Subject: Re: Create new arrays from series of subsequent integers in an existing array
Posted by JD Smith on Tue, 23 May 2006 01:19:15 GMT
View Forum Message <> Reply to Message

On Mon, 22 May 2006 15:10:08 -0700, Jonathan Wolfe wrote:

> Hello,
>
> I have been trying to average components of a time series when they meet
> certain criteria.
> For example,
>
> I have values in which I use a where statement to specify the critera
>
> criteria=where(time lt 1000)
>
> and I get an array returned which looks like this
>
> data=[0,1,2,3,4,6,8,9,10]
>
> Now, given this array, I would like to specify individual arrays for any
> "block" of subsequent numbers with size greater than three.

It's actually not as simple as it seems. Here's a method which uses LABEL_REGIONS and HISTOGRAM:

```
l=label_region([0L,(shift(data,-1)-data) eq 1,0L])
h=histogram(l[1:n_elements(l)-2],MIN=1,REVERSE_INDICES=ri)
wh=where(h ge 3-1,cnt)
for i=0,cnt-1 do print,data[ri[ri[wh[i]]]:ri[ri[wh[i]+1]-1]+1]
```

Note that LABEL_REGIONS is annoying in that it calls the end point as "no region", so we must temporarily surround it with buffer 0's.
Careful of my usage of REVERSE_INDICES there... instead of the normal semantic:

```
 data[ri[ri[i]:ri[i+1]-1]]
```

which is like data[index_vector], I instead use an explicit range:

```
 data[ri[r[i]]:ri[ri[i+1]-1]+1]
```

which is like data[low:high]. Stare at it until you see the difference. I did this because I knew that the elements in the labeled regions had consecutive indices, and because due to the SHIFT call, we're always missing one member of the "consecutive run" at the end (hence the +1). In general the indices in a given HISTOGRAM bucket

are not adjacent in the original array, so the first form is correct.

If you want to save these arrays as you go, perhaps a pointer array would be useful.

JD
