
Subject: Re: Anyway to avoid this last for loop
Posted by [R.Bauer](#) on Thu, 08 Jan 2009 13:54:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

There is no PEP8 for idl somewhere defined, or isn't it. And if so I probably have to clean also my code

The example is difficult to read.

Reimar

hldevil@gmx.de schrieb:

```
> Hi Everybody,  
>  
> after having worked my way through dfannings and jdsmiths histogram  
> tips I still haven't found a way to avoid one last (and time  
> consuming) FOR-loop. Maybe someone has an idea.  
>  
> I have to lists (Millions of entries): one containing detector hit-  
> times, one containing anticoincidence hit-times. Only those detector  
> hits which do not have a AC event at the same time are supposed to be  
> kept.  
>  
> I used a histogram approach:  
>  
> ;AClength is how many frames are thrown out after one AC Hit ->  
> multiple entries can fall into one bin  
>  
> detHist=histogram(data, REVERSE_INDICES=ri, /L64, binSize=AClength)  
> acHist=histogram(ACdeletes, /L64, binSize=AClength)  
>  
>  
> ;renormalize to one  
> dl=where(detHist GT 1, cntD)  
> al=where(acHist GT 1, cntA)  
> IF cntD NE 0 THEN detHist[dl]=1  
> IF cntA NE 0 THEN acHist[al]=1  
>  
> ;subtract the two histograms. All detector frames which have  
> corresponding AC frame should now have value 0, if AC frame exists but  
> not detector frame then value is -1. If only detector frame exists  
> (the ones we want) value stays 1!!!  
> diffHist=detHist-acHist  
>  
> ;keep the ones with one  
> keep=where(diffHist EQ 1, cnt)
```

>
> ;And now my problem. I need the indices of the keep-frames but there
> can be more than one in a bin so I'm stuck with this loop:
>
> FOR k=0L, cnt-1 DO BEGIN
> keepIndices=[keepIndices,ri[ri[keep[k]]:ri[keep[k]+1]-1]]
> ENDFOR
>
> Is there any way I can avoid this, especially because it scales with
> the number of hits which are kept.
>
> Best Regards,
>
> Steffen

Subject: Re: Anyway to avoid this last for loop
Posted by [Tom McGlynn](#) on Thu, 08 Jan 2009 14:28:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Jan 8, 4:59 am, hlde...@gmx.de wrote:

> ;AClength is how many frames are thrown out after one AC Hit ->
> multiple entries can fall into one bin
>
> detHist=histogram(data,REVERSE_INDICES=ri, /L64, binSize=AClength)
> acHist=histogram(ACdeletes, /L64, binSize=AClength)
>
> ;renormalize to one
> dl=where(detHist GT 1, cntD)
> al=where(acHist GT 1, cntA)
> IF cntD NE 0 THEN detHist[dl]=1
> IF cntA NE 0 THEN acHist[al]=1

Can't these last four lines be replaced by

```
detHist = detHist<1  
acHist = acHist<1
```

The < and > operators are useful though not necessarily intuitive.

>
> ;subtract the two histograms. All detector frames which have
> corresponding AC frame should now have value 0, if AC frame exists but
> not detector frame then value is -1. If only detector frame exists
> (the ones we want) value stays 1!!!
> diffHist=detHist-acHist
>
> ;keep the ones with one
> keep=where(diffHist EQ 1, cnt)

```
>  
> ;And now my problem. I need the indices of the keep-frames but there  
> can be more than one in a bin so I'm stuck with this loop:  
>  
> FOR k=0L, cnt-1 DO BEGIN  
>     keepIndices=[keepIndices,ri[ri[keep[k]]:ri[keep[k]+1]-1]]  
> ENDFOR  
>
```

To do it all at once, I think you can just go back to the data array as below:

```
keepHist = where (diffHist EQ 1, cnt) ; From the original  
if (cnt gt 0) then begin  
    keepers = intarr(n_elements(detHist))  
    keepers[keepHist] = 1;  
    keepIndices = where(keepers[data/ACLENGTH] eq 1)  
endif else begin  
    keepIndices = [-1]  
endelse
```

Haven't tested it, but it seems like it or something like it should work reasonably efficiently.

