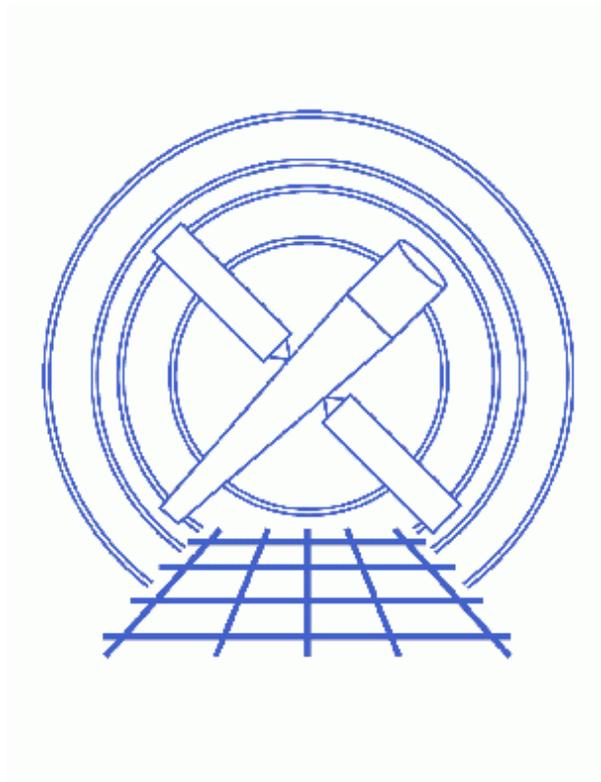


# Changing the grouping scheme of a dataset within Sherpa



**Sherpa Threads (CIAO 3.4)**

# Table of Contents

- **Getting Started**
  - ◆ Downloading the sherpa\_utils.sl script
  - ◆ Loading the sherpa\_utils.sl script
  - ◆ Loading the data
- **Regrouping the data**
- **Fit the data**
- **Notes**
- **Summary**
- **History**
- **Images**
  - ◆ Data grouped by 15 and 30 counts per group
  - ◆ Comparison of grouping by 15 and 30 counts per group
  - ◆ Fit to both datasets

---

# Changing the grouping scheme of a dataset within Sherpa

## *Sherpa Threads*

### Overview

*Last Update:* 1 Dec 2006 – reviewed for CIAO 3.4: no changes

#### *Synopsis:*

In order to use Gaussian statistics to fit a model to a dataset it is often necessary to "group" the data – i.e. combine channels until you have enough counts – before use. It is possible to set and change the grouping of a file *after* it has been read into *Sherpa* by using the `set_groups()` and `set_quality()` functions from *Sherpa* together with the routines from the `group` module.

Several routines – `groupByCounts()`, `groupBySNR()`, `groupAdaptively()`, and `groupAdaptiveSNR()` – have been added to the `sherpa_utils.sl` script (which is part of the CIAO Scripts distribution) to make re-grouping easier. This thread shows how you can use these functions when fitting PHA data.

*Proceed to the [HTML](#) or hardcopy (PDF: [A4](#) / [letter](#)) version of the thread.*

---

## Getting Started

### Downloading the `sherpa_utils.sl` script

This thread uses the `sherpa_utils.sl` script; for information about the script, consult the help file ("`ahelp sherpa_utils`"). The most recent version of `sherpa_utils.sl` is v1.26 (02 Nov 2004):

```
unix% grep Id $ASCDS_CONTRIB/share/slsh/local-packages/sherpa_utils.sl
% $Id: sherpa_utils.sl,v 1.26 2004/11/02 15:53:37 dburke Exp $
```

Note that `$ASCDS_CONTRIB/share/slsh/local-packages/` is the default path in the standard CIAO scripts installation; see the Scripts page for more information. ***Please check that you are using the most recent version before continuing.*** If you do not have the script installed or need to update to a newer version, please refer to the Scripts page.

---

## Loading the sherpa\_utils.sl script

The sherpa\_utils.sl script is loaded into a *Sherpa* session with the `evalfile` function:

```
sherpa> () = evalfile("sherpa_utils.sl");
```

The version may also be found by viewing the `_sherpa_utils_version` or `_sherpa_utils_version_string` variables once the file is loaded:

```
sherpa> _sherpa_utils_version
126
sherpa> _sherpa_utils_version_string
1.26
```

If you wish the functions to always be available to *Sherpa* add the following line to your `~/ .sherparc` file:

```
() = evalfile("sherpa_utils.sl");
```

For more information on configuring *Sherpa*, see the [Customizing Sherpa with a Resource File](#) thread.

