Subject: Re: Make_array() and using arrays as subscripts
Posted by btt on Mon, 09 Jan 2006 21:19:03 GMT
View Forum Message <> Reply to Message

Sheldon wrote:
> Hi everyone,
>
> I have a problem with the function MAKE_ARRAY() that I hope someone can
> help me with.
> I created a string array with the dimension (3, 344) and performed some
> basic assignment operations.
> Now I know that when I am finished with the operations I will perform
> on the array, the length 344 will be too long. Let's say that only the
> first 120 elements will be needed. Now I assigned that entire array the
> string '9999' and then when I am ready to reduce the size of the array
> I used the WHERE() function to find the indices where each element is
> not equal to '9999'. All went well until I form this array-as-subscript
> operation:
>
> my_array = my_array[good_indices]
>
> The dimensions all disappear and all I get is an array with the
> elements equalling the number of elements of good_indices, i.e.,
>
> print, size(my_array,/dimensions)
>
> IDL> 120
>
> Now I have used this type of array-as-subscript before on other types
> of array and I have never had this problem before. Does anyone know why
> this happens or how can I cut my_array without losing my 3 dimensions?
>

Hi,

You will want to keep tabs on the dimension(s) that WHERE operates since
a call to WHERE operates on the entire array unless restricted. For the
sake of simplicity you might try a loop. You haven't said if the 99999
flag could be in any one of the columns. If true then you must check
each column for the 99999 flag.

```
dims = SIZE(my_array, /DIM)
flag = MAKE_ARRAY(dims[1], VALUE = 0B)

For i = 0L, dims[0]-1 Do Begin

 A = WHERE(my_array[i,*] NE 9999, nA) ;check the ith column
```

```
 if nA GT 0 then flag[A] = flag[A] + 1B ;increment the ok values

EndFor

 ;any flag LT 3 must have had a 99999 somewhere
A = WHERE(flag EQ 3B, nA)
if nA GT 0 then my_array = my_array[*,A]
```

Hope that helps,
Ben

PS In about 2 minutes you'll hear a chorus of other (better) methods
from people more knowledgable than I about these things.

Subject: Re: Make_array() and using arrays as subscripts
Posted by b_gom on Mon, 09 Jan 2006 21:24:38 GMT
View Forum Message <> Reply to Message

I think you're looking for the ARRAY_INDICES function (This will
convert the 1-D indices returned by WHERE into a multidimensional
array), but it would be useful if you gave a more specific example of
your problem. For the details you gave, If you have a (3,344) array and
you only need the first 120 elements, then just use REFORM to change
my_array[good_indices] into a (3,40) array.

But, I think you have a more general problem. The result of your WHERE
call is not guaranteed to always fit into an array of the same
dimensionality as your original array, so
you would have to figure out what to do with 'empty' elements in your
final array.

```
IDL> x=findgen(3,4,5)
IDL> inds=where(x gt 30)
IDL> inds2=array_indices(x,inds)
IDL> help,inds2
INDS2           LONG      = Array[3, 29]
IDL> xind=inds2[0,*]
IDL> yind=inds2[1,*]
IDL> zind=inds2[2,*]
IDL> print,n_elements(uniq(xind,sort(xind)))
           3
IDL> print,n_elements(uniq(yind,sort(yind)))
           4
IDL> print,n_elements(uniq(zind,sort(zind)))
           3
```

So, we have 29 matches, but if we wanted the result in a 3-D array, we

would have 3*4*3-29=7 empty elements!

Brad

Sheldon wrote:
> Hi everyone,
>
> I have a problem with the function MAKE_ARRAY() that I hope someone can
> help me with.
> I created a string array with the dimension (3, 344) and performed some
> basic assignment operations.
> Now I know that when I am finished with the operations I will perform
> on the array, the length 344 will be too long. Let's say that only the
> first 120 elements will be needed. Now I assigned that entire array the
> string '9999' and then when I am ready to reduce the size of the array
> I used the WHERE() function to find the indices where each element is
> not equal to '9999'. All went well until I form this array-as-subscript
> operation:
>
> my_array = my_array[good_indices]
>
> The dimensions all disappear and all I get is an array with the
> elements equalling the number of elements of good_indices, i.e.,
>
> print, size(my_array,/dimensions)
>
> IDL> 120
>
> Now I have used this type of array-as-subscript before on other types
> of array and I have never had this problem before. Does anyone know why
> this happens or how can I cut my_array without losing my 3 dimensions?
>
> Sincerely,
> Sheldon

Subject: Re: Make_array() and using arrays as subscripts
Posted by peter.albert@gmx.de on Tue, 10 Jan 2006 08:13:13 GMT
View Forum Message <> Reply to Message

Hi Sheldon,

well, reading the title of this posting I don't see where Make_array()
comes into your question, but anyway ... :-)

Like said already, WEHRE always returns a 1-dimensional vector, so here
you loose your dimensions in my_array. If you can be sure that

good_indices always is a factor of 3 (i.e., if a line contains '9999',
all 3 elements are equal to "9999'), then you can use a simple

my_array = reform(my_array, 3, n_elements(good_indices)/3)

to get things right again.

However, if this is not the case, I can suggest yet another approach
without loops:

good_indices = where(strpos(strjoin(my_array), '9999')) eq -1)

Mind that in this case good_indices is only "valid" on the second
dimension, so at maximum it can have 344 elements. I am just
concatenating each line into one string each and look for the occurence
of '9999' within the string. If this substring does not occur at all,
it's a good line.

Then you can use

my_array = my_array[*, good_indices]

to shrink your array to the good lines without the nodata value.

Cheers,

Peter

Subject: Re: Make_array() and using arrays as subscripts
Posted by Liberum on Tue, 10 Jan 2006 09:07:09 GMT
View Forum Message <> Reply to Message

Thanks Peter,

Around midnight last night I solved the problem as you suggested (
my_array = my_array[*, good_indices]). I didn't think about the
asterisk *, i.e. cutting all dimensions with the one dimension array:
good_indices. I used the the MAKE_ARRAY because I had to sort my data
by individual satellites (NOAA15, NOAA16 and so forth). It was easier
to do this with the file strings and hence make_array([3,344],
/string). It may not be pretty but it works :).
Now I am tackling calling a python program from IDL.
Like Arne said: "I'll be back!"

Thanks,
Sheldon

Subject: Re: Make_array() and using arrays as subscripts
Posted by Liberum on Tue, 10 Jan 2006 09:08:39 GMT
View Forum Message <> Reply to Message

Thanks Ben,

Around midnight last night I solved the problem by simply do this (my_array = my_array[*, good_indices]). I didn't think about the asterisk *, i.e. cutting all dimensions with the one dimension array: good_indices.

Thanks,
Sheldon

Subject: Re: Make_array() and using arrays as subscripts
Posted by Liberum on Tue, 10 Jan 2006 09:09:34 GMT
View Forum Message <> Reply to Message

Thanks Brad,

Around midnight last night I solved the problem by simply do this (my_array = my_array[*, good_indices]). I didn't think about the asterisk *, i.e. cutting all dimensions with the one dimension array: good_indices.

Thanks,
Sheldon

Subject: Re: Make_array() and using arrays as subscripts
Posted by peter.albert@gmx.de on Tue, 10 Jan 2006 09:31:50 GMT
View Forum Message <> Reply to Message

> Now I am tackling calling a python program from IDL.

This might help if you try to go the other way round: calling IDL from within python:

http://www.astro.uio.no/~mcmurry/python-idl

Cheers,

Peter