Subject: Re: histogram question
Posted by alt on Fri, 10 Aug 2001 11:03:57 GMT
View Forum Message <> Reply to Message

JD Smith <jdsmith@astro.cornell.edu> wrote in message
news:<3B71C2AA.7E91E5BA@astro.cornell.edu>...
> for i=0,cnt-1 do begin
> low=r[r[bad[i]] & n=r[r[bad[i]+1]]-low
> inds=indgen(n)+low
> if n_elements(list) eq 0 then list=[inds] else list=[list,inds]
> endfor
>
> now you have the list of bad indices into X in hand, to perform whatever
> punishment is necessary.
>
It seems to be some misprint or my task misunderstanding. list is
indices into r, not into x.
for i=0,cnt-1 do begin
inds = r[r[bad[i]] : r[bad[i]+1]-1]
if n_elements(list) eq 0 then list=[inds] else list=[list,inds]
endfor
We needn't to check r[bad[i]] NE r[bad[i]+1] because we adding only
not empty bins.
> This brings up an interesting sub-problem though. If you have a list
> which consists of a series of pairs of indices, e.g.:
>
> [1,5,7,12,15,18]
>
> where each pair is intended to expand to the range within that pair:
>
> [1,2,3,4,5,7,8,9,10,11,12,15,16,17,18]
>
> how can you turn the former into the latter without a loop? This is
> somewhat similar to Pavel's running chunk index problem earlier in the
> year. Finding an answer is not trivial. It would apply directly to
> this problem, where the pairs are adjacent elements in the reverse
> indices vector. Any takers?
From my experience it is much faster in such cases to write a DLM
module then to rack brains on how implement something not standart
into IDL operations. It would be wonderful if IDL has build in C
compiler so we could write C code (with some limitations of course)
just inside of our IDL code. As we do in C writing ASM.

Regards,
Altyntsev Dmitriy