Subject: Re: Matching elements in two arrays of different sizes
Posted by Russell Ryan on Tue, 26 Jun 2012 20:28:20 GMT
View Forum Message <> Reply to Message

On Tuesday, June 26, 2012 12:07:29 PM UTC-4, pindsy wrote:
> Hi everyone,
>
> I am having trouble figuring out how to search through an array of size [1,17824] and match it to the points in another array of size [1,70].
>
> What I am trying to use to match the two arrays are the month, day, year, hour, and minute within each dataset. I have made two arrays using julday and have been trying to go through each of those arrays to find matches. However, it only searches up to the 70 first lines of the larger array and dosen't give all the possible matches.
>
> Any ideas on how to fix this would be helpful.
>
> Cheers,
>
> Meredith

On Tuesday, June 26, 2012 12:07:29 PM UTC-4, pindsy wrote:
> Hi everyone,
>
> I am having trouble figuring out how to search through an array of size [1,17824] and match it to the points in another array of size [1,70].
>
> What I am trying to use to match the two arrays are the month, day, year, hour, and minute within each dataset. I have made two arrays using julday and have been trying to go through each of those arrays to find matches. However, it only searches up to the 70 first lines of the larger array and dosen't give all the possible matches.
>
> Any ideas on how to fix this would be helpful.
>
> Cheers,
>
> Meredith

There are probably cleverer ways to do this, but the problem seems simple enough that a simple tool may be the best. Suppose the dates are d1 and d2 (d1 is the shorter vector)

```
maxmintime=0.01
n1=n_elements(d1)

mintime=fltarr(n1)
minid=lonarr(n1)
```

```
for i=0,n-1 do begin
  dt=abs(d1(i)-d2)
  mintime(i)=min(dt,loc)
  minid(i)=loc
endfor

g=where(mintime lt maxmintime)
if g(0) ne -1 then begin
  d1=d1(g)
  d2=d2(minid(g))
endif else print,'No matches!'
```

Subject: Re: Matching elements in two arrays of different sizes
Posted by lecacheux.alain on Wed, 27 Jun 2012 16:14:40 GMT
View Forum Message <> Reply to Message

On 26 juin, 18:07, pindsy <meredith.p...@gmail.com> wrote:
> Hi everyone,
>
> I am having trouble figuring out how to search through an array of size [1,17824] and match it to the points in another array of size [1,70].
>
> What I am trying to use to match the two arrays are the month, day, year, hour, and minute within each dataset. I have made two arrays using julday and have been trying to go through each of those arrays to find matches. However, it only searches up to the 70 first lines of the larger array and dosen't give all the possible matches.
>
> Any ideas on how to fix this would be helpful.
>
> Cheers,
>
> Meredith
>
>

Still use VALUE_LOCATE...
if jd1 and jd2 are the two arrays of julian dates, w will be the
vector of matching indices:

```
  w = where(jd1[Value_Locate(jd1, jd2)] eq jd2, /NULL)
```

For instance:

```
IDL> jd1 = julday(indgen(6)*2,1,2012)  ;month 1st day, every two
monthes
IDL> jd2 = julday(indgen(12),1,2012)   ;1st day of each month
IDL> print, where(jd1[Value_Locate(jd1, jd2)] eq jd2, /NULL)
         0         2         4         6         8
10
```

alain.

Subject: Re: Matching elements in two arrays of different sizes
Posted by Meredith Pind on Wed, 27 Jun 2012 16:38:51 GMT
View Forum Message <> Reply to Message

Awesome, thanks!

Here's a dumb question. What do I use as the variable that holds all the matching data to call the associated data in the data set to those dates?

On Tuesday, June 26, 2012 3:28:20 PM UTC-5, (unknown) wrote:
> On Tuesday, June 26, 2012 12:07:29 PM UTC-4, pindsy wrote:
>> Hi everyone,
>>
>> I am having trouble figuring out how to search through an array of size [1,17824] and match it to the points in another array of size [1,70].
>>
>> What I am trying to use to match the two arrays are the month, day, year, hour, and minute within each dataset. I have made two arrays using julday and have been trying to go through each of those arrays to find matches. However, it only searches up to the 70 first lines of the larger array and dosen't give all the possible matches.
>>
>> Any ideas on how to fix this would be helpful.
>>
>> Cheers,
>>
>> Meredith
>
>
>
> On Tuesday, June 26, 2012 12:07:29 PM UTC-4, pindsy wrote:
>> Hi everyone,
>>
>> I am having trouble figuring out how to search through an array of size [1,17824] and match it to the points in another array of size [1,70].
>>
>> What I am trying to use to match the two arrays are the month, day, year, hour, and minute within each dataset. I have made two arrays using julday and have been trying to go through each of those arrays to find matches. However, it only searches up to the 70 first lines of the larger array and dosen't give all the possible matches.
>>
>> Any ideas on how to fix this would be helpful.
>>
>> Cheers,
>>

