Subject: where function not finding value
Posted by mboggsk9 on Thu, 03 Apr 2014 17:44:00 GMT
View Forum Message <> Reply to Message

Hello,
I have a problem with a particular call to the 'where' function that I don't understand.

I have the following code snippet:

```
        print, 'dncom.dt.julian is: ', dncom.dt.julian
        print, 'dt.julian is: ', dt.julian

        ; end debug
        i = where(dncom.dt.julian eq dt.julian, count)  ; Find index of requested date & time
      if count gt 0 then begin
                data = dncom(i)
                ; debug stuff
                print, 'the data  ',data
        endif else begin
                print, 'ncom_table.pro: desired or most recent time not found'
                retall
                return
        endelse
```

and it says the requested value is not there everytime..

This is the output

```
dncom.dt.julian is:        95528.427       95528.438       95528.448
       95528.458       95528.469       95528.479       95528.490
       95528.500       95528.510       95528.521       95528.531
       95528.542       95528.552       95528.562       95528.573
       95528.583       95528.594       95528.604       95528.615
       95528.625       95528.635       95528.646       95528.656
       95528.667       95528.677       95528.688       95528.698
       95528.708       95528.719       95528.729       95528.740
       95528.750       95528.760       95528.771       95528.781
       95528.792       95528.802       95528.812       95528.823
       95528.833       95528.844       95528.854       95528.865
       95528.875       95528.885       95528.896       95528.906
       95528.917       95528.927       95528.938       95528.948
       95528.958       95528.969       95528.979       95528.990
       95529.000       95529.010       95529.021       95529.031
       95529.042       95529.052       95529.062       95529.073
       95529.083       95529.094       95529.104       95529.115
       95529.125       95529.135       95529.146       95529.156
       95529.167       95529.177       95529.188       95529.198
```

```
   95529.208      95529.219      95529.229      95529.240
   95529.250      95529.260      95529.271      95529.281
   95529.292      95529.302      95529.312      95529.323
   95529.333      95529.344      95529.354      95529.365
   95529.375      95529.385      95529.396      95529.406
   95529.417      95529.427      95529.438      95529.448
   95529.458      95529.469      95529.479      95529.490
   95529.500      95529.510      95529.521      95529.531
   95529.542      95529.552      95529.562      95529.573
   95529.583      95529.594      95529.604      95529.615
   95529.625      95529.635      95529.646      95529.656
   95529.667      95529.677      95529.688      95529.698
   95529.708      95529.719      95529.729      95529.740
   95529.750      95529.760      95529.771      95529.781
   95529.792      95529.802      95529.812      95529.823
   95529.833      95529.844      95529.854      95529.865
   95529.875      95529.885      95529.896      95529.906
   95529.917      95529.927      95529.938      95529.948
   95529.958      95529.969      95529.979      95529.990
   95530.000      95530.010      95530.021      95530.031
   95530.042      95530.052      95530.062      95530.073
   95530.083      95530.094      95530.104      95530.115
   95530.125      95530.135      95530.146      95530.156
   95530.167      95530.177      95530.188      95530.198
   95530.208      95530.219      95530.229      95530.240
   95530.250      95530.260      95530.271      95530.281
   95530.292      95530.302      95530.312      95530.323
   95530.333      95530.344      95530.354      95530.365
   95530.375      95530.385      95530.396      95530.406
   95530.417      95530.427

dt.julian is:       95530.427
ncom_table.pro: desired or most recent time not found
```

The value is there, It is the last value in the array.
Any idea what is going on?

Subject: Re: where function not finding value
Posted by on Thu, 03 Apr 2014 17:57:32 GMT

View Forum Message <> Reply to Message

Den torsdagen den 3:e april 2014 kl. 19:44:00 UTC+2 skrev Marsha Boggs:

>
> The value is there, It is the last value in the array.
>
> Any idea what is going on?

Try printing the difference between that last value in the array and the value you are looking for. They probably differ in decimal four or later.

Subject: Re: where function not finding value
Posted by David Fanning on Thu, 03 Apr 2014 18:29:57 GMT

View Forum Message <> Reply to Message

Mats Löfdahl writes:

>
> Den torsdagen den 3:e april 2014 kl. 19:44:00 UTC+2 skrev Marsha Boggs:
>>
>> The value is there, It is the last value in the array.
>>
>> Any idea what is going on?
>
> Try printing the difference between that last value in the array and the value you are looking for. They probably differ in decimal four or later.

