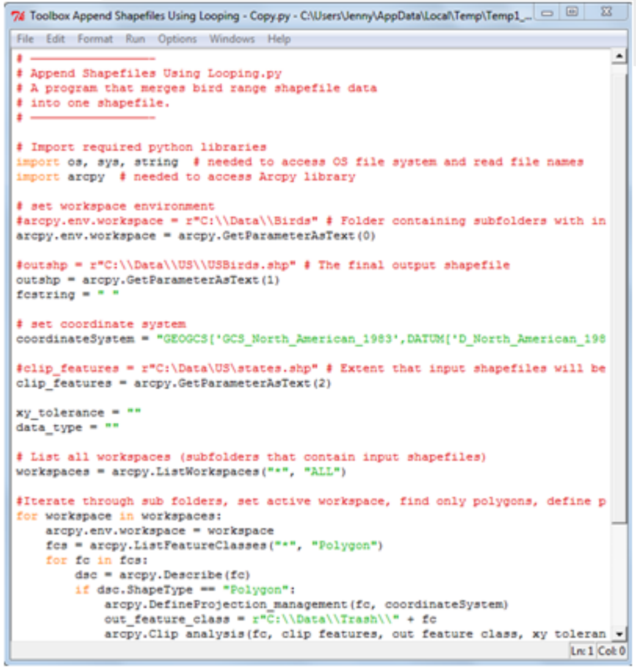


https://www.python.org/

Using Python: Traditional Install



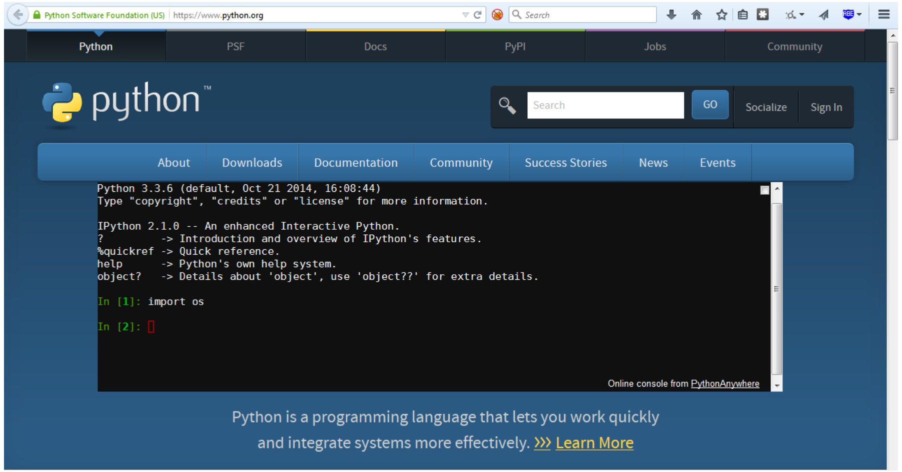
Via Command Line (terminal or shell)



Stand-alone script in a text editor (ex:IDLE)

Using Python: Modern Options

IPython
(browser-based shell but still needs to be installed locally)



Python Anywhere
(completely cloud-based)

http://nbviewer.ipython.org/
https://www.python.org/

Pre-packaged (and FREE) Python Distributions

WinPython for Python 2.7:

- numpy 1.9
- scipy 0.15
- PySAL: not included
- pandas 0.15
- shapely: not included
- fiona: not included
- six 1.8
- Windows only

Anaconda for Python 2.7:

