
xshape Documentation

Release 0.1.1

ClimateImpactLab

Feb 22, 2020

Contents

1	Features	3
2	Usage	5
2.1	Getting records for fields in a shapefile	5
2.2	Drawing shape boundaries on gridded data	6
2.3	Plotting regional data in a choropleth	6
3	Contents	9
3.1	Installation	9
3.2	Usage	10
3.3	Contributing	10
3.4	Credits	12
3.5	History	12
4	Indices and tables	13

Tools for working with shapefiles, topographies, and polygons in xarray

- Free software: MIT license
- Documentation: <https://xshape.readthedocs.io>.

CHAPTER 1

Features

- Read a shapefile and obtain an xarray DataArray of field records
- Draw shapefile boundaries on gridded data
- Plot xarray DataArray data indexed by shapefile records as a choropleth

2.1 Getting records for fields in a shapefile

```

In [1]: import xshape

In [2]: fields, polygons = xshape.parse_shapefile(
...:     '../tests/data/shapefiles/CA_counties/CA_counties',
...:     encoding='latin1')
...:

In [3]: fields
Out[3]:
<xarray.Dataset>
Dimensions:    (shape: 58)
Coordinates:
  * shape      (shape) int64 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 ...
Data variables:
  STATEFP     (shape) |S22 '06' '06' '06' '06' '06' '06' '06' '06' '06' '06' '06' ...
  COUNTYFP    (shape) |S22 '107' '009' '047' '079' '097' '041' '023' '051' ...
  COUNTYNS    (shape) |S22 '00277318' '01675885' '00277288' '00277304' ...
  GEOID       (shape) |S22 '06107' '06009' '06047' '06079' '06097' '06041' ...
  NAME        (shape) |S22 'Tulare' 'Calaveras' 'Merced' 'San Luis Obispo' ...
  NAMELSAD    (shape) |S22 'Tulare County' 'Calaveras County' ...
  LSAD        (shape) |S22 '06' '06' '06' '06' '06' '06' '06' '06' '06' '06' ...
  CLASSFP     (shape) |S22 'H1' 'H1' 'H1' 'H1' 'H1' 'H1' 'H1' 'H1' 'H1' 'H1' ...
  MTFCC       (shape) |S22 'G4020' 'G4020' 'G4020' 'G4020' 'G4020' 'G4020' ...
  CSAFP       (shape) |S22 '' '' '' '' '488' '488' '' '' '' '' '488' '472' ...
  CBSAFP      (shape) |S22 '47300' '' '32900' '42020' '42220' '41860' ...
  METDIVFP    (shape) |S22 '' '' '' '' '' '41884' '' '' '' '' '36084' '' '' ...
  FUNCSTAT    (shape) |S22 'A' 'A' 'A' 'A' 'A' 'A' 'A' 'A' 'A' 'A' 'A' ...
  ALAND       (shape) |S22 '12494707314' '2641820029' '5011554680' ...
  AWATER      (shape) |S22 '37391604' '43810423' '112760479' '820974619' ...
  INTPTLAT    (shape) |S22 '+36.2288317' '+38.1846184' '+37.1948063' ...
  INTPTLON    (shape) |S22 '-118.7810618' '-120.5593996' '-120.7228019' ...

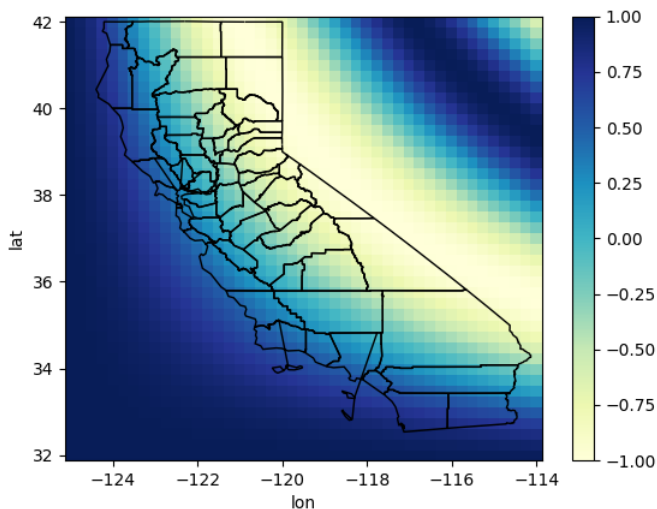
```

2.2 Drawing shape boundaries on gridded data

```
In [4]: import xarray as xr, numpy as np, xshape

# generate sample data
In [5]: da = xr.DataArray(
...:     np.cos((
...:         np.arange(41*45).reshape((41, 45)) * np.arange(41*45).reshape((45, 41)
...:         ).T)/4e5),
...:     dims=('lat', 'lon'),
...:     coords={
...:         'lon': np.linspace(-125, -114, 45),
...:         'lat': np.linspace(32, 42, 41)})
...:

In [6]: da.xshape.overlay(
...:     '../tests/data/shapefiles/CA_counties/CA_counties',
...:     encoding='latin1',
...:     cmap='YlGnBu');
...:
```



