
Subject: Re: HIST_ND() used for resampling point data onto grid

Posted by [JD Smith](#) on Mon, 21 Mar 2005 16:21:09 GMT

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

On Sat, 19 Mar 2005 11:25:33 -0800, Ed Hyer wrote:

> HIST_ND() is a godsend. Now that I have that, the rest is simple. Yet I
> can't think of how to do it. The answer, most likely, is somewhere in
> the HISTOGRAM tutorial, but I have not penetrated that far into its
> mysteries.
>
> I have DATA(N), with XYT position information given by LON(N),LAT(N),
> and TIMESTAMP(N). I can break this out beautifully with HIST_ND like
> so:
>
> hist_xyt=hist_nd(transpose([[mop_lon],[mop_lat],[mop_ts]]),[1,1,0.25],\$
> min=[-180,-90,154],max=[180,90,243],reverse=ri)
>
> Now, I need to get the mean of DATA(N) (and I wouldn't mind getting the
> variance either)for each bin in HIST_XYT(X,Y,T). I'm 100% confident
> this can be done without a loop, but my brain is not giving me the
> answer.
>
> I don't have complete data coverage, so I'll need also to skip empty
> bins.

Glad you like it. The answer is indeed in the histogram tutorial.

Just use the REVERSE_INDICES to loop through each bin, something like:

```
mean_hist_xyt=make_array(size(hist_xyt,/DIMENSIONS),VALUE=VALUES.F_NAN)
for j=0,n_elements(hist_xyt)-1 do if ri[j+1] gt ri[j] then $
    mean_hist_xyt[j]=mean(data[ri[ri[j]:ri[j+1]-1]])
```

Don't worry about the fact that the histogram is three-dimensional, the use of the reverse indices is the same as for a normal 1D histogram, and all the tricks listed in the tutorial apply. If you have an extremely large histogram, and don't want to loop over bins, you can perform various other tricks to cut down or eliminate the loops; see the tutorial.

JD
