Subject: Re: operation with array of NaNs
Posted by jschwab@gmail.com on Thu, 01 Feb 2007 01:06:12 GMT
View Forum Message <> Reply to Message

Just to summarize what has already been said.

WHERE returns a list of indicies. So if you do,

IDL> print, where([2, 3, 5, 7] eq 5)

2

If you do,

IDL> print, where([2, 3, 5, 7] gt 4)

2 3

But if you do,

IDL> print, where([2, 3, 5, 7] lt 0)

-1

In the case that the condition inside the WHERE is never satisfied,
IDL returns -1
And when you call, n_elements on the returned value of -1, you
rightfully get 1.

As others have said, IDL will store the number of times the condition
was satisfied in the variable name that you put after the condition in
the WHERE.

So you do,

IDL> void = where([2, 3, 5, 7] lt 0, count)
IDL> print, count

0

And there is the anticipated value of 0.

Using a count variable is a good practice when subscripting with the
result of WHERE.

Instead of

my_new_array= my_old_array[where(my_old_array gt foo)]

Do

new_indicies = where(my_old_array gt foo, count)
if count gt 0 then my_new_array = my_old_array[new_indicies]

I guarantee you that will save you a headache at somepoint.

Josiah

When you use the where statement and the condition is NEVER met, IDL
returns -1.
This is one
On Jan 31, 6:01 pm, inarde...@odu.edu wrote:
> Hello there,
>
> I am quite new to IDL and found an interesting problem.
>
> I have an array of elements (5x5).
> All the elements in the array are NaN(s).
>
> When I ask what is the number of locations (n_elements) where the
> elements of this array are e.g.: "finite", the answer is "1".
>
> If the array contains 1 finite elements (and the rest is Nan(s)), the
> answer to the same question is still "1".
> If the array contains 2 finite elements, the answer to the same
> question is (finally..) "2".
>
>> From there we progress normally.
>
> So why can't I have an answer "0" when the array is only filled with
> NaNs?
>
> The same problem appears with other forms of analysis of the same
> array. For example, "what is the number of elements greater than 10?"
> The answer is still "1"
>
> Thank you for your comments.
>
> I->