Subject: Re: Filtering out NaNs
Posted by bowman on Wed, 19 Mar 1997 08:00:00 GMT
View Forum Message <> Reply to Message

In article <5gp9oq$8hp$1@ratatosk.uio.no>, steinhh@rigil.uio.no (Stein Vidar Hagfors Haugan) wrote:

> NaNs are characterized by the fact that they are *not* equal
> to any number (that's what it says, isn't it :-)
>
> In fact, it's not even equal to itself - and this is the distinguishing
> feature that must be used to pick them out. I.e.,
>
> a(where(a ne a)) = missing_flag

I think you can also use the FINITE function.

Regards, Ken Bowman

--
Kenneth P. Bowman, Assoc. Prof. 409-862-4060
Department of Meteorology 409-862-4132 fax
Texas A&M University bowman@csrp.tamu.edu
College Station, TX 77843-3150

Subject: Re: Filtering out NaNs
Posted by davidf on Wed, 19 Mar 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Folks,

> |> There are some NaNs in my data which keep making my code fall over. I'd
> |> like to either filter them out or avoid them in the calculation, but doing
> |> something like
> |>
> |> a(where(a eq !values.f_nan) = missing_flag
> |>
> |> doesn't work. Does anyone have any suggestions?

>
> NaNs are characterized by the fact that they are *not* equal
> to any number (that's what it says, isn't it :-)
> In fact, it's not even equal to itself - and this is the distinguishing
> feature that must be used to pick them out. I.e.,
>
>> a(where(a ne a)) = missing_flag

I have to admit that I had convinced myself that this was a bug
in the implementation of NAN. I thank Stein Vidar for enlightening
me. It's always a good feeling when I am reminded that the universe
is far stranger than I imagine it to be most of the time! :-)

David

David Fanning, Ph.D.
Fanning Software Consulting
2642 Bradbury Court, Fort Collins, CO 80521
Phone: 970-221-0438 Fax: 970-221-4762
E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com

Subject: Re: Filtering out NaNs
Posted by Tim Patterson on Wed, 19 Mar 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Stein Vidar Hagfors Haugan wrote:
>
> In article <Iarla-1903971745010001@eorcu4.ch.cam.ac.uk>, Iarla@atm.ch.cam.ac.uk (Iarla Kilbane-Dawe) writes:
> |> Hello,
> |>
> |> There are some NaNs in my data which keep making my code fall over. I'd
> |> like to either filter them out or avoid them in the calculation, but doing
> |> something like
> |>
> |> a(where(a eq !values.f_nan) = missing_flag
> |>
> |> doesn't work. Does anyone have any suggestions?
> |>
>
> NaNs are characterized by the fact that they are *not* equal
> to any number (that's what it says, isn't it :-)
>
> In fact, it's not even equal to itself - and this is the distinguishing
> feature that must be used to pick them out. I.e.,
>
> a(where(a ne a)) = missing_flag
>
> Stein Vidar

A thing which is not even equal to itself !

I'm not sure if I'm reading the IDL newsgroup or the Zen mystics newsgroup...

Tim

Subject: Re: Filtering out NaNs
Posted by Dale Bailey on Thu, 20 Mar 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Is this what you want?:

if(NOT(FINITE(mh.isotope_halflife))) then print, "Halflife is NaN!"

Dale Bailey PhD
MRC Cyclotron Unit, Hammersmith Hospital
London. UK. W12 0NN
dale@cu.rpms.ac.uk
FAX: +44 (0)181 383 2029
Phone: +44 (0)181 383 3731 (direct)
+44 (0)181 383 3162 (reception)

Subject: Re: Filtering out NaNs
Posted by thompson on Thu, 20 Mar 1997 08:00:00 GMT
View Forum Message <> Reply to Message

sritcey@is.dal.ca (Stephen Ritcey) writes:

> What exactly does FINITE do on IDL?

In IDL, the FINITE function does filter out both NaN values and plus and minus infinity.

Personally, I don't like to write software that depends on IEEE arithmetic, because it makes the software somewhat less portable. There's probably only one platform that IDL runs on that doesn't use IEEE floating point numbers, and that's VMS (on either VAX or AXP). However, it's still critical for us to support VMS. Thus, when an XDR-formatted file is read containing NaN values, then those pixels are converted to an ordinary number that is used to signal missing data, and vice-versa. The question of how to handle +/-Inf has never arisen.

Bill Thompson

Subject: Re: Filtering out NaNs
Posted by peter on Thu, 20 Mar 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Stein Vidar Hagfors Haugan (steinhh@rigil.uio.no) wrote:

: NaNs are characterized by the fact that they are *not* equal
: to any number (that's what it says, isn't it :-)

: In fact, it's not even equal to itself - and this is the distinguishing
: feature that must be used to pick them out. I.e.,

: a(where(a ne a)) = missing_flag

I'd like to hereby nominate Stein for the "Cool Tip of the Week" award!

But is it platform independent?

Peter

Subject: Re: Filtering out NaNs
Posted by bowman on Thu, 20 Mar 1997 08:00:00 GMT
View Forum Message <> Reply to Message

In article <5grgr3$7ih@News.Dal.Ca>, sritcey@is.dal.ca (Stephen Ritcey) wrote:

> while Ken Bowman suggested using the FINITE function:
>
> b= a (finite (a)) ; exclude .... what? (see below)
>

>
> What exactly does FINITE do on IDL?

