
Subject: Re: PSYM=10 problem
Posted by [Liam Gumley](#) on Tue, 16 Nov 1999 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Laurent Chardon wrote:

> Is there a good reason why the first bin plotted by the plot/PSYM=10
> combination is half the size of all the others? Can I get around this
> behaviour? I want all the bins to be of equal size.

A few months ago in this newsgroup, David Fanning convinced me that PSYM=10 would never give an accurate representation of a histogram, and that the only way to do it right is to plot the histogram yourself. As you've noted, the problem is getting the edges of the bins in the right position. I came up with the following procedure which I believe computes and plots a 'correct' histogram (let me know if I'm wrong!):

;---cut here---

```
PRO HIST_PLOT, DATA, MIN=MIN_VALUE, MAX=MAX_VALUE, $  
  BINSIZE=BINSIZE, NORMALIZE=NORMALIZE, FILL=FILL, $  
  _EXTRA=EXTRA_KEYWORDS
```

```
; DATA      Array of data  
; MIN        Minimum data value for computing histogram  
;            (default is MIN(DATA))  
; MAX        Maximum data value for computing histogram  
;            (default is MAX(DATA))  
; BINSIZE     Binsize (default is to create 100 bins)  
; NORMALIZE   If set, normalize the histogram  
; FILL        If set, fill the histogram  
; Also accepts any keywords recognized by PLOT or POLYFILL
```

```
;- Check arguments
```

```
if n_params() ne 1 then message, 'Usage: HIST_PLOT, DATA'
```

```
if n_elements(data) eq 0 then message, 'DATA is undefined'
```

```
;- Check keywords
```

```
if n_elements(min_value) eq 0 then min_value = min(data)
```

```
if n_elements(max_value) eq 0 then max_value = max(data)
```

```
if n_elements(binsize) eq 0 then $
```

```
  binsize = (max_value - min_value) * 0.01
```

```
binsize = binsize > ((max_value - min_value) * 1.0e-5)
```

```
;- Compute histogram
```

```
hist = histogram(float(data), binsize=binsize, $
```

```
  min=min_value, max=max_value)
```

```
hist = [hist, 0L]
```

```
nhist = n_elements(hist)
```

```

;- Normalize histogram if required
if keyword_set(normalize) then $
    hist = hist / float(max(hist))

;- Compute bin values
bins = lindgen(nhist) * binsize + min_value

;- Create plot arrays
x = fltarr(2 * nhist)
x[2 * lindgen(nhist)] = bins
x[2 * lindgen(nhist) + 1] = bins
y = fltarr(2 * nhist)
y[2 * lindgen(nhist)] = hist
y[2 * lindgen(nhist) + 1] = hist
y = shift(y, 1)

;- Plot the histogram
plot, x, y, _extra=extra_keywords

;- Fill the histogram if required
if keyword_set(fill) then $
    polyfill, [x, x[0]], [y, y[0]], _extra=extra_keywords

END
;---cut here---

```

As verification, I submit the following example:

```

data = dist(256)
hist_plot, data
hist_plot, data, binsize=1.0, xrange=[-1, 10], yrange=[0, 20]
result = where(data ge 0.0 and data lt 1.0, count)
print, count
    1
result = where(data ge 1.0 and data lt 2.0, count)
print, count
    8
result = where(data ge 2.0 and data lt 3.0, count)
print, count
    16

```

Cheers,
Liam.

--

Liam E. Gumley
Space Science and Engineering Center, UW-Madison
<http://cimss.ssec.wisc.edu/~gumley>
