Subject: Re: Incorrect behavior of /NAN
Posted by Paul van Delst on Thu, 04 Apr 2002 18:19:00 GMT
View Forum Message <> Reply to Message

Kenneth Bowman wrote:
>
> I tried to incite some interest in this last week, but had no takers,
> so I'll try again. ;-)
>
> I believe that the behavior of the TOTAL function is incorrect in
> the case where all of the data are NaN and the /NAN keyword is set.
>
> Here are several possible cases:
>
> IDL> a = !values.f_nan
> IDL> print, total([a, 1.0])
> NaN
> IDL> print, total([a, 1.0], /nan)
> 1.00000
> IDL> print, total([a, a])
> NaN
> IDL> print, total([a, a], /nan)
> 0.00000
>
> I believe that the last case is incorrect.
>
> The documentation for /NAN says "Elements with the value NaN are treated
> as missing data." In the last case there are no valid data, so how can
> their sum be zero?
>
> I think this is probably a simple error in the TOTAL algorithm.
> Rather than computing the "sum of all the non-NaN values", it is
> probably doing "set sum to zero and then add all non-Nan values".

Seems reasonable to me. It only sums the data not flagged as missing. If all the data is missing, the sum of nothing is, well, nothing.

> For comparison, the MEAN function does not behave this way.
>
> IDL> print, mean([a, 1.0])
> NaN
> IDL> print, mean([a, 1.0], /nan)
> 1.00000
> IDL> print, mean([a, a])
> NaN
> IDL> print, mean([a, a], /nan)
> NaN

I think the last result *here* is wrong :o) Shouldn't it be 1.0? What is MEAN in this case but:

total([a,a],/NAN) / n_elements(non-nan values only)

Both the numerator and denominator are 0.

> Before I submit something (at least a question) to RSI, does anyone
> have comments?

Yes. Check your data *before* you TOTAL or MEAN them. :o)

paulv

--
Paul van Delst Religious and cultural
CIMSS @ NOAA/NCEP purity is a fundamentalist
Ph: (301)763-8000 x7274 fantasy
Fax:(301)763-8545 V.S.Naipaul