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

AHELP for CIAO 3.4

groupByCounts

Context: sherpa

Jump to: Description Examples NOTES HOW THE GROUPING WORKS See Also

Synopsis

Group a dataset by number of counts or signal−to−noise within Sherpa.

Syntax

```
groupByCounts( [dset,] numCounts )
groupBySNR( [dset,] minSNR )
groupAdaptively( [dset,] minCounts )
groupAdaptiveSNR( [dset,] minSNR )
```

Description

These functions allow a user to group a dataset from within Sherpa, without having to use the dmgroup tool. They make it easy to see how sensitive the fit results are to the details of how the data has been grouped. They are loaded into Sherpa with the call

```
require ("sherpa_utils");
```

Each function takes an optional first argument of dset, which is the number of the dataset to group; if not given it defaults to 1.

groupByCounts([dset,] numCounts)

Group the dataset so that each group contains at least numCounts counts. The grouping is done by the grpNumCounts() routine from the group module.

groupBySNR([dset,] minSNR)

Group the dataset so that each group has a signal−to−noise ratio of at least minSNR. The grouping is done by the grpSnr() routine from the group module.

groupAdaptively([dset,] minCounts)

Adaptively group the dataset so that each group contains at least minCounts counts. The grouping is done by the grpAdaptive() routine from the group module.

groupAdaptiveSNR([dset,] minSNR)

Adaptively group the dataset so that each group has a signal−to−noise ratio of at least minSNR. The grouping is done by the grpAdaptiveSnr() routine from the group module.

Example 1

```
sherpa> data src.pi
sherpa> groupByCounts( 20 )
```

In this example we have loaded in a file (src.pi), and then grouped the data so that there are at least 20 counts per group.

The screen output has been omitted from this, and the following, examples for clarity.

Example 2

```
sherpa> data src.pi
sherpa> subtract
sherpa> ignore all
sherpa> notice energy 0.5:7.0
sherpa> groupByCounts( 20 )
```

Here we background−sutract and filter the data before calling the function. The groupByCounts() routine uses the un−filtered, un−subtracted data, but re−applies the fitlers and background status before it finishes.

Example 3

```
sherpa> data 1 src.pi
sherpa> data 2 src.pi
sherpa> subtract 1:2
sherpa> ignore 1:2 all
sherpa> notice 1:2 energy 0.5:7.0
sherpa> groupByCounts( 1, 20 )
sherpa> groupByCounts( 2, 30 )
sherpa> paramprompt off
sherpa> source 1 = powlaw1d[pl1]
sherpa> source 2 = powlaw1d[pl2]
sherpa> fit 1
sherpa> fit 2
```

Here we load in the same file twice, perforem the same filtering on the data, and then group them so that they have a different number of counts per bin. They are then fit by the same model (a power law). The changes in the parameters of the source model show how sensitive the fit results are to the particular grouping scheme used.

NOTES

Please see "ahelp sherpa_utils" for information on how to load these routines into Sherpa.

HOW THE GROUPING WORKS

The routines are wrappers around routines from the group module ("ahelp modules group") and take advantage of the set_group() and set_quality() routines added to Sherpa in CIAO 3.1.

Each routine works in the same way. The dataset is checked to see if it has been background subtracted and what filters have been applied to it. If either have been set then they are removed and then any previous grouping is also removed (this is necessary so that the new grouping information can be applied to the dataset). The new grouping and quality arrays are found by calls to the relevant function from the group module, and then applied to the dataset. Finally the dataset is background−subtracted, if it was originally, and any filters re−applied.

During the routine, Sherpa may report:

```
WARNING: any applied filters are being deleted!
```

These warnings can be ignored unless there is an associated line saying:

```
Note: ignoring filter expression ...
```

If this latter message is seen then a previously applied filter has been ignored. This message will occur if a filter has been specified in units of bins or channels, since these values are not invariant when the dataset is re−grouped.

See Also

chandra

guide

sherpa

bye, calc_kcorr, dataspace, dcounts, dollarsign, echo, eflux, eqwidth, erase, flux, get, get_dcounts_sum, get_dir, get_eflux, get_eqwidth, get_filename, get_flux2d, get_flux_str, get_lfactorial, get_mcounts_sum, get_pflux, get_source_components, get_verbose, groupbycounts, guess, is, journal, list, list_par, mcounts, numbersign, paramest, plot_eprof, plot_rprof, prompt, reset, run, set, set_analysis, set_axes, set_coord, set_dataspace, set_dir, set_verbose, setplot, sherpa−module, sherpa_plotfns, sherpa_utils, show, simspec, use, version

---

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