
Subject: Re: histogram function question

Posted by [Helder Marchetto](#) on Fri, 06 Mar 2015 15:15:57 GMT

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

On Friday, March 6, 2015 at 3:31:31 PM UTC+1, Matthew Argall wrote:

>> Note: The data type of the value specified for MIN should match the data type of the input array. Since MIN is converted to the data type of the input array, specifying mismatched data types may produce undesired results.

>>

>> In the first case, 0.3 and 1.3 are truncated to 0 and 1, respectively.

>

> Maybe I should learn how to read, too :-p The MIN, MAX, and BINSIZE values are converted to the input type, not the other way around.

>

> However, the behavior is as if the input array is converted to integer type...

cgHistogram works because it calculates the histogram() **always** using binsize. When nbins is present instead of binsize, cgHistogram calculates binsize (after taking care of the types).

As far as the example goes, the result is the same also when the type of min, max and binsize are the same as the input data.

I find the result a bit strange, because the bin locations are not what I would expect them to be! I would have expected the bin locations to be:

nbins=3 -> [0-0.3, 0.3-0.6 and 0.6-1.0]

Because histogram should distribute 3 bins evenly between 0.0 and 1.0.

binsize=0.5 -> [0.0-0.5 and 0.5-1.0] (for only 2 bins... not 3)

Because the bins sizes should be starting at 0.0 and have a size of 0.5.

So I would have expected the following two results:

2, 0, 0

and

2, 0

Maybe this is just a problem with my understanding of the histogram function. I'll read about it over weekend and maybe I will have an idea of what's happening...

I guess, that generally speaking, the histogram function is wyginwye (well known acronym for "what you get is not what you expect").

```
IDL> data = [0, 0.3, 1.3d]
```

```
IDL> print, HISTOGRAM(data, nbin=3, min=0d, max=1d, loc=loc1), HISTOGRAM(data,  
binsize=0.5d, min=0d, max=1d, loc=loc2)
```

```
IDL> print, loc1, loc2
```

2	0	1
2	0	0
0.00000000	0.50000000	1.00000000
0.00000000	0.50000000	1.00000000

I'll stick to cgHistogram. At least it's consistent!

h
