Subject: Not where
Posted by bowman on Mon, 07 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

I often find myself using WHERE to divide an array into two parts. I do one operation on the first part and a different operation on the second part.

It would be nice to have an auxiliary array containing all the indices that are *not* returned by WHERE in i. For example, it would be nice to do

```
a = FLTARR(...)
i = WHERE((a...), count, NOT_WHERE = j, NOT_COUNT = not_count)
IF (count     GT 0L) THEN a[i] = ...
IF (not_count GT 0L) THEN a[j] = ...
```

Lacking the above changes to WHERE, can anyone suggest a fast and easy way to get j=NOT(i) ?

Thanks, Ken

--
Dr. Kenneth P. Bowman, Professor 409-862-4060
Department of Meteorology 409-862-4466 fax
Texas A&M University bowmanATcsrp.tamu.edu
College Station, TX 77843-3150 Replace AT with @

Subject: Re: Not where
Posted by bowman on Mon, 07 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

In article <bowman-0702000905130001@bowman.tamu.edu>, bowman@null.edu (Kenneth P. Bowman) wrote:

> It would be nice to have an auxiliary array containing all the indices
> that are *not* returned by WHERE in i.

Thanks for all the replies. Rather than appear to favor one, I decided to reply to my own posting. ;-)

The short answer seems to be no, there is no fast and easy way, although I tend to favor Liam's approach as being most similar to the existing WHERE function.

On another subject, I have been browsing through the "What's New in IDL 5.3" volume (thanks for earlier help from this group) and have discovered several improvements that have been proposed in this group (at least I thought they were good ideas!), although they are not necessarily easy to find.

The TOTAL function now includes a CUMULATIVE keyword. This makes it easy to convert histograms into cumulative histograms (among other things).

Speaking of not easy to find ;-), the new bisection search function is called VALUE_LOCATE. I do like the way it is implemented, though.`

Ken

Subject: Re: Not where
Posted by Martin Schultz on Mon, 07 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Alex Schuster wrote:
>
> David Fanning wrote:
>
>> Kenneth P. Bowman (bowman@null.edu) writes:
>>
>>> I often find myself using WHERE to divide an array into two parts. I do
>>> one operation on the first part and a different operation on the second
>>> part.
>>>
>>> It would be nice to have an auxiliary array containing all the indices
>>> that are *not* returned by WHERE in i. [...]
>
> Easy, yes, maybe not too fast, and not very elegant. Martin Schulz wrote
> (and probably posted) the routine INV_INDEX, which I attached.

I'm listening ;-)

> [...]
> This would add 1 to a[i] and subtract 5 from a[NOT(i)], for example. But
> I admit that
> j = inv_index(i)
> a[i] = a[i] + 1
> a[j] = a[j] - 5
> looks better. (Not cooler, but better.)

It might not work, though ;-(
1. You *must* supply the second parameter (totaln), e.g. as in

j = inv_index(i,n_elements(data))
2. I had to fix a bug which was that I used short ints instead of long.

So, please find the new version of this routine attached ...

Cheers,
Martin

--
[[[[[[[[[
[[Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie   [[
[[Bundesstr. 55, 20146 Hamburg   [[
[[phone: +49 40 41173-308   [[
[[fax: +49 40 41173-298   [[
[[martin.schultz@dkrz.de   [[
[[[[[[[[[

```
; $Id: inv_index.pro,v 1.11 1999/05/20 16:15:49 mgs Exp $
;------------------------------------------------------------ --
;+
; NAME:
;        INV_INDEX
;
; PURPOSE:
;        find the indices that do NOT match a WHERE condition
;
; CATEGORY:
;        array index handling
;
; CALLING SEQUENCE:
;        RESULT = INV_INDEX(INDEX,TOTALN)
;
; INPUTS:
;        INDEX : an index array, e.g. previously generated by a
;             WHERE command (may be -1)
;        TOTALN : the number of elements in the reference data
;             set, i.e. totaln = n_elements(index)+n_elements(result)
;
; KEYWORD PARAMETERS:
;
; OUTPUTS:
;        an integer array with all indices that were NOT in index
;        or -1 if index was complete
;
; SUBROUTINES:
;
; REQUIREMENTS:
;
```

```
; NOTES:
;       The function returns -1 if one of the following errors occurs:
;       - invalid number of arguments
;       - index variable is undefined
;       - totaln is less than n_elements(index)
;       - totaln less or equal 1, i.e. no associated data
;       The last error does not produce an error message, since this
;       feature was found to be very useful (in EXPLORE, the widget based
;       interactive data explorer)
;
; EXAMPLE:
;       data = findgen(50)
;       index = where(data ge 25)
;       invers = inv_index(index,n_elements(data))
;       print,invers
;
;       IDL prints numbers 0 through 24
;
; MODIFICATION HISTORY:
;       mgs, 10 May 1997: VERSION 1.00
;       mgs, 18 Aug 1997: - added template and check if n_elements(index) eq 0
;       mgs, 05 Apr 1999: - bug fix: needed to make sure result is type long
;
;-
; Copyright (C) 1997, Martin Schultz, Harvard University
; This software is provided as is without any warranty
; whatsoever. It may be freely used, copied or distributed
; for non-commercial purposes. This copyright notice must be
; kept with any copy of this software. If this software shall
; be used commercially or sold as part of a larger package,
; please contact the author to arrange payment.
; Bugs and comments should be directed to mgs@io.harvard.edu
; with subject "IDL routine inv_index"
 ;-------------------------------------------------------------- --


function inv_index,index,totaln

   newindex = -1L     ; default: nothing left

   ; check for errors:
   if (N_Params() ne 2) then begin
      print,'INV_INDEX: wrong number of arguments'
      return,newindex
   endif
   if (n_elements(index) eq 0) then begin
      print,'INV_INDEX: no valid index passed'
      return,newindex
```

```
  endif
  if (totaln lt n_elements(index)) then begin
    print,'INV_INDEX: totaln lt n_elements(index)'
    return,newindex
  endif
  if (totaln le 1) then return,newindex    ; no data there

  ; and handle the two situations:
  if (max(index) lt 0) then begin   ; no valid index passed
    newindex = lindgen(totaln)      ; create an integer array
    return,newindex                 ; with totaln elements
  endif

  ; else a valid indexarray was passed and we can construct the inverse
  newindex = lindgen(totaln)
  newindex(index) = -1
  i = where(newindex ge 0,count)
  if (count gt 0) then newindex = newindex(i)  $
  else newindex = -1L

  return, newindex
end
```

