
Subject: Re: IDL Way to Remove Rows of an Array
Posted by [Foldy Lajos](#) on Tue, 10 May 2011 13:01:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Tue, 10 May 2011, David Fanning wrote:

```
> Folks,
>
> I have other things demanding my attention this morning,
> so I thought I would offer an IDL challenge. What is
> the IDL Way to remove arbitrary rows of an array?
>
> For example, suppose I have this array:
>
> array = Indgen(3,6)
> print, array
>
>   0   1   2
>   3   4   5
>   6   7   8
>   9  10  11
>  12  13  14
>  15  16  17
>
> What is the IDL Way to remove rows 2 and 5?
>
> subarray = ???
> print, subarray
>
>   0   1   2
>   6   7   8
>   9  10  11
>  15  16  17
```

```
subarray=array[*,[0,2,3,5]]
```

regards,
Lajos

Subject: Re: IDL Way to Remove Rows of an Array
Posted by [David Fanning](#) on Tue, 10 May 2011 13:42:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

FÖLDY Lajos writes:

```
>
> On Tue, 10 May 2011, David Fanning wrote:
```

```

>
>> Folks,
>>
>> I have other things demanding my attention this morning,
>> so I thought I would offer an IDL challenge. What is
>> the IDL Way to remove arbitrary rows of an array?
>>
>> For example, suppose I have this array:
>>
>> array = Indgen(3,6)
>> print, array
>>
>>   0   1   2
>>   3   4   5
>>   6   7   8
>>   9  10  11
>>  12  13  14
>>  15  16  17
>>
>> What is the IDL Way to remove rows 2 and 5?
>>
>> subarray = ???
>> print, subarray
>>
>>   0   1   2
>>   6   7   8
>>   9  10  11
>>  15  16  17
>
> subarray=array[*,[0,2,3,5]]

```

OK, here then is a function that can remove arbitrary rows from a 2D array:

```

;*****
;
FUNCTION RemoveRows, array, rows

; array -- A 2D array from which rows will be removed.
; rows -- A vector of row indices to remove from array.

; Need both positional parameters.
IF N_Params() NE 2 THEN BEGIN
    Print, "Usage: 'RemoveRows, array, rowsToRemove'"
    RETURN, -1
ENDIF

; The array must be at least 2D.
ndims = Size(array, /N_DIMENSIONS)

```

```
IF ndims NE 2 THEN BEGIN
    void = Dialog_Message('Array must be 2D.')
    Print, "Usage: 'RemoveRows, array, rowsToRemove'"
    RETURN, -1
ENDIF
```

```
; The rows must be a vector.
IF N_Elements(rows) EQ 1 THEN rows = [rows]
```

```
; Find the rows we are keeping.
dims = Size(array, /DIMENSIONS)
allrows = Lindgen(dims[1])
goodRows = SetDifference(allrows, rows)
```

```
; Return the subscripted array.
RETURN, array[*,goodRows]
```

END

```
.***** *
,
```

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: IDL Way to Remove Rows of an Array
Posted by [David Fanning](#) on Tue, 10 May 2011 13:48:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning writes:

Sorry, but these lines:

```
> ; The rows must be a vector.
> IF N_Elements(rows) EQ 1 THEN rows = [rows]
```

Should be written like this:

```
; The rows must be a vector.
IF Size(rows, /N_DIMENSIONS) EQ 0 THEN rows = [rows]
```

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: IDL Way to Remove Rows of an Array

Posted by [Gray](#) on Tue, 10 May 2011 19:37:40 GMT

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

On May 10, 9:48 am, David Fanning <n...@idlcoyote.com> wrote:

> David Fanning writes:

>

> Sorry, but these lines:

>

>> ; The rows must be a vector.

>> IF N_Elements(rows) EQ 1 THEN rows = [rows]

>

> Should be written like this:

>

> ; The rows must be a vector.

> IF Size(rows, /N_DIMENSIONS) EQ 0 THEN rows = [rows]

>

> Cheers,

>

> David

>

> --

> David Fanning, Ph.D.

> Fanning Software Consulting, Inc.

> Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>

> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

JD Smith would probably say that it's faster to construct your own index arrays, right? Maybe like this?

```
nrow = n_elements(goodrows)
```

```
cols = rebin(lindgen(dims[0]),[dims[0],nrow],/sample)
```

```
rows = rebin(transpose(goodrows),[dims[0],nrow],/sample)
```

```
return, array[cols,rows]
```

Subject: Re: IDL Way to Remove Rows of an Array
Posted by [JDS](#) on Tue, 17 May 2011 16:59:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

> JD Smith would probably say that it's faster to construct your own
> index arrays, right? Maybe like this?

Probably not in this case. Typically it's faster to roll your own index lists when the list you are constructing is smaller than the list IDL has to construct, for example if you are further sub-indexing it. The rule of thumb is to try not to make IDL do lots of work you will then promptly throw away.

In this case, your list over columns, and the list IDL would create using "*" are identical. So I'd probably simply go with '*'. If you have a 'reject' list instead of 'keep' (as in David's example) a closely related but somewhat simpler/faster method is:

```
d=size(array,/DIMENSIONS)
array=array[*,where(~histogram(reject,MIN=0,MAX=d[1]-1),/NUL L)]
```
