Subject: Re: summation problem
Posted by Jaco van Gorkom on Thu, 05 Apr 2001 11:15:45 GMT
View Forum Message <> Reply to Message

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?

Yes, there is a way: IDL> test = [0,-1,3,4,!values.f_nan] 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:

IDL> print, test>0

0.00000 0.00000 3.00000 4.00000 NaN

% Program caused arithmetic error: Floating illegal operand

> I guess this is a simple problem, but would appreciate any help It is simple enough, but only once you know the solution.

cheers, Jaco

Subject: Re: summation problem
Posted by Manish on Thu, 05 Apr 2001 12:32:30 GMT
View Forum Message <> Reply to Message

Nice one mate, worked a treat.

Thanks.

Manish

"Jaco van Gorkom" <j.c.van.gorkom@fz-juelich.de> wrote in message news:3ACC53E1.A396D87C@fz-juelich.de...

> 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?
> Yes, there is a way:
> IDL> test = [0,-1,3,4,!values.f_nan]
> 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:
> IDL> print, test>0
      0.00000
                 0.00000
                              3.00000
                                         4.00000
                                                        NaN
  % Program caused arithmetic error: Floating illegal operand
>> I guess this is a simple problem, but would appreciate any help
  It is simple enough, but only once you know the solution.
> cheers,
   Jaco
```

Subject: Re: summation problem
Posted by Paul van Delst on Thu, 05 Apr 2001 15:00:28 GMT
View Forum Message <> Reply to Message

```
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?
>
> Yes, there is a way:
> IDL > test = [0,-1,3,4,]
> 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:
 IDL> print, test>0
      0.00000
                 0.00000
                             3.00000
                                         4.00000
                                                       NaN
```

> % Program caused arithmetic error: Floating illegal operand

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?

Why not do a

7.00000

```
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
??
IDL> print, sum
```

Oh, and make sure you set !EXCEPT = 2 in your idl setup file. That'll learn ya to remove errors from your code :o)

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.

Subject: Re: summation problem
Posted by Jaco van Gorkom on Thu.

Posted by Jaco van Gorkom on Thu, 05 Apr 2001 16:26:14 GMT

View Forum Message <> Reply to Message

```
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.

- > 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.

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

groetjes,

Jaco

- > ..
- > 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.