
SciKit Data Documentation

Release 0.1.3

Ivan Ogasawara

May 21, 2019

Contents

1	SciKit Data	3
1.1	Conda package current release info	3
1.2	About SciKit Data	3
1.3	Installing scikit-data	4
2	Installation	7
2.1	Using conda	7
2.2	Using pip	7
2.3	From sources	7
3	Usage	9
4	Contributing	11
4.1	Types of Contributions	11
4.2	Get Started!	12
4.3	Pull Request Guidelines	13
4.4	Tips	13
5	Indices and tables	15

Contents:

1.1 Conda package current release info

1.2 About SciKit Data

The propose of this library is to allow the data analysis process more easy and automatic.

General objectives:

- reduce boilerplate code;
- reduce time spent on data analysis tasks and;
- offer a reproducible data analysis workflow.

Generally, there is a lot of boilerplate code on data analysis task that could be resolved with reproducible mechanisms and easy data visualization methods. Another point is related to data publish. A lot of data analysts doesn't know about open data repositories or doesn't consider that in his/her scientific workflow communication.

Specifics objectives:

- optimize data visualization;
- integration with open data repositories to publish data;
- reproducibility on data analysis tasks through storing and recovery operations;

SkData should integrate with Pandas library (Python).

1.2.1 Books used as reference to guide this project:

- <https://www.packtpub.com/big-data-and-business-intelligence/clean-data>
- <https://www.packtpub.com/big-data-and-business-intelligence/python-data-analysis>
- <https://www.packtpub.com/big-data-and-business-intelligence/mastering-machine-learning-scikit-learn>
- <https://www.packtpub.com/big-data-and-business-intelligence/practical-data-analysis-second-edition>

1.2.2 Some other materials used as reference:

- https://github.com/rsouza/MMD/blob/master/notebooks/3.1_Kaggle_Titanic.ipynb
- <https://github.com/agconti/kaggle-titanic/blob/master/Titanic.ipynb>
- <https://github.com/donnemartin/data-science-ipython-notebooks/blob/master/kaggle/titanic.ipynb>

1.3 Installing scikit-data

1.3.1 Using conda

Installing *scikit-data* from the *conda-forge* channel can be achieved by adding *conda-forge* to your channels with:

```
$ conda config --add channels conda-forge
```

Once the *conda-forge* channel has been enabled, *scikit-data* can be installed with:

```
$ conda install scikit-data
```

It is possible to list all of the versions of *scikit-data* available on your platform with:

```
$ conda search scikit-data --channel conda-forge
```

1.3.2 Using pip

To install *scikit-data*, run this command in your terminal:

```
$ pip install skdata
```

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

1.3.3 More Information

- License: MIT
- Documentation: <https://skdata.readthedocs.io>

1.3.4 References

- CUESTA, Hector; KUMAR, Sampath. Practical Data Analysis. Packt Publishing Ltd, 2016.

Electronic materials

- [1] <http://www.datasciencecentral.com/profiles/blogs/introduction-to-outlier-detection-methods>

2.1 Using conda

Installing *scikit-data* from the *conda-forge* channel can be achieved by adding *conda-forge* to your channels with:

```
$ conda config --add channels conda-forge
```

Once the *conda-forge* channel has been enabled, *scikit-data* can be installed with:

```
$ conda install scikit-data
```

It is possible to list all of the versions of *scikit-data* available on your platform with:

```
$ conda search scikit-data --channel conda-forge
```

2.2 Using pip

To install *scikit-data*, run this command in your terminal:

```
$ pip install skdata
```

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

2.3 From sources

The sources for *scikit-data* can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/OpenDataScienceLab/skdata
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/OpenDataScienceLab/skdata/tarball/master
```

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

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

CHAPTER 3

Usage

To use SciKit Data Analysis in a project:

```
import skdata
```


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/OpenDataScienceLab/skdata/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

Jupyter Python Data Analysis could always use more documentation, whether as part of the official Jupyter Python Data Analysis 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/xmnlabs/skdata/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 *skdata* for local development.

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

```
$ git clone git@github.com:your_name_here/skdata.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 skdata
$ cd skdata/
$ 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 skdata 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 3.4 and 3.5. Check https://travis-ci.org/xmnlabs/skdata/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_skdata
```


CHAPTER 5

Indices and tables

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