File Attachments

1) inv_index.pro, downloaded 114 times

Subject: Re: Not where
Posted by John-David T. Smith on Mon, 07 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Alex Schuster wrote:
>
> David Fanning wrote:
>
>> Kenneth P. Bowman (bowman@null.edu) writes:
>>
>>> I often find myself using WHERE to divide an array into two parts. I do
>>> one operation on the first part and a different operation on the second
>>> part.
>>>
>>> It would be nice to have an auxiliary array containing all the indices
>>> that are *not* returned by WHERE in i. For example, it would be nice to
>>> do
>>>
>>> a = FLTARR(...)
>>> i = WHERE((a...), count, NOT_WHERE = j, NOT_COUNT = not_count)
>>> IF (count GT 0L) THEN a[i] = ...

>>> IF (not_count GT 0L) THEN a[j] = ...
>>>
>>> Lacking the above changes to WHERE, can anyone suggest a fast and easy
>>> way to get j=NOT(i) ?
>
> i = where(a gt something)
> j = where(a le something)
>
> Easy, yes, maybe not too fast, and not very elegant. Martin Schulz wrote
> (and probably posted) the routine INV_INDEX, which I attached.
>
>> Now *here* is a place where Alex's matrix operations will
>> really pay off!
>>
>> I'll leave it to Alex to handle this question. :-)
>
> Hmm, now here I would rather use INV_INDEX instead...
>
> mask = where(a gt something)
> a = (a+1) * mask + (a-5) * (1B-mask)
>

Did you mean:

mask=a gt something

JD

--
J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|

Subject: Re: Not where
Posted by Craig Markwardt on Mon, 07 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

bowman@null.edu (Kenneth P. Bowman) writes:
> I often find myself using WHERE to divide an array into two parts. I do
> one operation on the first part and a different operation on the second
> part.
>
> It would be nice to have an auxiliary array containing all the indices
> that are *not* returned by WHERE in i. For example, it would be nice to
> do
> ...

I think I've asked for this before on this list, and I agree with you wholeheartedly. I don't believe there's anyway to simulate the desired feature in IDL without doing another operation.

The matrix approach doesn't always seem to be appropriate. (For example, working on very large arrays, one wants to avoid as many computations as possible. The matrix approach actually *compounds* the number of computations).

Craig

--

---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------

Subject: Re: Not where
Posted by Alex Schuster on Mon, 07 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:

> Kenneth P. Bowman (bowman@null.edu) writes:
>
>> I often find myself using WHERE to divide an array into two parts. I do
>> one operation on the first part and a different operation on the second
>> part.
>>
>> It would be nice to have an auxiliary array containing all the indices
>> that are *not* returned by WHERE in i. For example, it would be nice to
>> do
>>
>> a = FLTARR(...)
>> i = WHERE((a...), count, NOT_WHERE = j, NOT_COUNT = not_count)
>> IF (count GT 0L) THEN a[i] = ...
>> IF (not_count GT 0L) THEN a[j] = ...
>>
>> Lacking the above changes to WHERE, can anyone suggest a fast and easy
>> way to get j=NOT(i) ?

i = where(a gt something)
j = where(a le something)

Easy, yes, maybe not too fast, and not very elegant. Martin Schulz wrote (and probably posted) the routine INV_INDEX, which I attached.

