
Subject: Histogram Bug

Posted by [John-David T. Smith](#) on Mon, 26 Nov 2001 16:43:38 GMT

[View Forum Message](#) <> [Reply to Message](#)

Here's another ugly IDL-crashing bug:

```
IDL> print,histogram([1,2],NBINS=1)
```

It happens in both 5.4 and 5.5. The problem is with 1 bin. It vanishes if BINSIZE is specified, except, of course, as:

```
IDL> print,histogram([1,2],NBINS=1,BINSIZE=0)
```

Not that this last one is meaningful, but still it should err gracefully. In fact any small enough binsize will cause the crash.

A workaround is to use a large enough nonzero BINSIZE for any histogram call for which NBINS could ever end up being unity. In reality, histogram(a,NBINS=1) should equal n_elements(a), and if you don't set your BINSIZE appropriately, you could miss values.

This all relates, I suspect, to the somewhat strange relation assumed by histogram() for BINSIZE, given NBINS:

$$\text{BINSIZE} = (\text{MAX} - \text{MIN}) / (\text{NBINS} - 1)$$

with the side effect of *changing* any input MAX, instead of the more sensible (in my opinion), and max-preserving:

$$\text{BINSIZE} = (\text{MAX} - \text{MIN}) / \text{NBINS}$$

This strange behavior we discussed long ago, with the conclusion that it's probably too late to change. Here's a snippet from my original complaint:

```
> I'm not sure why I didn't notice this before. The recently introduced
> NBINS keyword to histogram is incorrectly defined.
>
>   BINSIZE
>   Set this keyword to the size of the bin to
>   use. If this keyword is not specified, and
>   NBINS is not set, then a bin size of 1
>   is used. If NBINS is set, the default is
>   BINSIZE = (MAX - MIN) / (NBINS - 1).
>
> For example, consider 3 bins over the range [0,1]. Should be a binsize
> of 1/3, yes? Instead, you get a binsize of 1/2, and bins from:
>
```

```
> [0,.5]
> [.5,1.]
> [1.,1.5]
```

JD

P.S. Interestingly, the obvious failure case NBINS=0 is explicitly detected.
