
Subject: Re: vector of bin indices using histogram?

Posted by [btt](#) on Wed, 18 Oct 2006 13:19:13 GMT

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

greg michael wrote:

```
>
> Can anyone suggest a good way to get a vector of bin indices using
> histogram?
>
> IDL> x=randomu(0,10)
> IDL> print,x
>    0.415999  0.0919649  0.756410  0.529700  0.930436
> 0.383502  0.653919  0.0668422  0.722660  0.671149
>
> I make a histogram anyway:
>
> IDL> h=histogram(x,binsize=.1)
>
> And I also want to know which bin each element went into:
>
> i.e. b=[4,0,7,5,9,3,6,0,7,6]
>
> I could calculate that from the original data of course, but I'm sure
> there must be a trick to get it out of the reverse_indices more
> efficiently (when n_elements is huge).
>
```

Hi,

I think this does the trick by using the LOCATION output keyword to HISTOGRAM and then VALUE_LOCATE. I had to specify the MINIMUM bin value to get the indices lined up to match yours.

```
x = [ 0.415999, 0.0919649, 0.756410, 0.529700, 0.930436,$
      0.383502, 0.653919, 0.0668422, 0.722660, 0.671149]
```

```
h = HISTOGRAM(x, BINSIZE = 0.1, LOC = loc, MIN = 0.0)
```

```
print, VALUE_LOCATE(loc, x)
```

Cheers,
Ben

```
> ---
>
> While experimenting, I came across this, which is not nice...
```

```
>
> IDL> x=randomu(0,100)*1000.
> IDL> print,histogram(x,nbins=4)
>      31      34      34      1
>
> The max value sometimes ends up in a bin of its own (usually this last
> bin is zero - I suppose it's a rounding problem).
>
> ---
>
> And then a question about reverse_indices - (I think it's not touched
> in JD's tutorial):
>
> why are the two parts shoved into a single array? Is there an
> application where this arrangement gives some benefit? Wouldn't the
> first half make more sense indexing a second separate vector without
> the need for this offset?
>
>
> regards,
> Greg
>
```