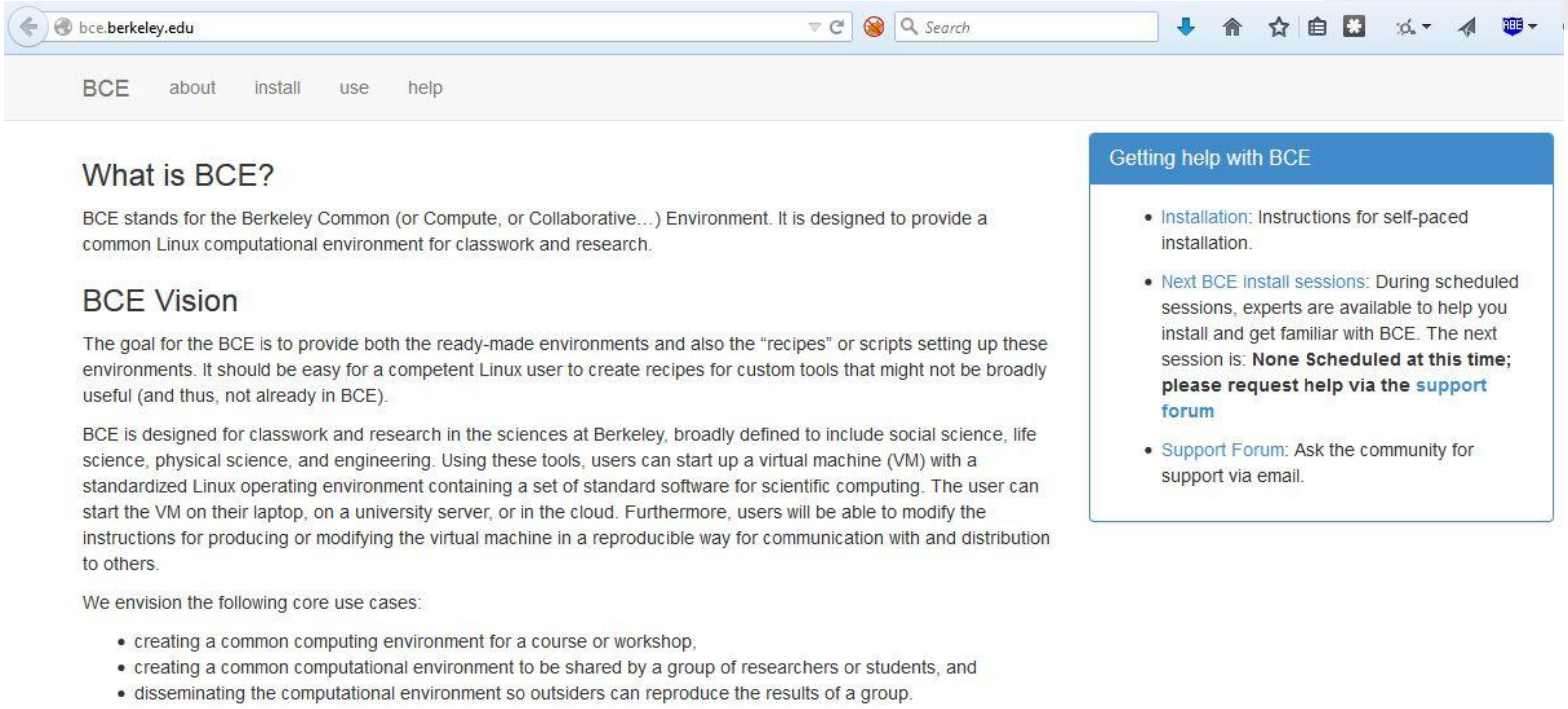
- numpy 1.9
- scipy 0.14
- PySAL 1.6
- pandas 0.14
- shapely: not included
- fiona: not included
- six 1.8
- Virtual Machine images
- Windows, Mac, Linux

Enthought Canopy for Python 2.7:

- numpy 1.8
- scipy 0.15
- PySAL 1.7 (in academic option)
- pandas 0.15
- shapely: 1.5.1 (in academic option)
- fiona: 1.4.8 (in academic option)
- six 1.9
- Windows, Mac, Linux

Other Python distribution options listed at: http://www.scipy.org/install.html

Berkeley Common Environment - Virtual Machine for Scientific Computing



bce.berkeley.edu

BCE about install use help

What is BCE?

BCE stands for the Berkeley Common (or Compute, or Collaborative...) Environment. It is designed to provide a common Linux computational environment for classwork and research.

BCE Vision

The goal for the BCE is to provide both the ready-made environments and also the "recipes" or scripts setting up these environments. It should be easy for a competent Linux user to create recipes for custom tools that might not be broadly useful (and thus, not already in BCE).

