

*AHELP for CIAO 3.4*

set_filter

Context: [sherpa](#)*Jump to:* [Description](#) [Examples](#) [Bugs](#) [See Also](#)

Synopsis

Module functions for assigning source and background filters from an array.

Syntax

```
Integer_Type set_filter([Integer_Type,]Array_Type)
Integer_Type set_bfilter([Integer_Type,]Array_Type)
```

Success/Error Return Values: 1/0

Arguments:

- (1) Dataset number (default 1)
- (2) An array of source/background filter values

Note that if only one argument is provided, it is assumed to be an array, and the dataset is assumed to be dataset 1.

Description

The `set_filter` and `set_bfilter` functions allow the user to set new filters for source and background datasets.

The input array is a `Short_Type`, `Long_Type`, or `Integer_Type` array of the same length as the input (unfiltered) dataset for which a new filter is to be defined. If an input array element is zero, then the corresponding data bin is to be ignored; non-zero, and the corresponding bin is to be noticed/used.

One may use these functions in combination with `get_filter` and `get_bfilter` to define new methods of data filtering that go beyond those currently coded in Sherpa; e.g., filtering based on data amplitude. See example below.

See the related Sherpa commands `NOTICE` and `IGNORE` for more information.

Example 1

Read in a 100-bin dataset, and filter out bins 10 to 60:

```
sherpa> DATA spec.dat
```

```

sherpa> a = [0:99]
sherpa> a[*] = 1
sherpa> a[[9:59]] = 0
sherpa> () = set_filter(1,a)

```

In this example, data are read into Sherpa, and then the filter array `a` defined: step 1 defines a 100-bin-length array, step 2 resets all elements to 1, and step 3 resets a subset of the elements to 0 (bins to be ignored). Note that while the bins are numbered 1 to 100 in the data, the analogous array indices are 0 to 99, so array elements 9 to 59 are reset. The array is then sent to Sherpa, where it sets the new filter.

Example 2

Read in a 100-bin dataset, and filter out all bins with counts greater than 65:

```

sherpa> DATA spec.dat
sherpa> WRITE DATA
Write X-Axis: Bin  Y-Axis: Flux (Counts)
      1          59
      2          46
      3          49
      4          65
      5          60
      6          60
      7          74
...
sherpa> d = get_data()
sherpa> a = where(d>65)
sherpa> printarr(a)
6
9
11
26
...
sherpa> f = get_filter()
sherpa> f[a] = 0
sherpa> set_filter(1,f)
1
sherpa> WRITE DATA
Write X-Axis: Bin  Y-Axis: Flux (Counts)
      1          59
      2          46
      3          49
      4          65
      5          60
      6          60
      8          58
...

```

In this example, data are read into Sherpa, and the counts amplitude is retrieved using `get_data`. A new array `a` is defined whose elements are the bins for which the data are greater than 65; the length of this array is 13 (13 bins have data greater than 65 and are thus to be ignored). The filter array is then retrieved with `get_filter` and the appropriate 13 elements of that array are reset to 0. The new filter array is then read back into Sherpa, and we see, e.g., that bin 7 is now ignored.

Bugs

See the [Sherpa bug pages](#) online for an up-to-date listing of known bugs.

See Also

sherpa

[analysis](#), [get_groups](#), [ignore](#), [notice](#), [set_groups](#), [set_ignore](#), [set_ignore2d](#), [set_ignore_all](#),
[set_ignore_bad](#), [set_notice](#), [set_notice2d](#), [set_notice_all](#)

The Chandra X-Ray Center (CXC) is operated for NASA by the Smithsonian Astrophysical Observatory.
60 Garden Street, Cambridge, MA 02138 USA.
Smithsonian Institution, Copyright © 1998–2006. All rights reserved.

URL:
http://cxc.harvard.edu/ciao3.4/set_filter.html
Last modified: December 2006

