
esal Documentation

Release 0.3.1

Aubrey Barnard, Akshay Sood

May 04, 2019

Contents:

1	Event Sequence Analysis Library	1
1.1	Documentation	1
1.2	Features	1
1.3	Requirements	2
1.4	Install	2
1.5	License	2
1.6	Concepts	2
1.7	Contact	2
2	Installation	3
2.1	Stable release	3
2.2	From sources	3
3	Usage	5
4	Contributing	7
4.1	Types of Contributions	7
4.2	Get Started!	8
4.3	Pull Request Guidelines	9
4.4	Tips	9
4.5	Deploying	9
5	Credits	11
6	History	13
6.1	0.3.1 (2019-05-03)	13
7	Indices and tables	15

Event Sequence Analysis Library

Esal (“easel”) is a library for the descriptive statistical analysis and manipulation of event sequences and timelines. Esal is intended to be used for exploring event sequence data and preparing data for modeling, but does not do any modeling itself. Conceptually, Esal is a representation for a dataset of sequences / timelines and an associated set of meaningful operations (selection, counting, transformation).

1.1 Documentation

<https://esal.readthedocs.io>

1.2 Features

This project is in the early design and implementation stages. The implemented features are:

- Event objects
- Event sequence data structure for the efficient querying of event sequences
- Interval objects
- Allen’s interval algebra (constant-time implementation that uses the minimum number of comparisons)

The planned features are:

- Selection and counting of sequences
- Selection and counting of events
- Sampling events and sequences

- Temporal statistics
- Reading/Writing various relational and flat representations

Some of the planned features have already been implemented but haven't yet been organized into an official API. Feel free to look through the code. Suggestions are welcome.

1.3 Requirements

- Python 3

1.4 Install

```
pip3 install [--user] https://github.com/cloudbopper/esal/archive/<name>.zip#egg=esal
```

Replace `<name>` with the name of the tag, branch, or commit you want to install, e.g. “master” or “v0.2.0”. If you don't have a `pip3`, replace it with `python3 -m pip`. For more information, see the [Pip documentation](#).

1.5 License

Esal is free, open source software. It is released under the MIT license. See the *LICENSE* file for details.

1.6 Concepts

See the [package documentation](#) for a conceptual overview of events and sequences.

1.7 Contact

- [Aubrey Barnard](#)
- [Akshay Sood](#)

[Open an issue](#) to report a bug or ask a question. To contribute, use the regular fork and pull request work flow.

Copyright (c) 2019 Aubrey Barnard, Akshay Sood. This is free software. See LICENSE for details.

2.1 Stable release

To install esal, run this command in your terminal:

```
$ pip install esal
```

This is the preferred method to install esal, 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.

2.2 From sources

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

You can either clone the public repository:

```
$ git clone git://github.com/cloudbopper/esal
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/cloudbopper/esal/tarball/master
```

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

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


CHAPTER 3

Usage

To use esal in a project:

```
import esal
```


Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at <https://github.com/cloudbopper/esal/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.

4.1.2 Fix Bugs

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

4.1.3 Implement Features

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

4.1.4 Write Documentation

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

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/cloudbopper/esal/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 :)

4.2 Get Started!

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

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

```
$ git clone git@github.com:your_name_here/esal.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 esal
$ cd esal/
$ 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 esal tests
$ python setup.py test or py.test
$ 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.

4.3 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 3.5 and 3.6, and for PyPy. Check https://travis-ci.org/cloudbopper/esal/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ py.test tests.test_esal
```

4.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

CHAPTER 5

Credits

- Aubrey Barnard
- Akshay Sood

CHAPTER 6

History

6.1 0.3.1 (2019-05-03)

- Forked from <https://github.com/afbarnard/esal.git>

CHAPTER 7

Indices and tables

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