

---

Subject: Re: Multi-dimensions without for loop ?  
Posted by [btt](#) on Thu, 25 Mar 2004 14:09:06 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Emmanuel Christophe wrote:

> Hi,  
>  
> I'm trying to optimize some IDL code, removing for loops.  
>  
> I'm using the mean function to get the average of each line (this is for  
> the example, could be another function). From a 2 dimensional array, I  
> want to remove the mean of each column.  
>  
> Here is a sample using a loop:  
> -----  
> for j=0,size-1 do begin  
> vect=data[j,\*]  
> datac[j,\*]=vect-mean(vect)  
> endfor  
> -----  
>  
> How to do it in one instruction: if i'm using something like  
> 'mean(data)', i'll get the average for the whole array, and not line by  
> line.  
>  
> the instruction 'total' give me something similar to what I want:  
> 'total(data,1)' will make the sum in only one direction.  
>  
> How to get that with ordinary function ? and is it possible ? I'm  
> thinking of something like 'data[0:size-1,\*]'...  
>

Hello,

You can do it without a loop, but unfortunately not with the built-in statistics routines.

```
IDL> data = Findgen(4,6)
IDL> colTotal = TOTAL(data,2)
IDL> dim = Size(data, /dim)
IDL> colMean = Rebin(colTotal/dim[1], dim)
IDL> newData = data-colMean
```

But you could write your own routine to add the neat dimension handling to MEAN, MOMENT, etc that you already get with TOTAL, MAX, etc.

Ben

---

---

Subject: Re: Multi-dimensions without for loop ?

Posted by [Andrea Pitacco](#) on Thu, 25 Mar 2004 14:22:12 GMT

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

---

On Thu, 25 Mar 2004 11:47:31 +0100, Emmanuel Christophe  
<melaneum555@yahoo.fr> wrote:

```
> Hi,  
>  
> I'm trying to optimize some IDL code, removing for loops.  
>  
> I'm using the mean function to get the average of each line (this is for  
> the example, could be another function). From a 2 dimensional array, I  
> want to remove the mean of each column.  
>  
> Here is a sample using a loop:  
> -----  
> for j=0,size-1 do begin  
>   vect=data[j,*]  
>   datac[j,*]=vect-mean(vect)  
> endfor  
> -----  
>  
> How to do it in one instruction: if i'm using something like  
> 'mean(data)', i'll get the average for the whole array, and not line by  
> line.
```

Hi Emmanuel,

I have not understood clearly if you like to have the column mean  
being removed from the data column, or the row (line?) mean removed  
from the data row. 'Seems more likely the first. Anyway:

In the first case you may want to try:

```
datac = data - Total(data,2) / (Size(data))[2] #  
Replicate(1D,(Size(data))[2])
```

In the second:

```
datac = data - Total(data,1) / (Size(data))[1] ##  
Replicate(1D,(Size(data))[1])
```

Regards, Andrea

Andrea Pitacco  
University of Padova, Italy

---

---

Subject: Re: Multi-dimensions without for loop ?

Posted by [Emmanuel Christophe](#) on Fri, 26 Mar 2004 15:01:07 GMT

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

---

ok, I guess my example using 'mean' was not well choosen. I understand how to make it using 'total' now and it's working because this is a sum.

But in this case:

Let define a simple array (dimension 2x2 to explain, but i'd like to make it work on bigger array):

```
|a00 a10|  
|a01 a11|
```

Is it possible to get in one step without using 'for' loops

dimension (1x2)

```
| a00.a10 |  
| a01.a11 |
```

or

```
| max([a00,a10]) |  
| max([a01,a11]) |
```

Thanks,  
Emmanuel

---

Subject: Re: Multi-dimensions without for loop ?

Posted by [JD Smith](#) on Fri, 26 Mar 2004 23:36:22 GMT

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

---

On Fri, 26 Mar 2004 16:01:07 +0100, Emmanuel Christophe wrote:

> ok, I guess my example using 'mean' was not well choosen. I understand  
> how to make it using 'total' now and it's working because this is a sum.

>

> But in this case:

>

> Let define a simple array (dimension 2x2 to explain, but i'd like to  
> make it work on bigger array):

```
> |a00 a10|  
> |a01 a11|
```

>

> Is it possible to get in one step without using 'for' loops

>

```
> dimension (1x2)
>
> | a00.a10 |
> | a01.a11 |
```

```
p=product(a,1)
```

```
> or
>
> | max([a00,a10]) |
> | max([a01,a11]) |
>
```

```
m=max(a,DIMENSION=1)
```

A pleasingly IDL-esque lack of symmetry there. Probably requires IDL>5.6.

JD

---