Subject: Re: Histogram
Posted by David Grier on Mon, 11 Oct 2010 14:16:08 GMT
View Forum Message <> Reply to Message

On 10/11/10 9:39 AM, silje wrote:
> Hey!
> I'm trying to get a grip of the Histogram function in IDL, but have
> run into a problem that I don't understand. As you can see from the
> code below I have created an array with six differend elements,but
> when I call HISTOGRAM with a binsize equal to 0.1 then it seems like
> 1.2 and 1.3 is put in the same bin. Why is this?
>
>
```
> IDL>  arr = [1.1,1.2,1.3,1.4,1.5,1.6]
> IDL>  print, HISTOGRAM(FLOAT(arr), binsize=0.1)
>            1          2          1          1          1
```
>
>
> Thanks!
> Best regards Silje

This question hinges on (1) where HISTOGRAM places the origin of
its bins and (2) the representation of floating point numbers.
If you explicitly specify where you want the bins to start, then
HISTOGRAM does what you were expecting:

```
IDL> print, HISTOGRAM(arr, binsize=0.1, min=1.05)
 1 1 1 1 1 1
```

The command you tried automatically set the origin of the first bin
to 1.1. Each data point then falls right on the dividing line between
two bins. Which way they should fall is clear in decimal notation.
The answer can be different when the same numbers are represented at
finite numerical precision. Your example exercises this distinction.
Here's why:

```
IDL> print, 1.1 + 2.*0.1 - 1.3
1.19209e-7
```

which is not zero. Therefore, your third data point falls into the
second bin because of round-off error.

Doing the same calculation in double precision "solves" the problem
```
IDL> print, 1.1d + 2.d*0.1d - 1.3d
0.0000000
```

TTFN,

David