
Subject: Re: Replicate array over columns - using transpose?

Posted by [Lajos Foldy](#) on Thu, 22 Mar 2012 10:50:39 GMT

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

On Mar 22, 11:11 am, Robin Wilson <ro...@rtwilson.com> wrote:

> Hi all,

>

> I've got an array, a, as follows:

>

> IDL> print, a

> 10.0000 35.0000 60.0000 85.0000 110.000

>

> I want to replicate this across 4 columns to produce an array like the

> following:

>

> 10 10 10 10

> 35 35 35 35

> 60 60 60 60

> 85 85 85 85

> 110 110 110 110

>

> I'm using the CMREPLICATE routine to do this, and get the following:

>

> IDL> print, CMREPLICATE(a, 4)

> 10.0000 35.0000 60.0000 85.0000 110.000

> 10.0000 35.0000 60.0000 85.0000 110.000

> 10.0000 35.0000 60.0000 85.0000 110.000

> 10.0000 35.0000 60.0000 85.0000 110.000

>

> However, I want to replicate it in the columns as in the example above.

> As far as I can see it there are two ways to deal with this: transpose

> the array after I've done it (computationally expensive, as I'll be

> doing it with far larger arrays in the real program) or transpose it

> before running CMREPLICATE. However, if I do this I get an extra

> dimension added:

>

> IDL> help, CMREPLICATE(transpose(a), 4)

> <Expression> FLOAT = Array[1, 5, 4]

>

> Is there a way to replicate over the columns without adding this extra

> dimension? I'm sure I'm missing something simple here.

>

> Cheers,

>

> Robin

b=rebin(reform(a,1,5), 5,5)

regards,
Lajos

Subject: Re: Replicate array over columns - using transpose?

Posted by [dplatten](#) on Thu, 22 Mar 2012 10:57:56 GMT

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

Hi Robin,

You can use a combination of REBIN and TRANSPOSE to do this. The following should work for you:

```
nElements = 5
```

```
nCols = 4
```

```
test = INDGEN(nElements)
```

```
print, test
```

```
    0    1    2    3    4
```

```
print, REBIN(TRANSPOSE(test), nCols, nElements)
```

```
    0    0    0    0
```

```
    1    1    1    1
```

```
    2    2    2    2
```

```
    3    3    3    3
```

```
    4    4    4    4
```

Regards,

David

Subject: Re: Replicate array over columns - using transpose?

Posted by [Heinz Stege](#) on Thu, 22 Mar 2012 11:25:52 GMT

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

Hi Robin,

I don't know CMREPLICATE. Does the following work?

```
help,transpose(cmreplicate(a,4))
```

Another question: What is the advantage of using CMREPLICATE instead of the built-in REBIN function?

```
help,transpose(rebin(a,n_elements(a),4,/sample))
```

Heinz

Subject: Re: Replicate array over columns - using transpose?

Posted by [Heinz Stege](#) on Thu, 22 Mar 2012 11:42:33 GMT

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

On Thu, 22 Mar 2012 12:25:52 +0100, Heinz Stege wrote:

```
>  
> help,transpose(rebin(a,n_elements(a),4,/sample))
```

Don't use the above. Lajos' solution without the TRANSPOSE operation is much more efficient:

On Thu, 22 Mar 2012 03:50:39 -0700 (PDT), Lajos Foldy wrote:

```
>  
> b=rebin(reform(a,1,5), 5,5)
```

Heinz

Subject: Re: Replicate array over columns - using transpose?

Posted by [Craig Markwardt](#) on Thu, 22 Mar 2012 12:01:30 GMT

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

On Thursday, March 22, 2012 6:11:54 AM UTC-4, robintw wrote:

```
> Hi all,  
>  
> I've got an array, a, as follows:  
>  
> IDL> print, a  
>    10.0000    35.0000    60.0000    85.0000    110.000  
>  
> I want to replicate this across 4 columns to produce an array like the  
> following:  
>  
> 10 10 10 10  
> 35 35 35 35  
> 60 60 60 60  
> 85 85 85 85  
> 110 110 110 110  
>  
> I'm using the CMREPLICATE routine to do this, and get the following:  
>  
> IDL> print, CMREPLICATE(a, 4)  
>    10.0000    35.0000    60.0000    85.0000    110.000  
>    10.0000    35.0000    60.0000    85.0000    110.000
```

```
> 10.0000 35.0000 60.0000 85.0000 110.000
> 10.0000 35.0000 60.0000 85.0000 110.000
>
> However, I want to replicate it in the columns as in the example above.
> As far as I can see it there are two ways to deal with this: transpose
> the array after I've done it (computationally expensive, as I'll be
> doing it with far larger arrays in the real program) or transpose it
> before running CMREPLICATE. However, if I do this I get an extra
> dimension added:
>
> IDL> help, CMREPLICATE(transpose(a), 4)
> <Expression>  FLOAT  = Array[1, 5, 4]
>
> Is there a way to replicate over the columns without adding this extra
> dimension? I'm sure I'm missing something simple here.
```

I'm the CM in CMreplicate, but I wouldn't use CMreplicate. I use this instead.

```
print, [1,1,1,1,1] # a
```

Usually I'm matching up X and Y labels to a 2D grid, so I do this,

```
XX = (X*0 + 1) # Y
YY = X # (Y*0 + 1)
```

Now XX and YY are 2D arrays with X running along one axis and Y running along the other axis. It looks crazy, but it works.

