Subject: Re: HISTOGRAM data type bug?
Posted by Timm Weitkamp on Mon, 14 Jun 2010 08:57:45 GMT
View Forum Message <> Reply to Message

chris wrote:
> Hi,
> I observed that IDL 6.4 x64 and 7.1 x64 (Windows 7 x64) crashes each
> time I'd like to do the following:
>
> IDL> test_histo_bug
> % Compiled module: TEST_HISTO_BUG.
> L INT = Array[5, 5]
> L DOUBLE = Array[5, 5]
> H LONG = Array[7]
>
> IDL> test_histo_bug,/crashme
> ->IDL crashes completely
>
>
> pro test_histo_bug,crashme=crashme
> l=$
> [[0,0,0,0,0],$
> [0,1,1,1,1],$
> [0,1,1,1,1],$
> [0,1,1,1,1],$
> [0,1,1,1,2]]
> help,l
> l=keyword_set(crashme)? ulong(l) : double(l)
> h=histogram(l,reverse_indices=r,nbins=7,min=1)
> help,l,h
> end
>
> Is it possible that HISTOGRAMM can't *handle* ULONG values? May be
> someone can shed some light on this behaviour?

This may not shed much light on it, but:

(1) I can reproduce this crash:

```
IDL> print, !version
{ x86_64 linux unix linux 7.0 Oct 25 2007      64      64}
IDL> test_histo_bug
% Compiled module: TEST_HISTO_BUG.
L               INT       = Array[5, 5]
L               DOUBLE    = Array[5, 5]
H               LONG      = Array[7]
IDL> test_histo_bug,/crashme
L               INT       = Array[5, 5]
```

Floating point exception

(2) In the past, I had already seen IDL systematically crashing when I did certain operations with unsigned integer data types (not only ULONG, but also UINT; but I don't remember the operations).

Timm

Subject: Re: HISTOGRAM data type bug?
Posted by Foldy Lajos on Mon, 14 Jun 2010 09:45:03 GMT
View Forum Message <> Reply to Message

On Mon, 14 Jun 2010, chris wrote:

> Hi,
> I observed that IDL 6.4 x64 and 7.1 x64 (Windows 7 x64) crashes each
> time I'd like to do the following:
>
> IDL> test_histo_bug
> % Compiled module: TEST_HISTO_BUG.
> L INT = Array[5, 5]
> L DOUBLE = Array[5, 5]
> H LONG = Array[7]
>
> IDL> test_histo_bug,/crashme
> ->IDL crashes completely
>
>
> pro test_histo_bug,crashme=crashme
> l=$
> [[0,0,0,0,0],$
> [0,1,1,1,1],$
> [0,1,1,1,1],$
> [0,1,1,1,1],$
> [0,1,1,1,2]]
> help,l
> l=keyword_set(crashme)? ulong(l) : double(l)
> h=histogram(l,reverse_indices=r,nbins=7,min=1)
> help,l,h
> end
>
> Is it possible that HISTOGRAMM can't *handle* ULONG values? May be
> someone can shed some light on this behaviour?
>
> Thanks in advance
>
> CR

>

From the docs:

BINSIZE
Set this keyword to the size of the bin to use. If this keyword is not specified, and NBINS is not set, then a bin size of 1 is used. If NBINS is set, the default is BINSIZE = (MAX - MIN) / (NBINS - 1).

Note
The data type of the value specified for BINSIZE should match the data type of the Array argument. Since BINSIZE is converted to the data type of Array, specifying mismatched data types may produce undesired results.

