
Subject: Speed-up of code

Posted by [philipelson](#) on Tue, 25 Aug 2009 14:16:40 GMT

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

Dear All,

I have a question relating to the optimization of some code which averages an array based on the values in another array.

Its much easier to explain in an example:

```
day   = [ 1, 1, 2, 3, 3, 3, 3]
```

```
value = [ 2, 4, 5, 2, 3, 2, 1]
```

Which should return, depending on which is easier, either

```
avg   = [ 3, 5, 2]
```

or

```
avg   = [ 3, 3, 5, 2, 2, 2, 2]
```

This is fairly straightforward using a for loop, but how to do it in the IDL way?

You can see two examples of the basic code below:

```
; =====  
;  
;           FIRST EXAMPLE  
;  
; =====  
unique = uniq(day)  
avg = intarr(n_elements(unique))  
FOR i=0, n_elements(unique) -1 DO BEGIN  
    res = WHERE(day EQ day[unique[i]], count)  
    if count GT 0 THEN avg[i] = total(value[res],/DOUBLE) / count  
ENDFOR  
print, avg
```

```
; =====  
;  
;           SECOND EXAMPLE  
;  
; =====  
h = histogram(day, REVERSE_INDICES=ri)  
avg = h*0  
FOR i=0, n_elements(h)-1 DO BEGIN  
    data_inds = ri[ri[i]:ri[i+1]-1]  
    avg[i] = total(value[data_inds],/DOUBLE) / h[i]  
ENDFOR  
print, avg
```

At this stage I open the floor; I essentially want to achieve the results as above without the need for the for loop.

My assumption is that the HISTOGRAM function will be helpful, but having spent quite some time on this I am beginning to think that it cannot be done - though I would love to be proved wrong by any histogram guru out there.

Thanks,

Philip