>> Meredith
>
> There are probably cleverer ways to do this, but the problem seems simple enough that a simple tool may be the best. Suppose the dates are d1 and d2 (d1 is the shorter vector)
>
> maxmintime=0.01
> n1=n_elements(d1)
>
> mintime=fltarr(n1)
> minid=lonarr(n1)
> for i=0,n-1 do begin
> dt=abs(d1(i)-d2)
> mintime(i)=min(dt,loc)
> minid(i)=loc
> endfor
>
> g=where(mintime lt maxmintime)
> if g(0) ne -1 then begin
> d1=d1(g)
> d2=d2(minid(g))
> endif else print,'No matches!'

On Tuesday, June 26, 2012 3:28:20 PM UTC-5, (unknown) wrote:
> On Tuesday, June 26, 2012 12:07:29 PM UTC-4, pindsy wrote:
>> Hi everyone,
>>
>> I am having trouble figuring out how to search through an array of size [1,17824] and match it to the points in another array of size [1,70].
>>
>> What I am trying to use to match the two arrays are the month, day, year, hour, and minute within each dataset. I have made two arrays using julday and have been trying to go through each of those arrays to find matches. However, it only searches up to the 70 first lines of the larger array and dosen't give all the possible matches.
>>
>> Any ideas on how to fix this would be helpful.
>>
>> Cheers,
>>
>> Meredith
>
>
>
> On Tuesday, June 26, 2012 12:07:29 PM UTC-4, pindsy wrote:
>> Hi everyone,
>>
>> I am having trouble figuring out how to search through an array of size [1,17824] and match it

to the points in another array of size [1,70].
>>
>> What I am trying to use to match the two arrays are the month, day, year, hour, and minute within each dataset. I have made two arrays using julday and have been trying to go through each of those arrays to find matches. However, it only searches up to the 70 first lines of the larger array and dosen't give all the possible matches.
>>
>> Any ideas on how to fix this would be helpful.
>>
>> Cheers,
>>
>> Meredith
>
> There are probably cleverer ways to do this, but the problem seems simple enough that a simple tool may be the best. Suppose the dates are d1 and d2 (d1 is the shorter vector)
>
> maxmintime=0.01
> n1=n_elements(d1)
>
> mintime=fltarr(n1)
> minid=lonarr(n1)
> for i=0,n-1 do begin
>   dt=abs(d1(i)-d2)
>   mintime(i)=min(dt,loc)
>   minid(i)=loc
> endfor
>
> g=where(mintime lt maxmintime)
> if g(0) ne -1 then begin
>   d1=d1(g)
>   d2=d2(minid(g))
> endif else print,'No matches!'

Subject: Re: Matching elements in two arrays of different sizes
Posted by Meredith Pind on Wed, 27 Jun 2012 16:48:02 GMT
View Forum Message <> Reply to Message

Hey Alain,

thanks. I did this and got 16 matches
w =   0   1   2   3   4   10   11   12   13   14   15
       16   17   18   19   20

When I go and print off jd2(w) (smaller array) and jd1(w) (larger array) I get this

IDL> print, jd1(w)
2455771.2   2455771.2   2455771.2   2455771.2   2455771.2   2455771.2

2455771.2 2455771.2 2455771.2 2455771.2 2455771.2 2455771.2
2455771.2 2455771.2 2455771.2 2455771.2

IDL> print, jd2(w)
2455774.0 2455773.6 2455772.9 2455773.0 2455772.7 2455778.0
2455778.0 2455776.9 2455776.9 2455783.8 2455782.6 2455781.9
2455781.9 2455780.7 2455775.0 2455774.8

Which aren't the same values. I'm not sure I understand this.

