

---

**ticts**

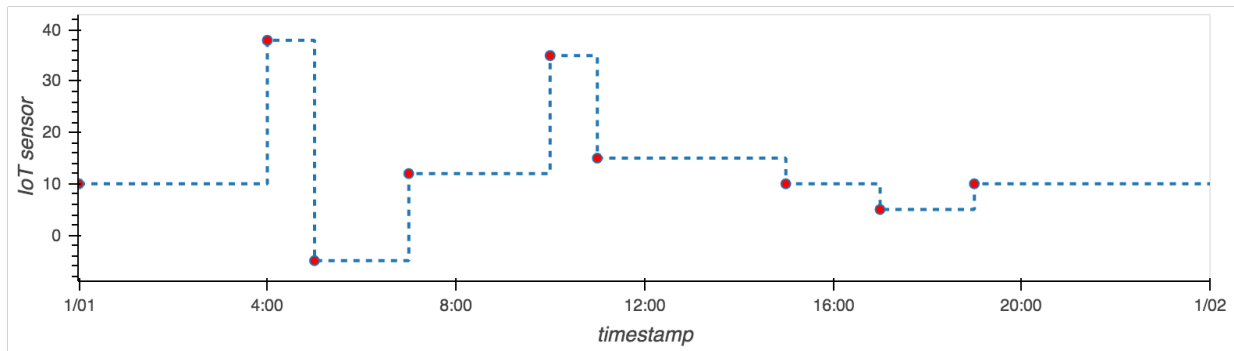
**Jun 28, 2022**



<b>1</b>	<b>Why ?</b>	<b>3</b>
<b>2</b>	<b>About TicTs</b>	<b>5</b>
2.1	QuickStart . . . . .	5
2.2	TimeSeries . . . . .	9
2.3	Authors . . . . .	10
2.4	CHANGES . . . . .	10
<b>3</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>
	<b>Index</b>	<b>21</b>



A Python library for unevenly-spaced timeseries analysis.



Notebooks to play around are available [here](#).



# CHAPTER 1

---

## Why ?

---

Sensors become omnipresents, and often the measurements are done at irregular time intervals. I found it hard to work with using pandas, and found myself always swithing back to the evenly-spaced timeseries world, by resampling at the lowest frequency applying a fill forward. This is definitely not how it should works ! And you loose information.





# CHAPTER 2

---

## About TicTs

---

The purpose is to provide a simple and intuitive interface to manipulate this kind of data, including:

- operations (sum, sub, mul, div, max, min)
- setting intervals
- methods to pass from evenly-spaced to unevenly-spaced and vice-versa

You have to keep in mind the following rules:

- intervals are open left, closed right
- timestamp are always localized, defaulting to UTC
- setting a default value to your timeseries might be important

## 2.1 QuickStart

### 2.1.1 Instanciation

```
import ticts
ticts.TimeSeries({'2019-01-01': 1, '2019-01-02': 2})
ticts.TimeSeries({'2019-01-01': 1, '2019-01-02': 2}, default=0)

import datetime
dt1 = datetime.datetime(2019, 1, 1)
ticts.TimeSeries({'dt1': 1, '2019-01-02': 2})
```

### 2.1.2 Samples

```
In [1]: dt1
Out[1]: Timestamp('2019-01-01 00:00:00+0000', tz='UTC')
```

```
In [2]: onehour
Out[2]: datetime.timedelta(seconds=3600)
```

```
In [3]: smallts
Out[3]:
<TimeSeries>
2019-01-01T01:00:00+00:00: 1,
2019-01-01T03:00:00+00:00: 3,
2019-01-01T06:00:00+00:00: 10,
```

```
In [4]: otherts
Out[4]:
<TimeSeries> (default=0)
2019-01-01T02:00:00+00:00: 2,
2019-01-01T03:00:00+00:00: 3,
2019-01-01T05:00:00+00:00: 5,
```

