
unithon Documentation

Release 0.1

David Garcia-Retuerta

Sep 18, 2019

Contents:

1	Introduction	1
2	Installation	3
2.1	First Installation	3
2.2	Update Package	3
3	API Reference	5
3.1	unithon	5
4	Contribute	9
5	Indices and tables	11
	Python Module Index	13
	Index	15

CHAPTER 1

Introduction

unithon is a python library for unifying heterogeneous datasets, primarily focused on IoT data processing. Several datasets with different structure and refresh times can easily be combined into a single well-formatted dataset. Furthermore, it provides some commonly used functions for data processing.

Note: After installing the package you are now available to use it! As unithon's latest release is 0.1 the installation is optimized for it. If you try installing another unithon release, some features may not work.

2.1 First Installation

In order to get this package working you will need to install it on its last version. To install the package on either way you will need to have a Python 3.7 version installed and pip. So, to install the latest release of `unithon`, you can either do it:

- via Python Package Indexer (PyPI):

```
$ python -m pip install unithon
```

- from GitHub via PyPI:

```
$ python -m pip install https://github.com/dvidgar/unithon/archive/master.zip
```

2.2 Update Package

If you already had `unithon` installed and you want to update it you can do it:

- via PyPI:

```
$ python -m pip install --upgrade unithon
```

- from GitHub via PyPi:

```
$ python -m pip install --upgrade https://github.com/dvidgar/unithon/archive/  
↪master.zip
```

All the dependencies are already listed on the setup file of the package, but to sum them up, when installing `unithon`, it will install the following dependencies:

- `pandas 0.25.1`

3.1 unithon

`unithon.adapt_frequency(df_, new_frequency=60, start_date=None, end_date=None, time_column_name='date_time')`

This function changes the refresh frequency of a dataframe.

Parameters

- **df** (`pandas.DataFrame`) – dataframes of all houses.
- **new_frequency** (`int`, optional) – refresh frequency in minutes of the output.
- **start_date** (`datetime`, optional) – left extreme of the selected time interval.
- **end_date** (`datetime`, optional) – right extreme of the selected time interval.
- **time_column_name** (`str`, optional) – name of the column containing the time information.

Returns The function returns a pandas dataframe with the selected refresh frequency.

Return type `pandas.DataFrame`

`unithon.df_house_sensor(df_, house_number, sensor)`

This function extracts the information of a specific sensor in a certain house from a dataframe.

Parameters

- **df** (`pandas.DataFrame`) – dataframe containing all data.
- **house_number** (`int`) – number of the house which data is getting extracted.
- **sensor** (`int` or `str`) – name/number of the sensor which data is getting extracted.

Returns The function returns a dataframe containing the data of a specific sensor in a certain house.

Return type `pandas.DataFrame`

`unithon.fix_date_format(df_, date_format='%d %m %Y %H:%M', date_column_name='hour')`

This function converts the data column into 'datetime' format.

Parameters

- **df** (`pandas.DataFrame`) – Input dataset.
- **date_format** (`str`, optional) – format in which the dates are embedded.
- **date_column_name** (`str`, optional) – name of the column containing the dates.

Returns The function returns a dataframe with the date column in datetime format.

Return type `pandas.DataFrame`

`unithon.fix_empty_weeks(df_, column)`

This function fulfills the empty gaps found in a dataset.

Parameters

- **df** (`pandas.DataFrame`) – Input dataset.
- **column** (`int`) – column's position (as number in the dataframe).

Returns The function returns the fixed dataframe.

Return type `bool`

`unithon.fix_month_format(element)`

This function converts the abbreviation of a Spanish month into its corresponding month number.

Parameters **element** (`str`) – name of the month in Spanish. Abbreviation of the first 3 letters.

Returns The function returns the corresponding number as string.

Return type `str`

`unithon.get_df_house(df_, house_number, frequency=60, time_column_name='date_time')`

This function extracts the dataframe of a specific house from a general dataframe.

Parameters

- **df** (`pandas.DataFrame`) – dataframes of all houses.
- **house_number** (`str`) – number of the selected house.
- **frequency** (`int`, optional) – refresh frequency in minutes of the output.
- **time_column_name** (`str`, optional) – name of the column containing the time information.

Returns The function returns a pandas dataframe with all the information of the selected house.

Return type `pandas.DataFrame`

`unithon.load_data(original_path='./concatenado', date_format='%d %m %Y %H:%M',
sort_values=True, date_column_name='hour')`

This function loads information from several files and outputs a single dataset containing all the information.

Parameters

- **original_path** (`str`, optional) – path where all the files are located.
- **date_format** (`str`, optional) – format in which the dates are embedded.
- **sort_values** (`bool`, optional) – sort the values by data or preserve the original order.
- **date_column_name** (`str`, optional) – name of the column containing the dates.

Returns The function returns a `pandas.DataFrame` containing all the loaded data.

Return type `pandas.DataFrame`

`unithon.write_df_all_houses(df_, output_path='./', frequency=60)`

This function writes in a csv the dataframes of all houses.

Parameters

- **df** (`pandas.DataFrame`) – dataframes of all houses.
- **output_path** (`str`, optional) – relative output path.
- **frequency** (`int`, optional) – refresh frequency in minutes of the output.

Returns The function returns `True` if the operation is successful.

Return type `bool`

CHAPTER 4

Contribute

As this is an open source project it is open to contributions, bug reports, bug fixes, documentation improvements, enhancements and ideas.

Also there is an open tab of [issues](#) where anyone can contribute opening new issues if needed or navigate through them in order to solve them or contribute to its solving.

CHAPTER 5

Indices and tables

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

u

`unithon`, 5

A

`adapt_frequency()` (*in module unithon*), 5

D

`df_house_sensor()` (*in module unithon*), 5

F

`fix_date_format()` (*in module unithon*), 5

`fix_empty_weeks()` (*in module unithon*), 6

`fix_month_format()` (*in module unithon*), 6

G

`get_df_house()` (*in module unithon*), 6

L

`load_data()` (*in module unithon*), 6

U

`unithon` (*module*), 5

W

`write_df_all_houses()` (*in module unithon*), 7