Subject: Re: Compare two variables
Posted by Joost Aan de Brugh on Fri, 11 Jul 2008 15:41:14 GMT
View Forum Message <> Reply to Message

On Jul 11, 2:05 pm, d.po...@gmail.com wrote:
> Hi everyone
> I have got a problem and I don't know how to solve it:
> I have two variables like this:
> A, which col*row=2*4
> B, which col*row=4*100.
> All A's couples (x,y)'s are somewhere in the 2th and 3th column of
> Result. I want to take this rows (I want to extract that rows in
> result which 2th and 3th columns are settled in the B's).for
> example:
> A=[[1,2], [3,4] ,[5,6], [7,8]]
>
> B=[….[11,22,1,2,],….. [44,55,3,4]….. ,[22,99,5,6], ….[77,66,7,8]…]
>
> …..=means there are some other data
> I just want to extract these rows and put them in a new variable
> Any help?
> Cheers

There are two problems. One is that you have two variables you have to compare and one is that you have a whole vector of values they can be.

Probably you are familiar with the Where-function

idx = Where(array <condition considering array as a scalar>)

idx will be a vector of indices where the array has values that meet the condition. If there are none, idx will be -1.

The problem that you have to check both the 2nd and the 3rd column will force constructions like

idx1 = Where(vector1 eq value1)
idx2 = Where(vector2 eq value2)
if idx1[0] eq -1 or idx2[0] eq -1 then screw it
...

It is better to create another vector. As you work with integers, you compress two integers into one.
totalB = B[2,*] + B[3,*]
DB = (total*(total+1))/2 + B[2,*] ; The value in DB[j] determines exactly what B[2,j] and B[3,j] are
totalA = A[0,*] + A[1,*]
DA = (total*(total+1))/2 + A[0,*] ; Same joke, so if DA[i] eq DB[j]

then A[0,i] eq B[2,j] and A[1,i] eq B[3,j]

And now we are left with two vectors. As long as DA only contains four values (like your example), you can use a forloop

```
for j=0,4-1 do begin
  idx = Where(DB eq DA[j])
  if idx[0] eq -1 then continue ; Subscripting [0] is required if idx
can have more than one value.
  if N_Elements(idxs) eq 0 then idxs = idx else idxx = [idxx,idx] ;
Add the index to the result vector.
end

result = B[*,idxs]
```

There must be a more efficient way to do the last part, but Where(DB eq DA) does not work.

You can also do it at the array way, create one array like. Watch out where to use Transposes (Check it out first)

```
MA =

DA[0]  DA[1]  DA[2]  DA[3]
DA[0]  DA[1]  DA[2]  DA[3]
DA[0]  DA[1]  DA[2]  DA[3]
DA[0]  DA[1]  DA[2]  DA[3]
DA[0]  DA[1]  DA[2]  DA[3]
...
```

and one like

```
MB =

DB[0]  DB[0]  DB[0]  DB[0]
DB[1]  DB[1]  DB[1]  DB[1]
DB[2]  DB[2]  DB[2]  DB[2]
DB[3]  DB[3]  DB[3]  DB[3]
DB[4]  DB[4]  DB[4]  DB[4]
...

MC = LonArr(4,100) ; Comparison
idx = Where(MB eq MA)
MC[idx] = 1

IDL> print,MC

0  0  0  0 ; Drop this B-index
```

0 1 0 0 ; There is an A-index that fits
0 0 0 0
0 0 1` 0
1 0 0 0
...

idxs = total(MC,1)

IDL> print,idxs

0,1,0,1,1,...

result = B[*,idxs]

; This is more work, but probably faster than a for-loop if we have
more than 4 rows in A.

I hope this helps you. Probably there are better ways, but this should
work.

