
Subject: Re: Complementary set.

Posted by [Martin Schultz](#) on Tue, 22 Sep 1998 07:00:00 GMT

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

Imanol Echave wrote:

>
> Hi people:
>
> I want to eliminate certain elements from an array. I've the index of the
> elements to be erased, and I want to obtain the "complementary" index (with the
> elements to maintain) to do `new_array=array[index]`. I can do this with a FOR
> loop, but... is it a better way?

Try out the attached INV_INDEX function. It is meant to serve exactly your needs.

Regards,
Martin.

--

Dr. Martin Schultz
Department for Earth&Planetary Sciences, Harvard University
109 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318
fax : (617)-495-4551

e-mail: mgs@io.harvard.edu
Internet-homepage: <http://www-as.harvard.edu/people/staff/mgs/>

```
;------ --  
;  
;+  
; 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 97 times

Subject: Re: Complementary set.
 Posted by [David Kastrop](#) on Thu, 24 Sep 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Imanol Echave <ccaeccai@sc.ehu.es> writes:

> Hi people:

>
> I want to eliminate certain elements from an array. I've the index of the
> elements to be erased, and I want to obtain the "complementary" index (with the
> elements to maintain) to do `new_array=array[index]`. I can do this with a FOR
> loop, but... is it a better way?

Assume that the array size is in `sz`, and the indices in `ind`. Assume that `ind` is non-empty. Then

```
invind = replicate(1,sz)
invind[ind] = 0
invind=where(invind)
```

does the trick in $O(n)$ time.

--

David Kastrup Phone: +49-234-700-5570
Email: dak@neuroinformatik.ruhr-uni-bochum.de Fax: +49-234-709-4209
Institut für Neuroinformatik, Universitätsstr. 150, 44780 Bochum, Germany
