
Subject: Re: Histogram question

Posted by [David Fanning](#) on Sun, 08 Aug 2004 13:40:15 GMT

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

Michael Wallace writes:

```
> I have some data I need to histogram. I have two vectors, v1 and v2
> which define a two-dimensional density. Normally I could just use
> HIST_2D and and I'd be set. However, this time around I have a third
> array, v3. v3[i] corresponds to a distinct number of counts at the
> position [v1[i], v2[i]]. So, when I do the histogram, I want to use the
> value found in v3 rather than just simply calculating a density based
> only on occurrences of v1 and v2 pairs.
>
> For example, let's say that I have...
> v1 = [0, 1, 0, 2, 0, 2, 2, 1, 0]
> v2 = [1, 1, 2, 2, 0, 1, 2, 0, 0]
> v3 = [3, 0, 2, 0, 1, 1, 4, 2, 1]
>
> Doing a HIST_2D against v1 and v2 should yield something like...
> [[2, 1, 0],
>  [1, 1, 1],
>  [1, 0, 2]]
>
> But what I really want would use the counts in v3 instead of
> incrementing by 1 for each occurrence of [v1[i], v2[i]]...
>
> [[2, 2, 0],
>  [3, 0, 1],
>  [2, 0, 4]]
>
> Anyone know of an efficient way to do this? I figure there's some trick
> you can do with histogram to achieve this effect, but I am no where near
> the histogram guru like others on this list.
```

I don't know if this is the most "efficient" way,
but the idea is that you have to "replicate" the
numbers in v1 and v2 by the number of counts in v3.

For a quick and dirty method, I used this:

```
,*****
,
v1 = [0, 1, 0, 2, 0, 2, 2, 1, 0]
v2 = [1, 1, 2, 2, 0, 1, 2, 0, 0]
v3 = [3, 0, 2, 0, 1, 1, 4, 2, 1]
```

```
; Replicate each index by the number of counts.
```

```

v1_expand = Ptr_New(/Allocate_Heap)
v2_expand = Ptr_New(/Allocate_Heap)

*v1_expand = [Replicate(v1[0], v3[0])]
*v2_expand = [Replicate(v2[0], v3[0])]
FOR j=1,N_Elements(v3)-1 DO BEGIN
  IF v3[j] NE 0 THEN *v1_expand = [*v1_expand, Replicate(v1[j], v3[j])]
  IF v3[j] NE 0 THEN *v2_expand = [*v2_expand, Replicate(v2[j], v3[j])]
ENDFOR

final = Hist_2D(*v1_expand, *v2_expand)
Ptr_Free, v1_expand, v2_expand
Print, final
,*****

```

Which gave me the answer:

2	2	0
3	0	1
2	0	4

Cheers,

David

--

David Fanning, Ph.D.
 Fanning Software Consulting, Inc.
 Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
