Subject: Finding elements in Two Arrays (THE FUNCTION)
Posted by stl on Fri, 03 Jun 1994 13:36:09 GMT
View Forum Message <> Reply to Message

Hi, as I promised, here it is. This function can be used to
find the intersecting values, and more importantly their indecies.
The thanks should go to Dan Carr who posted most of this in an
interactive mode yesterday. I just tinkered with little things
like error checking and making it into a function. It seems to
run pretty DARN fast!

Let me know if there are any problems and or suggestions! This should
be a pretty useful function.

-stephen Strebel

```
;
;+
; NAME:
; where_array.pro
;
; PURPOSE:
; return the indecies of where vector B exists in vector A.
;  Basicly a where(B eq A) where B and A are 1 dimensional arrays.
;
;
; CATEGORY:
;       Array
;
;
; CALLING SEQUENCE:
;  result = where_array(A,B)
;
; INPUTS:
;  A vector the might contains elements of vector B
;  B vector the we would like to know which of its
;    elements exist in A
;
; OPTIONAL INPUTS:
;
; KEYWORD PARAMETERS:
;  Common_num array to be filled with the numbers common to
;     both arrays
;  Com_Dup  Array of all values that are alike.  This is
;     different from Common_num because values can be repeated.
;
; OUTPUTS:
;  Index into B of elements found in vector A.  If no
```

```
;  matches are found -1 is returned.  If the function is called
;  with incorrect arguments, a warning is displayed, and -2 is
;  returned (see side effects for more info)
;
; OPTIONAL OUTPUTS:
;
; COMMON BLOCKS:
;  None
;
; SIDE EFFECTS:
;  If the function is called incorrectly, a message is diplayed
;  to the screen, and the !ERR_STRING is set to the warning
;  message.  No error code is set, because the program returns
;  -2 already
;
; RESTRICTIONS:
;  This should be used with only Vectors.  Matricies other then
;  vectors will result in -2 being returned.  Also, A and B must
;  be defined
;
;
; PROCEDURE:
;
; EXAMPLE:
; idl> A=[2,1,3,5,3,8,2,5]
;   IDL> B=[3,4,2,8,7,8]
; IDL> result = where_array(a,b,COMMON_NUM=nums,COM_DUP=all_nums)
; IDL> print,result, nums,all_nums
;
; SEE ALSO:
; where
;
; MODIFICATION HISTORY:
;  Written by:  Dan Carr at RSI (command line version) 2/6/94
;    Stephen Strebel  3/6/94
;     made into a function, but really DAN did all
;     the thinking on this one!
;
;
;-
FUNCTION where_array,A,B,COMMON_NUM=common_num,COM_DUP=com_dup

; Check for: correct number of parameters
;    that A and B have each only 1 dimension
;    that A and B are defined
if (n_params() ne 2 or (size(A))(0) ne 1 or (size(B))(0) ne 1 $
 or n_elements(A) eq 0 or n_elements(B) eq 0) then begin
 message,'Inproper parameters',/Continue
```

```
 message,'Usage: result = where_array(A,B,[COMMON_NUM=com],[COM_DUP=dup]',/Continue
 return,-2
endif

; PART 1.  Find the numbers common to both sets.

set_length = Max([Max(A), Max(B)]) + 1L
set_1 = Bytarr(set_length)
set_2 = Bytarr(set_length)
set_1(A) = 1B
set_2(B) = 1B
common_num = Where(set_1 AND set_2)

if common_num(0) eq -1 then return,-1

; PART 2.  Find where the common numbers exist in array B.
;  This is the slick part!  (way to go Dan!)

index_arr1 = Replicate(1, N_Elements(B)) # common_num
index_arr2 = b # Replicate(1, N_Elements(common_num))
common_sub = Where(index_arr1 EQ index_arr2) MOD N_Elements(B)


; PART 3.  Get ALL the common numbers in array B (including duplicates)

com_dup = B(common_sub(SORT(common_sub)))

return,common_sub

END
```

--
Stephen C Strebel / SKI TO DIE
stl@maz.sma.ch / and
Swiss Meteorological Institute, Zuerich / LIVE TO TELL ABOUT IT
01 256 93 85 / (and pray for snow)