Subject: Determining number of good data points over certain dimensions of an array
Posted by cb16 on Wed, 13 Jul 2016 14:37:40 GMT
View Forum Message <> Reply to Message

Hello!

I have a four dimensional array in IDL. The dimensions are as follows:

(longitude, latitude, time, parameter type)

There are a lot of missing values over space in my data, so I'm trying to to get the number of good values in a lon-lat area. To do this, I'm finding missing values and setting them to NaNs, then using knowledge about the total number of missing values, the total number of good values, and the sizes of my dimensions to get what I want.

In doing this - more for my own peace of mind than anything else - I also check that there are no missing values over the time dimension (there shouldn't be since I'm working with climatological data, but the dataset documentation doesn't explicitly say this) or between the parameters I'm working with (the nature of the data implies that there also shouldn't be, but again, I want to double check because documentation is not explicit).

Anyways, I believe my code does all of the above, but I'm not sure. I'd very much appreciate some confirmation and/or help.

Code below:

```
;Find missing values and set them to NaNs
bad = WHERE(data EQ -9999.0, count, COMPLEMENT = good, NCOMPLEMENT = count_g)
IF (count GT 0) THEN data[bad] = !Values.F_NaN

;Check that there are only missing values over space, not time or between BRDF params
IF ((count MOD ntime) NE 0) OR ((count MOD 3) NE 0) THEN MESSAGE, 'There are time series values missing, or missing values between BRDF params. Check data.'

;Get number of good values in a lat-lon area
count_g = (count_g/3)/ntime
```

(To clarify, the 3 corresponds to the size of the fourth dimension, the parameter type.)

I use MOD because I figure that if missing values are only over lon and lat dimensions, then the total number of missing values is the number of missing values in a lon-lat area multiplied by the sizes of the other dimensions. So, the total number of missing values should divide evenly into the time and parameter type dimensions. If this is true, then the number of good values in a lon-lat area is just the total number of good values divided by the time and parameter type dimensions.

Many thanks in advance.

---

Subject: Re: Determining number of good data points over certain dimensions of an array
Posted by Helder Marchetto on Wed, 13 Jul 2016 15:38:01 GMT
View Forum Message <> Reply to Message

Hi,
this is not my type of thing, but I think that the line:

;Check that there are only missing values over space, not time or between BRDF params
IF ((count MOD ntime) NE 0) OR ((count MOD 3) NE 0) THEN MESSAGE, 'There are time series values missing, or missing values between BRDF params. Check data.'

is not doing what you want it to do. If I understand correctly, your data variable looks like:
IDL> help, data
DATA FLOAT = Array[4, 100]
assuming that there are 100 data points.
Now notice that the result of the where() function refers to the one dimensional "version" of your data array (so an array that looks like FLOAT = Array[400]).

You now have two options:
1) use the function array_indices to convert the one dimensional to 2 dimensional array (http://www.harrisgeospatial.com/docs/ARRAY_INDICES.html)
2) do the math. Any time data will have the modulo equal to 2 (indexing starts at 0, so 0 = longitude, 1 = latitude, 2 = time, 3 =type).
To check things, then use:

```
if (count gt 0) then begin
  badPositions = bad mod 4
  badTime = where(badPositions eq 2, cntbadTime)
  badType = where(badPositions eq 3, cntbadType)
  if (cntbadTime gt 0) || (cntbadType gt 0) then MESSAGE, '...'
endif
```

I'm not sure what you meant with BRDF, so the second check (badType) might be wrong.

Hope it helps. Cheers,
Helder

On Wednesday, July 13, 2016 at 4:37:45 PM UTC+2, cb16 wrote:
> Hello!
>
> I have a four dimensional array in IDL. The dimensions are as follows:
>
> (longitude, latitude, time, parameter type)

>
> There are a lot of missing values over space in my data, so I'm trying to to get the number of good values in a lon-lat area. To do this, I'm finding missing values and setting them to NaNs, then using knowledge about the total number of missing values, the total number of good values, and the sizes of my dimensions to get what I want.
>
> In doing this - more for my own peace of mind than anything else - I also check that there are no missing values over the time dimension (there shouldn't be since I'm working with climatological data, but the dataset documentation doesn't explicitly say this) or between the parameters I'm working with (the nature of the data implies that there also shouldn't be, but again, I want to double check because documentation is not explicit).
>
> Anyways, I believe my code does all of the above, but I'm not sure. I'd very much appreciate some confirmation and/or help.
>
>
> Code below:
>
> ;Find missing values and set them to NaNs
> bad = WHERE(data EQ -9999.0, count, COMPLEMENT = good, NCOMPLEMENT = count_g)
> IF (count GT 0) THEN data[bad] = !Values.F_NaN
>
> ;Check that there are only missing values over space, not time or between BRDF params
> IF ((count MOD ntime) NE 0) OR ((count MOD 3) NE 0) THEN MESSAGE, 'There are time series values missing, or missing values between BRDF params. Check data.'
>
> ;Get number of good values in a lat-lon area
> count_g = (count_g/3)/ntime
>
> (To clarify, the 3 corresponds to the size of the fourth dimension, the parameter type.)
>
> I use MOD because I figure that if missing values are only over lon and lat dimensions, then the total number of missing values is the number of missing values in a lon-lat area multiplied by the sizes of the other dimensions. So, the total number of missing values should divide evenly into the time and parameter type dimensions. If this is true, then the number of good values in a lon-lat area is just the total number of good values divided by the time and parameter type dimensions.
>
>
> Many thanks in advance.

---

Subject: Re: Determining number of good data points over certain dimensions of an array
Posted by cb16 on Thu, 14 Jul 2016 15:55:18 GMT
View Forum Message <> Reply to Message

My data array actually looks a little different...

IDL> help, read.data
<Expression>    FLOAT     = Array[2401, 811, 46, 3]

...but I think the ARRAY_INDICES function should help, since it does exactly what I'm looking to do.

Thanks!

---