Subject: WHERE problems (longish)
Posted by Benjamin Panter on Tue, 22 Jul 2003 15:23:50 GMT

View Forum Message <> Reply to Message

Hiya,

This is puzzling me, and I've been through all that I can think of. I have a look up table called "dust_lookup". It is a 2 x 300ish array and has wavelengths and the corresponding correction factor. I need to pluck a few values out, so I'm using where:

```
print, where(2900. eq reform(dust_lookup[*,0]))
```

which works absolutly fine for most values: unfortunatly not for all:

If I write a little test:

```
PRO tester, dust_lookup

print, where(2900. eq reform(dust_lookup[*,0]))
print, where(2920. eq reform(dust_lookup[*,0]))
print, where(2940. eq reform(dust_lookup[*,0]))
print, where(2960. eq reform(dust_lookup[*,0]))
print, where(2980. eq reform(dust_lookup[*,0]))
print, where(3000. eq reform(dust_lookup[*,0]))
print, where(3020. eq reform(dust_lookup[*,0]))
print, where(3040. eq reform(dust_lookup[*,0]))
print, where(3060. eq reform(dust_lookup[*,0]))
print, where(3080. eq reform(dust_lookup[*,0]))
print, where(3100. eq reform(dust_lookup[*,0]))

END
```

it comes out with

```
IDL> tester, dust_lookup
          10
          11
          12
          13
          -1
          -1
          -1
          17
          18
          19
          20
```

The values which have -1 certainly exist - and were generated in exactly
the same way as the others. I've put the array online if anyone fancies
looking at it - http://www.roe.ac.uk/~bdp/where_problem.idl

Am I being stupid again? What is special about 2980,3000 and 3020??

Cheers,

Ben

--
Ben Panter, Edinburgh
My name (no spaces)@bigfoot which is a com.

Subject: Re: Where problem
Posted by Foldy Lajos on Thu, 04 Feb 2010 09:33:22 GMT
View Forum Message <> Reply to Message

On Thu, 4 Feb 2010, Dave_Poreh wrote:

> Folks
> I can?t solve this problem. Will somebody tells me what is going on?
> x=findgen(100)
> y=[30,40,50,80]*1.0
> index=where(x eq y)
> but every time gives me:
>> index=-1
> Any help highly appreciated
> Cheers
> Dave
>

Try 'help, x eq y' and 'print, x eq y'.

('x eq y' is equivalent to '[0.,1.,2.,3.] eq [30.,40.,50.,80.]')

regards,
lajos

Subject: Re: Where problem
Posted by d.poreh on Thu, 04 Feb 2010 09:47:02 GMT
View Forum Message <> Reply to Message

On Feb 4, 1:33 am, FÖLDY Lajos <fo...@rmki.kfki.hu> wrote:
> On Thu, 4 Feb 2010, Dave_Poreh wrote:

>> Folks
>> I can?t solve this problem. Will somebody tells me what is going on?
>> x=findgen(100)
>> y=[30,40,50,80]*1.0
>> index=where(x eq y)
>> but every time gives me:
>>> index=-1
>> Any help highly appreciated
>> Cheers
>> Dave
>
> Try 'help, x eq y' and 'print, x eq y'.
>
> ('x eq y' is equivalent to '[0.,1.,2.,3.] eq [30.,40.,50.,80.]')
>
> regards,
> lajos

Thanks. What I am trying to do is to extract index of 30,40,50, and
80 in the array of x. I mean
X[index]=30, X[index]=40, and so on.
What I have to do?
Cheers

Subject: Re: Where problem
Posted by Foldy Lajos on Thu, 04 Feb 2010 09:59:44 GMT
View Forum Message <> Reply to Message

On Thu, 4 Feb 2010, Dave_Poreh wrote:

> On Feb 4, 1:33 am, FÖLDY Lajos <fo...@rmki.kfki.hu> wrote:
>> On Thu, 4 Feb 2010, Dave_Poreh wrote:
>>> Folks
>>> I can?t solve this problem. Will somebody tells me what is going on?
>>> x=findgen(100)
>>> y=[30,40,50,80]*1.0
>>> index=where(x eq y)
>>> but every time gives me:
>>>> index=-1
>>> Any help highly appreciated
>>> Cheers
>>> Dave
>>
>> Try 'help, x eq y' and 'print, x eq y'.
>>
>> ('x eq y' is equivalent to '[0.,1.,2.,3.] eq [30.,40.,50.,80.]')
>>

>> regards,
>> lajos
>
> Thanks. What I am trying to do is to extract index of 30,40,50, and
> 80 in the array of x. I mean
> X[index]=30, X[index]=40, and so on.
> What I have to do?
> Cheers
>

```
IDL> print, value_locate(x,y)
          30          40          50          80
```

regards,
lajos

Subject: Re: Where problem
Posted by Wout De Nolf on Thu, 04 Feb 2010 10:22:30 GMT
View Forum Message <> Reply to Message

On Thu, 4 Feb 2010 01:27:26 -0800 (PST), Dave_Poreh
<d.poreh@gmail.com> wrote:

> Folks
> I canï¿½t solve this problem. Will somebody tells me what is going on?
> x=findgen(100)
> y=[30,40,50,80]*1.0
> index=where(x eq y)
> but every time gives me:
>> index=-1
> Any help highly appreciated
> Cheers
> Dave

Two issues:

1. wrong use of where:

```
IDL> x=findgen(100)
IDL> y=[30,40,50,80]
IDL> b=x eq y
```

b will have 4 elements (smallest of the x and y dimension) and will
b[i] will only be 1 when x[i] eq y[i]. So this is not what you want.