### 2.1.3 GetItem

An **interval** is closed left, open right: [, [

```
In [5]: smallts.index
Out[5]: SortedKeysView(SortedDict({Timestamp('2019-01-01 01:00:00+0000', tz='UTC'): 1,
→ Timestamp('2019-01-01 03:00:00+0000', tz='UTC'): 3, Timestamp('2019-01-01_
→ 06:00:00+0000', tz='UTC'): 10}))

# Accessing value at key
In [6]: smallts[dt1 + onehour]
Out[6]: 1

In [7]: smallts[dt1] # no default set

In [8]: otherts[dt1] # default set
Out[8]: 0

# Accessing values sliced
In [9]: smallts[dt1: dt1 + 5 * onehour] # calls TimeSeries.slice
Out[9]:
<TimeSeries>
2019-01-01T01:00:00+00:00: 1,
2019-01-01T03:00:00+00:00: 3,

# Getting previous is the default
In [10]: key = dt1 + 2 * onehour

In [11]: smallts[key, 'previous'] == smallts[key] == 1
Out[11]: True

# linear interpolation is available
In [12]: smallts[key, 'linear']
Out[12]: 2.0
```

## 2.1.4 Set Item, Set Interval

Setting a default to your TimeSeries object is mandatory in order to set intervals.

```
In [13]: ts = TimeSeries(smallts, default=0)
```

```
In [14]: ts[dt1 + onehour] = 2
```

```
In [15]: print(ts)
<TimeSeries> (default=0)
2019-01-01T01:00:00+00:00: 2,
2019-01-01T03:00:00+00:00: 3,
2019-01-01T06:00:00+00:00: 10,
```

```
In [16]: start = dt1 + 4 * onehour
```

```
In [17]: end = dt2 + 7 * onehour
```

```
In [18]: ts.set_interval(start, end, 7)
```

```
In [19]: print(ts)
<TimeSeries> (default=0)
2019-01-01T01:00:00+00:00: 2,
2019-01-01T03:00:00+00:00: 3,
2019-01-01T04:00:00+00:00: 7,
2019-01-02T07:00:00+00:00: 10,
```

```
# same as
```

```
In [20]: ts[start: end] = 7
```

```
In [21]: print(ts)
<TimeSeries> (default=0)
2019-01-01T01:00:00+00:00: 2,
2019-01-01T03:00:00+00:00: 3,
2019-01-01T04:00:00+00:00: 7,
2019-01-02T07:00:00+00:00: 10,
```

## 2.1.5 Operations

Sum

```
In [22]: smallts + 10
```

```
Out[22]:
<TimeSeries> (default=None)
2019-01-01T01:00:00+00:00: 11,
2019-01-01T03:00:00+00:00: 13,
2019-01-01T06:00:00+00:00: 20,
```

```
In [23]: smallts + otherts
```

```
Out[23]:
<TimeSeries>
2019-01-01T01:00:00+00:00: 1,
2019-01-01T02:00:00+00:00: 3,
2019-01-01T03:00:00+00:00: 6,
2019-01-01T05:00:00+00:00: 8,
2019-01-01T06:00:00+00:00: 15,
```

(continues on next page)

(continued from previous page)

```
In [24]: sum([smallts, smallts, smallts])
Out[24]:
<TimeSeries>
2019-01-01T01:00:00+00:00: 3,
2019-01-01T03:00:00+00:00: 9,
2019-01-01T06:00:00+00:00: 30,
```

## Sub

```
In [25]: smallts - otherts
Out[25]:
<TimeSeries>
2019-01-01T01:00:00+00:00: 1,
2019-01-01T02:00:00+00:00: -1,
2019-01-01T03:00:00+00:00: 0,
2019-01-01T05:00:00+00:00: -2,
2019-01-01T06:00:00+00:00: 5,
```