Good luck,
Tom McGlynn

Subject: Re: Anyway to avoid this last for loop
Posted by [Tom McGlynn](#) on Thu, 08 Jan 2009 14:36:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

After posting I realized that this means that one doesn't have to use histogram's reverse indices at all in the solution. That may be illegal in this news group and is a strong indicator that the solution must be wrong. :)

Tom

Subject: Re: Anyway to avoid this last for loop
Posted by [David Fanning](#) on Thu, 08 Jan 2009 14:40:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

Tom McGlynn writes:

> After posting I realized that this means that one doesn't have to use

> histogram's reverse indices at all in the solution. That may be
> illegal in this news group and is a strong indicator that the solution
> must be wrong. :)

At the very least, it violates our charter! :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Anyway to avoid this last for loop
Posted by [hldevil](#) on Thu, 08 Jan 2009 17:31:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 8 Jan., 15:28, Tom McGlynn <t...@milkyway.gsfc.nasa.gov> wrote:

> On Jan 8, 4:59 am, hlde...@gmx.de wrote:

>

>> ;AClength is how many frames are thrown out after one AC Hit ->

>> multiple entries can fall into one bin

>

>> detHist=histogram(data,REVERSE_INDICES=ri, /L64, binSize=AClength)

>> acHist=histogram(ACdeletes, /L64, binSize=AClength)

>

>> ;renormalize to one

>> dl=where(detHist GT 1, cntD)

>> al=where(acHist GT 1, cntA)

>> IF cntD NE 0 THEN detHist[dl]=1

>> IF cntA NE 0 THEN acHist[al]=1

>

> Can't these last four lines be replaced by

> detHist = detHist<1

> acHist = acHist<1

>

> The < and > operators are useful though not necessarily intuitive.

>

>

>

>

>

>

>> ;subtract the two histograms. All detector frames which have

>> corresponding AC frame should now have value 0, if AC frame exists but

>> not detector frame then value is -1. If only detector frame exists

```

>> (the ones we want) value stays 1!!!
>> diffHist=detHist-acHist
>
>> ;keep the ones with one
>> keep=where(diffHist EQ 1, cnt)
>
>> ;And now my problem. I need the indices of the keep-frames but there
>> can be more than one in a bin so I'm stuck with this loop:
>
>> FOR k=0L, cnt-1 DO BEGIN
>>         keepIndices=[keepIndices,ri[ri[keep[k]]:ri[keep[k]+1]-1]]
>> ENDFOR
>
> To do it all at once, I think you can just go back to the data array
> as below:
>
> keepHist = where (diffHist EQ 1, cnt) ; From the original
> if (cnt gt 0) then begin
>     keepers = intarr(n_elements(detHist))
>     keepers[keepHist] = 1;
>     keepIndices = where(keepers[data/ACLENGTH] eq 1)
> endif else begin
>     keepIndices = [-1]
> endelse
>
> Haven't tested it, but it seems like it or something like it should
> work reasonably efficiently.
>
> Good luck,
> Tom McGlynn

```

Seems reasonable. I'll try it and see if I get the same results as with the for loop.

Thanks,

Steffen

Subject: Re: Anyway to avoid this last for loop
 Posted by [hldevil](#) on Fri, 09 Jan 2009 08:46:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

Okay, I tested it. Works perfectly. Thank you very much!

Cheers,

Steffen

Subject: Re: Anyway to avoid this last for loop
Posted by [Jeremy Bailin](#) on Fri, 09 Jan 2009 18:28:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

> Can't these last four lines be replaced by
> detHist = detHist<1
> acHist = acHist<1
>
> The < and > operators are useful though not necessarily intuitive.

For that matter, those operators are valid for the op= construction, so you can write the very efficient (in both performance and typing time):

```
detHist <= 1
```

but which will confuse the hell out of people who normally think in most other languages where <= is a comparison. ;-)

-Jeremy.
