Subject: Re: Histogram question
Posted by Chris Lee on Mon, 09 Aug 2004 14:39:17 GMT
View Forum Message <> Reply to Message

> .. 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
> .. 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]
>
> ans=[[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. -Mike

There are some things that HISTOGRAM can't do (no, really). TOTAL and WHERE can help you a bit. I can get the loop down to the total number of bins in the histogram (in this case there really isn't any speed inprovement, hopefully you have more elements than bins in the real data?)

If your data has lots of zeros in v3 then this method gets more efficient than the 'count everything' approach because the zeros are silently ignored. (If your resulting histogram is sparse another, more tedious optimisation, can be made by doing a HIST_2D(v1,v2) before hand)

```
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]

n=n_elements(v1)
;build an index array
index=fltarr(n)
index=v2*(max(v1)+1)+v1
;m=the number of bins in the 2d histogram
m=max(index)+1
;tot=the 2d histogram
tot=fltarr(m)

for i=0, m-1 do begin & $
  w=where(index eq i,c)  & $
  if(c gt 0) then tot[i]=tot[i]+total(v3[w]) & $
endfor

tot=reform(tot, max(v1)+1, max(v2)+1)
```

;
:
;

Chris.