BCE is designed for classwork and research in the sciences at Berkeley, broadly defined to include social science, life science, physical science, and engineering. Using these tools, users can start up a virtual machine (VM) with a standardized Linux operating environment containing a set of standard software for scientific computing. The user can start the VM on their laptop, on a university server, or in the cloud. Furthermore, users will be able to modify the instructions for producing or modifying the virtual machine in a reproducible way for communication with and distribution to others.

We envision the following core use cases:

- creating a common computing environment for a course or workshop,
- creating a common computational environment to be shared by a group of researchers or students, and
- disseminating the computational environment so outsiders can reproduce the results of a group.

Getting help with BCE

- Installation: Instructions for self-paced installation.
- Next BCE install sessions: During scheduled sessions, experts are available to help you install and get familiar with BCE. The next session is: **None Scheduled at this time; please request help via the support forum**
- Support Forum: Ask the community for support via email.

http://bce.berkeley.edu/

OSGeo Live - Virtual Machine for Open Source and Web GIS

http://live.osgeo.org/en/index.html

Useful Open Source Python Spatial Libraries

Data Handling:

shapely

GDAL/OGR

pyQGIS

pyshp

pyproj

Analysis:

shapely

numpy, scipy

pandas, GeoPandas

PySAL

Rasterio

scikit-learn, scikit-image

Plotting:

matplotlib, prettyplotlib

descartes

cartopy

https://github.com/SpatialPython/spatial_python/blob/master/packages.md
http://spatialdemography.org/essential-python-geospatial-libraries/
http://carsonfarmer.com/2013/07/essential-python-geo-libraries/

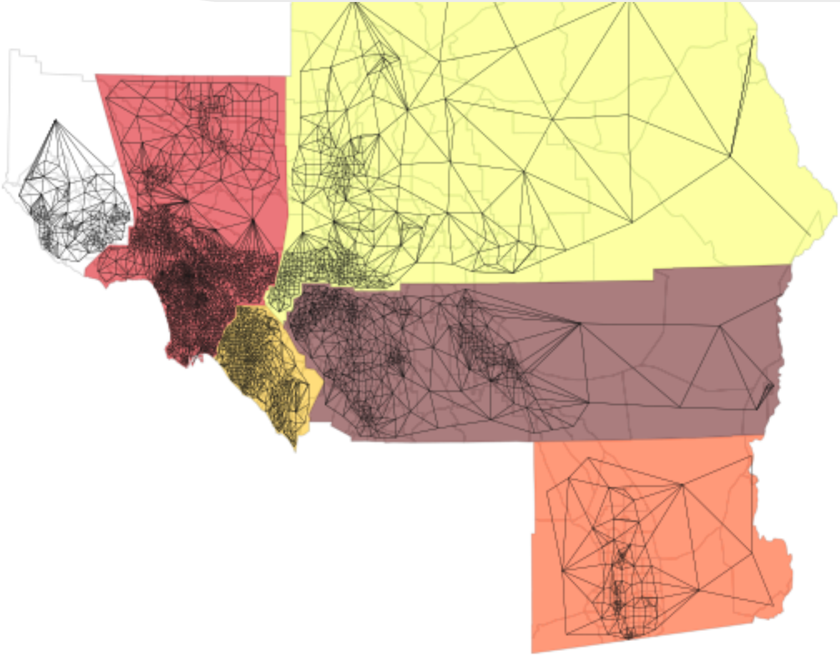
PySAL - Python Spatial Analysis Library

Vector and Raster Data Analysis

- Spatial Autocorrelation
- Spatial Econometrics
- Spatial Smoothing
- Regionalization
- Markov Chains

Requires Python 2.6 or 2.7

- numpy (1.3 or later)
- scipy (0.11 or later)
- add shapely for more options



https://pysal.readthedocs.org/en/latest/users/index.html

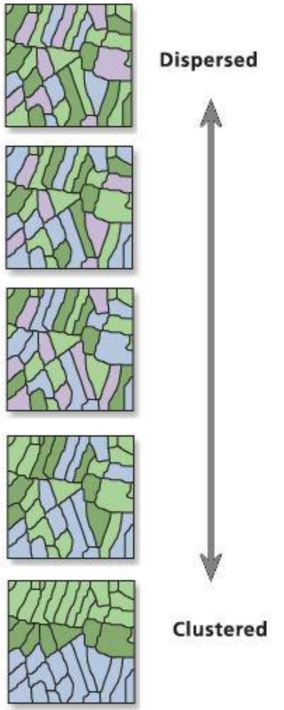
What is Spatial Autocorrelation?