---

## Loading the data

This thread uses the same dataset as used in the [Introduction to Fitting PHA Spectra](#) thread. We load the dataset twice so that we can easily see the effect of changing the grouping scheme (the screen output has been omitted for clarity):

```
sherpa> DATA 1 3c273.pi
...
sherpa> DATA 2 3c273.pi
...
```

We note that the data in `3c273.pi` was created so that each group contained at least 15 counts – by using the `NUM_CTS grouptype option` of `dmgroup` – as can be seen by using the `dmhistory` tool.

```
sherpa> !dmhistory 3c273.pi dmgroup | tr ' ' "\012"
dmgroup
infile="3c273.pi"
outfile="./3c273.tmp"
grouptype="NUM_CTS"
grouptypeval="15"
binspec=""
xcolumn="channel"
ycolumn="counts"
tabspec=""
tabcolumn=""
stopspec=""
stopcolumn=""
errcolumn=""
clobber="no"
verbose="0"
maxlength="0"
```

The command was preceded by `!` to tell *Sherpa* to execute it as a shell command, and `| tr ' ' "\012"` was added to add a new-line character between parameter values (to make the screen output easier to read).

---

## Regrouping the data

We use the `groupByCounts` function to group the second dataset by 30 counts per group, which is double that of the first dataset. The "WARNING" messages can be ignored as there no filters have been applied to the data. The two datasets are then plotted using logarithmic scaling on both axes:

```
sherpa> groupByCounts( 2, 30 )
WARNING: any applied filters are being deleted!
WARNING: any applied filters are being deleted!
sherpa> set log
sherpa> lplot 2 data 1 data 2
```

The resulting plot  shows how the data looks before and after re-grouping. We can also plot them on the same graph by setting the `sherpa.multiplot.newarea` field to 0:

```
sherpa> sherpa.multiplot.newarea = 0
sherpa> lp 2 data 1 data 2
sherpa> symbol red
sherpa> symbol circle
sherpa> redraw
sherpa> sherpa.multiplot.newarea = 1
```

which creates this plot .

## Fit the data

Once the data has been re-grouped you can use it just like any other dataset. Here we repeat the fit made in the [Introduction to fitting PHA spectra](#) to see what difference the different grouping scheme makes.

```
sherpa> NOTICE 2 ENERGY 0.1:6.0
sherpa> SUBTRACT 2
sherpa> PARAMPROMPT OFF
Model parameter prompting is off
sherpa> SOURCE 2 = xsphabs[abs] * powlaw1d[p2]
sherpa> abs.nh = 0.07
sherpa> FREEZE abs
sherpa> FIT 2
LVMQT: V2.0
LVMQT: initial statistic value = 389.097
LVMQT: final statistic value = 28.0675 at iteration 10
      p2.gamma  2.16148
      p2.ampl  0.000232544

sherpa> GOODNESS 2
Goodness: computed with Chi-Squared Gehrels

DataSet 2: 22 data points -- 20 degrees of freedom.
Statistic value      = 28.0675
Probability [Q-value] = 0.107811
Reduced statistic    = 1.40338

sherpa> COVAR 2

Computed for sherpa.cov.sigma = 1
-----
```

## Re-grouping PHA data in Sherpa – Sherpa

Parameter Name	Best-Fit	Lower Bound	Upper Bound
p2.gamma	2.16148	-0.0763935	+0.0763935
p2.ampl	0.000232544	-1.4084e-05	+1.4084e-05

which can be compared to the original results:

```
sherpa> NOTICE 1 ENERGY 0.1:6.0
sherpa> SUBTRACT 1
sherpa> SOURCE 1 = abs * powlaw1d[p1]
sherpa> FIT 1
LVMQT: V2.0
LVMQT: initial statistic value = 355.854
LVMQT: final statistic value = 37.9079 at iteration 10
      p1.gamma  2.1585
      p1.ampl  0.000224838

sherpa> COVAR 1

Computed for sherpa.cov.sigma = 1
-----
Parameter Name      Best-Fit Lower Bound      Upper Bound
-----
      p1.gamma      2.1585  -0.0827852  +0.0827852
      p1.ampl      0.000224838  -1.48257e-05  +1.48257e-05
```

The following shows both fits (the extra commands are used to make the two plots have the same X axis):

```
sherpa> LP 2 fit 1 fit 2
sherpa> d 1 location 0.15 0.9 0.475 0.9
sherpa> d 2 location 0.15 0.9 0.1 0.475
sherpa> d 1 tickvals off
sherpa> d 1:2 limits x 0.1 7.0
sherpa> d 1 label 0.5 0.0006 "15 counts per group"
sherpa> label size 1.2
sherpa> d 2 label 0.5 0.0006 "30 counts per group"
sherpa> label size 1.2
sherpa> redraw
```

which creates [this plot](#) .

