
Subject: Re: Removing equal elements from an array
Posted by [Rick Towler](#) on Tue, 15 Aug 2006 16:30:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

I'm sure there is a built in function that I am unaware of but here is one way:

```
b = a - shift(a,2)
c = a[* , where(b[1,*] ne 0)]
```

```
IDL> print, a
    20.4000    30.2000
    40.3000    60.2000
    50.2000    32.4000
    50.2000    32.4000
IDL> b = a - shift(a,2)
IDL> print, b
   -29.8000   -2.20000
    19.9000    30.0000
    9.90000   -27.8000
    0.000000    0.000000
IDL> c = a[* , where(b[1,*] ne 0)]
IDL> print, c
    20.4000    30.2000
    40.3000    60.2000
    50.2000    32.4000
```

-Rick

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
>

Subject: Re: Removing equal elements from an array
Posted by [Julio\[1\]](#) on Tue, 15 Aug 2006 17:36:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ok Rick, it worked!

Thanks,
Julio

Rick Towler escreveu:

> I'm sure there is a built in function that I am unaware of but here is
> one way:
>
> b = a - shift(a,2)
> c = a[*, where(b[1,*] ne 0)]
>
> IDL> print, a
> 20.4000 30.2000
> 40.3000 60.2000
> 50.2000 32.4000
> 50.2000 32.4000
> IDL> b = a - shift(a,2)
> IDL> print, b
> -29.8000 -2.20000
> 19.9000 30.0000
> 9.90000 -27.8000
> 0.000000 0.000000
> IDL> c = a[*, where(b[1,*] ne 0)]
> IDL> print, c
> 20.4000 30.2000
> 40.3000 60.2000
> 50.2000 32.4000
>
> -Rick
>
> 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
>>
```

Subject: Re: Removing equal elements from an array
Posted by [Jean H.](#) on Tue, 15 Aug 2006 19:14:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Rick Towler wrote:

```
> I'm sure there is a built in function that I am unaware of but here is
> one way:
>
> b = a - shift(a,2)
> c = a[*, where(b[1,*] ne 0)]
>
> IDL> print, a
>    20.4000    30.2000
>    40.3000    60.2000
>    50.2000    32.4000
>    50.2000    32.4000
> IDL> b = a - shift(a,2)
> IDL> print, b
>   -29.8000   -2.20000
>    19.9000    30.0000
>     9.9000   -27.8000
>    0.000000    0.000000
> IDL> c = a[*, where(b[1,*] ne 0)]
```

this seems to be incomplete...

you can have a value repeted in a column but not in another...

So you would have to get the indices on the 1st col and on the 2nd col.

```
IDL> a = transpose([[1,2,3,3,3],[5,5,4,4,1]])
```

```
IDL> print,a
  1   5
  2   5
  3   4
  3   4
  3   1
IDL> b = a - shift(a,2)
IDL> c = a[* , where(b[1,*] ne 0)]
IDL> print,c
  1   5
  3   4
  3   1
```

==>> 2;5 is missing!

so you should do

```
c = a[* , where(b[0,*] ne 0 or b[1,*] ne 0)]
```

```
IDL> print,c
  1   5
  2   5
  3   4
  3   1
```

we now have all entries..

Jean

```
> IDL> print, c
>    20.4000    30.2000
>    40.3000    60.2000
>    50.2000    32.4000
>
> -Rick
>
> 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
>>
```

Subject: Re: Removing equal elements from an array
Posted by [Mike\[2\]](#) on Tue, 15 Aug 2006 22:40:16 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.

Take a look at the uniq function. Here's an example:

```
IDL> a=[1,2,3,4,5,6,7,7,8,8,8]
IDL> print, a
      1      2      3      4      5      6      7      7
8      8      8
IDL> print, a[uniq(a)]
      1      2      3      4      5      6      7      8
```

Using your data:

```
IDL> A = [[20.4, 40.3, 50.2, 50.2], [30.2, 60.2, 32.4, 32.4]]
IDL> i = uniq(A[:,0])
IDL> B = [[(A[:,0])[i]], [(A[:,1])[i]]]
IDL>
IDL> print, A
      20.4000      40.3000      50.2000      50.2000
      30.2000      60.2000      32.4000      32.4000
IDL> print, B
      20.4000      40.3000      50.2000
      30.2000      60.2000      32.4000
```

Mike

Subject: Re: Removing equal elements from an array
Posted by [Julio\[1\]](#) on Wed, 16 Aug 2006 12:20:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ok guys... Excellent tips, problem solved!

Mike escreveu:

```
> 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.
>
> Take a look at the uniq function. Here's an example:
>
> IDL> a=[1,2,3,4,5,6,7,7,8,8,8]
> IDL> print, a
>   1   2   3   4   5   6   7   7
>  8   8   8
> IDL> print, a[uniq(a)]
>   1   2   3   4   5   6   7   8
>
> Using your data:
>
> IDL> A = [[20.4, 40.3, 50.2, 50.2], [30.2, 60.2, 32.4, 32.4]]
> IDL> i = uniq(A[:,0])
> IDL> B = [[(A[:,0])[i]], [(A[:,1])[i]]]
> IDL>
> IDL> print, A
>   20.4000   40.3000   50.2000   50.2000
>   30.2000   60.2000   32.4000   32.4000
> IDL> print, B
>   20.4000   40.3000   50.2000
>   30.2000   60.2000   32.4000
>
> Mike
```

Subject: Re: Removing equal elements from an array
Posted by [Maarten\[1\]](#) on Wed, 16 Aug 2006 12:55:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mike wrote:

```
> Julio wrote:
>> 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.
>
> Take a look at the uniq function. Here's an example:
```

[snip]

Which still doesn't take into account the following situation:

```
A = [[20.4, 40.3, 50.2, 50.2], $
      [30.2, 60.2, 32.4, 32.5]]
```

in which case no items should be removed. Just thinking out aloud here.

With:

```
lat = A[:,0] & lon = A[:,1]
```

we have

```
idx_lat = uniq(lat) & idx_lon = uniq(lon)
```

At the very least both index arrays should be the same, if you want to apply this automagically.

If the precision of the coordinates is limited, you can try to combine the lat and lon in a single number. If the coordinates are floats, the following ought to work:

```
l = lat + (2.0D0^23)*lon
idx = uniq(l)
lat = lat[idx] & lon = lon[idx]
```

Maarten

Subject: Re: Removing equal elements from an array
Posted by [Julio\[1\]](#) on Wed, 16 Aug 2006 15:24:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Maarten,

I used your code to remove equal elements from an array. It worked fine for a small array. But I tested using a greater amount of points and some equal elements (pairs of coords) remains. There are 434 pairs... an example of them:

```
234.000    208.000
228.000    208.000
234.000    208.000
234.000    208.000
```

```
178.000    209.000
....
153.000    314.000
146.000    318.000
181.000    318.000
```

The pair (234.000, 208.000) repeats 3 times, so 2 pairs should be removed. In the output array for these 434 input pairs I have:

```
234.000    208.000
228.000    208.000
234.000    208.000
178.000    209.000
... and so on
```

We see the pair (234.000, 208.000) repeats 2 times! Do you have any idea about what is going on??

Julio

Maarten escreveu:

```
> Mike wrote:
>> Julio wrote:
>>> 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.
>>
>> Take a look at the uniq function. Here's an example:
>
> [snip]
>
> Which still doesn't take into account the following situation:
>
> A = [[20.4, 40.3, 50.2, 50.2], $
>       [30.2, 60.2, 32.4, 32.5]]
>
> in which case no items should be removed. Just thinking out aloud here.
> With:
> lat = A[:,0] & lon = A[:,1]
>
> we have
> idx_lat = uniq(lat) & idx_lon = uniq(lon)
>
> At the very least both index arrays should be the same, if you want to
> apply this automagically.
>
> If the precision of the coordinates is limited, you can try to combine
```

> the lat and lon in a single number. If the coordinates are floats, the
> following ought to work:
>
> I = lat + (2.0D0^23)*lon
> idx = uniq(I)
> lat = lat[idx] & lon = lon[idx]
>
> Maarten

Subject: Re: Removing equal elements from an array
Posted by [Jean H.](#) on Wed, 16 Aug 2006 16:12:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

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]

Jean

Julio wrote:

> Dear Maarten,
>
> I used your code to remove equal elements from an array. It worked fine
> for a small array. But I tested using a greater amount of points and
> some equal elements (pairs of coords) remains. There are 434 pairs...
> an example of them:
>
> 234.000 208.000
> 228.000 208.000
> 234.000 208.000
> 234.000 208.000
> 178.000 209.000
>
> 153.000 314.000
> 146.000 318.000
> 181.000 318.000
>
> The pair (234.000, 208.000) repeats 3 times, so 2 pairs should be
> removed. In the output array for these 434 input pairs I have:
>
> 234.000 208.000
> 228.000 208.000

```

> 234.000    208.000
> 178.000    209.000
> ... and so on
>
> We see the pair (234.000, 208.000) repeats 2 times! Do you have any
> idea about what is going on??
>
> Julio
>
> Maarten escreveu:
>
>
>> Mike wrote:
>>
>>> Julio wrote:
>>>
>>>> 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.
>>>
>>> Take a look at the uniq function. Here's an example:
>>
>> [snip]
>>
>> Which still doesn't take into account the following situation:
>>
>> A = [[20.4, 40.3, 50.2, 50.2], $
>>      [30.2, 60.2, 32.4, 32.5]]
>>
>> in which case no items should be removed. Just thinking out aloud here.
>> With:
>> lat = A[:,0] & lon = A[:,1]
>>
>> we have
>> idx_lat = uniq(lat) & idx_lon = uniq(lon)
>>
>> At the very least both index arrays should be the same, if you want to
>> apply this automagically.
>>
>> If the precision of the coordinates is limited, you can try to combine
>> the lat and lon in a single number. If the coordinates are floats, the
>> following ought to work:
>>
>> I = lat + (2.0D0^23)*lon
>> idx = uniq(I)
>> lat = lat[idx] & lon = lon[idx]
>>
>> Maarten

```

>
>

Subject: Re: Removing equal elements from an array
Posted by [Julio\[1\]](#) on Wed, 16 Aug 2006 16:54:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ok... :-) Now it really worked.

Jean H. escreveu:

> 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]
>
> Jean
>
> Julio wrote:
>> Dear Maarten,
>>
>> I used your code to remove equal elements from an array. It worked fine
>> for a small array. But I tested using a greater amount of points and
>> some equal elements (pairs of coords) remains. There are 434 pairs...
>> an example of them:
>>
>> 234.000 208.000
>> 228.000 208.000
>> 234.000 208.000
>> 234.000 208.000
>> 178.000 209.000
>>
>> 153.000 314.000
>> 146.000 318.000
>> 181.000 318.000
>>
>> The pair (234.000, 208.000) repeats 3 times, so 2 pairs should be
>> removed. In the output array for these 434 input pairs I have:
>>
>> 234.000 208.000
>> 228.000 208.000
>> 234.000 208.000