> Now *here* is a place where Alex's matrix operations will
> really pay off!
>
> I'll leave it to Alex to handle this question. :-)

Hmm, now here I would rather use INV_INDEX instead...

mask = where(a gt something)
a = (a+1) * mask + (a-5) * (1B-mask)

This would add 1 to a[i] and subtract 5 from a[NOT(i)], for example. But
I admit that
 j = inv_index(i)
 a[i] = a[i] + 1
 a[j] = a[j] - 5
looks better. (Not cooler, but better.)

Alex
--
Alex Schuster Wonko@weird.cologne.de PGP Key available
alex@pet.mpin-koeln.mpg.de

```
;------------------------------------------------------------ --
;+
; NAME:
;       INV_INDEX
;
; PURPOSE:
;       find the indices that do NOT match a WHERE condition
;
; CATEGORY:
;       array index handling
;
; CALLING SEQUENCE:
;       RESULT = INV_INDEX(INDEX,TOTALN)
;
; INPUTS:
;       INDEX : an index array, e.g. previously generated by a
;            WHERE command (may be -1)
;       TOTALN : the number of elements in the reference data
;            set, i.e. totaln = n_elements(index)+n_elements(result)
;
; KEYWORD PARAMETERS:
;
; OUTPUTS:
;       an integer array with all indices that were NOT in index
;       or -1 if index was complete
```

```
;
; SUBROUTINES:
;
; REQUIREMENTS:
;
; NOTES:
;       The function returns -1 if one of the following errors occurs:
;       - invalid number of arguments
;       - index variable is undefined
;       - totaln is less than n_elements(index)
;       - totaln less or equal 1, i.e. no associated data
;       The last error does not produce an error message, since this
;       feature was found to be very useful (in EXPLORE, the widget based
;       interactive data explorer)
;
; EXAMPLE:
;       data = findgen(50)
;       index = where(data ge 25)
;       invers = inv_index(index,n_elements(data))
;       print,invers
;
;       IDL prints numbers 0 through 24
;
; MODIFICATION HISTORY:
;       mgs, 10 May 1997: VERSION 1.00
;       mgs, 18 Aug 1997: added template and check if n_elements(index) eq 0
;
;-
; Copyright (C) 1997, Martin Schultz, Harvard University
; This software is provided as is without any warranty
; whatsoever. It may be freely used, copied or distributed
; for non-commercial purposes. This copyright notice must be
; kept with any copy of this software. If this software shall
; be used commercially or sold as part of a larger package,
; please contact the author to arrange payment.
; Bugs and comments should be directed to mgs@io.harvard.edu
; with subject "IDL routine inv_index"
 ;-------------------------------------------------------------- --


function inv_index,index,totaln

   newindex = -1     ; default: nothing left

   ; check for errors:
   if (N_Params() ne 2) then begin
      print,'INV_INDEX: wrong number of arguments'
      return,newindex
```

```
  endif
  if (n_elements(index) eq 0) then begin
    print,'INV_INDEX: no valid index passed'
    return,newindex
  endif
  if (totaln lt n_elements(index)) then begin
    print,'INV_INDEX: totaln lt n_elements(index)'
    return,newindex
  endif
  if (totaln le 1) then return,newindex    ; no data there

  ; and handle the two situations:
  if (max(index) lt 0) then begin   ; no valid index passed
    newindex = indgen(totaln)      ; create an integer array
    return,newindex                ; with totaln elements
  endif

  ; else a valid indexarray was passed and we can construct the inverse
  newindex = indgen(totaln)
  newindex(index) = -1
  i = where(newindex ge 0,count)
  if (count gt 0) then newindex = newindex(i)  $
  else newindex = -1

  return, newindex
end
```

File Attachments

1) inv_index.pro, downloaded 114 times

Subject: Re: Not where
Posted by Alex Schuster on Tue, 08 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

"J.D. Smith" wrote:

> Alex Schuster wrote:

>> mask = where(a gt something)
>> a = (a+1) * mask + (a-5) * (1B-mask)
>
> Did you mean:
>
> mask=a gt something

Um. Well. Yes.

Alex
--
Alex Schuster Wonko@weird.cologne.de PGP Key available
alex@pet.mpin-koeln.mpg.de

Subject: Re: Not where
Posted by Paul Krummel on Mon, 21 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Sorry about that, upgraded my news reader and the default is to uuencode
attachments! Have now set it to not encode plain text. Thanks for pointing
it out.
Cheers Paul

"Mark Fardal" <fardal@weka.astro.umass.edu> wrote in message
news:7vg0uqz363.fsf@weka.phast.umass.edu...
> "Paul Krummel" <paul.krummel@dar.csiro.au> writes:

<snip>

>> begin 666 sets.pro
>> M.RL-"CL@3D%-13H-"CL)4T544PT*.PT*.R!3970@;W!E<F%T;W)S+B @56YI
>> M;VXL($EN=&5R<V5C=&EO;BP@86YD($1I9F9E<F5N8V4@*&DN92X@ <F5T=7)N
> etc...
>
>
> Looks very useful. But what exactly is the point in uuencoding
> ascii text? :->
>
> thanks,
> Mark Fardal
> UMass
>
