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

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?

Many thanks,

Iarla.

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

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

Subject: Re: Filtering out NaNs
Posted by Christian Marquardt on Fri, 21 Mar 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Peter Webb wrote:
>
> 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!

Well - it will result in an error message from IDL if there are _no_
NaN's or +/-Infinites in that array. Better try something like

idx = where(a ne a) & if idx(0) ne -1 then a(idx) = missing_flag

or

idx = where(finite(a) ne 1) & if idx(0) ne -1 then a(idx) =
missing_flag

I wonder if there's a way to use where without checking if the
condition was met at all, given that you do not know this in
advance.

Regards,

Chris.

---------- ----------
Christian Marquardt

Meteorologisches Institut der | tel.: (+49) 30-838-71170
Freien Universitaet Berlin | fax.: (+49) 30-838-71167
Carl-Heinrich-Becker-Weg 6-10 | email:
marq@strat01.met.fu-berlin.de
D-12165 Berlin |
---------- ----------

Subject: Re: Filtering out NaNs
Posted by Mark Hadfield on Sun, 23 Mar 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Stein Vidar Hagfors Haugan <steinhh@rigil.uio.no> wrote in
article <5gp9oq$8hp$1@ratatosk.uio.no>...
>
> 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've always used the finite function, eg:

  a(where(1-finite(a))) = missing_flag

However FINITE returns 0 for infinities as well as NaNs and I suppose this
could be a problem in some instances.

I just tried 4 different comparison operators against pairs of NaN's and
found that on my system (IDL 5.0 prerelease on WinNT/Intel) NaN IS equal to
itself, but it's also greater than itself.

IDL> a = !values.f_nan
IDL> print, a eq a, a ne a, a gt a, a lt a
  1 0 1 0
% Program caused arithmetic error: Floating illegal operand

Curious.

 ========== ==
Mark Hadfield NIWA (Taihoro Nukurangi)
 PO Box 14-901
m.hadfield@niwa.cri.nz Wellington, New Zealand