

```

#!/usr/bin/env python
"""
heapy [heeahpy] - High Energy Astrophysics in PYthon
Usage Example file
Author: Derek Blue
"""

# %% Import required packages
# heapy manages all required internal imports itself
# so apart from required imports to load and clean your
# data no other imports are required
from __future__ import division, print_function
import heapy

# %% Load example data
# This is an example time-series included with the heapy
# package stored as a csv file, cleaned and returned as a
# standard Python dictionary.
ex_data = heapy.ExampleData()

# %% Clean data into required columns
# heapy requires that you load and clean your data prior to
# use. Once you have loaded and cleaned your data heapy only
# requires that you store time, value, and error in standard
# Python lists
time = ex_data['time']
value = ex_data['value']
error = ex_data['error']

# %% Create a heapy TimeSeries Object
# Once you have cleaned your data into the required lists
# create a heapy TimeSeries object to begin analysis
ts = heapy.TimeSeries(time, value, error)

# %% Plot the time-series to see what it looks like
# heapy TimeSeries objects can be plotted with multiple options
# Options:
#     original    [bool]- Plot the original data
#     detrended   [bool]- Plot the detrended data, will only be different if detrended
#     trendline   [bool]- Plot a trendline fit to the original data
#     fmt         [char]- Format option for matplotlib, follows matplotlib convention
#     save        [bool]- Save the plot as a figure
ts.Plot()
ts.Plot(original=True, detrended=False, trendline=False, fmt='.', save=False)
ts.Plot(original=True, detrended=False, trendline=True, fmt='.', save=False)
ts.Plot(original=True, detrended=True, trendline=False, fmt='.', save=False)
ts.Plot(original=True, detrended=True, trendline=True, fmt='.', save=False)
ts.Plot(original=False, detrended=True, trendline=False, fmt='.', save=False)
ts.Plot(original=False, detrended=True, trendline=True, fmt='.', save=False)

# %% Remove any linear trend in the time-series
# Any linear trend in the time-series will artificially steepen the
# resulting structure function so we will use the Detrend function
# to remove any linear trend in the data
ts.Detrend()
# Since we plotted the time-series detrended data above with no visible
# difference we will plot it again here to show the difference
ts.Plot(original=True, detrended=True, trendline=True, fmt='.', save=False)

# %% Generate the structure function
# Once the time-series has been detrended we can create a StructureFunction
# object from the detrended time-series
sf = heapy.StructureFunction(ts)
print('Structure function calculation complete')

# %% Plot the resulting StructureFunction
# heapy StructureFunction objects have a built-in method for quick and dirty
# plotting to get a quick idea of what your object's structure function looks
# like
sf.Plot(cutoff=1e3)

# %% Extract the structure function data table
sf_data = sf.sf
print(type(sf_data))

```