The Journal of Open Source Software

EarthPy: A Python package that makes it easier to explore and plot raster and vector data using open source Python tools.

Leah Wasser[1], Maxwell B. Joseph[1], Joe McGlinchy[1], Jenny Palomino[1], Korinek, Nathan[1], Chris Holdgraf[2], and Tim Head[3]

1 Earth Lab, University of Colorado - Boulder **2** University of California - Berkeley, Project Jupyter **3** Wild Tree Tech

DOI: 10.21105/joss.01886

Software

- Review
- Repository
- Archive

Submitted: 06 November 2019
Published: 13 November 2019

Summary & Purpose

`EarthPy` makes commonly performed spatial data exploration tasks easier for scientist by building upon functions in the widely used packages: `Rasterio` and `GeoPandas`. `EarthPy` is designed for users who are new to `Python` and spatial data with a focus on scientific data.

When a user is working with spatial data for research, there are a suite of data exploration activities that are often performed including:

1. Viewing histograms and plots of single bands within a remote sensing image to explore potential calibration and other data quality issues.
2. Creating basemaps that have legends with unique symbology.
3. Creating plots of images with colorbars.
4. Rendering RGB (and other composite) images of multi band spectral remote sensing images.
5. Masking clouds from a remote sensing image.
6. Limiting geographic extent of spatial data

The above operations are crucial to understanding a dataset and identifying issues that may need to be addressed with further data processing when beginning an analysis. In the `R` world, these tasks are quickly performed using the `raster` and `sp` packages. However, there isn't a tool that makes these tasks easy for users in the `Python` open source package landscape.

EarthPy Audience

`EarthPy` was originally designed to support the Earth Analytics Education program at Earth Lab - University of Colorado, Boulder. Our program teaches students how to work with a suite of earth and environmental data using open source `Python`. All lessons are published as free open education resources on our online learning portal (https://www.earthdatascience.org). Through this publication process, we identified that many spatial data exploration and cleanup tasks which were performed regularly required many steps that could be easily wrapped into helper functions. We modeled these functions after those available in the `R` ecosystem, given the experience of the developers' many years of working and teaching with `R`.

`EarthPy` allows the user to streamline common geospatial data operations in a modular way. This reduces the amount of repetitive coding required to open and stack raster datasets, clip the data to a defined area, and in particular, plotting data for investigation.

EarthPy Functionality

EarthPy is organized into several modules:

- io: Input/output for data: utility functions to download existing teaching data subsets or other data into a user's working directory (by default, this directory is: `~/earth-analytics/data`). The IO module supports downloading data for the Earth Lab Earth Analytics courses as well as any user with a URL to a compressed file.
- mask: Mask out cloud and shadow covered pixels from raster data: helper functions to mask remote sensing images using a cloud mask or QA (i.e. quality) layer.
- plot: Visualizing spatial data: plotting utilities including plotting a set of bands saved in a numpy array format, creating a custom colorbar, and custom legends with unique symbology.
- spatial: Raster processing and analysis: utilities to crop a set of bands to a defined spatial extent, create a hillshade, stack bands, and calculate normalized difference rasters.
- clip: Vector data subsetting: A module to clip vector data using GeoPandas. Allows for clipping of points, lines, and polygon data within a specified polygon.

EarthPy Vignettes

In addition to detailed API documentation and example code executed by doctest, EarthPy documentation includes a long-form examples gallery that demonstrates functionality using case studies. These longer case studies provide an opportunity to document how to integrate the functionality contained in different EarthPy modules, with an emphasis on compelling visualizations that convey key concepts for spatial data processing.

EarthPy in Context

EarthPy Focus on Integration of Spatial Data By Scientists

EarthPy is an open source Python package that makes it easier to plot and work with both spatial raster and vector data using open source tools. EarthPy's goal is to make working with spatial data easier for scientists who want to use open source Python tools for analysis and visualization.

Earthpy depends upon GeoPandas (Jordahl et al., 2019), which has a focus on vector data handling and analysis, and Rasterio (Gillies & others, n.d.), which facilitates input and output of raster data files as numpy arrays. It also requires Matplotlib for plotting operations.

To simplify dependency management and installation for non-experts, we maintain a version of EarthPy on the conda-forge channel, which installs the system libraries upon which EarthPy depends. This combined with high-level wrapper around GeoPandas, Rasterio, and Matplotlib (Hunter, 2007) lowers the barrier to entry for people, particularly scientists, who are learning how to work with spatial data in Python.

While there are other useful Python packages for working with vector data such as PySAL (Rey & Anselin, 2007) or raster data such as GeoRasters, EarthPy draws from both GeoPandas and Rasterio to integrate functionality for vector and raster into one package.

EarthPy in the Classroom

EarthPy also supports education and teaching. The io module makes it easier for a student to download a suite of teaching data subsets and other data to a standard working directory

(that is automatically created if it does not exist). This supports reproducibility of workflows in a classroom (or other) setting.

The `plot` module facilitates quick and early data exploration by introductory-level students for whom the intricancies of customizing plots with `Matplotlib` might be overwhelming. The `mask` and `spatial` modules both reduce the technical learning curve for spatial analysis with `Python`, which supports instructors in focusing on the key scientific concepts behind the code.

The vignettes developed with `EarthPy` also provide easily adaptable starting points for in-class exercises that help students learn key spatial concepts using scientific data.

Acknowledgements

There have been many contributors to earthpy that we are thankful for. We are also thankful for the feedback that we recieved through the software review implemented by pyOpenSci. Specifically we thank Luiz Irber who has served as an editor for this review and the two reviewers: Sean Gillies and Rohit Goswami.

References

Gillies, S., & others. (n.d.). *Rasterio: Geospatial raster i/o for Python programmers*. Mapbox. Retrieved from https://github.com/mapbox/rasterio

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. doi:10.1109/MCSE.2007.55

Jordahl, K., Bossche, J. V. den, Wasserman, J., McBride, J., Gerard, J., Fleischmann, M., Tratner, J., et al. (2019). *Geopandas/geopandas: V0.6.1*. Zenodo. doi:10.5281/zenodo.3483425

Rey, S. J., & Anselin, L. (2007). PySAL: A Python Library of Spatial Analytical Methods. *The Review of Regional Studies*, *37*(1), 5–27.