---

## Notes

The [groupByCounts\(\)](#) and related functions can also be used on data that was not grouped before being read into *Sherpa*. This can be useful for two reasons:

- You can use the routines to find the best grouping scheme for your data without having to re-run the *dmgroup* tool and re-load the data into *Sherpa*.
- You can fit the un-grouped data with the [Cash statistic](#) and then use the functions to make it easier to compare the fit to the data in plots.

## Summary

The thread shows how you can use the `groupByCounts()` function from the `sherpa_utils.sl` file to change the grouping scheme of a PHA file once it has been read into *Sherpa*. This allows you to see how sensitive the fit results are to the grouping scheme by changing the number of counts per group or using a different method for grouping the data.

---

## History

- 14 Dec 2004 updated for CIAO 3.2: script version and path
- 17 Jun 2005 updated information in [Get Started](#) on loading the script
- 21 Dec 2005 reviewed for CIAO 3.3: no changes
- 01 Dec 2006 reviewed for CIAO 3.4: no changes

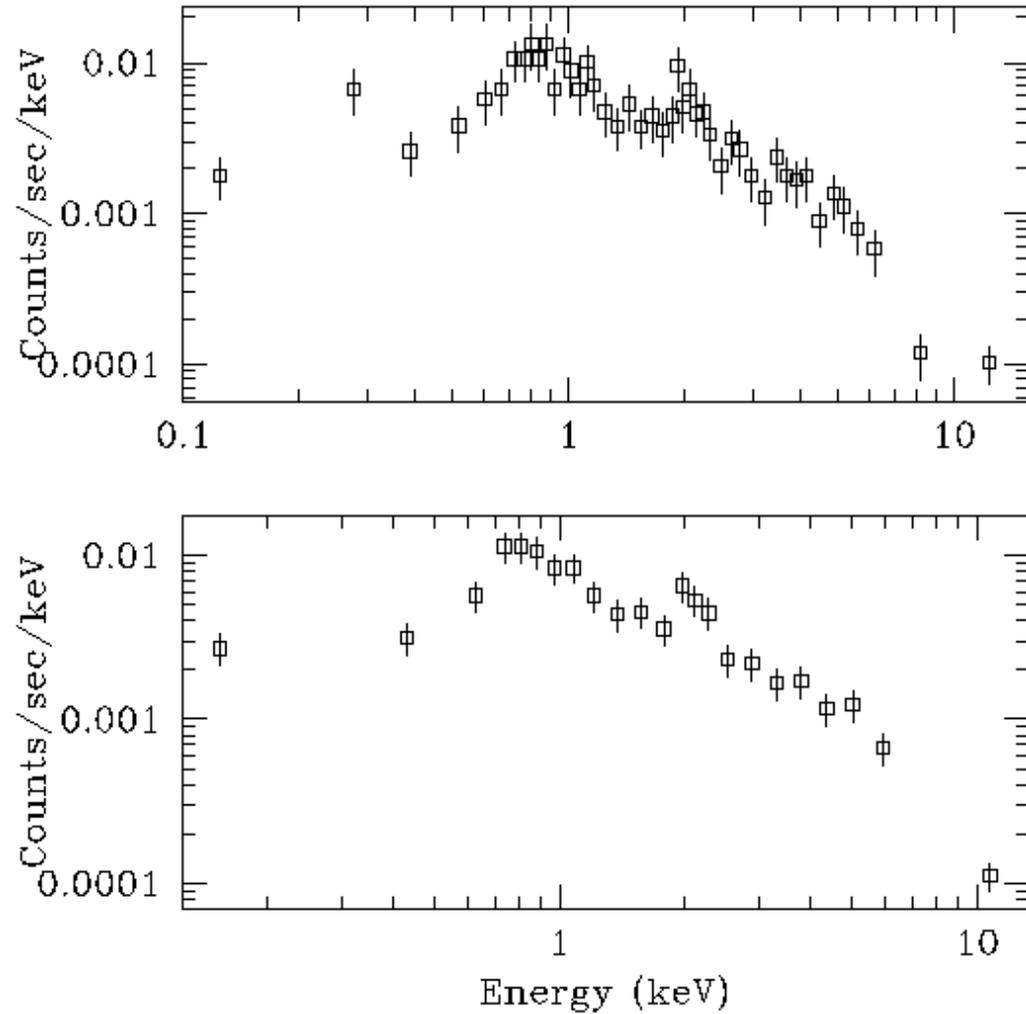
---

URL: [http://cxc.harvard.edu/sherpa/threads/pha\\_regroup/](http://cxc.harvard.edu/sherpa/threads/pha_regroup/)

