Subject: summation problem
Posted by Manish on Thu, 05 Apr 2001 10:58:27 GMT
View Forum Message <> Reply to Message

Hi, I'm pretty new to IDL (well, actually completely new!) and was wondering if anyone could help with a small problem.

I have a program which produces an array of values of sunlight flux during the day. Unfortunately, it produces crazy numbers before the sunrises and after it sets(as expected). I'm summing the values throughout the day to get a total integrated day flux, but here's the problem - is there a way of telling the TOTAL function to ignore negative numbers and NaN numbers?

I guess this is a simple problem, but would appreciate any help

Many thanks,

Manish

Subject: Re: summation problem
Posted by davidf on Thu, 05 Apr 2001 16:48:30 GMT
View Forum Message <> Reply to Message

Jaco van Gorkom (j.c.van.gorkom@fz-juelich.de) writes:

> Geez! Thanks for pointing that out! This solves everything Manish: !EXCEPT = 0.

Shut your Command Output Log window and IDL will also run flawlessly! :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: summation problem
Posted by Paul van Delst on Thu, 05 Apr 2001 16:54:59 GMT
View Forum Message <> Reply to Message

Jaco van Gorkom wrote:
>

> Paul van Delst wrote:
>> Jaco van Gorkom wrote:
>>> Manish wrote:
>>> ...
>>>> I have a program which produces an array of values of sunlight flux during
>>>> the day. Unfortunately, it produces crazy numbers before the sunrises and
>>>> after it sets(as expected). I'm summing the values throughout the day to
>>>> get a total integrated day flux, but here's the problem - is there a way of
>>>> telling the TOTAL function to ignore negative numbers and NaN numbers?
>>>
>>> IDL> print, total(test>0,/nan)
>>> 7.00000
>>> % Program caused arithmetic error: Floating illegal operand
>>>
>>> The 'illegal operand' error appears to be harmless, caused by comparing test>0:
>>
>> Harmless maybe, but I for one don't like seeing illegal operand errors, let alone ignoring
>> them. What if you add some code that tries to take the log of a -ve number, see the same
>> error and shrug it off?
>
> That is *exactly* what I'd do: shrug it off! The log of a -ve number gives me
> -Inf as a
> result, which will propagate through any calculation to be NaN, Inf, or -Inf.
> Just another
> missing data point.

Yoicks. Please prepend any procedures/functions you distribute with your initials, JVG_
:o)

>> loc_finite = WHERE(FINITE(test) EQ 1, count_finite)
>> IF (count_finite GT 0) THEN $
>> sum = TOTAL(test[loc_finite] > 0.0) $
>> ELSE $
>> sum = !VALUES.F_NAN ; Or some other suitable flag
>
> Of course I see your point. I would probably implement a FINITE() check if I
> were writing a
> full program, but the annoying special-case programming for when WHERE() returns
> -1 held me
> back here.

Hence the use of count_finite. Then it doesn't matter what is returned by WHERE() in
loc_finite.

> On second thought: WHERE() could only return -1 for a day without sunrise...

I'm sure there are some Laplanders that use IDL :o) And there's those Antarctic folk.

>> Oh, and make sure you set !EXCEPT = 2 in your idl setup file. That'll learn ya to remove
>> errors from your code :o)
>
> Geez! Thanks for pointing that out! This solves everything Manish: !EXCEPT = 0.

Oof! That hurtz.

paulv

--
Paul van Delst A little learning is a dangerous thing;
CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian spring;
Ph: (301)763-8000 x7274 There shallow draughts intoxicate the brain,
Fax:(301)763-8545 And drinking largely sobers us again.
paul.vandelst@noaa.gov Alexander Pope.