Two Measures of Spatial Autocorrelation:

1. Global - quantifies clustering/dispersion across a region
   a. values ~ 1.0: highly clustered
   b. values ~0.0: no spatial autocorrelation
   c. values ~ -1.0: highly dispersed

2. Local - identifies clusters (hot-spots) within the region

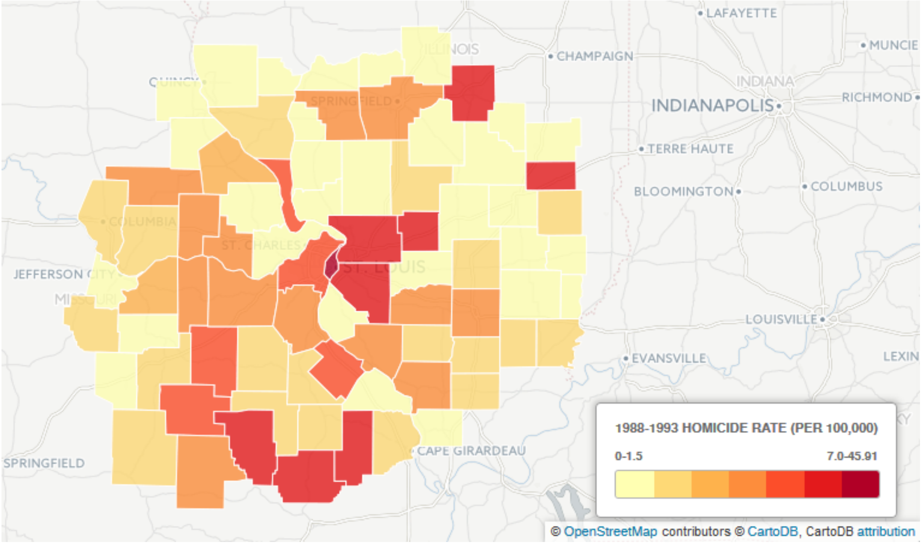
Example in presentation is based on Moran's I Global and Local Indicators (other indicators detailed here).



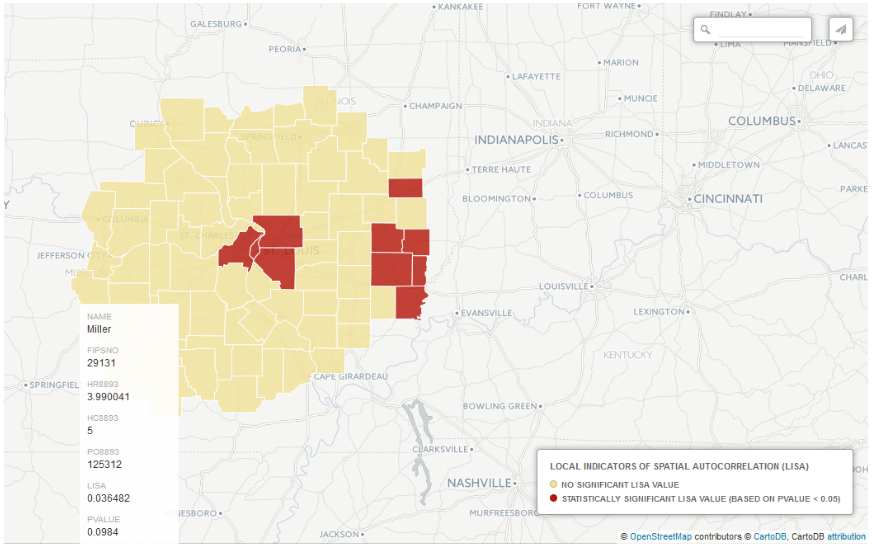
http://www.pysal.org/users/tutorials/autocorrelation.html
Image: ESRI help

Results of Spatial Autocorrelation in PySAL

Clustered with Global Moran's I = 0.24 (CartoDB)



Three Clusters of Hot-spots (CartoDB)



Why Use PySAL for this?