You may also find the new IDL system variable, !VALUES handy.

```
IDL> HELP, !VALUES, /STRUCT
** Structure !VALUES, 4 tags, length=24:
   F_INFINITY      FLOAT             INF
   F_NAN           FLOAT             NaNQ
   D_INFINITY      DOUBLE                INF
   D_NAN           DOUBLE            NaNQ


IDL> x = FINDGEN(3)
IDL> x(1) = !VALUES.F_NAN
IDL> x(2) = !VALUES.F_INFINITY
```

```
IDL> PRINT, x
     0.00000        NaNQ         INF
IDL> PRINT, WHERE(x EQ !VALUES.F_NAN)
         -1
IDL> PRINT, WHERE(x EQ !VALUES.F_INFINITY)
          2
IDL> PRINT, FINITE(x)
  1   0   0
```

So in IDL, at least, infinity is equal to infinity, but NAN is not equal to NAN. FINITE returns false for both infinity and NAN with no error message.

BTW, this is on:
IDL. Version 4.0.1a (OSF alpha).
Your mileage may vary by platform.

One last tidbit

```
IDL> x(2) = -!VALUES.F_INFINITY
IDL> print, x
     0.00000        NaNQ        -INF
IDL> PRINT, WHERE(x EQ !VALUES.F_INFINITY)
         -1
```

So you can test for + or - infinity.

Ken Bowman

--
Kenneth P. Bowman, Assoc. Prof. 409-862-4060
Department of Meteorology 409-862-4132 fax
Texas A&M University bowman@csrp.tamu.edu
College Station, TX 77843-3150

Subject: Re: Filtering out NaNs
Posted by f055 on Thu, 20 Mar 1997 08:00:00 GMT
View Forum Message <> Reply to Message

-> |> There are some NaNs in my data which keep making my code fall over.
-> |> I'd like to either filter them out or avoid them in the calculation,
-> |> but doing something like
-> |>
-> |> a(where(a eq !values.f_nan) = missing_flag

-> |>
-> |> doesn't work. Does anyone have any suggestions?
-> |>
->
-> NaNs are characterized by the fact that they are *not* equal
-> to any number (that's what it says, isn't it :-)
->
-> In fact, it's not even equal to itself - and this is the distinguishing
-> feature that must be used to pick them out. I.e.,
->
->   a(where(a ne a)) = missing_flag
->

Somewhat less confusing, perhaps, to someone who later tries to debug/modify
your code might be:

a(where(finite(a) eq 0)) = missing_flag

finite returns true (ne 0) only for numbers and not for things that are
"Not a Number" (or is it "Not a Numerical Quantity" = NanQ ?)

Tim

..... Dr Tim Osborn . t.osborn@uea.ac.uk
.... ___/.. __ /.. /.. /. Senior Research Associate . phone:01603 592089
... /..... /. /.. /.. /.. Climatic Research Unit . fax: 01603 507784
.. /..... __/.. /.. /... School of Environmental Sciences.
. /..... /\ ... /.. /.... University of East Anglia .
____/.._/.._.._____/..... Norwich NR4 7TJ .
..... UK .

Subject: Re: Filtering out NaNs
Posted by sritcey on Thu, 20 Mar 1997 08:00:00 GMT

View Forum Message <> Reply to Message

Iarla Kilbane-Dawe asked about filetering out NaN's.

The issue of detecting NaNs is a bit of a swamp; having just come out of
the muck a while ago, I'd like to add my two cents. [I'm running PVWave,
so some of the following may not apply to IDL.]

Stein Vidar suggested using that magical property of IEEE NaN's that they
are not even equal to themselves:

```
  b= a (where (a eq a))   ; exclude NaN's
```

while Ken Bowman suggested using the FINITE function:

```
b= a (finite (a))       ; exclude .... what?  (see below)
```

In IEEE floating point specification, in addition to ordinary floating point numbers, there are a bunch of special values, among them NaN (``not a number'') and infinity, both positive and negative.

Vidar's technique will exclude NaN's, but not infinities. The approach is widely used, but has been criticized in traditional languages like Fortran, where it has been known to fail because the compiler optimized it away. There is a general consensus that a function to test for NaN-ship is desirable in any language which supports IEEE arithmetic.

Unfortunately, FINITE is not that function. In my PV-Wave documentation, FINITE is described as ``Returns a value indicating if the input is finite or not ... [FINITE(x)] returns 1 if x is infinite or ... NaN''. Thus it seems to do more than the original poster wanted, since it not only flags NaNs, it also flags infinities.

Now maybe that's OK; probably you want to exclude both pathologies when cleaning up a data set. Unfortunately, the PV-Wave implementation is at odds with the documentation (PV-WAVE v6.01 (Advantage), IRIX Release 5.3):

Consider

```
WAVE> mach= machine (/float)
WAVE> a= [0.0, 1.0, mach.nan, 3.0, mach.neg_inf]
WAVE> goodguys= finite (a)
% Program caused arithmetic error: Floating illegal operand
% Detected at $MAIN$   (FINITE).
WAVE> print, goodguys
   1   1   1   1   0
```

So at least on the platform mentioned, FINITE cannot be used to flag NaN's. Question: Is the documentation wrong, or the implementation? I personally think that the implementation does what I would prefer; I would like to see a function ISNAN added.

What exactly does FINITE do on IDL?

Stephen Ritcey (902)494-3313 (voice) (902)494-5191 (fax)
Physics Dept., Dalhousie Univ., Halifax, N.S., Canada B3H 3J5