
Subject: Count duplicate elements in an array but keep their order!

Posted by [Steve Daal](#) on Fri, 30 Aug 2013 23:23:09 GMT

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

Hi,

I have a long array that looks like:

```
a =  
-120.09796  
-120.09796  
-108.18972  
-108.18972  
-108.18972  
-121.98  
-95.23  
-95.23  
-95.23  
-95.23  
-95.23  
-95.23  
-95.23  
-95.23  
-95.23  
-95.23  
-95.23  
-95.23  
-95.23  
-95.03  
-95.03  
-95.03  
-95.03  
-71.303756  
-122.39084  
-122.39084  
-122.39084  
-122.24503]
```

with many more elements inside. I need to create two new arrays with the consecutive counts corresponding to the values of the first array after removing the duplicates, for instance, the above array would give me:

```
a_new = [-120.09796, -108.18972, -121.98, -95.23, -71.303756, -122.24503]
```

and

```
b_new = [2, 3, 1, 13, 1, 4]
```

where, elements "-120.09796" existed twice in the original array and so forth.

Any help is much appreciated.

Subject: Re: Count duplicate elements in an array but keep their order!

Posted by [Josh Sixsmith](#) on Sat, 31 Aug 2013 15:03:44 GMT

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

The following might work.

Using the UNIQ function will give the indices, which you can use to get the value counts.

To get the counts for each value you'll then need to use the SHIFT function to offset the indices in order subtract from the original indices.

Assuming that a is already sorted, other wise use SORT(a) first

```
un = UNIQ(a)
```

In order to get the proper count for the first value, the unique array will need to be extended by 1 element.

```
un_extra = intarr(n_elements(un)+1)
un_extra[1:n_elements(un_extra)-1] = un
```

Add 1 to the array to account for the fact that indices start at 0

```
un_extra += 1
un_offset = SHIFT(un_extra, -1)
```

```
counts = un_offset - un_extra
```

Remove the last element in counts

```
counts = count[0:n_elements(counts)-2]
values = a[un]
```

Hope that helps ,if it works :)

Cheers

Josh

Subject: Re: Count duplicate elements in an array but keep their order!

Posted by [David Fanning](#) on Tue, 03 Sep 2013 12:58:33 GMT

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

Josh Sixsmith writes:

> The following might work.

> Using the UNIQ function will give the indices, which you can use to get the value counts.

> To get the counts for each value you'll then need to use the SHIFT function to offset the indices in order subtract from the original indices.

Using the UNIQ function on a floating point array is probably a bad idea. You might find that ALL your floating point numbers are unique, depending upon how you arrive at them.

I think to solve this problem you are going to have to convert the floats to integers of whatever degree of "floating" accuracy makes sense for the problem. Then work with the integers to find, count, and remove duplicates.

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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

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

Subject: Re: Count duplicate elements in an array but keep their order!

Posted by [Steve Daal](#) on Mon, 16 Sep 2013 17:51:58 GMT

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

On Tuesday, September 3, 2013 7:58:33 AM UTC-5, David Fanning wrote:

> Josh Sixsmith writes:

>

>

>

>> The following might work.

>

>> Using the UNIQ function will give the indices, which you can use to get the value counts.

>

>> To get the counts for each value you'll then need to use the SHIFT function to offset the indices in order subtract from the original indices.

>

>

>

> Using the UNIQ function on a floating point array is probably a bad

>

> idea. You might find that ALL your floating point numbers are unique,

>

> depending upon how you arrive at them.

>

>

>

> I think to solve this problem you are going to have to convert the

>

> floats to integers of whatever degree of "floating" accuracy makes sense

>

> for the problem. Then work with the integers to find, count, and remove

>

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

Thanks Josh and David. That helped a lot.