Last modified: 1 Dec 2006

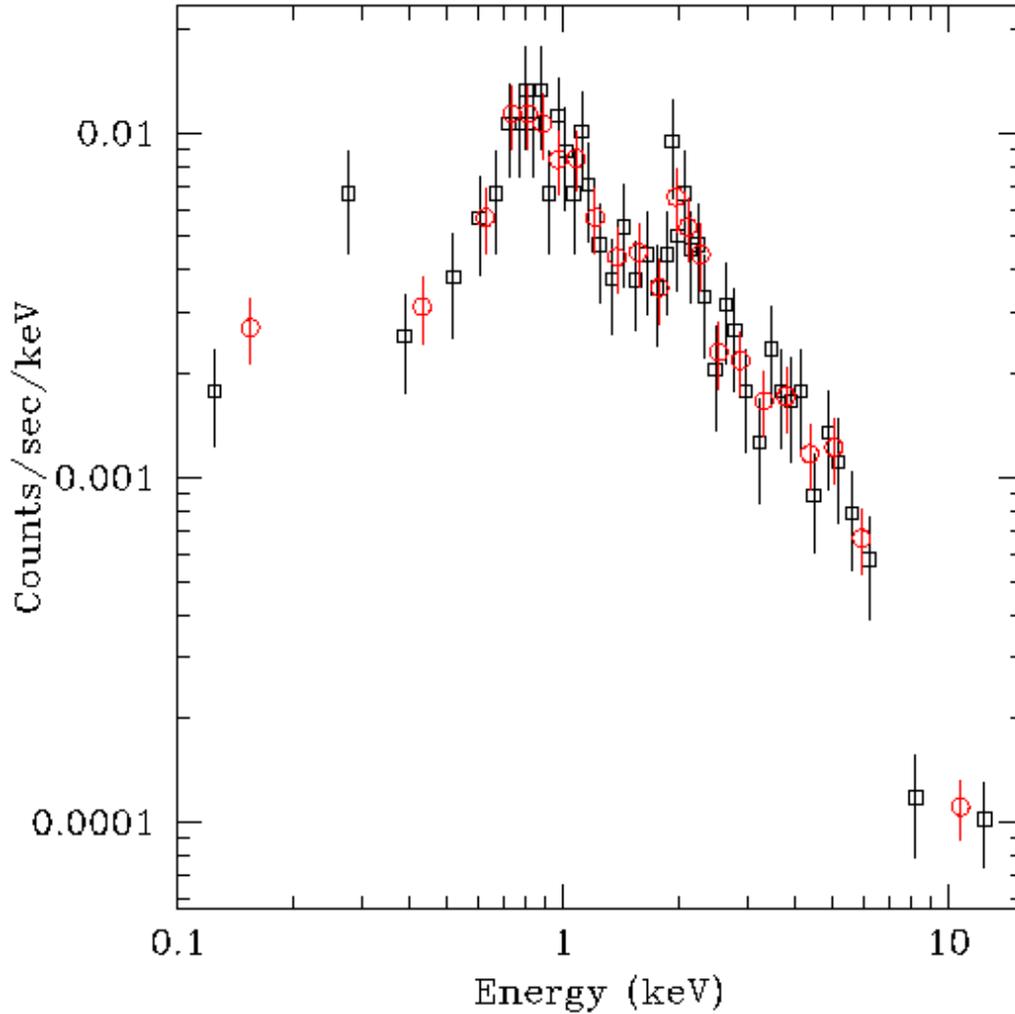
### Image 1: Data grouped by 15 and 30 counts per group

The top plot shows the data as read in to *Sherpa* – which is binned by 15 counts per group – and the bottom plot shows the dataset after `groupByCounts()` has been called to re-group the data to 30 counts per group.



## Image 2: Comparison of grouping by 15 and 30 counts per group

This plot shows the same data as previously  except that both datasets are on the same graph. The black (square) points show the data binned by 15 counts per group and the red (circle) points show the the data binned by 30 counts per group.



### Image 3: Fit to both datasets

This plot shows the fits to the data when grouped by 15 counts per group (top plot) and 30 counts per group (bottom plot).