On Wednesday, June 27, 2012 11:14:40 AM UTC-5, alx wrote:
> On 26 juin, 18:07, pindsy <meredith.p...@gmail.com> wrote:
>> Hi everyone,
>>
>> I am having trouble figuring out how to search through an array of size [1,17824] and match it to the points in another array of size [1,70].
>>
>> What I am trying to use to match the two arrays are the month, day, year, hour, and minute within each dataset. I have made two arrays using julday and have been trying to go through each of those arrays to find matches. However, it only searches up to the 70 first lines of the larger array and dosen't give all the possible matches.
>>
>> Any ideas on how to fix this would be helpful.
>>
>> Cheers,
>>
>> Meredith
>>
>>
>
> Still use VALUE_LOCATE...
> if jd1 and jd2 are the two arrays of julian dates, w will be the
> vector of matching indices:
> w = where(jd1[Value_Locate(jd1, jd2)] eq jd2, /NULL)
>
> For instance:
> IDL> jd1 = julday(indgen(6)*2,1,2012) ;month 1st day, every two
> monthes
> IDL> jd2 = julday(indgen(12),1,2012) ;1st day of each month
> IDL> print, where(jd1[Value_Locate(jd1, jd2)] eq jd2, /NULL)
> 0 2 4 6 8
> 10
>
> alain.

On Wednesday, June 27, 2012 11:14:40 AM UTC-5, alx wrote:

> On 26 juin, 18:07, pindsy <meredith.p...@gmail.com> wrote:
>> Hi everyone,
>>
>> I am having trouble figuring out how to search through an array of size [1,17824] and match it to the points in another array of size [1,70].
>>
>> What I am trying to use to match the two arrays are the month, day, year, hour, and minute within each dataset. I have made two arrays using julday and have been trying to go through each of those arrays to find matches. However, it only searches up to the 70 first lines of the larger array and dosen't give all the possible matches.
>>
>> Any ideas on how to fix this would be helpful.
>>
>> Cheers,
>>
>> Meredith
>>
>>
>
> Still use VALUE_LOCATE...
> if jd1 and jd2 are the two arrays of julian dates, w will be the
> vector of matching indices:
> w = where(jd1[Value_Locate(jd1, jd2)] eq jd2, /NULL)
>
> For instance:
> IDL> jd1 = julday(indgen(6)*2,1,2012) ;month 1st day, every two
> monthes
> IDL> jd2 = julday(indgen(12),1,2012) ;1st day of each month
> IDL> print, where(jd1[Value_Locate(jd1, jd2)] eq jd2, /NULL)
> 0 2 4 6 8
> 10
>
> alain.

Subject: Re: Matching elements in two arrays of different sizes
Posted by lecacheux.alain on Wed, 27 Jun 2012 21:28:24 GMT
View Forum Message <> Reply to Message

On 27 juin, 18:48, pindsy <meredith.p...@gmail.com> wrote:
> Hey Alain,
>
> thanks. I did this and got 16 matches
> w = 0 1 2 3 4
10 11 12 13 14 15
16 17 18 19 20
>
> When I go and print off jd2(w) (smaller array) and jd1(w) (larger array) I get this

>
> IDL> print, jd1(w)
> 2455771.2   2455771.2   2455771.2   2455771.2   2455771.2
  2455771.2   2455771.2   2455771.2   2455771.2   2455771.2
  2455771.2   2455771.2   2455771.2   2455771.2   2455771.2
  2455771.2
>
> IDL> print, jd2(w)
> 2455774.0   2455773.6   2455772.9   2455773.0   2455772.7
  2455778.0   2455778.0   2455776.9   2455776.9   2455783.8
  2455782.6   2455781.9   2455781.9   2455780.7   2455775.0
  2455774.8
>
> Which aren't the same values. I'm not sure I understand this.
>
>
>
> On Wednesday, June 27, 2012 11:14:40 AM UTC-5, alx wrote:
>> On 26 juin, 18:07, pindsy <meredith.p...@gmail.com> wrote:
>>> Hi everyone,
>
>>> I am having trouble figuring out how to search through an array of size [1,17824] and match it to the points in another array of size [1,70].
>
>>> What I am trying to use to match the two arrays are the month, day, year, hour, and minute within each dataset. I have made two arrays using julday and have been trying to go through each of those arrays to find matches. However, it only searches up to the 70 first lines of the larger array and dosen't give all the possible matches.
>
>>> Any ideas on how to fix this would be helpful.
>
>>> Cheers,
>
>>> Meredith
>
>> Still use VALUE_LOCATE...
>> if jd1 and jd2 are the two arrays of julian dates, w will be the
>> vector of matching indices:
>>   w = where(jd1[Value_Locate(jd1, jd2)] eq jd2, /NULL)
>
>> For instance:
>> IDL> jd1 = julday(indgen(6)*2,1,2012) ;month 1st day, every two
>> monthes
>> IDL> jd2 = julday(indgen(12),1,2012)   ;1st day of each month
>> IDL> print, where(jd1[Value_Locate(jd1, jd2)] eq jd2, /NULL)
>>   0   2   4   6   8
>> 10
>

