
Subject: Re: Histogram and bin sizes

Posted by [Conor](#) on Thu, 21 Feb 2008 17:20:17 GMT

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

On Feb 21, 12:19 pm, Conor <cmanc...@gmail.com> wrote:

> On Feb 21, 11:29 am, jeffnettles4...@gmail.com wrote:

>

>

>

>> On Feb 21, 9:05 am, Conor <cmanc...@gmail.com> wrote:

>

>>> On Feb 20, 2:43 pm, pgri...@gmail.com wrote:

>

>>>> jeffnettles4...@gmail.com wrote:

>>>> > I've always wondered why you have to use a constant bin size with

>>>> > HISTOGRAM().

>>>> > To quote J.D.'s famous tutorial: "a histogram

>>>> > represents nothing more than a fancy way to count." Doesn't an

>>>> > imposed constant bin size imply that this is the only way it's ok to

>>>> > count? I can think of several reasons i wouldn't want to do this - I

>>>> > used logarithmic bin sizes in my dissertation, for example (now i'm

>>>> > hoping someone isn't going to answer this post saying i screwed up in

>>>> > my dissertation :-)).

>

>>>> I use logarithmic bins myself quite often, and the fact that a

>>>> logarithmic bin

>>>> size is the same as a constant bin size in log space, makes it is easy

>>>> to use histogram to get that. Less regulars binning don't work with

>>>> histogram, but nobody stops you from writing your own version to work

>>>> with them (it will not be as fast as histogram though).

>

>>>> Ciao,

>>>> Paolo

>

>>>> > And besides, Excel lets you use arbitrary bin

>>>> > sizes....and if Excel lets you do it, it has to be ok, right???? ;-)

>

>>>> > Jeff

>

>>> You can always do whatever binning you want, you just have to

>>> transform your data to the new space and then bin it constantly. Why

>>> doesn't histogram let you use arbitrary binsizes? Not being an IDL

>>> developer I don't know for sure, but I would guess it's a speed

>>> issue. The simpler a program is the faster it is. I use histogram

>>> all the time because it's one of the speedier programs in IDL. It

>>> would make me very sad if in order to make histogram more flexible, it

>>> also became much slower, especially since by transforming my data set

>>> I can use arbitrary bin sizes for histogram.

```
>
>> That actually sounds like what i've done in the past. For my
>> dissertation i needed two kinds of histograms: logarithmic bins
>> (which was fine, no trouble there) and bins that had arbitrary sizes.
>> For the latter, i would either do the histograms in Excel (yuck) or
>> compute two or three histograms in IDL using histogram() with
>> different bin sizes and sort of do some "mixing and matching" of the
>> resulting arrays to get what i wanted. Of course, David hadn't
>> written his awesome histoplot routine yet then either :( Anyway, i'm
>> up against the arbitrary bin sizes problem again for a project i'm
>> doing for someone, and it got me wondering whether this situation is
>> just so rare it wasn't worth supporting in histogram(). I wouldn't
>> want to lose histogram's speed either though.
```

```
>
>> Jeff
>
> Arbitrary bin sizes should be pretty easy to program. You just need
> to map your data points appropriately. For instance if you had the
> data set:
>
> x = randomu(seed,100)
>
> and you wanted bins from:
> [0-.1,.1-.3,.3-.35,.35-.8,.8-1]
>
> you might do something like this:
>
> x = randomu(seed,100)
> bins = [ [0,.1], [.1,.3], [.3,.35], [.35,.8], [.8,1] ]
> newx = fltarr(n_elements(x))
> for i=0,n_elements(bins[0,*])-1 do begin
>   w = where( x ge bins[0,i] and x lt bins[1,i], c )
>   if c gt 0 then newx[w] = i+.5
> endfor
>
> hist = histogram(newx,binsize=1.0,min=0)
> plothist,newx
```

Obviously you'll have to manually set the x-axis labels... This might need a little tweaking but it should give the general idea.