Or, my guess, you are running into the famous Where gotcha:

http://www.idlcoyote.com/misc_tips/noidea.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 thue. ("Perhaps thou speakest truth.")

Subject: Re: where function not finding value
Posted by Phillip Bitzer on Thu, 03 Apr 2014 18:44:30 GMT

View Forum Message <> Reply to Message

On Thursday, April 3, 2014 12:57:32 PM UTC-5, Mats Löfdahl wrote:
> Try printing the difference between that last value in the array and the value you are looking for. They probably differ in decimal four or later.

As a follow on, testing equality in floats is not a good idea. This goes under the "sky is falling" category:

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

http://www.cygnus-software.com/papers/comparingfloats/compar ingfloats.htm

Subject: Re: where function not finding value
Posted by mboggsk9 on Thu, 03 Apr 2014 21:15:49 GMT

View Forum Message <> Reply to Message

Thanks for the feed back. I will review all of this and hope my solution is there. I am not a idl programmer (but will have to figure it out) I was tasked to port this code over from a HPUX platform (circa 1995) with wave 7.5 (where it works)to a new redhat linux platform with wave 10.0... where it doesn't. This is a large code with NO comments. But, now I know where to start looking. Thank you again.

Subject: Re: where function not finding value
Posted by Jahvasc on Tue, 10 Mar 2015 18:12:41 GMT

View Forum Message <> Reply to Message

Hi, guys,

I have a similar problem but I'm comparing intervals to avoid the float-precision problem. Still, I can't get the right answer...

I have a vector with 500 values ranging from 0.01 to 0.4 (0.01 to 0.1 by 0.01; 0.2 to 0.4 by 0.1). When I use the histogram function I get a certain number of counts for all intervals:

```
      79        57        48        44        43        43
      30        43        31        34        25        14     9
```

i.e., the value 0.01 appears 79 times, the value 0.02, 57 times, etc.

However, when I use the where function, I get some "holes". This is the bit of the code I'm using:

```
mmod=[findgen(10)*0.01 + 0.01,findgen(3)*0.1 + 0.2]
massi=0.
for j=0,12 do begin
  a=where(mass gt massi and mass le mmod(j),count)
  print,massi,mmod(j),count
  massi=mmod(j)
endfor
```

In this case I get the following numbers:

```
0.00000    0.0100000          79
0.0100000    0.0200000        57
```

```
0.0200000   0.0300000      48
0.0300000   0.0400000      44
0.0400000   0.0500000       0
0.0500000   0.0600000      43
0.0600000   0.0700000      73
0.0700000   0.0800000      43
0.0800000   0.0900000       0
0.0900000    0.100000      31
0.100000    0.200000       59
0.200000    0.300000       14
0.300000    0.400000        9
```

When I manually try, for example,

```
a=where(mass gt 0.04 and mass le 0.05,count)
print,count
```

I get count=43. What's is going on?

