
Subject: Removing equal elements from an array
Posted by [Julio\[1\]](#) on Tue, 15 Aug 2006 14:46:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

Another question... please help me!!

I have an array 'A' with two columns, latitudes and longitudes, and several lines. A need to make another array with the elements of A that don't repeat. An example:

```
A[0]=[20.4, 40.3, 50.2, 50.2]
A[1]=[30.2, 60.2, 32.4, 32.4]
```

Note that the third and fourth pairs are the same (50.2, 32.4). So, I need to make another array and remove one of the pairs. So, I would have:

```
A[0]=[20.4, 40.3, 50.2]
A[1]=[30.2, 60.2, 32.4]
```

Do you have any idea how to do that??

Thanks!

Julio

Subject: Re: Removing equal elements from an array
Posted by [Julio\[1\]](#) on Wed, 16 Aug 2006 19:42:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

I'm using only integer values, so the Maarten's code and the Jean's tip solved my problem. But I agree with you JD, when using float values it won't work.

Thanks for all the comments!
Julio

JD Smith escreveu:

```
> On Wed, 16 Aug 2006 10:12:43 -0600, Jean H. wrote:
>
>> Hi,
>>
>> just sort your array based on the 2 fields...
>> you can do something like:
>> maxCol2 = max(a[1,*])
>> sortedIndices = sort([a[0,*]*maxCol2 + a[1,*]])
```

>> now do as you did before, but using a[0,sortedIndices] and
>> a[1,sortedIndices]
>
>
> This will only work in general for integer valued coordinates, but
> will get you into trouble with floating point. Note the following
> degeneracy, for maxCol2=180.:
>
> $.1 * 180 + 1 == .05 * 180 + 10.$
>
> thus, e.g., [1,1] and [.05,10.] would be considered the same
> coordinates in your method.
>
> The only possibility for arbitrary floats over some range is to cast
> them to integers using a useful precision, and then shifting one set
> of numbers clear of the other.
>
> JD

Subject: Re: Removing equal elements from an array
Posted by news.qwest.net on Thu, 17 Aug 2006 16:07:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Julio" <julio@cpa.unicamp.br> wrote in message
news:1155653196.084429.65000@74g2000cwt.googlegroups.com...
> Another question... please help me!!
>
> I have an array 'A' with two columns, latitudes and longitudes, and
> several lines. A need to make another array with the elements of A that
> don't repeat. An example:
>
> A[0]=[20.4, 40.3, 50.2, 50.2]
> A[1]=[30.2, 60.2, 32.4, 32.4]
>
> Note that the third and fourth pairs are the same (50.2, 32.4). So, I
> need to make another array and remove one of the pairs. So, I would
> have:
>
> A[0]=[20.4, 40.3, 50.2]
> A[1]=[30.2, 60.2, 32.4]
>
> Do you have any idea how to do that??
>
> Thanks!
>
> Julio

If I follow you correctly, you want to find unique pairs of numbers right?
How about combining the pairs into one number, and running uniq() on that?

For instance:

```
A = fltarr(2,4)
A[0,*]=[20.4, 40.3, 50.2, 50.2]
A[1,*]=[30.2, 60.2, 32.4, 32.4]
combo = A[0,*]*1000 + A[1,*]
indices = UNIQ(combo, SORT(combo))
```

{and of course, keep in mind the comments by JD and others about
using uniq on floats. You might want to change the lats and lon to integers
and then use uniq: (something like)
int_a = round(a*100)

Cheers,
bob

Subject: Re: Removing equal elements from an array
Posted by [JD Smith](#) on Thu, 17 Aug 2006 17:59:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thu, 17 Aug 2006 10:07:59 -0600, R.G. Stockwell wrote:

```
>
> "Julio" <julio@cpa.unicamp.br> wrote in message
> news:1155653196.084429.65000@74g2000cwt.googlegroups.com...
>> [quoted text muted]
>
> If I follow you correctly, you want to find unique pairs of numbers right?
> How about combining the pairs into one number, and running uniq() on that?
>
> For instance:
>
> A = fltarr(2,4)
> A[0,*]=[20.4, 40.3, 50.2, 50.2]
> A[1,*]=[30.2, 60.2, 32.4, 32.4]
> combo = A[0,*]*1000 + A[1,*]
> indices = UNIQ(combo, SORT(combo))
```

This method is so deceptively simple that it really seems like it should work, and sometimes it does, but it doesn't, in general, for floating point numbers, even if they fall in the proper range. I gave a similar example before, but it's worth repeating:

[.2,10] => combo=210
[.1,110] => combo=210
[.15,60] => combo=210

or what about:

[5.5,5.5] => combo= 5505.5
[5.2,305.] => combo= 5505.5

The only sure fire way to map a given pair of numbers over a finite range to a single unique combination, is to scale them to integers, then offset one set entirely from the other (64bit integers are nice for this). For small sets of numbers you might not run into this form of collision, but for many numbers, it's quite likely you would.

