Subject: Re: TOTAL() and NaNs, again
Posted by Jeremy Bailin on Tue, 26 Mar 2013 14:30:30 GMT
View Forum Message <> Reply to Message

On 3/25/13 5:30 PM, Fabien wrote:
> Dear IDLers,
>
> I know my problems with NaNs do bother just my little person. Maybe I am
> the only one having crappy data to deal with, or I am the only one using
> NaNs to mask out things in my 3D data arrays... Anyways, I'll try to
> make this is my last post about NaNs.
>
> I've been complaining lately about MEAN throwing a math error in this case:
> IDL> array = [!VALUES.F_NAN, !VALUES.F_NAN]
> IDL> print, MEAN(array, /NAN)
>   -NaN
> % Program caused arithmetic error: Floating illegal operand
>
> I think, personally, that the result of mean in this case should be a
> NaN, and that this should not throw a math error. Because, afterwards,
> if I do:
> IDL> print, (array[0] + array[1]) / 0.
>   NaN
>
> This is an ugly divide by zero but there is no math warning here. But
> the problems comes from TOTAL:
> IDL> print, TOTAL(array, /NAN)
>   0.00000
> And of course:
> IDL> print, TOTAL(array, /NAN) / TOTAL(FINITE(array))
>   -NaN
> % Program caused arithmetic error: Floating illegal operand
>
> To be honest, this is described in the Doc: "Since the value NaN is
> treated as missing data, if Array contains only NaN values the TOTAL
> routine will return 0."
>
> But I don't see why it is so. This is not coherent with what one would
> expect TOTAL to do: the sum of all the elements in the array...
>

Actually, this is exactly what I would expect TOTAL to do... when you use the /NAN flag, you are saying "ignore NaNs". If there are no non-NaN elements, then the total is zero.

Now I'd be willing to believe that the behaviour of MEAN in this case is funny.

-Jeremy.