Subject: Re: manipulating structures

Posted by Kenneth P. Bowman on Sun, 08 Apr 2007 14:06:07 GMT
View Forum Message <> Reply to Message

In article <1175997980.025055.256650@e65g2000hsc.googlegroups.com>,
"metachronist" <rkombiyil@gmail.com> wrote:

> Thanks much! I don't trust myself in such circumstances and hence I
> agree with Dr.B :-) This prompts me to ask another trivial question,
> if I may..Since I have lots of missing data, and I do lots of math
> operations (array ops, fft etc. etc.), will these (NaN) propagate all
> the way through in such situations? Should I be using them in
> conjunction with finite statement? Any pointers as to where one oughta
> be careful with these NaNs?
>
> Thanks in advance for your time and sharing your experience,
> ~rk

Many IDL functions include /NAN keywords to skip NaNs in operations (TOTAL, MEAN, etc.). In other cases, you will have to find the good data with WHERE(FINITE(...), COUNT = count).

There is one special case that you have to watch out for when using TOTAL with the /NAN keyword. If *all* of the elements are NaNs, the result returned is not a NaN, but a zero.

```
IDL> x = replicate(!values.f_nan, 5)
IDL> print, x
         NaN         NaN         NaN         NaN         NaN
IDL> print, total(x)
         NaN
IDL> print, total(x, /nan)
     0.00000
```

I think this is a serious implementation bug because it renders the /NAN keyword useless in most circumstances, but I guess we are stuck with it.

Inconsistently, this happens with TOTAL, but not with MEAN

```
IDL> print, mean(x, /nan)
         NaN
```

Ken