What you can do is:

```
ind=value_locate(x,y)
```

```
ind2=where(x[ind] ne y,ct)
if ct ne 0 then ind[ind2]=-1
```

2. comparing floating point numbers, see:
www.dfanning.com/code_tips/comparearray.html

```
IDL> x=findgen(100)
IDL> y=[30,40,50,80]*1.0
```

I would do something like

```
small=1e-6
ind=value_locate(x,y)
ind2=where(abs(x[ind] - y) gt small,ct)
if ct ne 0 then begin
   ind[ind2]++
   ind2=where(abs(x[ind] - y) gt small,ct)
   if ct ne 0 then ind[ind2]=-1
endif
```

See David's page on what "small" should be.

You could also do something like this

```
x=rebin(x,n_elements(x),n_elements(y),/sample)
y=rebin(transpose(y),n_elements(x),n_elements(y),/sample)
ind=where(abs(x - y) gt small, ct)
```

... and so on ... which is ok for small arrays but not for large arrays (memory issues)

Subject: Re: Where problem
Posted by Wout De Nolf on Thu, 04 Feb 2010 10:26:57 GMT

View Forum Message <> Reply to Message

On Thu, 4 Feb 2010 10:59:44 +0100, Fï¿½LDY Lajos <foldy@rmki.kfki.hu>
wrote:

```
> IDL> print, value_locate(x,y)
>           30          40          50          80
>
> regards,
> lajos
```

I see I was too late :-).

Subject: Re: Where problem
Posted by d.poreh on Thu, 04 Feb 2010 10:30:35 GMT
View Forum Message <> Reply to Message

```
On Feb 4, 2:22 am, Wox <s...@nomail.com> wrote:
>  On Thu, 4 Feb 2010 01:27:26 -0800 (PST), Dave_Poreh
>
>  <d.po...@gmail.com> wrote:
>> Folks
>> I can’t solve this problem. Will somebody tells me what is going on?
>> x=findgen(100)
>> y=[30,40,50,80]*1.0
>> index=where(x eq y)
>> but every time gives me:
>>> index=-1
>> Any help highly appreciated
>> Cheers
>> Dave
>
>  Two issues:
>
>  1. wrong use of where:
>
>  IDL> x=findgen(100)
>  IDL> y=[30,40,50,80]
>  IDL> b=x eq y
>
>  b will have 4 elements (smallest of the x and y dimension) and will
>  b[i] will only be 1 when x[i] eq y[i]. So this is not what you want.
>
>  What you can do is:
>  ind=value_locate(x,y)
>  ind2=where(x[ind] ne y,ct)
>  if ct ne 0 then ind[ind2]=-1
>
>  2. comparing floating point numbers, see:www.dfanning.com/code_tips/comparearray.html
>
>  IDL> x=findgen(100)
>  IDL> y=[30,40,50,80]*1.0
>
>  I would do something like
>
>  small=1e-6
>  ind=value_locate(x,y)
>  ind2=where(abs(x[ind] - y) gt small,ct)
>  if ct ne 0 then begin
>     ind[ind2]++
>     ind2=where(abs(x[ind] - y) gt small,ct)
>     if ct ne 0 then ind[ind2]=-1
```

> endif
>
> See David's page on what "small" should be.
>
> You could also do something like this
>
> x=rebin(x,n_elements(x),n_elements(y),/sample)
> y=rebin(transpose(y),n_elements(x),n_elements(y),/sample)
> ind=where(abs(x - y) gt small, ct)
> ... and so on ... which is ok for small arrays but not for large
> arrays (memory issues)

Thanks Guys. This is exactly what I want.
Cheers

Subject: Re: Where problem
Posted by JJ on Thu, 04 Feb 2010 16:20:21 GMT

View Forum Message <> Reply to Message

The value_locate() solution may work for you, but it will really only work in pretty specific cases. The X vector must be monotonically increasing (though you could first sort the list). Even if the values you're looking for are not in X, a valid result will be returned - since value_locate() returns the index of the start of the "interval into which the given value falls" (this issue addressed by wox).

If you want a simple and robust solution to this kind of problem (and many others), I highly recommend Craig Markwardt's cmset_op routine (follow the Array/Set link from the Markwardt IDL Library page at http://www.physics.wisc.edu/~craigm/idl/idl.html).

With cmset_op you would do something like:

```
index = cmset_op (x, 'and', y, /index)
```

Of course, if there are multiple entries in your array that have the same value, and you want to find the indices of all of those locations, you might want to do a where on each element of y.

-JJ

Subject: Re: Where problem
Posted by d.poreh on Fri, 05 Feb 2010 12:01:33 GMT

View Forum Message <> Reply to Message

On Feb 4, 8:20 am, JJ <j...@cornell.edu> wrote:

> The value_locate() solution may work for you, but it will really only
> work in pretty specific cases. The X vector must be monotonically
> increasing (though you could first sort the list). Even if the values
> you're looking for are not in X, a valid result will be returned -
> since value_locate() returns the index of the start of the "interval
> into which the given value falls" (this issue addressed by wox).
>
> If you want a simple and robust solution to this kind of problem (and
> many others), I highly recommend Craig Markwardt's cmset_op routine
> (follow the Array/Set link from the Markwardt IDL Library page
athttp://www.physics.wisc.edu/~craigm/idl/idl.html).
>
> With cmset_op you would do something like:
>
> index = cmset_op (x, 'and', y, /index)
>
> Of course, if there are multiple entries in your array that have the
> same value, and you want to find the indices of all of those
> locations, you might want to do a where on each element of y.
>
> -JJ

Thanks JJ.
Cheers