Craig

Subject: Re: Replicate array over columns - using transpose?

Posted by [Russell\[1\]](#) on Thu, 22 Mar 2012 15:17:35 GMT

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

It's way easier than that.

```
Suppose,
a=[10,35,60,85,110.]
```

and you want it to have n entries, then all you need is

```
b=(1+bytarr(n))#a
```

voila!

Russell

On Mar 22, 6:11 am, Robin Wilson <ro...@rtwilson.com> wrote:

```
> Hi all,
>
> I've got an array, a, as follows:
>
> IDL> print, a
>    10.0000    35.0000    60.0000    85.0000    110.000
>
> I want to replicate this across 4 columns to produce an array like the
> following:
>
> 10    10    10    10
> 35    35    35    35
> 60    60    60    60
> 85    85    85    85
> 110   110   110   110
>
> I'm using the CMREPLICATE routine to do this, and get the following:
>
> IDL> print, CMREPLICATE(a, 4)
>    10.0000    35.0000    60.0000    85.0000    110.000
>    10.0000    35.0000    60.0000    85.0000    110.000
>    10.0000    35.0000    60.0000    85.0000    110.000
>    10.0000    35.0000    60.0000    85.0000    110.000
>
> However, I want to replicate it in the columns as in the example above.
> As far as I can see it there are two ways to deal with this: transpose
> the array after I've done it (computationally expensive, as I'll be
> doing it with far larger arrays in the real program) or transpose it
> before running CMREPLICATE. However, if I do this I get an extra
> dimension added:
>
> IDL> help, CMREPLICATE(transpose(a), 4)
> <Expression>  FLOAT  = Array[1, 5, 4]
>
> Is there a way to replicate over the columns without adding this extra
> dimension? I'm sure I'm missing something simple here.
>
> Cheers,
>
> Robin
```

Subject: Re: Replicate array over columns - using transpose?

On Mar 22, 8:01 am, Craig Markwardt <craig.markwa...@gmail.com> wrote:

> On Thursday, March 22, 2012 6:11:54 AM UTC-4, robintw wrote:

>> Hi all,

>

>> I've got an array, a, as follows:

>

>> IDL> print, a

>> 10.0000 35.0000 60.0000 85.0000 110.000

>

>> I want to replicate this across 4 columns to produce an array like the
>> following:

>

>> 10 10 10 10

>> 35 35 35 35

>> 60 60 60 60

>> 85 85 85 85

>> 110 110 110 110

>

>> I'm using the CMREPLICATE routine to do this, and get the following:

>

>> IDL> print, CMREPLICATE(a, 4)

>> 10.0000 35.0000 60.0000 85.0000 110.000

>> 10.0000 35.0000 60.0000 85.0000 110.000

>> 10.0000 35.0000 60.0000 85.0000 110.000

>> 10.0000 35.0000 60.0000 85.0000 110.000

>

>> However, I want to replicate it in the columns as in the example above.

>> As far as I can see it there are two ways to deal with this: transpose

>> the array after I've done it (computationally expensive, as I'll be

>> doing it with far larger arrays in the real program) or transpose it

>> before running CMREPLICATE. However, if I do this I get an extra

>> dimension added:

>

>> IDL> help, CMREPLICATE(transpose(a), 4)

>> <Expression> FLOAT = Array[1, 5, 4]

>

>> Is there a way to replicate over the columns without adding this extra

>> dimension? I'm sure I'm missing something simple here.

>

> I'm the CM in CMreplicate, but I wouldn't use CMreplicate. I use this instead.

>

> print, [1,1,1,1,1] # a

>

> Usually I'm matching up X and Y labels to a 2D grid, so I do this,

>

> XX = (X*0 + 1) # Y

> YY = X # (Y*0 + 1)
>
> Now XX and YY are 2D arrays with X running along one axis and Y running along the other
axis. It looks crazy, but it works.
>
> Craig

Ha! I should have read on. I didn't see someone posted this answer.

Sorry for the repetition,
Russell

Subject: Re: Replicate array over columns - using transpose?
Posted by [Yngvar Larsen](#) on Fri, 23 Mar 2012 12:06:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

I don't think this variant is covered yet:

```
a = reform(float([10,35,60,85,110]), 1, 5, /overwrite)
print, rebin(a, 4, 5)
```

--
Yngvar