```

>> 178.000    209.000
>> ... and so on
>>
>> We see the pair (234.000, 208.000) repeats 2 times! Do you have any
>> idea about what is going on??
>>
>> Julio
>>
>> Maarten escreveu:
>>
>>
>>> Mike wrote:
>>>
>>>> Julio wrote:
>>>>
>>>> >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.
>>>>
>>>> Take a look at the uniq function. Here's an example:
>>>
>>> [snip]
>>>
>>> Which still doesn't take into account the following situation:
>>>
>>> A = [[20.4, 40.3, 50.2, 50.2], $
>>>      [30.2, 60.2, 32.4, 32.5]]
>>>
>>> in which case no items should be removed. Just thinking out aloud here.
>>> With:
>>> lat = A[:,0] & lon = A[:,1]
>>>
>>> we have
>>> idx_lat = uniq(lat) & idx_lon = uniq(lon)
>>>
>>> At the very least both index arrays should be the same, if you want to
>>> apply this automagically.
>>>
>>> If the precision of the coordinates is limited, you can try to combine
>>> the lat and lon in a single number. If the coordinates are floats, the
>>> following ought to work:
>>>
>>> l = lat + (2.0D0^23)*lon
>>> idx = uniq(l)
>>> lat = lat[idx] & lon = lon[idx]
>>>
>>> Maarten
>>

```

>>

Subject: Re: Removing equal elements from an array
Posted by [JD Smith](#) on Wed, 16 Aug 2006 17:29:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wed, 16 Aug 2006 08:24:31 -0700, Julio wrote:

> Dear Maarten,
>
> I used your code to remove equal elements from an array. It worked fine
> for a small array. But I tested using a greater amount of points and
> some equal elements (pairs of coords) remains. There are 434 pairs...
> an example of them:
>
> 234.000 208.000
> 228.000 208.000
> 234.000 208.000
> 234.000 208.000
> 178.000 209.000
>
> 153.000 314.000
> 146.000 318.000
> 181.000 318.000
>
> The pair (234.000, 208.000) repeats 3 times, so 2 pairs should be
> removed. In the output array for these 434 input pairs I have:
>
> 234.000 208.000
> 228.000 208.000
> 234.000 208.000
> 178.000 209.000
> ... and so on
>
> We see the pair (234.000, 208.000) repeats 2 times! Do you have any
> idea about what is going on??

It's almost always a bad idea to rely on two floating numbers being precisely equal (as UNIQ does). See http://www.dfanning.com/math_tips/sky_is_falling.html. A better method is to test if they differ by less than some small number, epsilon.

Sadly, IDL's SORT isn't very flexible, since it only works on a single vector at a time, and you can't specify a generic sorting function. If your vectors cover a small range, and densely fill it, you can use HIST_2D or HIST_ND to bin them, taking the populated bin centers as

your unique set, but this will become awkward for small bin sizes or very widely spaced data.

Maarten's code needs to add sort:

```
idx=uniq(l,sort(l))
```

but will still suffer from "almost equal" issues w.r.t round-off.

In this case, you know the (small) maximum range of your variables: lon from 0-360, lat from -90 to 90 (I'm guessing; if not, it's easy to generalize). It's therefore straightforward to create a single lat_lon unsigned long long integer index which uniquely encodes the latitude and longitude, and allows comparing coordinates to some precision epsilon.

```
epsilon=1.e-7 ; difference in degrees for equality  
lat_lon = ulong64((lat+90.)/epsilon) + ishft(ulong64(lon/epsilon),32)
```

As long as the "maximum value/epsilon" of either variable does not overflow a 32 bit unsigned integer (e.g. 4294967295 or less), you'll have a nice unique index. You may or may not actually want to use epsilon=1.e-7; your data might be binned and truncated such that, e.g. 1.e-3 or 1.e-4 is more appropriate. If comparing values to a precision of 0.01 degrees is good enough, you can even squeeze lat_lon into a normal 32 bit integer for some speedup (except on 64-bit systems), since $360/.01 < 2^{16}$. In any case, this should work:

```
u=uniq(lat_lon,sort(lat_lon))  
lat=lat[u] & lon=lon[u]
```

If you have three or more variables with similar range you want to compare to some small epsilon, you should first understand MACHAR's output, and especially that an absolute precision of, e.g., .01 is not available at all floating point values, but in any case, you'll need something else. Similarly if you have two floating-point numbers that vary over a wide range.

JD

Subject: Re: Removing equal elements from an array
Posted by [JD Smith](#) on Wed, 16 Aug 2006 17:40:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

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