Subject: Re: Compare two variables
Posted by Jean H. on Fri, 11 Jul 2008 15:49:33 GMT
View Forum Message <> Reply to Message

d.poreh@gmail.com wrote:
> Hi everyone
> I have got a problem and I donï¿½t know how to solve it:
> I have two variables like this:
> A, which col*row=2*4
> B, which col*row=4*100.
> All Aï¿½s couples (x,y)ï¿½s are somewhere in the 2th and 3th column of
> Result. I want to take this rows (I want to extract that rows in
> result which 2th and 3th columns are settled in the Bï¿½s).for
> example:
> A=[[1,2], [3,4] ,[5,6], [7,8]]
>
> B=[ï¿½.[11,22,1,2,],ï¿½.. [44,55,3,4]ï¿½.. ,[22,99,5,6],
ï¿½.[77,66,7,8]ï¿½]
>
> ï¿½..=means there are some other data
> I just want to extract these rows and put them in a new variable
> Any help?
> Cheers

If you only have integers, you may want to re-create the cell index....
either the "real" cell index, or one of your own (like cell 1;2 ==> 12,
or 0102 or...). Then a simple intersect will do the trick

Jean

Subject: Re: Compare two variables
Posted by d.poreh on Fri, 11 Jul 2008 17:14:59 GMT
View Forum Message <> Reply to Message

On Jul 11, 5:49 pm, "Jean H." <jghas...@DELTHIS.ucalgary.ANDTHIS.ca>
wrote:
> d.po...@gmail.com wrote:
>> Hi everyone
>> I have got a problem and I don't know how to solve it:
>> I have two variables like this:
>> A, which col*row=2*4
>> B, which col*row=4*100.
>> All A's couples (x,y)'s are somewhere in the 2th and 3th column of
>> Result. I want to take this rows (I want to extract that rows in
>> result which 2th and 3th columns are settled in the B's).for
>> example:
>> A=[[1,2], [3,4] ,[5,6], [7,8]]
>
>> B=[….[11,22,1,2,],….. [44,55,3,4]….. ,[22,99,5,6], ….[77,66,7,8]…]
>
>> …..=means there are some other data
>> I just want to extract these rows and put them in a new variable
>> Any help?
>> Cheers
>
> If you only have integers, you may want to re-create the cell index....
> either the "real" cell index, or one of your own (like cell 1;2 ==> 12,
> or 0102 or...). Then a simple intersect will do the trick
>
> Jean

no ny data is not integer

Subject: Re: Compare two variables
Posted by d.poreh on Sat, 12 Jul 2008 13:18:34 GMT
View Forum Message <> Reply to Message

> ; This is more work, but probably faster than a for-loop if we have
> more than 4 rows in A.
>
> I hope this helps you. Probably there are better ways, but this should
> work.

hi Joost
but i work with float data. as you said this metheod works for
integer. could we modify it to work for non integer data?
Cheers
Dave

Subject: Re: Compare two variables
Posted by Sven Geier on Sat, 12 Jul 2008 19:54:53 GMT
View Forum Message <> Reply to Message

d.poreh@gmail.com wrote:

> Hi everyone
> I have got a problem and I don?t know how to solve it:
> I have two variables like this:
> A, which col*row=2*4
> B, which col*row=4*100.
> All A?s couples (x,y)?s are somewhere in the 2th and 3th column of
> Result. I want to take this rows (I want to extract that rows in
> result which 2th and 3th columns are settled in the B?s).for
> example:
> A=[[1,2], [3,4] ,[5,6], [7,8]]
>
> B=[?.[11,22,1,2,],?.. [44,55,3,4]?.. ,[22,99,5,6], ?.[77,66,7,8]?]
>
> ?..=means there are some other data
> I just want to extract these rows and put them in a new variable
> Any help?
> Cheers

I'm probably just misunderstanding something, but what is wrong with

line_1_index = where (b[2,*] eq a[0,0] and b[3,*] eq a[1,0])
line_2_index = ...
...

result = [b[*,line_1_index],b[*,line_2_index],b[...],...]

It's not terribly elegant but if it's only 5 lines and has to be typed only
once, what's the point of optimizing the hell out of it?

--
http://www.sgeier.net
My real email address does not contain any "Z"s.

Subject: Re: Compare two variables
Posted by Joost Aan de Brugh on Mon, 14 Jul 2008 08:09:37 GMT
View Forum Message <> Reply to Message

Hello Dave,

> hi Joost
> but i work with float data. as you said this metheod works for
> integer. could we modify it to work for non integer data?
> Cheers
> Dave

That is a pity. But isn't that dangerous in any case. If A and B are floats then the expression A eq B is not reliable because of continuous rounding. It may still be reliable if absolutely no arithmetic is involved.

The compression trick does not work for floats, because of the degree of infinity.