## Comparisons

```
In [26]: smallts <= 10
Out[26]:
<TimeSeries> (default=None)
2019-01-01T01:00:00+00:00: True,
2019-01-01T03:00:00+00:00: True,
2019-01-01T06:00:00+00:00: True,

In [27]: smallts <= otherts
Out[27]:
<TimeSeries>
2019-01-01T01:00:00+00:00: False,
2019-01-01T02:00:00+00:00: True,
2019-01-01T03:00:00+00:00: True,
2019-01-01T05:00:00+00:00: True,
2019-01-01T06:00:00+00:00: False,

In [28]: smallts < 10
Out[28]:
<TimeSeries> (default=None)
2019-01-01T01:00:00+00:00: True,
2019-01-01T03:00:00+00:00: True,
2019-01-01T06:00:00+00:00: False,

In [29]: smallts < otherts
Out[29]:
<TimeSeries>
2019-01-01T01:00:00+00:00: False,
2019-01-01T02:00:00+00:00: True,
2019-01-01T03:00:00+00:00: False,
2019-01-01T05:00:00+00:00: True,
2019-01-01T06:00:00+00:00: False,

In [30]: smallts >= 10
Out[30]:
<TimeSeries> (default=None)
2019-01-01T01:00:00+00:00: False,
```

(continues on next page)

(continued from previous page)

```
2019-01-01T03:00:00+00:00: False,
2019-01-01T06:00:00+00:00: True,
```

```
In [31]: smallts > 10
```

```
Out[31]:
```

```
<TimeSeries> (default=None)
2019-01-01T01:00:00+00:00: False,
2019-01-01T03:00:00+00:00: False,
2019-01-01T06:00:00+00:00: False,
```

## 2.2 TimeSeries

**class** ticts.timeseries.**TimeSeries** (*data=None, default=None, name='value', permissive=True, tz='UTC'*)

TimeSeries object.

### Parameters

- **default** – The default value of timeseries.
- **permissive** (*bool*) – Whether to allow accessing non-existing values or not. If is True, getting non existing item returns None. If is False, getting non existing item raises.

### compact ()

Convert this instance to a compact version: consecutive measurement of the same value are discarded.

**Returns** TimeSeries

### empty

Return whether the TimeSeries is empty or not.

### iterintervals (*end=None*)

Iterator that contain start, end of intervals.

**Parameters** **end** (*datetime*) – right bound of last interval.

### lower\_bound

Return the lower bound time index.

### set\_interval (*start, end, value*)

Set a value for an interval of time.

### Parameters

- **start** (*datetime or str*) – lower bound
- **end** (*datetime or str*) – upper bound
- **value** – the value to be set

**Returns** self

**Raises** NotImplementedError – when no default is set.

### slice (*start, end*)

Slice your timeseries for give interval.

### Parameters

- **start** (*datetime or str*) – lower bound

- **end** (*datetime or str*) – upper bound

**Returns** TimeSeries sliced

**upper\_bound**

Return the upper bound time index.

## 2.3 Authors

Bramver <BramVer@users.noreply.github.com> Jeusel Guillaume <guillaume.jeusel@renewex.co> Remy  
<remy.moutie@engie.com> gjeusel <jeusel.guillaume@gmail.com>

## 2.4 CHANGES

### 2.4.1 0.4.0

- [FIX] iterintervals on new python3.7+ (PEP 479)

### 2.4.2 0.3.5

- Add .keys method to TimeSeries (#21)

### 2.4.3 0.3.4

- Add update method to TimeSeries (#20)

### 2.4.4 0.3.3

- Add iplot method (#18)
- Some renames and ValueError instead of Exception
- Make doc build works again
- solve it adding \_kwargs\_special\_keys property
- Add check to try inferring the frequency
- Try to infer the frequencies when to\_dataframe

### 2.4.5 0.3.1

- Be permissive on division with zero as default

### 2.4.6 0.3.0

- Update Readme
- Create TictsMagicMixin for builtin methods of TimeSeries
- add it
- Linting
- Rework it, no longer inherit from SortedDict
- add test on values
- First implementation