>> alain.
> On Wednesday, June 27, 2012 11:14:40 AM UTC-5, alx wrote:
>> On 26 juin, 18:07, pindsy <meredith.p...@gmail.com> wrote:
>>> Hi everyone,
>
>>> I am having trouble figuring out how to search through an array of size [1,17824] and match it to the points in another array of size [1,70].
>
>>> What I am trying to use to match the two arrays are the month, day, year, hour, and minute within each dataset. I have made two arrays using julday and have been trying to go through each of those arrays to find matches. However, it only searches up to the 70 first lines of the larger array and dosen't give all the possible matches.
>
>>> Any ideas on how to fix this would be helpful.
>
>>> Cheers,
>
>>> Meredith
>
>> Still use VALUE_LOCATE...
>> if jd1 and jd2 are the two arrays of julian dates, w will be the
>> vector of matching indices:
>> w = where(jd1[Value_Locate(jd1, jd2)] eq jd2, /NULL)
>
>> For instance:
>> IDL> jd1 = julday(indgen(6)*2,1,2012) ;month 1st day, every two
>> monthes
>> IDL> jd2 = julday(indgen(12),1,2012) ;1st day of each month
>> IDL> print, where(jd1[Value_Locate(jd1, jd2)] eq jd2, /NULL)
>> 0 2 4 6 8
>> 10
>
>> alain.

When you calculate where(jd1[Value_Locate(jd1, jd2)] eq jd2), you can call w2, you get a vector w2 of indices into jd2 pointing on those values of jd2 - by definition of the Value_Locate function - which are equal to the lower bound of the intervals, into jd1, containing each of them.
Therefore, the days of interest are given by jd2[w2].
If you would like to get them from jd1, the simplest way is to reverse the statement:
w1 = where(jd2[Value_Locate(jd2, jd1)] eq jd1). This time, w1 is a vector of indices into jd1 and the days of interest are given by jd1[w1].
Of course, you will easily check that w1 and w2 vectors have same number of elements and that jd1[w1] is identical to jd2[w2].
alain.

Subject: Re: Matching elements in two arrays of different sizes
Posted by Meredith Pind on Thu, 28 Jun 2012 14:53:44 GMT
View Forum Message <> Reply to Message

Ok, that makes sense.

I set jd1 to the larger dataset and jd2 to the smaller. When I compute w2 and w1 i get w2 = long[16] and w1 = !NULL. What does that mean??

On Wednesday, June 27, 2012 4:28:24 PM UTC-5, alx wrote:
> On 27 juin, 18:48, pindsy <meredith.p...@gmail.com> wrote:
>> Hey Alain,
>>
>> thanks. I did this and got 16 matches
>> w = 0 1 2 3 4
10 11 12 13 14 15
16 17 18 19 20
>>
>> When I go and print off jd2(w) (smaller array) and jd1(w) (larger array) I get this
>>
>> IDL> print, jd1(w)
>> 2455771.2 2455771.2 2455771.2 2455771.2 2455771.2
2455771.2 2455771.2 2455771.2 2455771.2 2455771.2
2455771.2 2455771.2 2455771.2 2455771.2 2455771.2
2455771.2
>>
>> IDL> print, jd2(w)
>> 2455774.0 2455773.6 2455772.9 2455773.0 2455772.7
2455778.0 2455778.0 2455776.9 2455776.9 2455783.8
2455782.6 2455781.9 2455781.9 2455780.7 2455775.0
2455774.8
>>
>> Which aren't the same values. I'm not sure I understand this.
>>
>>
>>
>> On Wednesday, June 27, 2012 11:14:40 AM UTC-5, alx wrote:
>>> On 26 juin, 18:07, pindsy <meredith.p...@gmail.com> wrote:
>>>> Hi everyone,
>>
>>>> I am having trouble figuring out how to search through an array of size [1,17824] and match it to the points in another array of size [1,70].
>>
>>>> What I am trying to use to match the two arrays are the month, day, year, hour, and minute within each dataset. I have made two arrays using julday and have been trying to go through each of those arrays to find matches. However, it only searches up to the 70 first lines of the larger array and dosen't give all the possible matches.
>>
>>>> Any ideas on how to fix this would be helpful.

