Subject: Re: extracting values from an array
Posted by Francois L. on Wed, 01 Jun 2005 19:19:09 GMT
View Forum Message <> Reply to Message

Hello,

Actually I thought maybe there is a shortcut like:
B = A gt -1

But if
A = [2,-1, -1, 3,-1,-1, 4, -1]
the command
B = A gt -1
leads to
B = [1,0,0,1,0,0,1,0]
which this doesn't help me as I want B = [2,3,4]

The array A is never bigger than 50 elements.
So I will keep my three lines of code.

Thank you,

"R.G. Stockwell" <no@email.please> wrote in message
news:6znne.23$3A1.1578@news.uswest.net...
> "Francois L." <fleduc@lycos.com> wrote in message
> news:1117651509.818205@news.drenet.dnd.ca...
>> Hello,
>>
>> Probably a simple question...
>>
>> I have an array A and I want to extract values greater than -1.
>>> A = [2,-1, -1, 3,-1,-1, 4, -1]
>>> w = where(a gt -1)
>>> if w[0] gt -1 then b = a[w]
>>
>> Is there a cleaner and faster way to do this ?
>>
>> Thanks,
>>

>
>
> Hi,
>
> cleaner? I doubt it, I think where() is pretty clean :) and I don't think
> writing

> two lines of code is being excessively verbose.
>
> faster? you could try a histogram call, with the reverse indices keyword,
> since histogram is a magic routine that, when used non-intuitively,
> usually leads to superior results. Use 2 bins, -1000 to -1, and 0 to 1000
> for
> instance, where 1000 is chosen to be larger than any of your data.
> This may be less clean than the where() call.
>
> If, however, you are looping through a large number of A vectors,
> perhaps you could do a histogram call on all of them (or something like
> that).
> Or if you are looping over different criteria (gt -1, then gt 1, etc) that
> could
> be accomplished in one histogram call.
>
> Exactly how big is this A array that you are not satisfied with the speed
> of where?
>
> Cheers,
> bob
>
