



pyutils

January 27, 2025

Abstract

The library `pyutils` contains several general utilities in Python that can be used in other SAS packages, ranging from date time utilities to practical FTOOLS transcriptions in Python. This utilities can be used in a stand-alone format by importing the library in any Python environment once a SAS session has been started.

1 Use

In order to use `pyutils` simply import it in a Python session:

```
import pysas.pyutils.pyutils as pyutils  
  
pyutils.(function)...
```

A further explanation of the different functions present in `pyutils` can be found in the section below. For a more detailed description on how to run each function, use the `help(...)` command.

2 Description

`pyutils` contains several functions that are not specific to a single package, script or task, but rather, common utilities that can be used by SAS tasks or by users within a Python interface. Albeit limited in its current state, it should evolve into a more complex common library easy to import.

The preferred use of the module is through `pysas`, as in `'import pysas.pyutils.pyutils as pyutils'`.

2.1 Current functions

At the moment, these are the current functions present in `pyutils`:

- *pyadd*: Sums the image data from the primary HDU of the given FITS files. Does not copy the information passed through the HDU header. The images must have the same size.



- *date_time*: Returns the current time in the following format: Current time: %Y - %m - %d - %H:%M:%S.
- *pyselect*: Returns the rows from the data of a FITS file that suits the conditions imposed by the arguments of the function.
- *pymodhdu*: Modifies the header of the extension of a FITS file using the parameters found in the arguments passed to the function.
- *get_key_word*: Returns the keyword from the header of a FITS file. If the extension is passed, will only search in that particular extension. If not, it'll loop over all the extensions until a match is found. If it's not found, an alternative output (in the form of a non-mandatory parameter) will be returned.
- *pydump*: Returns the raw values (as a list) of the column of the FITS file passed as argument.
- *is_notebook*: Returns true or false depending on whether or not is executed in a Jupyter Notebook environment.
- *delete_data*: Deletes rows or columns from the extension of a FITS file.
- *create_fits_from_data*: Creates a FITS file from the input data using an array-label system. NOTE: the extension will be nameless.
- *filter_data*: Filters 2D NumPy arrays or `astropy.table.Table` objects using the criteria passed to the function.
- *imgstat*: Provides statistics from FITS image files. Can be dumped into a text file.
- *merge_fits*: Merges the specified data columns from several FITS files into a new file.
- *add_column_to_fits*: Adds a data columns to an existing data extension of a FITS file.
- *add_row_to_fits*: Adds a row or rows to the existing FITS file.
- *sort_fits*: Rewrites the given FITS file according to the order of one given column or columns.

2.2 Errors

Will raise the usual Python exceptions through the AstroPy/NumPy frame. Other errors will be notified to the user accordingly in each function.

3 Input Files

1. The standard input is mentioned in each function. The common recommendation is to use the FITS path as the input. In the future this will be changed to also take Python objects.

4 Output Files

1. Each utility has its own output files (or any). This is mentioned in the documentation for each function.



5 Comments

- Please report any bug found or any extra utilities that may seem useful for the purpose of SAS and/or **pyutils**.

References