>>
>>>> Cheers,
>>
>>>> Meredith
>>
>>> Still use VALUE_LOCATE...
>>> if jd1 and jd2 are the two arrays of julian dates, w will be the
>>> vector of matching indices:
>>>   w = where(jd1[Value_Locate(jd1, jd2)] eq jd2, /NULL)
>>
>>> For instance:
>>> IDL> jd1 = julday(indgen(6)*2,1,2012) ;month 1st day, every two
>>> monthes
>>> IDL> jd2 = julday(indgen(12),1,2012)   ;1st day of each month
>>> IDL> print, where(jd1[Value_Locate(jd1, jd2)] eq jd2, /NULL)
>>>   0   2   4   6   8
>>> 10
>>
>>> alain.
>> On Wednesday, June 27, 2012 11:14:40 AM UTC-5, alx wrote:
>>> On 26 juin, 18:07, pindsy <meredith.p...@gmail.com> wrote:
>>>> Hi everyone,
>>
>>>> I am having trouble figuring out how to search through an array of size [1,17824] and match it to the points in another array of size [1,70].
>>
>>>> What I am trying to use to match the two arrays are the month, day, year, hour, and minute within each dataset. I have made two arrays using julday and have been trying to go through each of those arrays to find matches. However, it only searches up to the 70 first lines of the larger array and dosen't give all the possible matches.
>>
>>>> Any ideas on how to fix this would be helpful.
>>
>>>> Cheers,
>>
>>>> Meredith
>>
>>> Still use VALUE_LOCATE...
>>> if jd1 and jd2 are the two arrays of julian dates, w will be the
>>> vector of matching indices:
>>>   w = where(jd1[Value_Locate(jd1, jd2)] eq jd2, /NULL)
>>
>>> For instance:
>>> IDL> jd1 = julday(indgen(6)*2,1,2012) ;month 1st day, every two
>>> monthes
>>> IDL> jd2 = julday(indgen(12),1,2012)   ;1st day of each month
>>> IDL> print, where(jd1[Value_Locate(jd1, jd2)] eq jd2, /NULL)
>>>   0   2   4   6   8

>>> 10
>>
>>> alain.
>
> When you calculate where(jd1[Value_Locate(jd1, jd2)] eq jd2), you can
> call w2, you get a vector w2 of indices into jd2 pointing on those
> values of jd2 - by definition of the Value_Locate function - which are
> equal to the lower bound of the intervals, into jd1, containing each
> of them.
> Therefore, the days of interest are given by jd2[w2].
> If you would like to get them from jd1, the simplest way is to reverse
> the statement:
> w1 = where(jd2[Value_Locate(jd2, jd1)] eq jd1). This time, w1 is a
> vector of indices into jd1 and the days of interest are given by
> jd1[w1].
> Of course, you will easily check that w1 and w2 vectors have same
> number of elements and that jd1[w1] is identical to jd2[w2].
> alain.

Subject: Re: Matching elements in two arrays of different sizes
Posted by Meredith Pind on Thu, 28 Jun 2012 16:33:56 GMT
View Forum Message <> Reply to Message

Nevermind! Got it. Thanks for all the help!

On Thursday, June 28, 2012 9:53:44 AM UTC-5, pindsy wrote:
> Ok, that makes sense.
>
> I set jd1 to the larger dataset and jd2 to the smaller. When I compute w2 and w1 i get w2 = long[16] and w1 = !NULL. What does that mean??
>
> On Wednesday, June 27, 2012 4:28:24 PM UTC-5, alx wrote:
>> On 27 juin, 18:48, pindsy <meredith.p...@gmail.com> wrote:
>>> Hey Alain,
>>>
>>> thanks. I did this and got 16 matches
>>> w = 0 1 2 3 4
10 11 12 13 14 15
16 17 18 19 20
>>>
>>> When I go and print off jd2(w) (smaller array) and jd1(w) (larger array) I get this
>>>
>>> IDL> print, jd1(w)
>>> 2455771.2 2455771.2 2455771.2 2455771.2 2455771.2
2455771.2 2455771.2 2455771.2 2455771.2 2455771.2
2455771.2 2455771.2 2455771.2 2455771.2 2455771.2
2455771.2

