
snowplot Documentation

Release 0.2.0

Micah Johnson

Dec 13, 2021

CONTENTS:

1	snowplot	1
1.1	Usage	1
1.2	Features	1
1.3	Credits	1
2	Installation	3
2.1	Stable release	3
2.2	From sources	3
3	Usage	5
4	Configuration File Reference	7
4.1	lyte_probe	7
4.2	snow_microopen	10
4.3	hand_hardness	11
4.4	output	13
5	Contributing	15
5.1	Types of Contributions	15
5.2	Get Started!	16
5.3	Pull Request Guidelines	17
5.4	Tips	17
5.5	Deploying	17
6	Credits	19
6.1	Development Lead	19
6.2	Contributors	19
7	History	21
7.1	0.1.0 (2020-02-10)	21
8	Indices and tables	23

SNOWPLOT

SnowPlot is a command line tool for snow professionals who need to make vertical plots to display various snowpack data. This was made from the need to make figures to validate the Lyte Probe. To that end we are adding density profiles, hand hardness profiles, snow micropen profiles, and more. The tool is based on a configuration file where the user can use a simple text editor to generate complicated figures.

1.1 Usage

```
snowplot config.ini
```

- Free software: BSD license
- Documentation: <https://snowplot.readthedocs.io>.

1.2 Features

- Manage repeatable figures using a config file
- Add annotations to plots with simple text
- Plot Lyte Probe Data
- Plot hand hardness data
- Plot Snow micropen data

1.3 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

INSTALLATION

2.1 Stable release

To install snowplot, run this command in your terminal:

```
$ pip install snowplot
```

This is the preferred method to install snowplot, 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 snowplot can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/micahjohnson150/snowplot
```

Or download the [tarball](#):

```
$ curl -OJL https://github.com/micahjohnson150/snowplot/tarball/master
```

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

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

CHAPTER THREE

USAGE

To use snowplot from the commandline:

```
snowplot config.ini
```


CONFIGURATION FILE REFERENCE

The snowplot's configuration file is described in detail below. This information is all based on the master.ini file stored under the top level of the package.

For configuration file syntax information please visit <http://inicheck.readthedocs.io/en/latest/>

4.1 lyte_probe

add_average_column

If true will average together all the columns

Default: False

Type: bool

assumed_depth

Depth in cms to assumed a linear depth profile to with the timeseries data

Default: None

Type: int

bottom_depth

Depth in CM where the movement stopped

Default: None

Type: float

calibration_coefficients

Polynomial coefficients to use for processing the column_to_plot data

Default: 1

Type: float

color

Decimal RGB Color to use for the plot

Default: 0.455.784 1

Type: float

columns_to_plot

List of columns to plot in the data

Default: sensor_1

Type: string

detect_surface

Use the NIR sensors to determine the location of the snow surface.

Default: False

Type: bool

filename

Filename to be plotted

Default: None

Type: criticalfilename

fill_solid

Determines whether to fill in the profile solid to the y axis

Default: True

Type: bool

plot_labels

a list of tuples containing labels to add to the plot

Default: None

Type: string

problem_layer

Depth in centimeters to place a red horizontal line on the plot.

Default: None

Type: float

smoothing

Rolling window over each column to apply an averaging filter

Default: None

Type: int

surface_depth

Depth in CM where the snow surface begins

Default: 0

Type: float

title

Plot title for the Lyte probe

Default: lyte probe

Type: string

use_filename_title

Use the filename to for the lyte plot title

Default: True

Type: bool

xlabel

Label to put under the x axis

Default: Hardness (mN)

Type: string

xlimits

Range in the X Axis to plot

Default: None

Type: floatpair

ylabel

Label on the y axis

Default: Depth from surface (cm)

Type: string

ylimits

Range of depths in cm to plot

Default: -100 0

Type: floatpair

4.2 snow_micropen

color

Decimal RGB Color to use for the plot

Default: 0.211 .27 .31 1

Type: float

filename

Filename to be plotted

Default: None

Type: criticalfilename

fill_solid

Determines whether to fill in the profile solid to the y axis

Default: False

Type: bool

plot_labels

a list of tuples containing labels to add to the plot

Default: None

Type: string

problem_layer

Depth in centimeters to place a red horizontal line on the plot.

Default: None

Type: float

smoothing

Rolling window over each column to apply an averaging filter

Default: None

Type: int

title

Plot title the snow micro pen

Default: snow micropen

Type: string

use_filename_title

Use the filename to for the SMP title

Default: True

Type: bool

xlabel

Label to put under the x axis

Default: Force (mN)

Type: string

xlimits

Range in the X Axis to plot

Default: None

Type: floatpair

ylabel

Label on the y axis

Default: Depth from surface (cm)

Type: string

ylimits

Range of depths in cm to plot

Default: -100 0

Type: floatpair

4.3 hand_hardness

color

Decimal RGB Color to use for the plot

Default: 0.603 0.6 0.84 1

Type: float

filename

Filename to be plotted

Default: None

Type: criticalfilename

plot_labels

a list of tuples containing labels to add to the plot

Default: None

Type: string

problem_layer

Depth in centimeters to place a red horizontal line on the plot.

Default: None

Type: float

title

Plot title for the hand hardness profile

Default: Hand Hardness

Type: string

xlabel

Label to put under the x axis

Default: Hand Hardness

Type: string

xlimits

Range in the X Axis to plot

Default: None

Type: floatpair

ylabel

Label on the y axis

Default: Depth from surface (cm)

Type: string

ylimits

Range of depths in cm to plot

Default: -100 0

Type: floatpair

4.4 output

dpi

Resolution of the image to produce

Default: 500

Type: int

figure_size

Must be a list of two items specifying figure size in inches

Default: 6 10

Type: float

file_type

File format of the figure to be outputted

Default: png

Type: string

Options: png jpg pdf svg eps

filename

filename for the outputted figure

Default: None

Type: string

output_dir

Location to save figures

Default: ./output

Type: directory

show_plot

Show the plot to be outputted

Default: True

Type: bool

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:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at <https://github.com/micahjohnson150/snowplot/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.

5.1.2 Fix Bugs

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

5.1.3 Implement Features

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

5.1.4 Write Documentation

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

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/micahjohnson150/snowplot/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 :)

5.2 Get Started!

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

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

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

5.3 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, 3.6, 3.7 and 3.8, and for PyPy. Check https://travis-ci.com/micahjohnson150/snowplot/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ pytest tests.test_snowplot
```

5.5 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:

```
$ bump2version patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

CREDITS

6.1 Development Lead

- Micah Johnson <micah.johnson150@gmail.com>

6.2 Contributors

None yet. Why not be the first?

HISTORY

7.1 0.1.0 (2020-02-10)

- First release on PyPI.

INDICES AND TABLES

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