JD

Subject: Re: Removing equal elements from an array
Posted by news.qwest.net on Thu, 17 Aug 2006 20:06:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

"JD Smith" <jdsmith@as.arizona.edu> wrote in message
news:pan.2006.08.17.17.59.05.354360@as.arizona.edu...

...

> This method is so deceptively simple that it really seems like it should
> work, and sometimes it does, but it doesn't, in general, for floating
> point numbers

...

right, i agree completely
(that is what I was trying to say in my last paragraph
with the suggestion to do something like using round(a*100) if appropriate).

Cheers,
bob

Subject: Re: Removing equal elements from an array
Posted by news.qwest.net on Thu, 17 Aug 2006 20:32:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

"JD Smith" <jdsmith@as.arizona.edu> wrote in message
news:pan.2006.08.17.17.59.05.354360@as.arizona.edu...

...

> [5.5,5.5] => combo= 5505.5

```
> [5.2,305.] => combo= 5505.5
```

Good point, but I thought it was obvious to choose the correct multiplication factor such that the 2 numbers will not overlap. Six digits of lat or lon will give 10 meter resolution, which is probably good enough for many geophysical applications. So, to be more precise:

```
; make data array
A = fltarr(2,9)
A[0,*]=[20.4, 40.3, 50.2, 50.2, 5.5,5.2, .2,.1,.15]
A[1,*]=[30.2, 60.2, 32.4, 32.4, 5.5, 302,10,110,60]
original_a = a

factor = 10L^6L ; 6 digits for each value
intfactor = 1000 ; take 3 decimal places, 10m resolution
a = round(a * intfactor,/164)
combo = A[0,*]*factor + A[1,*]
result = UNIQ(combo) ; the indices

; output results
print
print, 'result'
for i = 0, n_elements(result)-1 do begin
  print,original_a[*,result[i]]
endfor

end
```

Subject: Re: Removing equal elements from an array
Posted by [JD Smith](#) on Thu, 17 Aug 2006 21:21:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thu, 17 Aug 2006 14:32:30 -0600, R.G. Stockwell wrote:

```
>
> "JD Smith" <jdsmith@as.arizona.edu> wrote in message
> news:pan.2006.08.17.17.59.05.354360@as.arizona.edu...
> ...
>
>> [5.5,5.5] => combo= 5505.5
>> [5.2,305.] => combo= 5505.5
>
> Good point, but I thought it was obvious to choose the correct
> multiplication
> factor such that the 2 numbers will not overlap. Six digits of lat
```

- > or lon will give 10 meter resolution, which is probably good enough
- > for many geophysical applications.

It's not just a matter of choosing the multiplication factor, it's actually truncating or rounding to integer values, to keep the "low digits" and "high digits" from polluting each other's waters.

- > So, to be more precise:
- >
- > ; make data array
- > A = fltarr(2,9)
- > A[0,*]=[20.4, 40.3, 50.2, 50.2, 5.5,5.2, .2,.1,.15]
- > A[1,*]=[30.2, 60.2, 32.4, 32.4, 5.5, 302,10,110,60]
- > original_a = a
- >
- > factor = 10L^6L ; 6 digits for each value
- > intfactor = 1000 ; take 3 decimal places, 10m resolution
- > a = round(a * intfactor,/l64)
- > combo = A[0,*]*factor + A[1,*]
- > result = UNIQ(combo) ; the indices

That's a similar "cast to integers with a given precision" approach as I recommend, only using decimal digits instead of binary bits. Just wanted to make clear that it's dangerous with full precision floating points in general (a point I hadn't appreciated until playing with it a bit).

JD

Subject: Re: Removing equal elements from an array
Posted by [R.Bauer](#) on Thu, 14 Sep 2006 06:32:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

Julio wrote:

- > Another question... please help me!!
- >
- > I have an array 'A' with two columns, latitudes and longitudes, and
- > several lines. A need to make another array with the elements of A that
- > don't repeat. An example:
- >
- > A[0]=[20.4, 40.3, 50.2, 50.2]
- > A[1]=[30.2, 60.2, 32.4, 32.4]
- >
- > Note that the third and fourth pairs are the same (50.2, 32.4). So, I
- > need to make another array and remove one of the pairs. So, I would
- > have:
- >

```
> A[0]=[20.4, 40.3, 50.2]
> A[1]=[30.2, 60.2, 32.4]
>
> Do you have any idea how to do that??
>
> Thanks!
>
> Julio
>
```

```
a=[20.4, 40.3, 50.2, 50.2]
ix=uniq(a)
b=[30.2, 60.2, 32.4, 32.4]
print,a[ix],b[ix]
```

cheers
Reimar

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de

a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro.html
=====