So, your BINSIZE is 0 (calculated as ULONG). This may cause a 'divide by zero' crash. I think IDL forgets to check BINSIZE.

regards,
lajos

Subject: Re: HISTOGRAM data type bug?
Posted by rogass on Mon, 14 Jun 2010 12:29:26 GMT
View Forum Message <> Reply to Message

On 14 Jun., 11:45, FÖLDY Lajos <fo...@rmki.kfki.hu> wrote:
> On Mon, 14 Jun 2010, chris wrote:
>> Hi,
>> I observed that IDL 6.4 x64 and 7.1 x64 (Windows 7 x64) crashes each
>> time I'd like to do the following:
>
>> IDL> test_histo_bug
>> % Compiled module: TEST_HISTO_BUG.
>> L INT = Array[5, 5]
>> L DOUBLE = Array[5, 5]
>> H LONG = Array[7]
>
>> IDL> test_histo_bug,/crashme
>> ->IDL crashes completely
>
>> pro test_histo_bug,crashme=crashme
>> l=$
>> [[0,0,0,0,0],$
>> [0,1,1,1,1],$
>> [0,1,1,1,1],$
>> [0,1,1,1,1],$
>> [0,1,1,1,2]]

>> help,l
>> l=keyword_set(crashme)? ulong(l) : double(l)
>> h=histogram(l,reverse_indices=r,nbins=7,min=1)
>> help,l,h
>> end
>
>> Is it possible that HISTOGRAMM can't *handle* ULONG values? May be
>> someone can shed some light on this behaviour?
>
>> Thanks in advance
>
>> CR
>
> From the docs:
> ----------
> BINSIZE
> Set this keyword to the size of the bin to use. If this keyword is not
> specified, and NBINS is not set, then a bin size of 1 is used. If NBINS is
> set, the default is BINSIZE = (MAX - MIN) / (NBINS - 1).
>
> Note
> The data type of the value specified for BINSIZE should match the data
> type of the Array argument. Since BINSIZE is converted to the data type of
> Array, specifying mismatched data types may produce undesired results.
> ----------
>
> So, your BINSIZE is 0 (calculated as ULONG). This may cause a 'divide by
> zero' crash. I think IDL forgets to check BINSIZE.
>
> regards,
> lajos

Uh, thank you.
I forgot the influence of the data type.
So, anyway, this means also that if you don't determine the binsize
exactly
the histogram might be erroneous due to the rounding of nbins and
binsize....

I used the histogram together with a label_region(/ulong).

However, the missing check is now a *feature* for me :)

Subject: Re: HISTOGRAM data type bug?
Posted by David Fanning on Mon, 14 Jun 2010 12:36:37 GMT
View Forum Message <> Reply to Message

FÖLDY Lajos writes:

> From the docs:
> ----------
> BINSIZE
> Set this keyword to the size of the bin to use. If this keyword is not
> specified, and NBINS is not set, then a bin size of 1 is used. If NBINS is
> set, the default is BINSIZE = (MAX - MIN) / (NBINS - 1).
>
> Note
> The data type of the value specified for BINSIZE should match the data
> type of the Array argument. Since BINSIZE is converted to the data type of
> Array, specifying mismatched data types may produce undesired results.
> ----------
>
> So, your BINSIZE is 0 (calculated as ULONG). This may cause a 'divide by
> zero' crash. I think IDL forgets to check BINSIZE.

Crashing is only the most extreme of the bad things that can happen if you don't match the binsize variable with the data type of the argument. If you don't believe me, try writing a wrapper for the Histogram command and make it available on a public forum. Your phone will soon be ringing off the hook with people facing inexplicably wrong results. :-)

It seems to me this is something ITTVIS might be able to fix, especially in such a high profile routine like HISTOGRAM. It's pretty disconcerting to read the Note above after you have spent five maddening hours trying to figure out why your program doesn't work.

In the meantime, you might find the Coyote Library routine Convert_to_Type of interest. It was created specifically to solve this problem.

http://www.dfanning.com/programs/convert_to_type.pro

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: HISTOGRAM data type bug?
Posted by rogass on Tue, 15 Jun 2010 18:38:44 GMT
View Forum Message <> Reply to Message

Dear all,
thank you for your suggestions and hints.

David, I use your routines as often and as much as I can, but anyway
thank you for your offer :) (On my first place is undefine.pro which
does its job pretty well.)

Regards
CR