



I/O Bugs: general

Caveats

1. *If the boundaries of a bin (e.g. 20 to 25) are partially included in a filter interval (e.g. 24 to 26), the entire bin is retained in the output.*

For instance, using this sample data:

```
[17.500000] = 0.186
[22.500000] = 0.1487
[27.500000] = 0.1209
```

Note that *Sherpa* uses the *center* of the bin as its nominal position. When *Sherpa* interprets the filter `ignore filter 24:28`, there is only one bin in the above list that falls in that interval: bin 5, with a position of 27.5. In the case of the filter `ignore filter 24:26`, both bin 4 (position of 22.5) and bin 5 (position of 27.5) are selected.

In both cases there are counts from bin 4 and bin 5 that fall within the interval indicated by the filter. Since the filter only overlaps part of the bin, *Sherpa* keeps the entire bin. This allows the user to look at the results and modify the filter to include (or exclude) more points if desired.

Bugs

1. *When working in WCS coordinates, a region file may be applied incorrectly to the data in *Sherpa*.*

Regions can be saved from ds9 to file, and later read into *Sherpa* so that regions can be noticed or ignored (e.g., with `"notice filter \"region(foo.reg)\""`, where `foo.reg` was saved from an earlier ds9 session). When working in WCS coordinates, a region file may be applied incorrectly; it is better to work in image coordinates if possible.

2. *Attempts to input data from directories other than the present working directory may fail.*

Workaround:

delimit the input filename with quotes:

```
sherpa> data ../img.fits
Syntax Error: data ../img.fits;
sherpa> data "../img.fits"
sherpa>
```

3. *Filter commands with extra extraneous characters are applied successfully, although it causes a parse error.*

```
sherpa> notice filter 10:30 foofoofoo
Parse Error/Undefined Name: notice filter 10:30 foofoofoo;
sherpa> show
...
Current filters for dataset 1:
```

```
notice source 1 filter 10 : 30 foofoofoo
Noticed filter size: 21 bins
...
```

The filter is applied, but cannot be successfully read back in if saved.

4. *If a 2–D filter is assigned to Sherpa via the Sherpa/S–Lang module command `set_filter()`, and subsequently the command `SAVE ALL` is issued, then no information about the filter is saved.*

This is because the shape or source cannot be rendered in a line of text. One possible solution for this in future versions of Sherpa is to write out the filter to disk, and `SAVE` a `READ FILTER` command expression.

5. *Actions (e.g. defining filters) using regions with quantities in arcseconds will fail.*

For example:

```
sherpa> notice world "ellipse(325,-23,9.9",8.2",94)"
```

will not work because of the extra quote marks.

Workaround:

convert the quantities to arcminutes.

6. *Applying the same (2–D) region filter to source and background data yields an "Illegal instruction".*

```
sherpa> data "img_bin8_7_70.fits[440:600,392:557]" fits
sherpa> read back "smbg_bin8_7_70.fits[440:600,392:557]" fits
sherpa> notice filter "circle(80,90,26)"
sherpa> quit
Goodbye.
Illegal instruction
```

Workaround:

specify the filter twice:

```
sherpa> data "img_bin8_7_70.fits[440:600,392:557]" fits
sherpa> read back "smbg_bin8_7_70.fits[440:600,392:557]" fits
sherpa> notice source filter "circle(80,90,26)"
sherpa> notice back filter "circle(80,90,26)"
sherpa> quit
Goodbye.
```