Thanks,

Jaqueline

On Thursday, April 3, 2014 at 7:44:00 PM UTC+2, Marsha Boggs wrote:

```
>  Hello,
>  I have a problem with a particular call to the 'where' function that I don't understand.
>
>  I have the following code snippet:
>
>         print, 'dncom.dt.julian is: ', dncom.dt.julian
>         print, 'dt.julian is: ', dt.julian
>
>         ; end debug
>         i = where(dncom.dt.julian eq dt.julian, count)  ; Find index of requested date & time
>       if count gt 0 then begin
>                 data = dncom(i)
>                 ; debug stuff
>                 print, 'the data  ',data
>         endif else begin
>                 print, 'ncom_table.pro: desired or most recent time not found'
>                 retall
>                 return
>         endelse
>
>
>  and it says the requested value is not there everytime..
>
>  This is the output
```

```
>
> dncom.dt.julian is:      95528.427     95528.438     95528.448
>        95528.458     95528.469     95528.479     95528.490
>        95528.500     95528.510     95528.521     95528.531
>        95528.542     95528.552     95528.562     95528.573
>        95528.583     95528.594     95528.604     95528.615
>        95528.625     95528.635     95528.646     95528.656
>        95528.667     95528.677     95528.688     95528.698
>        95528.708     95528.719     95528.729     95528.740
>        95528.750     95528.760     95528.771     95528.781
>        95528.792     95528.802     95528.812     95528.823
>        95528.833     95528.844     95528.854     95528.865
>        95528.875     95528.885     95528.896     95528.906
>        95528.917     95528.927     95528.938     95528.948
>        95528.958     95528.969     95528.979     95528.990
>        95529.000     95529.010     95529.021     95529.031
>        95529.042     95529.052     95529.062     95529.073
>        95529.083     95529.094     95529.104     95529.115
>        95529.125     95529.135     95529.146     95529.156
>        95529.167     95529.177     95529.188     95529.198
>        95529.208     95529.219     95529.229     95529.240
>        95529.250     95529.260     95529.271     95529.281
>        95529.292     95529.302     95529.312     95529.323
>        95529.333     95529.344     95529.354     95529.365
>        95529.375     95529.385     95529.396     95529.406
>        95529.417     95529.427     95529.438     95529.448
>        95529.458     95529.469     95529.479     95529.490
>        95529.500     95529.510     95529.521     95529.531
>        95529.542     95529.552     95529.562     95529.573
>        95529.583     95529.594     95529.604     95529.615
>        95529.625     95529.635     95529.646     95529.656
>        95529.667     95529.677     95529.688     95529.698
>        95529.708     95529.719     95529.729     95529.740
>        95529.750     95529.760     95529.771     95529.781
>        95529.792     95529.802     95529.812     95529.823
>        95529.833     95529.844     95529.854     95529.865
>        95529.875     95529.885     95529.896     95529.906
>        95529.917     95529.927     95529.938     95529.948
>        95529.958     95529.969     95529.979     95529.990
>        95530.000     95530.010     95530.021     95530.031
>        95530.042     95530.052     95530.062     95530.073
>        95530.083     95530.094     95530.104     95530.115
>        95530.125     95530.135     95530.146     95530.156
>        95530.167     95530.177     95530.188     95530.198
>        95530.208     95530.219     95530.229     95530.240
>        95530.250     95530.260     95530.271     95530.281
>        95530.292     95530.302     95530.312     95530.323
>        95530.333     95530.344     95530.354     95530.365
>       95530.375      95530.385      95530.396      95530.406
>       95530.417      95530.427
>
>  dt.julian is:       95530.427
>  ncom_table.pro: desired or most recent time not found
>
>
>  The value is there, It is the last value in the array.
>  Any idea what is going on?
```

Subject: Re: where function not finding value
Posted by David Fanning on Tue, 10 Mar 2015 18:42:34 GMT
View Forum Message <> Reply to Message

Jahvasc writes:

```
>
>  However, when I use the where function, I get some "holes". This is the bit of the code I'm using:
>
>  mmod=[findgen(10)*0.01 + 0.01,findgen(3)*0.1 + 0.2]
>  massi=0.
>  for j=0,12 do begin
>    a=where(mass gt massi and mass le mmod(j),count)
>    print,massi,mmod(j),count
>    massi=mmod(j)
>  endfor
>
>  In this case I get the following numbers:
>
>  0.00000    0.0100000          79
>  0.0100000    0.0200000        57
>  0.0200000    0.0300000        48
>  0.0300000    0.0400000        44
>  0.0400000    0.0500000         0
>  0.0500000    0.0600000        43
>  0.0600000    0.0700000        73
>  0.0700000    0.0800000        43
>  0.0800000    0.0900000         0
>  0.0900000     0.100000        31
>  0.100000     0.200000         59
>  0.200000     0.300000         14
>  0.300000     0.400000          9
>
>  When I manually try, for example,
>
>  a=where(mass gt 0.04 and mass le 0.05,count)
>  print,count
>
```

> I get count=43. What's is going on?

Given the evidence, I would say you are making incorrect assumptions
about your data.

Cheers,

David

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

Subject: Re: where function not finding value
Posted by chris_torrence@NOSPAM on Tue, 10 Mar 2015 19:01:58 GMT

View Forum Message <> Reply to Message

On Tuesday, March 10, 2015 at 12:12:46 PM UTC-6, Jahvasc wrote:
> Hi, guys,
>
> I have a similar problem but I'm comparing intervals to avoid the float-precision problem. Still, I can't get the right answer...
>
> I have a vector with 500 values ranging from 0.01 to 0.4 (0.01 to 0.1 by 0.01; 0.2 to 0.4 by 0.1).
> When I use the histogram function I get a certain number of counts for all intervals:

```
>
>          79          57          48          44          43          43
>          30          43          31          34          25          14      9
>
```

