



Search projects

[Help](#)[Sponsors](#)[Log in](#)[Register](#)

# lttb 0.3.1

✓ Latest version

pip install lttb 

Released: Oct 14, 2020

Largest-Triangle-Three-Buckets algorithm for downsampling time series-like data

## Navigation

 [Project description](#) [Release history](#) [Download files](#)

## Project links

 [Homepage](#) [Source](#) [Tracker](#)

## Statistics

View statistics for this project via [Libraries.io](#), or by using [our public dataset on Google BigQuery](#)

## Meta

License: MIT License

Author: [JA Viljoen](#) 

## Maintainers

[javiljoen](#)

## Classifiers

Development Status

- 4 - Beta

## Project description

Numpy implementation of Steinarsson's *Largest-Triangle-Three-Buckets* algorithm for downsampling time series-like data while retaining the overall shape and variability in the data

LTTB is well suited to filtering time series data for visual representation, since it reduces the number of *visually redundant* data points, resulting in smaller file sizes and faster rendering of plots.

Note that it is not a technique for statistical aggregation, cf. regression models or non-parametric curve fitting / smoothing.

This implementation is based on the original JavaScript code at <https://github.com/sveinn-steinsson/plot-downsample> and Sveinn Steinarsson's 2013 MSc thesis *Downsampling Time Series for Visual Representation*.

Licence: MIT

## Usage

Install the `lttb` package into your (virtual) environment:

```
$ pip install lttb
```

The function `lttb.downsample()` can then be used in your Python code:

```
import numpy as np
import lttb

# Generate an example data set of 100 random points:
# - column 0 represents time values (strictly increasing)
# - column 1 represents the metric of interest: CPU usage, stock price, etc.
data = np.array([range(100), np.random.random(100)]).T

# Downsample it to 20 points:
small_data = lttb.downsample(data, n_out=20)
assert small_data.shape == (20, 2)
```

A test data set is provided in the source repo in `tests/timeseries.csv`. It was downloaded from <http://flot.base.is/> and converted from JSON to CSV.

👤 Love Python? Support the PSF in 2023! [Give Today](#)

👤 2FA will soon be required on PyPI [Read more](#)

#### License

- [OSI Approved :: MIT License](#)

#### Operating System

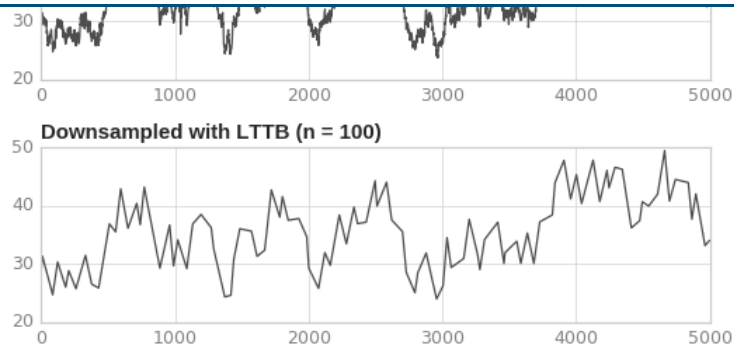
- [OS Independent](#)

#### Programming Language

- [Python :: 2](#)
- [Python :: 2.7](#)
- [Python :: 3](#)
- [Python :: 3.5](#)
- [Python :: 3.6](#)
- [Python :: 3.7](#)
- [Python :: 3.8](#)
- [Python :: 3.9](#)

#### Topic

- [Office/Business :: Financial :: Investment](#)
- [Scientific/Engineering :: Visualization](#)



### Input validation

By default, `downsample()` checks that the input data satisfies the following constraints:

- it is a two-dimensional array of two columns;
- the values in the first column are strictly increasing; and
- there are no missing (NaN) values in the data.

These checks can be skipped (e.g. if you know that your data will always meet these conditions), or additional checks can be added (e.g. that the time values must be evenly spaced), by passing in a different list of validation functions, e.g.:

```
# No input validation:
small_data = lttb.downsample(data, n_out=20, validators=[])

# Stricter check on x values:
from lttb.validators import *
small_data = lttb.downsample(data, n_out=20, validators=[has_two_columns, x_is_regular])
```

### History

#### 0.3.1 / 2020-10-14

- All modules and functions now have docstrings.
- [dev] The library is now also tested against Python 3.9. No changes to the code were required.
- [dev] The project is now hosted on SourceHut; the links on the PyPI page have been updated.
- [dev] CI testing has been migrated from Travis to builds.sr.ht.

#### 0.3.0 / 2020-09-15

- Validation of input data is now configurable.
- New default: `downsample()` raises `ValueError` if input data contains NaN values. This can be disabled by removing `contains_no_nans()` from the list of validators.
- [dev] Imports are now sorted with `isort`.

#### 0.2.2 / 2020-01-08

- `setup.py` was fixed so that this package can be installed in Python 2 again.

#### 0.2.1 / 2019-11-25

🐍 Love Python? Support the PSF in 2023!

Give Today ↗

👤 2FA will soon be required on PyPI

Read more ↗

- Performance improvements
- Released on PyPI (on 2019-11-06)

0.1.0 / 2017-03-18

- Initial implementation

Contributors

- JA Viljoen – original Numpy implementation
- Guillaume Bethouart – performance improvements
- Jens Krüger – fix for py27



Help

- Installing packages ↗
- Uploading packages ↗
- User guide ↗
- Project name retention ↗
- FAQs

About PyPI

- PyPI on Twitter ↗
- Infrastructure dashboard ↗
- Statistics
- Logos & trademarks
- Our sponsors

Contributing to PyPI

- Bugs and feedback
- Contribute on GitHub ↗
- Translate PyPI ↗
- Sponsor PyPI
- Development credits ↗

Using PyPI

- Code of conduct ↗
- Report security issue
- Privacy policy ↗
- Terms of use
- Acceptable Use Policy

Status: Service Under Maintenance ↗

Developed and maintained by the Python community, for the Python community.  
Donate today!

"PyPI", "Python Package Index", and the blocks logos are registered trademarks of the Python Software Foundation ↗.

© 2023 Python Software Foundation ↗  
Site map

Switch to desktop version

➤ English   español   français   日本語   português (Brasil)   українська   Ελληνικά   Deutsch   中文 (简体)   中文 (繁體)   русский   עברית   esperanto

PyPI logo

PyPI logo

AWS  
Cloud computing  
and Security  
Sponsor

Datadog  
Monitoring

Fastly  
CDN

Google  
Download Analytics

Microsoft  
PSF Sponsor

Pingdom  
Monitoring

Sentry  
Error logging

StatusPage  
Status page