>>>
>>> IDL> print, jd2(w)
>>> 2455774.0 2455773.6 2455772.9 2455773.0 2455772.7
2455778.0 2455778.0 2455776.9 2455776.9 2455783.8
2455782.6 2455781.9 2455781.9 2455780.7 2455775.0
2455774.8
>>>
>>> Which aren't the same values. I'm not sure I understand this.
>>>
>>>
>>>
>>> On Wednesday, June 27, 2012 11:14:40 AM UTC-5, alx wrote:
>>>> On 26 juin, 18:07, pindsy <meredith.p...@gmail.com> wrote:
>>>> > Hi everyone,
>>>
>>>> > I am having trouble figuring out how to search through an array of size [1,17824] and match it to the points in another array of size [1,70].
>>>
>>>> > What I am trying to use to match the two arrays are the month, day, year, hour, and minute within each dataset. I have made two arrays using julday and have been trying to go through each of those arrays to find matches. However, it only searches up to the 70 first lines of the larger array and dosen't give all the possible matches.
>>>
>>>> > Any ideas on how to fix this would be helpful.
>>>
>>>> > Cheers,
>>>
>>>> > Meredith
>>>
>>>> Still use VALUE_LOCATE...
>>>> if jd1 and jd2 are the two arrays of julian dates, w will be the
>>>> vector of matching indices:
>>>> w = where(jd1[Value_Locate(jd1, jd2)] eq jd2, /NULL)
>>>
>>>> For instance:
>>>> IDL> jd1 = julday(indgen(6)*2,1,2012) ;month 1st day, every two
>>>> monthes
>>>> IDL> jd2 = julday(indgen(12),1,2012) ;1st day of each month
>>>> IDL> print, where(jd1[Value_Locate(jd1, jd2)] eq jd2, /NULL)
>>>> 0 2 4 6 8
>>>> 10
>>>
>>>> alain.
>>> On Wednesday, June 27, 2012 11:14:40 AM UTC-5, alx wrote:
>>>> On 26 juin, 18:07, pindsy <meredith.p...@gmail.com> wrote:
>>>> > Hi everyone,
>>>
>>>> > I am having trouble figuring out how to search through an array of size [1,17824] and

match it to the points in another array of size [1,70].
>>>
>>>> > What I am trying to use to match the two arrays are the month, day, year, hour, and minute within each dataset. I have made two arrays using julday and have been trying to go through each of those arrays to find matches. However, it only searches up to the 70 first lines of the larger array and dosen't give all the possible matches.
>>>
>>>> > Any ideas on how to fix this would be helpful.
>>>
>>>> > Cheers,
>>>
>>>> > Meredith
>>>
>>>> Still use VALUE_LOCATE...
>>>> if jd1 and jd2 are the two arrays of julian dates, w will be the
>>>> vector of matching indices:
>>>>   w = where(jd1[Value_Locate(jd1, jd2)] eq jd2, /NULL)
>>>
>>>> For instance:
>>>> IDL> jd1 = julday(indgen(6)*2,1,2012) ;month 1st day, every two
>>>> monthes
>>>> IDL> jd2 = julday(indgen(12),1,2012)   ;1st day of each month
>>>> IDL> print, where(jd1[Value_Locate(jd1, jd2)] eq jd2, /NULL)
>>>>   0   2   4   6   8
>>>> 10
>>>
>>>> alain.
>>
>> When you calculate where(jd1[Value_Locate(jd1, jd2)] eq jd2), you can
>> call w2, you get a vector w2 of indices into jd2 pointing on those
>> values of jd2 - by definition of the Value_Locate function - which are
>> equal to the lower bound of the intervals, into jd1, containing each
>> of them.
>> Therefore, the days of interest are given by jd2[w2].
>> If you would like to get them from jd1, the simplest way is to reverse
>> the statement:
>> w1 = where(jd2[Value_Locate(jd2, jd1)] eq jd1). This time, w1 is a
>> vector of indices into jd1 and the days of interest are given by
>> jd1[w1].
>> Of course, you will easily check that w1 and w2 vectors have same
>> number of elements and that jd1[w1] is identical to jd2[w2].
>> alain.