Compared to ArcMap, allows for more control over inputs and more robust testing of the data by including more advanced statistical methods:

- permutations of data (multiple runs of the random distribution)
- choice between one-tailed and two-tailed tests
- adjustments for populations of different sizes

Compared to R, leverage the benefits of working within Python:

- Object-oriented programming (advanced PySAL examples)
- Easily embed within desktop or web applications
- Easily work with Database Systems (e.g. PostgreSQL/PostGIS)
- Work with other useful Python packages (e.g. numpy, scipy, shapely)

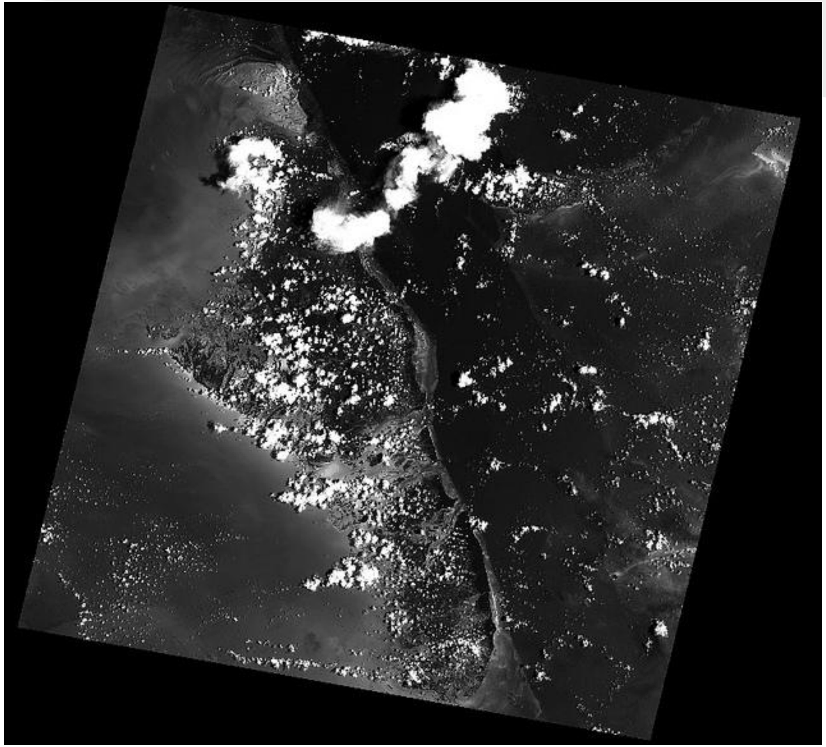
Rasterio

Raster Geoprocessing and Data Analysis

- Raster calculation across bands
- Raster stacking and merging
- Histograms and color maps
- Raster file conversions

Requires Python 2.7, 3.3 or 3.4

- GDAL (1.9 or later)
- C compiler (more info)
- numpy (1.7 or later)
- enum34
- cligj
- affine (1.0 or later)



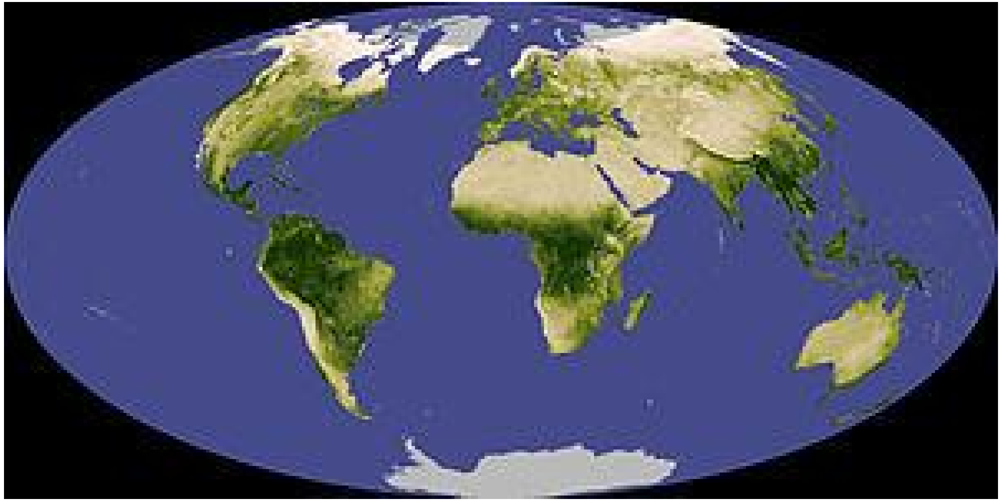
https://github.com/mapbox/rasterio

Normalized Difference Vegetation Index (NDVI)

