
Subject: array indexing question
Posted by [mmiller3](#) on Thu, 11 Jul 2002 22:57:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

I'm doing some calculations to calculate mutual information and I have a question about the most efficient/slickest way to calculate the sum. I wonder if there is a loop-free way to do this.

I have three arrays, pa, pb and pab. pa and pb are both 1D arrays of length N and pab is a 2D NxN array. I want to calculate the sum of $pab[i,j] * \log(pab[i,j]/pa[i]/pb[j])$. I know that I can do things like $\text{total}(pa * \log(pa))$ when I'm dealing with a single array. Any suggestions for how to do my first sum most efficiently?

Mike

--

Michael A. Miller mmiller3@iupui.edu
Imaging Sciences, Department of Radiology, IU School of Medicine

Subject: Re: Array indexing
Posted by [marc schellens\[1\]](#) on Tue, 16 Jul 2002 08:50:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Andre Kyme wrote:

> Hi everyone,
>
> I have a 2D image and want to index a bunch of (x,y) pairs.
> X is the set of x coords and Y the set of y coords. Say I want
> to set all (x,y) pairs to 200:
>
> image[X,Y]=200
>
> This is fine. But if image has multiple slices and I do:
>
> image[X,Y,0]=200
>
> all possible combinations of points in X and Y get set to 200,
> whereas I only want corresponding values in X and Y set.
> Is there some array notation I'm missing that enables this?

All possible combinations you get when the indexing arrays are of different size.

So here you want to write:

```
Z=lonarr(n_elements(X))  
image[X,Y,Z]=200
```

cheers,
:-) marc
