Subject: Re: manipulating structures
Posted by Kenneth P. Bowman on Sat, 07 Apr 2007 02:59:15 GMT

View Forum Message <> Reply to Message

In article <1175911428.077370.279650@n59g2000hsh.googlegroups.com>,
"metachronist" <rkombiyil@gmail.com> wrote:

```
>  a=fltarr(dim)           ;dim is dimension, i.e., #days * #data/day
```

The preceding line is unnecessary, as the following line will create the array "a" automatically.

```
>  a=float(data[*].mydat)    ;data[dim].mydat is data variable

>  a[where(a[*] eq 999999.)]=!values.f_nan
```

The line above is not a great idea, though, as it will crash when there are no missing data. Plus, the [*] is unnecessary. You should do something like this instead

```
i = WHERE(a EQ 999999.0, count)
IF (count GT 0) THEN a[i] = !VALUES.F_NAN
```

Other than that, the concept seems fine. You have to create a FLOAT variable in order to use NaNs, which I heartily endorse.

The only alternative is to create the original data structure using a FLOAT instead of a LONG (presumably when you read the data). I prefer to replace missing data codes with NaNs at the point I read the data. That way I don't use them inadvertently.

Ken Bowman

Subject: Re: manipulating structures
Posted by David Fanning on Sat, 07 Apr 2007 06:20:14 GMT

View Forum Message <> Reply to Message

metachronist writes:

> To all ye who have attained idl nirvana and to the one who speaketh
> the truth :p
>
> Missing data points in my 'binary' data (coded as 32bit words) are
> denoted by large #'s, like 999999. In order "not" to plot these
> missing values, I am using !values.f_nan. But the array ought to be
> float to set it to 'NaN' directly. Besides, the type of a variable
> within a structure can't be modified.

>
> What I did was:
> --
> a=fltarr(dim)   ;dim is dimension, i.e., #days *
> #data/day
> a=float(data[*].mydat)   ;data[dim].mydat is data variable
> a[where(a[*] eq 999999.)]=!values.f_nan
> --
>
> It works, but I was wondering if there is a "better way" to do this?

Ken's explanation is absolutely correct, but if it is
really just the plotting of the data that is causing
you problems, I'd forget about NANs and just use the
MAX_VALUE keyword to set a value less than the "missing"
value.

```
   PLOT, data, MAX_VALUE=999999.0 - 1
```

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: manipulating structures
Posted by Kenneth P. Bowman on Sat, 07 Apr 2007 14:33:29 GMT

View Forum Message <> Reply to Message

In article <MPG.2080e60786ec3c36989f18@news.frii.com>,
David Fanning <news@dfanning.com> wrote:

> Ken's explanation is absolutely correct, but if it is
> really just the plotting of the data that is causing
> you problems, I'd forget about NANs and just use the
> MAX_VALUE keyword to set a value less than the "missing"
> value.
>
>   PLOT, data, MAX_VALUE=999999.0 - 1

This is true, but using "special numbers" to indicate missing data
is rife with the possibility using the missing value as valid
data with noticing it. I'm a big advocate of using NaNs
because they ensure that if you use them by mistake, your result

will be a NaN (which is usually hard to ignore :-)).

Ken

Subject: Re: manipulating structures
Posted by rkombiyil on Sun, 08 Apr 2007 02:06:20 GMT

View Forum Message <> Reply to Message

On Apr 7, 11:33 pm, "Kenneth P. Bowman" <k-bow...@removethis.tamu.edu>
wrote:
> In article <MPG.2080e60786ec3c36989...@news.frii.com>,
> David Fanning <n...@dfanning.com> wrote:
>
>> Ken's explanation is absolutely correct, but if it is
>> really just the plotting of the data that is causing
>> you problems, I'd forget about NANs and just use the
>> MAX_VALUE keyword to set a value less than the "missing"
>> value.
>
>> PLOT, data, MAX_VALUE=999999.0 - 1
>
> This is true, but using "special numbers" to indicate missing data
> is rife with the possibility using the missing value as valid
> data with noticing it. I'm a big advocate of using NaNs
> because they ensure that if you use them by mistake, your result
> will be a NaN (which is usually hard to ignore :-)).
>
> Ken

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

Subject: Re: manipulating structures
Posted by David Fanning on Sun, 08 Apr 2007 03:31:45 GMT

View Forum Message <> Reply to Message

metachronist writes:

> I don't trust myself in such circumstances and hence I
> agree with Dr.B :-)

If safety is a big issue, I would NOT recommend playing
with a loaded IDL program! C or JAVA would be much
safer and would be much less likely to fire at an
unexpected target. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

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

Subject: Re: manipulating structures
Posted by David Fanning on Sun, 08 Apr 2007 18:23:39 GMT

View Forum Message <> Reply to Message

Kenneth P. Bowman writes:

> 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. I think this is a serious
> implementation bug because it renders the /NAN keyword useless in
> most circumstances.

Has this been reported to ITTVIS, Ken?