Maybe a two-step filtering is apropriate

idx1 = Where(B[2,*] = A[0,j]) ; in for-loop or with the matrix-trick I did with DA and DB.
inbetweenresult = B[*,idx1]

idx2 = Where(inbetweenresult[3,*] = A[1,j]) ; in for-loop or with the matrix-trick I did with DA and DB.
result = inbetweenresult[*,idx2]

Cheers,
Joost

Subject: Re: Compare two variables
Posted by ben.bighair on Mon, 14 Jul 2008 11:57:20 GMT
View Forum Message <> Reply to Message

On Jul 14, 4:09 am, Joost Aan de Brugh <joost...@gmail.com> wrote:
> Hello Dave,
>
>> hi Joost
>> but i work with float data. as you said this metheod works for
>> integer. could we modify it to work for non integer data?
>> Cheers
>> Dave
>
> That is a pity. But isn't that dangerous in any case. If A and B are

> floats then the expression A eq B is not reliable because of
> continuous rounding. It may still be reliable if absolutely no
> arithmetic is involved.
>
> The compression trick does not work for floats, because of the degree
> of infinity.
>
> Maybe a two-step filtering is apropriate
>
> idx1 = Where(B[2,*] = A[0,j]) ; in for-loop or with the matrix-trick I
> did with DA and DB.
> inbetweenresult = B[*,idx1]
>
> idx2 = Where(inbetweenresult[3,*] = A[1,j]) ; in for-loop or with the
> matrix-trick I did with DA and DB.
> result = inbetweenresult[*,idx2]

Hi,

It doesn't seem to me that Dave has provided sufficient information.
I think the question was not fully fleshed out so it is hard to
provide helpful answers.

For example, is it possible that the coordinates could be temporarily
coerced into integers with losing unique pairings? If that is that
case then he can use the method described by Joost. Or, here is
another tack, is there a certain granularity (or precision) to the
coordinates - measured to the nearest tenth or hundreth perhaps? If
that is the case then he could simply promote the coordinates by
multiplying by 10 (or 100 or whatever) and then convert to integer.

Cheers,
Ben

Subject: Re: Compare two variables
Posted by d.poreh on Mon, 14 Jul 2008 13:00:17 GMT
View Forum Message <> Reply to Message

On 14 Jul., 04:57, "ben.bighair" <ben.bigh...@gmail.com> wrote:
> On Jul 14, 4:09 am, Joost Aan de Brugh <joost...@gmail.com> wrote:
>
>
>
>
>
>> Hello Dave,
>

>>> hi Joost
>>> but i work with float data. as you said this metheod works for
>>> integer. could we modify it to work for non integer data?
>>> Cheers
>>> Dave
>
>> That is a pity. But isn't that dangerous in any case. If A and B are
>> floats then the expression A eq B is not reliable because of
>> continuous rounding. It may still be reliable if absolutely no
>> arithmetic is involved.
>
>> The compression trick does not work for floats, because of the degree
>> of infinity.
>
>> Maybe a two-step filtering is apropriate
>
>> idx1 = Where(B[2,*] = A[0,j]) ; in for-loop or with the matrix-trick I
>> did with DA and DB.
>> inbetweenresult = B[*,idx1]
>
>> idx2 = Where(inbetweenresult[3,*] = A[1,j]) ; in for-loop or with the
>> matrix-trick I did with DA and DB.
>> result = inbetweenresult[*,idx2]
>
> Hi,
>
> It doesn't seem to me that Dave has provided sufficient information.
> I think the question was not fully fleshed out so it is hard to
> provide helpful answers.
>
> For example, is it possible that the coordinates could be temporarily
> coerced into integers with losing unique pairings? If that is that
> case then he can use the method described by Joost. Or, here is
> another tack, is there a certain granularity (or precision) to the
> coordinates - measured to the nearest tenth or hundreth perhaps? If
> that is the case then he could simply promote the coordinates by
> multiplying by 10 (or 100 or whatever) and then convert to integer.
>
> Cheers,
> Ben- Zitierten Text ausblenden -
>
> - Zitierten Text anzeigen -

Ben
Actually my coordinates are float numbers with 2 decimal. I think your proposed way is good and I can multiply that numbers whit 100 to take integers and after finishing I can divide them to 100 to take actual coordinate.

Cheers
Dave