
Subject: Re: compare 2-d array with vector

Posted by [Helder Marchetto](#) on Mon, 24 Sep 2012 08:20:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Monday, September 24, 2012 9:32:17 AM UTC+2, Danxia wrote:

> Hi all, I'm having problem when comparing a 2-d array with a given vector. For example, if giving a 2-d array

>
> a=[[1,1,1],

>
> [2,2,2],

>
> [3,3,3],

>
> [4,4,4]]

>
> and a vector b=[1,2],

>
> how can I get all the subscripts of array a that's equal to any element in vector b, which in this example is [0,1,2,3,4,5]. Please let me know if I failed to make this clear. Looking forward to any reply. Thank you very much!

Hi,
I think you should go for the histogram function. It is counter-intuitive at first, but is very fast and works well.

Example on how to start out:

```
IDL> a=[[1,1,1],[2,2,2],[4,4,4],[5,5,5]]
```

```
IDL> b = [2,4]
```

```
IDL> ha = histogram(a, REVERSE_INDICES=ra, OMAX=oMax, OMIN=oMin)
```

```
IDL> hb = histogram(b, MAX=oMax, MIN=oMin)
```

```
IDL> print, ha,hb
```

```
    3    3    0    3    3
    0    1    0    1    0
```

If you then loof for non zero values in the product of ha and hb then you can trace back the common values. If you use the reverse indices, you can find out where such values were in the original array.

Something like this (you have to make some extra controls for non existing matches and similar):

```
IDL> CommonPos = where(ha*hb)
```

```
IDL> FOR I=0,N_ELEMENTS(CommonPos)-1 DO PRINT,
```

```
r[r[CommonPos[I]]:r[CommonPos[I]+1]-1]
```

```
    3    4    5
    6    7    8
```

Use array_indices if you want the positions in two dimensions.

Hope it helps,
Helder

Subject: Re: compare 2-d array with vector
Posted by [Heinz Stege](#) on Mon, 24 Sep 2012 11:39:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Mon, 24 Sep 2012 00:32:17 -0700 (PDT), Danxia wrote:

> Hi all, I'm having problem when comparing a 2-d array with a given vector. For example, if giving a 2-d array
> a=[[1,1,1],
> [2,2,2],
> [3,3,3],
> [4,4,4]]
> and a vector b=[1,2],
> how can I get all the subscripts of array a that's equal to any element in vector b, which in this example is [0,1,2,3,4,5]. Please let me know if I failed to make this clear. Looking forward to any reply. Thank you very much!

Hi Danxia,

the use of the value_locate function is very helpful:

```
subscripts=where(a eq b[value_locate(b,a)>0])
```

Note that b must be sorted and must have 2 or more elements. ">0" may be omitted, if a is an array (and you have not enabled the STRICTARRSUBS compiler option).

Cheers, Heinz

Subject: Re: compare 2-d array with vector
Posted by [Heinz Stege](#) on Mon, 24 Sep 2012 11:53:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Mon, 24 Sep 2012 13:39:44 +0200, Heinz Stege wrote:

> the use of the value_locate function is very helpful:
>
> subscripts=where(a eq b[value_locate(b,a)>0])
>
> Note that b must be sorted and must have 2 or more elements.

AFAIK some IDL Developers are reading this newsgroup. Here is a request for an enhancement of the value_locate function:

Please let value_locate return result=-long(a lt b[0]) instead of throwing an error message, if b has 1 element only.

Would be nice. Thank you.

Heinz

Subject: Re: compare 2-d array with vector
Posted by [Craig Markwardt](#) on Mon, 24 Sep 2012 15:19:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Monday, September 24, 2012 7:52:55 AM UTC-4, Heinz Stege wrote:

> On Mon, 24 Sep 2012 13:39:44 +0200, Heinz Stege wrote:
>
>
>
>> the use of the value_locate function is very helpful:
>
>>
>
>> subscripts=where(a eq b[value_locate(b,a)>0])
>
>>
>
>> Note that b must be sorted and must have 2 or more elements.
>
>
>
> AFAIK some IDL Developers are reading this newsgroup. Here is a
>
> request for an enhancement of the value_locate function:
>
>
>
> Please let value_locate return result=-long(a lt b[0]) instead of
>
> throwing an error message, if b has 1 element only.