$$\text{NDVI} = \frac{\text{(Near Infrared Band - Red Band)}}{\text{(Near Infrared Band + Red Band)}}$$

Values range from -1.0 to 1.0:

- water ~ -1.0
- barren area ~ 0.0
- shrub/grass ~ 0.2-0.4
- forest ~ 1.0

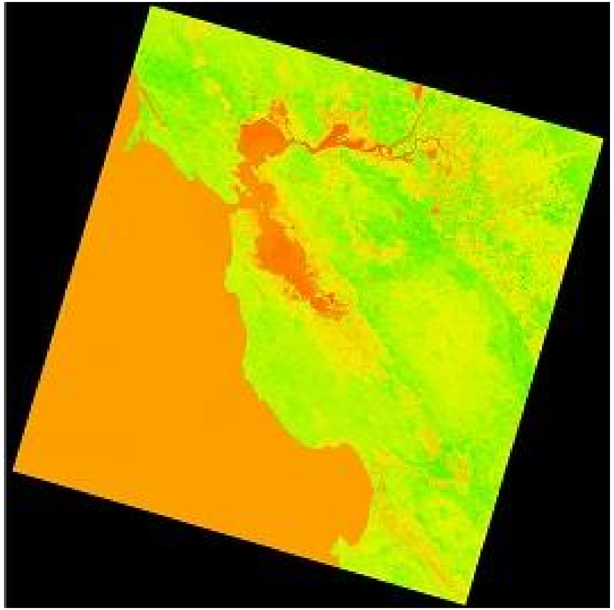


http://en.wikipedia.org/wiki/Normalized_Difference_Vegetation_Index

Results of Raster Calculation in Rasterio

GeoTIFF of Landsat 8 image for the San Francisco Bay Area

NDVI result from ~ -1.0 (red/orange) to ~1.0 (green = vegetation)



GeoPandas: Pandas + Shapely

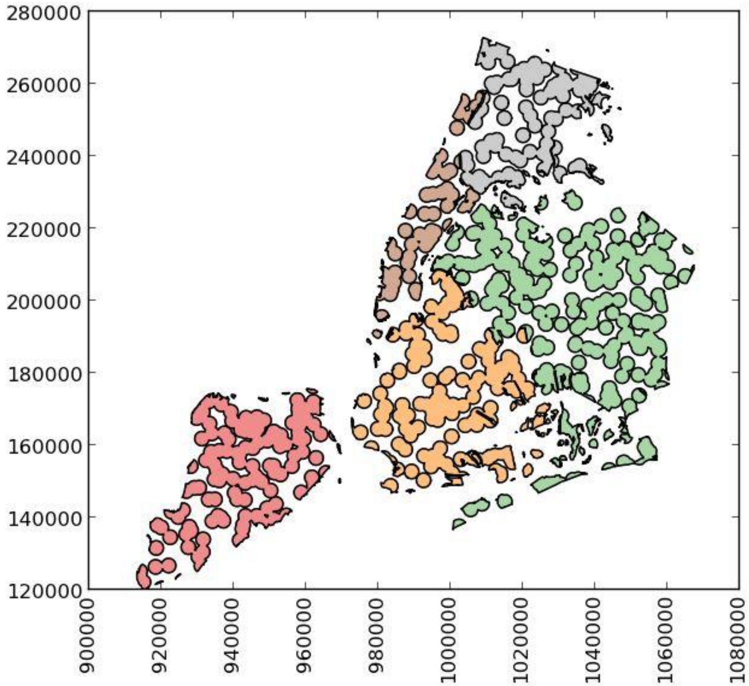
Vector Geoprocessing

- buffer
- intersect
- union
- difference

Requires Python 2.6, 2.7 or 3.2+

- numpy
- pandas (0.13 or later)
- shapely
- fiona
- six

```
>>> holes = boros['geometry'].intersection(mp)
```