2.3 Plotting regional data in a choropleth

Using the xarray extension, we can plot DataArray data directly:

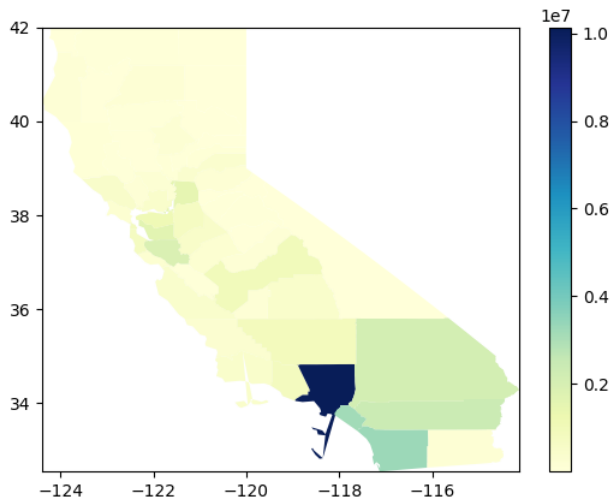
```
In [7]: import xshape, xarray as xr, pandas as pd

In [8]: df = pd.read_csv('../tests/data/datasets/co-est2016.csv', encoding='latin1')
...: ca = df[(df['STATE'] == 6) & (df['COUNTY'] > 0)].copy()
...: ca['fips'] = df['STATE'] * 1000 + df['COUNTY']
...: da = ca.set_index(['fips'])['POPESTIMATE2016'].to_xarray()
...: da.coords['GEOID'] = ('fips', ), list(map('{:05}'.format, da.fips.values))
...: da = da.swap_dims({'fips': 'GEOID'})
...:
```

(continues on next page)

(continued from previous page)

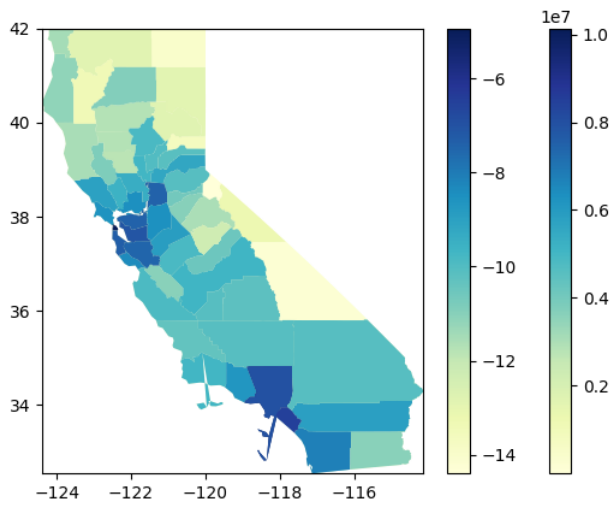
```
In [14]: da.xshape.plot(
.....:     '../tests/data/shapefiles/CA_counties/CA_counties',
.....:     encoding='latin1',
.....:     cmap='YlGnBu');
.....:
```



We can also combine the information from the fields with the data contained in the DataArray:

```
In [15]: land_area = (
.....:     fields
.....:     .set_coords('GEOID')
.....:     .swap_dims({'shape': 'GEOID'})
.....:     .ALAND.astype(float))
.....:

In [16]: np.log(da / land_area).xshape.plot(
.....:     '../tests/data/shapefiles/CA_counties/CA_counties',
.....:     encoding='latin1',
.....:     cmap='YlGnBu');
.....:
```



3.1 Installation

3.1.1 Stable release

To install xshape, run this command in your terminal:

```
$ pip install xshape
```

This is the preferred method to install xshape, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

3.1.2 From sources

The sources for xshape can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/ClimateImpactLab/xshape
```

Or download the `tarball`:

```
$ curl -OL https://github.com/ClimateImpactLab/xshape/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

3.2 Usage

To use xshape in a project:

```
import xshape
```

3.3 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

3.3.1 Types of Contributions

Report Bugs

Report bugs at <https://github.com/ClimateImpactLab/xshape/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

Write Documentation

xshape could always use more documentation, whether as part of the official xshape docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/ClimateImpactLab/xshape/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

3.3.2 Get Started!

Ready to contribute? Here's how to set up *xshape* for local development.

1. Fork the *xshape* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/xshape.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv xshape
$ cd xshape/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 xshape tests
$ pytest -m tests.run_tests
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

3.3.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/ClimateImpactLab/xshape/pull_requests and make sure that the tests pass for all supported Python versions.

3.3.4 Tips

To run a subset of tests:

```
$ pytest tests.test_xshape
```

3.4 Credits

This repository is a project of the [Climate Impact Lab](#)

3.4.1 Development Lead

- Michael Delgado <delgado.michaelt@gmail.com>

3.4.2 Contributors

None yet. Why not be the first?

3.5 History

3.5.1 0.1.0 (2018-01-13)

- First release on PyPI.

CHAPTER 4

Indices and tables

- `genindex`
- `modindex`
- `search`