Yeah, I filed a support request on this several years ago, but no action so far.

Craig

Subject: Re: compare 2-d array with vector
Posted by [Mark Piper](#) on Mon, 24 Sep 2012 15:30:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Monday, September 24, 2012 9:19:38 AM UTC-6, Craig Markwardt wrote:

>

> Yeah, I filed a support request on this several years ago, but no action so far.
>

Heinz & Craig,

This looks eminently reasonable. I'll look at getting it into the backlog today.

mp

Subject: Re: compare 2-d array with vector
Posted by chris_torrence@NOSPAM on Mon, 17 Dec 2012 19:25:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Monday, September 24, 2012 9:30:25 AM UTC-6, Mark Piper wrote:
> On Monday, September 24, 2012 9:19:38 AM UTC-6, Craig Markwardt wrote:
>
>>
>
>> Yeah, I filed a support request on this several years ago, but no action so far.
>
>>
>
>
>
> Heinz & Craig,
>
>
>
> This looks eminently reasonable. I'll look at getting it into the backlog today.
>
>
>
> mp

This is now fixed and will be in IDL 8.8.2. You can now pass a 1-element array or a scalar into value_locate.

-Chris
ExelisVIS

Subject: Re: compare 2-d array with vector
Posted by [Fabzi](#) on Mon, 07 Jan 2013 10:59:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 12/17/2012 08:25 PM, Chris Torrence wrote:

> This is now fixed and will be in IDL 8.8.2. You can now pass a 1-element array or a scalar into value_locate.
>
> -Chris
> ExelisVIS

Hi Chris,

While you are looking at VALUE_LOCATE, could you also look into this?

```
IDL> data = FINDGEN(10) & data[0] = !VALUES.F_NAN
IDL> p = VALUE_LOCATE(INDGEN(10), data) & print, p[0]
      0
% Program caused arithmetic error: Floating illegal operand
IDL> p = VALUE_LOCATE(LINDGEN(10), data) & print, p[0]
      -1
% Program caused arithmetic error: Floating illegal operand
IDL> p = VALUE_LOCATE(FINDGEN(10), data) & print, p[0]
      -1
```

This was discussed in this thread:

https://groups.google.com/forum/?hl=fr&fromgroups=#!searchin/comp.lang.idl-pvwave/fab/comp.lang.idl-pvwave/DBbUmnAaB Mc/kYdniU_d3koJ

Thanks!

Fab

Subject: Re: compare 2-d array with vector
Posted by [Jeremy Bailin](#) on Mon, 07 Jan 2013 22:30:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 1/7/13 4:59 AM, Fab wrote:

```
> On 12/17/2012 08:25 PM, Chris Torrence wrote:
>> This is now fixed and will be in IDL 8.8.2. You can now pass a
>> 1-element array or a scalar into value_locate.
>>
>> -Chris
>> ExelisVIS
>
> Hi Chris,
>
> While you are looking at VALUE_LOCATE, could you also look into this?
>
> IDL> data = FINDGEN(10) & data[0] = !VALUES.F_NAN
> IDL> p = VALUE_LOCATE(INDGEN(10), data) & print, p[0]
>      0
```

> % Program caused arithmetic error: Floating illegal operand
> IDL> p = VALUE_LOCATE(LINDGEN(10), data) & print, p[0]
> -1
> % Program caused arithmetic error: Floating illegal operand
> IDL> p = VALUE_LOCATE(FINDGEN(10), data) & print, p[0]
> -1
>
> This was discussed in this thread:
> https://groups.google.com/forum/?hl=fr&fromgroups=#!searchin/comp.lang.idl-pvwave/fab/comp.lang.idl-pvwave/DBbUmnAaB Mc/kYdniU_d3koJ
>
>
> Thanks!
>
> Fab
>

What do you expect him to do? Our conclusion was that it's inherently undefined, so there's not much point in asking for consistency.

-Jeremy.
