Subject: Using Minimum operator with NaN values
Posted by wlandsman on Tue, 22 Oct 2013 00:25:34 GMT
View Forum Message <> Reply to Message

Most of this information is in the IDL docs
http://www.exelisvis.com/docs/Minimum_and_Maximum_Oper.html#
expressions_2976722315_1032669 but it is easy to forget.   The minimum and maximum operators do not work nicely with NaN values.

```
IDL> print,!Version
{ x86_64 darwin unix Mac OS X 8.2.3 May  2 2013      64      64}
IDL> a = findgen(5)
IDL> a[2] = !values.f_nan
IDL> print,a
      0.00000      1.00000          NaN      3.00000      4.00000
IDL> print,a>2
      2.00000      2.00000      2.00000      3.00000      4.00000
% Program caused arithmetic error: Floating illegal operand
```

I think most people would say that the proper behavior would be for the NaN value to be preserved.   The IDL doc says the behavior is actually hardware dependent, and on some machines the NaN is preserved! It suggests instead using WHERE and FINITE() instead of ">", whenever NaN values are present.   This also eliminates the annoying "illegal operand" error message.   The ">" operator does not check for NaN values for speed reasons, and the syntax does not allow having a /NaN keyword with ">" ;-)

A related problem occurs with the HIST_EQUAL() function.

```
IDL> print,hist_equal(a)
   0  85   0 170 255
% Program caused arithmetic error: Floating illegal operand
```

In this case, it is correct that the NaN is not preserved, since the output of HIST_EQUAL() is a byte array.   But I think it is a bug that the illegal operand error occurs.   HIST_EQUAL() *does* check for NaN values internally -- for example, HISTOGRAM() is called with the NaN keyword. It is only at the last step in HIST_EQUAL() where a ">" operator is used, that the NaN value are ignored.   In this case, I think ITTVIS should follow their own advice and use WHERE and FINITE() rather than the ">" operator.

--Wayne

Subject: Re: Using Minimum operator with NaN values
Posted by suicidaleggroll on Tue, 22 Oct 2013 15:08:58 GMT
View Forum Message <> Reply to Message

On Monday, October 21, 2013 6:25:34 PM UTC-6, wlandsman wrote:
> Most of this information is in the IDL docs
http://www.exelisvis.com/docs/Minimum_and_Maximum_Oper.html#

expressions_2976722315_1032669 but it is easy to forget.   The minimum and maximum operators do not work nicely with NaN values.

```
>
>
>
>  IDL> print,!Version
>
>  { x86_64 darwin unix Mac OS X 8.2.3 May  2 2013      64      64}
>
>  IDL> a = findgen(5)
>
>  IDL> a[2] = !values.f_nan
>
>  IDL> print,a
>
>       0.00000      1.00000          NaN      3.00000      4.00000
>
>  IDL> print,a>2
>
>       2.00000      2.00000      2.00000      3.00000      4.00000
>
>  % Program caused arithmetic error: Floating illegal operand
>
>
>
```

> I think most people would say that the proper behavior would be for the NaN value to be preserved.   The IDL doc says the behavior is actually hardware dependent, and on some machines the NaN is preserved! It suggests instead using WHERE and FINITE() instead of ">", whenever NaN values are present.   This also eliminates the annoying "illegal operand" error message.   The ">" operator does not check for NaN values for speed reasons, and the syntax does not allow having a /NaN keyword with ">" ;-)

```
>
>
>
>  A related problem occurs with the HIST_EQUAL() function.
>
>  IDL> print,hist_equal(a)
>
>    0  85   0 170 255
>
>  % Program caused arithmetic error: Floating illegal operand
>
>
>
```

> In this case, it is correct that the NaN is not preserved, since the output of HIST_EQUAL() is a byte array.   But I think it is a bug that the illegal operand error occurs.   HIST_EQUAL() *does* check for NaN values internally -- for example, HISTOGRAM() is called with the NaN keyword. It is only at the last step in HIST_EQUAL() where a ">" operator is used, that the NaN value are

ignored.   In this case, I think ITTVIS should follow their own advice and use WHERE and FINITE() rather than the ">" operator.
>
>
>
> --Wayne

I do dislike that the > and < operators do not work with NaNs, however there is a simple workaround if you need it (this is what I do when I need to preserve NaNs through those operations).

```
print,(a>2)+a*0
```

Subject: Re: Using Minimum operator with NaN values
Posted by wlandsman on Tue, 22 Oct 2013 21:48:04 GMT

View Forum Message <> Reply to Message

> I do dislike that the > and < operators do not work with NaNs, however there is a simple workaround if you need it (this is what I do when I need to preserve NaNs through those operations).
>
>>
> print,(a>2)+a*0

Thanks.   The above code is almost twice as fast for me compared with

```
g=where(finite(a))
a[g] = a[g]>2
```

though it still gives an illegal operand error on the Mac.   --Wayne