OBJECTIVE LOW FLOW THRESHOLD IDENTIFICATION USING BREAKPOINT ANALYSIS

SUPPLEMENTARY MATERIALS

Table of Contents

Description

objective_thresholds is a Python package for objective threshold level calculation based on a flow duration curve (FDC). At the time of publication this package contains tools for threshold calculation for low flow analysis, contained in the accompanying *lowflow* module. The functions included in the module allow for the computation of a single threshold value (function *threshold*) on a given lower range of the FDC, or multiple (function *multiple*) thresholds if additional division is needed (i.e., “shallow” and “deep” low flows). Further implementations and modules will be added in the future to allow for additional threshold calculations and analysis at full range of FDC.

How to cite.

If you use *objective_thresholds* in a scientific publication, please include the reference to article Raczyński K., Dyer J., 2022, Development of an Objective Low Flow Identification Method Using Breakpoint Analysis

Website:

Official repository website address:

```
https://github.com/chrisrac/objective_thresholds/
```

Installation

With pip:

```
Pip install objective_thresholds
```

Dependencies

The *objective_thresholds* package requires the following:

- Python 3.4+
- Pandas
- NumPy
- Jenkspy

Usage

The *objective_thresholds* package currently includes the module *lowflow* with two functions available. To use the package, once installed it should be imported to Python:

```
import objective_thresholds.lowflow as olf
```

The function *threshold* allows for the computation of low flow thresholds using breakpoints through the application of the Fisher-Jenks algorithm, which is applied to the lower range of the FDC (calculated from provided dataset: *input_data*). By default, the lowest 35% of the FDC is used for low flow threshold calculations, and one threshold value is returned. The function accepts as input any array-like data, including: lists, pandas Series, numpy array, or slices of pandas DataFrame. Optionally, the *limit* of the lower FDC can be set by the user if a value other than the default 35th percentile is needed. If the input data are highly discretized (contains numerous repetitive values), the *method* argument controls the result:

- *leave* – breaks the computation, Exception is returned;
- *max* – the maximum value in the lower FDC range is returned;
- *median* – the median value in the lower FDC range is returned.

The *threshold* function returns one float or int type value representing the identified threshold, depending on the input data type. Some examples of usage:

```
# importing lowflow module from objective_thresholds package

>>> import objective_thresholds.lowflow as olf

>>> import pandas as pd


# importing example data from testdata.csv file (available at source repository)

>>> data = pd.read_csv('home/testdata.csv')
```

```
# checking data structure
>>> data.head()
           id  flow1  flow2
0           0   2.71   2.84
1           1   0.31   0.34
2           2   0.16   0.18
3           3   0.23   0.26
4           4   0.26   0.28

# calculating threshold value for first data series using default settings
(35th percentile; limit=0.35)
>>> threshold_value = olf.threshold(data['flow1'])
# results in: 0.13

# modifying range of lower FDC to 50th percentile
>>> threshold_50p = olf.threshold(data['flow1'], limit=0.5)
# results in: 0.19

# an Exception is raised if fewer than 5 unique values are present
>>> threshold_10p = olf.threshold(data['flow2'], limit=0.10)
Exception: Unique values of lower FDC range are less than 5
and used method is "leave", no thresholds are returned.

# controlling output by changing the method for discretized data
threshold_max = olf.threshold(data['flow2'], limit=0.10, method = 'max')
# results in: 0.11
```

The function *multiple* allows for the computation of multiple low flow thresholds using breakpoints through the application of the Fisher-Jenks algorithm, which is applied to the lower range of the FDC (provided as *input_data*). By default, the lowest 35% of the FDC is used for low flow threshold calculations, and multiple threshold values are returned. This function is to be used when at least two thresholds are needed, such as when dividing the lower FDC range into "shallow" and "deep" low flow events. The function accepts as input any array-like data, including: lists, pandas Series, numpy array, or slices of pandas DataFrame. Argument *breaks* controls the number of thresholds to be returned. Optionally, the *limit* of the lower FDC can be set by the user if a value other than the default 35th percentile is needed.. If the input data are highly discretized (contains numerous repetitive values), the *method* argument controls the result:

- *leave* – breaks the computation, Exception is returned;
- *max* – the maximum value in the lower FDC range is returned;
- *median* – the median value in the lower FDC range is returned.

Note that the implementation of the *max* or *median* arguments causes the *multiple* function to return a single value, regardless of the number of breakpoints requested.

The *multiple* function returns a list of float or int type values, depending on the input data type, representing the identified thresholds. Some examples of usage:

```
# importing lowflow module from objective_thresholds package
>>> import objective_thresholds.lowflow as olf
>>> import pandas as pd

# importing example data from testdata.csv file (available at source
repository)
>>> data = pd.read_csv('home/testdata.csv')
# checking data structure
>>> data.head()
           id  flow1  flow2
0           0   2.71   2.84
1           1   0.31   0.34
2           2   0.16   0.18
3           3   0.23   0.26
4           4   0.26   0.28

# calculating threshold value for first data series using default settings
(limit=0.35; breaks=2)
>>> threshold_value = olf.multiple(data['flow1'])
# results in: [0.1, 0.16]

# modifying range of lower FDC to 50th percentile
>>> threshold_50p = olf.multiple(data['flow1'], limit=0.5)
# results in: [0.14, 0.24]

# modifying the number of thresholds to be returned
```

```
>>> threshold_3th = olf.multiple(data['flow1'], breaks=3)
# results in: [0.09, 0.13, 0.17]

# an Exception is raised if fewer than 5 unique values are present
>>> threshold_10p = olf.multiple(data['flow2'], limit=0.10)
Exception: Unique values of lower FDC range are less than 5
and used method is "leave", no thresholds are returned.

# controlling output by changing the method for discretized data
threshold_max  =  olf.multiple(data['flow2'],  limit=0.10,  method  =  'median',
breaks=6)
# results in: 0.08 as only one value is returned by max or median method,
regardless of breaks setting
```

License