Subject: Re: histogram produces extra bin in 64-bit IDL 8.0
Posted by David Fanning on Fri, 13 May 2011 14:08:58 GMT
View Forum Message <> Reply to Message

Eric Tittley writes:

> Consider the following code:
> IDL> D=randomu(32,3200);
> IDL> N=histogram(D,min=0.,max=1.,binsize=0.1,Locations=X)
>
> In IDL 32-bit:
> IDL> print, size(N)
> 1 10 3 10
> IDL> print, X
> 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
> IDL> print, !version
> { x86 linux unix linux 8.0 Jun 18 2010 32 64}
>
> In 64-bit IDL:
> IDL> print, size(N)
> 1 11 3 11
> IDL> print, X
> 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
> IDL> print, !version
> { x86_64 linux unix linux 8.0 Jun 18 2010 64 64}
>
> As you can see, the 64-bit version produces an extra erroneous bin, contrary to the what is expected from the help pages for histogram.

Undoubtedly a consequence of the razor's edge:

http://www.idlcoyote.com/math_tips/razoredge.html

Cheers,

David

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

Subject: Re: histogram produces extra bin in 64-bit IDL 8.0

Posted by penteado on Fri, 13 May 2011 14:13:22 GMT
View Forum Message <> Reply to Message

On May 13, 11:08 am, David Fanning <n...@idlcoyote.com> wrote:
> Eric Tittley writes:
>> Consider the following code:
>> IDL> D=randomu(32,3200);
>> IDL> N=histogram(D,min=0.,max=1.,binsize=0.1,Locations=X)
>
>> In IDL 32-bit:
>> IDL> print, size(N)
>> 1 10 3 10
>> IDL> print, X
>> 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
>> IDL> print, !version
>> { x86 linux unix linux 8.0 Jun 18 2010 32 64}
>
>> In 64-bit IDL:
>> IDL> print, size(N)
>> 1 11 3 11
>> IDL> print, X
>> 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
>> IDL> print, !version
>> { x86_64 linux unix linux 8.0 Jun 18 2010 64 64}
>
>> As you can see, the 64-bit version produces an extra erroneous bin, contrary to the what is expected from the help pages for histogram.
>
> Undoubtedly a consequence of the razor's edge:
>
> http://www.idlcoyote.com/math_tips/razoredge.html

That is what I was thinking, considering both results to be normal and
expected. But I was wondering what changed between versions.

Subject: Re: histogram produces extra bin in 64-bit IDL 8.0
Posted by EddE on Fri, 13 May 2011 14:27:27 GMT
View Forum Message <> Reply to Message

On May 13, 3:13 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:

> That is what I was thinking, considering both results to be normal and
> expected. But I was wondering what changed between versions.

http://pastebin.com/aTMJXZtC

Curiouser and curiouser - 32 and 64 bit modes on the same machine act

identically, but another machine behaves differently....

(found in discussion with http://twitter.com/allinthegutter)

--
Edd

Subject: Re: histogram produces extra bin in 64-bit IDL 8.0
Posted by Foldy Lajos on Fri, 13 May 2011 14:28:40 GMT
View Forum Message <> Reply to Message

On Fri, 13 May 2011, Paulo Penteado wrote:

> That is what I was thinking, considering both results to be normal and
> expected. But I was wondering what changed between versions.

I guess x86 uses the x87 floating point coprocessor with 80 bit precision,
while x86_64 uses SSE2 with 64 bit precision.

regards,
Lajos

Subject: Re: histogram produces extra bin in 64-bit IDL 8.0
Posted by David Fanning on Fri, 13 May 2011 14:50:57 GMT
View Forum Message <> Reply to Message

EddE writes:

> Curiouser and curiouser - 32 and 64 bit modes on the same machine act
> identically, but another machine behaves differently....

I do notice in this code that there is a discrepancy in the
type of data and the type of the binsize. Histogram is
notorious for having difficulty when the data types
don't match. Hence, the Convert_To_Type function
in the Coyote Library.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.

Subject: Re: histogram produces extra bin in 64-bit IDL 8.0
Posted by war on Wed, 18 May 2011 15:04:53 GMT
View Forum Message <> Reply to Message

I just posted the following message but then I saw there just was a topic about more or less the similar problem. In my case, it is just about using different binsize and max.

Can somebody explain why the following gives different number of elements with the HISTOGRAM function?

```
IDL> hist_spd= HISTOGRAM(velocity[indb],binsize= .20,min=0.,max=1.40)
IDL> print,hist_spd
         6         16         23          0          0
0          0
IDL> hist_spd= HISTOGRAM(velocity[indb],binsize= 20.,min=0.,max=140.)
IDL> print,hist_spd
        45          0          0          0          0
0          0          0
```

(whatever velocity content is)
My main concern is the number of elements returned by the function. The only difference between the 2 command is that I multiplied the binsize and max with 100. BUT they return 7 and 8 elements?

I would expect they both return the same number of elements.

Thanks,

Andry