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: manipulating structures
Posted by Kenneth P. Bowman on Sun, 08 Apr 2007 18:40:36 GMT

View Forum Message <> Reply to Message

In article <MPG.2082e11919d147fa989f1c@news.frii.com>,
David Fanning <news@dfanning.com> wrote:

> Kenneth P. Bowman writes:
>
>> 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. I think this is a serious
>> implementation bug because it renders the /NAN keyword useless in
>> most circumstances.
>
> Has this been reported to ITTVIS, Ken?

We discussed this in this group several years ago, and *I think* I had some communication with them about this. The response was to add a note in the TOTAL documentation pointing out this behavior.

The rules for IEEE arithmetic say that the result of any operation involving a NaN is a NaN. That is not true in the case of TOTAL. If you sum an array that is all NaNs, the result is zero.

Ken

Subject: Re: manipulating structures
Posted by Foldy Lajos on Sun, 08 Apr 2007 19:22:30 GMT

View Forum Message <> Reply to Message

On Sun, 8 Apr 2007, Kenneth P. Bowman wrote:

> The rules for IEEE arithmetic say that the result of any operation
> involving a NaN is a NaN. That is not true in the case of TOTAL. If
> you sum an array that is all NaNs, the result is zero.

You are summing an empty array, if the /NaN keyword (= discard NaNs) was set. So, what is the sum of an empty array: 0 or Nan? :-)

regards,
lajos

Subject: Re: manipulating structures
Posted by David Fanning on Mon, 09 Apr 2007 00:17:17 GMT

View Forum Message <> Reply to Message

Lajos = writes:

> You are summing an empty array, if the /NaN keyword (= discard NaNs) was
> set. So, what is the sum of an empty array: 0 or Nan? :-)

Well, whatever it is, it probably ought be to the same
as MEAN, don't you think?

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: manipulating structures
Posted by rkombiyil on Mon, 09 Apr 2007 01:45:13 GMT
View Forum Message <> Reply to Message

On Apr 9, 9:17 am, David Fanning <n...@dfanning.com> wrote:
> Lajos = writes:
>> You are summing an empty array, if the /NaN keyword (= discard NaNs) was
>> set. So, what is the sum of an empty array: 0 or Nan? :-)
>
> Well, whatever it is, it probably ought be to the same
> as MEAN, don't you think?
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

I agree with Dr.Fanning completely. Consistency should be maintained
in such situations. Else, it will lead to inaccurate results and the
end user will have to spent much time trying to debug, when such
inconsistent behaviour is "not" obvious!
Thanks,
/rk

Subject: Re: manipulating structures
Posted by Foldy Lajos on Mon, 09 Apr 2007 10:21:55 GMT

View Forum Message <> Reply to Message

On Sun, 8 Apr 2007, David Fanning wrote:

> Lajos = writes:
>
>> You are summing an empty array, if the /NaN keyword (= discard NaNs) was
>> set. So, what is the sum of an empty array: 0 or Nan? :-)
>
> Well, whatever it is, it probably ought be to the same
> as MEAN, don't you think?
>
> Cheers,
>
> David

MEAN has an extra division, 0/0 and NaN/0 are both NaN, so it can differ
from the result of TOTAL.

I don't know which result is better, and I guess different people have
different preferences. There is a third option: IDL reports an error, as
empty arrays are illegal.

regards,
lajos

Subject: Re: manipulating structures
Posted by Kenneth Bowman on Mon, 09 Apr 2007 15:02:36 GMT

View Forum Message <> Reply to Message

In article <Pine.LNX.4.64.0704082118360.6079@bifur.rmki.kfki.hu>,
FOLDY Lajos <foldy@rmki.kfki.hu> wrote:

> On Sun, 8 Apr 2007, Kenneth P. Bowman wrote:
>
>> The rules for IEEE arithmetic say that the result of any operation
>> involving a NaN is a NaN. That is not true in the case of TOTAL. If
>> you sum an array that is all NaNs, the result is zero.
>
> You are summing an empty array, if the /NaN keyword (= discard NaNs) was
> set. So, what is the sum of an empty array: 0 or Nan? :-)
>
> regards,
> lajos

How can the sum of no numbers be a number?

I know we went through this same philosophical argument before, but in the end
the /NAN keyword is there as a programming convenience. Zero is
a valid answer when there are good data in the array. It should not be a
valid answer when there are no good data in the array.

As I said, it renders the /NAN keyword useless to me. I don't want to
get a valid number back when there are no valid input data.

Instead of

```
   tot = TOTAL(x, /NAN)
```

I must do

```
  i = WHERE(FINITE(x), COUNT = count)
  IF (COUNT EQ 0) THEN tot = !VALUES.F_NAN $
             ELSE tot = TOTAL(x[i])
```

Ken

Subject: Re: manipulating structures
Posted by James Kuyper on Mon, 09 Apr 2007 17:05:52 GMT

View Forum Message <> Reply to Message

Kenneth Bowman wrote:
...
> How can the sum of no numbers be a number?

Consider the total human population of all cities on the moon.
Naively, I would think that is is 0 (unless there is something very
interesting going on 8-), calculated as a sum of no numbers. Would you
argue that it is not a number? Would you argue that it is the sum of
some numbers? If so, what is the meaning of the numbers being added up?