http://geopandas.org/user.html#examples

Comparing Python to other Spatial tools

	ArcDesktop Suite	*Open Source GIS (e.g., QGIS, uDig)*	*Python*	*R Spatial*
Setting Up Working Environment	Easy Install; Very clear GUI including Model Builder; restricted to Windows	**Easy Install; Relatively clear GUIs; OS independent**	Often many dependencies for installation of packages; (too?) many GUI and text editor options; OS independent	Requires installation of packages, but relatively easy; R Studio is nice GUI; OS independent
Spatial Databases	Needs ArcSDE for DBMS	work with major open source DBMS	**Works well with major open source DBMS**	Can work with major open source DBMS
Analyzing Data	Good for beginners and advanced users who use ArcPy; not much statistical power but solid data handling	Good for beginners; Grass allows more advanced analysis	**Statistical platform of choice for most sciences; handles "Big Data" the best**	**Statistical platform of choice for env and ecological sciences**
Making a Map	**Easiest option for Beginners**	Very Good option for Beginners	Requires willingness to learn programming fundamentals	Good for scientists already using R
Publishing Data (web)	ArcGIS Server: expensive but really easy to use	Doesn't have built-in option	**Web full of options** (details on next slide)	R Shiny is getting a lot of attention

Open source options to publish spatial data

Easy Free Options: CartoDB, Google Maps API and Fusion Tables, ArcGIS Online and ArcGIS Open Data (both open to non-ESRI license holders)

Python-based: GeoDjango, Kartograph, Mapnik (C++ with Python bindings), Mapserver and Open Street Map API (support many languages including Python), Flask in combination with spatial database like MongoDB

JavaScript-based: Kartograph, Leaflet, PolyMaps, MapBox, OpenLayers

Spatial Data Servers and Suite of Tools: Geoserver, OpenGeoSuite

Spatial Databases: PostGIS, MongoDB, SpatiaLite

OS Geo Live - Virtual Machine for Open Source and Web GIS

http://live.osgeo.org/en/index.html

Online (and free!) Resources for Python

Code Academy (programming tutorials): http://www.codecademy.com/

Coursera (full courses): https://www.coursera.org/courses?query=python

Python wiki pages: https://wiki.python.org/moin/BeginnersGuide/NonProgrammers

https://docs.python.org/2/tutorial/

Python at Berkeley (DLab): http://python.berkeley.edu/learning_resources.html

Python Books and Training: http://pythonbooks.revolunet.com/

http://www.learnpython.org/

ArcPy tutorials from ESRI: http://training.esri.com/gateway/index.cfm?fa=catalog.webCourseDetail&courseid=2520

http://training.esri.com/gateway/index.cfm?fa=catalog.webCourseDetail&courseid=2523

UC Berkeley: Spatial Data Science Bootcamp

As technology is rapidly changing, the goal is not to teach a specific suite of tools but rather to teach participants how to develop and refine repeatable and testable workflows for spatial data using common standard programming practices.

Set Up Your Environment

- Virtual machine environments (Linux-based)
- Spatial databases (PostgreSQL/PostGIS) with multi-user editing and versioning (GeoGig)

Wrangle Data

- APIs (Google Maps, OpenStreetMap)
- Modern data formats and tools (GeoJSON, GDAL)

Analyze Data

- Vector-based analysis using ArcPy and PySAL (within IPython)
- Raster-based analysis using Rasterio (within IPython) and R

Visualize and Publish Data

- Web-based visualizations (D3)
- Map publication (geostack, CartoDB)

Next Session: May 20-22, 2015 on UC Berkeley campus
http://iep.berkeley.edu/spatialdatascience

GEOSPATIAL INNOVATION FACILITY
Cutting-Edge Mapping Technology at UC Berkeley

Questions or Comments?

jpalomino@berkeley.edu

Evaluate the sessions

Sign in: 2015.foss4g-na.org/