### 2.4.7 0.2.4

- Small fix on div

### 2.4.8 0.2.3

- Precise which super for equals

### 2.4.9 0.2.2

- Render testing available at ticts package place

### 2.4.10 0.2.1

- Re-organize code
- Add extensions to pandas objects 'to\_ticts'
- Fix sphinx doc building
- Add io to\_json / from\_json

### 2.4.11 0.2.0

- Linting
- fix pytest version
- update to\_dataframe method to use self.name
- Add operations
- Remove arrow dep
- Rework init and copy methods to accept pd.DataFrame or pd.Series
- Renames and pandas as mandatory dep
- Clean up small garbage
- Go for a no default object with repr
- Add \_\_eq\_\_ as operation per key

**2.4.12 0.1.7**

- Add iterintervals method
- Remove useless timestamp\_converter
- use bisect

**2.4.13 0.1.6**

- Fix bug of set\_interval on emptyts
- Add some more tests
- only go for python 3.6 at the moment
- Add other python version

**2.4.14 0.1.5**

- Add deploy as stage
- Add ignore on /prof

**2.4.15 0.1.4**

- Make use of SortedDict methods for performances

**2.4.16 0.1.3**

- restructured text of PyPI does not allow image in title
- Skip cleanup in deploy
- Improve notebook
- change strategy to be able to build doc
- Update changelog

**2.4.17 0.1.2**

- Fix on rule of slice out of right bound
- Improving index
- Add more quickstart
- Fix issue on set\_interval
- Add scale
- Work on Docs



### 2.4.18 0.1.1

- Better handle of default value
- Adding picture to README
- Small fix on bounds of tutorial
- Small fix README
- Adapt plot ranges in Tutorial notebook
- Add reference to traces

### 2.4.19 0.1.0

- Enhance Readme and Add jupyter notebook
- cleaning
- Allow operations on mixed ts with default or not
- Add custom repr

### 2.4.20 0.0.7

- Add permissive mode

### 2.4.21 0.0.6

- Authorize getitem on str
- Improving \_\_init\_\_
- move the test to a more appropriate class
- make 0 a possible value for default

### 2.4.22 0.0.5

- Improve conditional\_update and rename it

### 2.4.23 0.0.4

- Fix bug on slice with bounds
- Add operators le lt ge gt and conditional\_update
- Fix several bugs:
- Add recursive operations
- Fix bug on set\_interval

### 2.4.24 0.0.2

- Add pandas potential dep

### 2.4.25 0.0.1

- Seed isort to get consistent behaviour in CI
- Linter
- Update travis cfg to deploy on tag
- Authorize setitem on slice calling set\_interval
- re-organize tests
- Implement \_get\_linear\_interpolate
- Linting
- fix tests
- add show diff on failure
- Add sample method
- cleaning up
- Add floor and ceil operators
- Update license in setup.cfg
- Add test
- fix pre-commit version
- Fix bug on set\_interval
- Comment it
- Change the init to get copy and deepcopy works
- Add logging
- Add tests for set\_interval on empty
- Add set\_interval method
- Linting
- Comment cov when locally run
- Linting
- Make compact work again
- Update precommit
- Improve init and add slicing capabilities
- Change of design
- Linter
- Go for yapf instead of black
- Linting

- Update pre-commit
- Configuring travis
- Linter
- put first stones
- First commit thx to cookiecutter



## CHAPTER 3

---

### Indices and tables

---

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



**t**

`ticts.timeseries`, 9





## C

`compact()` (*ticts.timeseries.TimeSeries* method), 9

## E

`empty` (*ticts.timeseries.TimeSeries* attribute), 9

## I

`iterintervals()` (*ticts.timeseries.TimeSeries*  
method), 9

## L

`lower_bound` (*ticts.timeseries.TimeSeries* attribute), 9

## S

`set_interval()` (*ticts.timeseries.TimeSeries*  
method), 9

`slice()` (*ticts.timeseries.TimeSeries* method), 9

## T

`ticts.timeseries` (*module*), 9

`TimeSeries` (*class in ticts.timeseries*), 9

## U

`upper_bound` (*ticts.timeseries.TimeSeries* attribute),  
10