> i.e., the value 0.01 appears 79 times, the value 0.02, 57 times, etc.
>
> However, when I use the where function, I get some "holes". This is the bit of the code I'm using:

```
>
>  mmod=[findgen(10)*0.01 + 0.01,findgen(3)*0.1 + 0.2]
>  massi=0.
>  for j=0,12 do begin
>    a=where(mass gt massi and mass le mmod(j),count)
>    print,massi,mmod(j),count
>    massi=mmod(j)
>  endfor
```

```
>
>  In this case I get the following numbers:
>
>  0.00000     0.0100000          79
>  0.0100000     0.0200000        57
>  0.0200000     0.0300000        48
>  0.0300000     0.0400000        44
>  0.0400000     0.0500000         0
>  0.0500000     0.0600000        43
>  0.0600000     0.0700000        73
>  0.0700000     0.0800000        43
>  0.0800000     0.0900000         0
>  0.0900000      0.100000        31
>  0.100000     0.200000          59
>  0.200000     0.300000          14
>  0.300000     0.400000           9
>
>  When I manually try, for example,
>
>  a=where(mass gt 0.04 and mass le 0.05,count)
>  print,count
>
>  I get count=43. What's is going on?
>
```

Hi Jaqueline,

I think you are running into issues with floating-point precision. On every computer platform (in any language), there are floating-point numbers which are not exactly representable. For example, in IDL, try:

```
IDL> print, 0.1, format='(f25.16)'
       0.1000000014901161
```

If you use a number like 0.1, which isn't exactly representable, for math operations, and you then try to compare that to other numbers, you will get surprising results:

```
IDL> x = 0 & for i=0,99 do x = x + 0.1
IDL> print,x
      10.0000
IDL> print,x eq 10
   0
IDL> print,x, format='(f25.16)'
      10.0000019073486330
```

This is just a limitation of doing floating-point math on a computer (nothing to do with IDL).

Hope this helps.

-Chris

Subject: Re: where function not finding value
Posted by Jahvasc on Tue, 10 Mar 2015 20:06:45 GMT
View Forum Message <> Reply to Message

I'm sure about the data. Moreover, I would be glad to send them to you (or to whoever may want them) if you can/want to make your own tests. But my point is: why histogram and where functions give different results?

Subject: Re: where function not finding value
Posted by Jahvasc on Tue, 10 Mar 2015 20:15:16 GMT
View Forum Message <> Reply to Message

Hi, Chris, thank you for your answer. But it was to avoid this floating point problems that I'm using the intervals. Besides, why when I write down the interval (mass gt 0.04 and mass le 0.05) I got the the right answer?

Subject: Re: where function not finding value
Posted by David Fanning on Tue, 10 Mar 2015 20:15:25 GMT
View Forum Message <> Reply to Message

Jahvasc writes:

> I'm sure about the data. Moreover, I would be glad to send them to you (or to whoever may want them) if you can/want to make your own tests. But my point is: why histogram and where functions give different results?

Here is one reason:

   http://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 thue. ("Perhaps thou speakest truth.")

Subject: Re: where function not finding value

Posted by Burch on Tue, 10 Mar 2015 20:19:15 GMT
View Forum Message <> Reply to Message

On Tuesday, March 10, 2015 at 3:15:18 PM UTC-5, Jahvasc wrote:
> Hi, Chris, thank you for your answer. But it was to avoid this floating point problems that I'm using the intervals. Besides, why when I write down the interval (mass gt 0.04 and mass le 0.05) I got the the right answer?

Unfortunately you're not quite avoiding those issues. For instance, analogous to what you're doing try this

```
IDL> print, 0.05 le 4.0*0.01+0.01
   0

IDL> print, 0.05 le 0.05
   1
```

The first example *should* return 1 in a perfect world, but once you do mathematical operations (such as the multiplication and addition) you will introduce floating point arithmetic issues.

Subject: Re: where function not finding value
Posted by chris_torrence@NOSPAM on Wed, 11 Mar 2015 02:34:48 GMT
View Forum Message <> Reply to Message

On Tuesday, March 10, 2015 at 2:19:16 PM UTC-6, Jeff B wrote:
> On Tuesday, March 10, 2015 at 3:15:18 PM UTC-5, Jahvasc wrote:
>> Hi, Chris, thank you for your answer. But it was to avoid this floating point problems that I'm using the intervals. Besides, why when I write down the interval (mass gt 0.04 and mass le 0.05) I got the the right answer?
>
> Unfortunately you're not quite avoiding those issues. For instance, analogous to what you're doing try this
>
> IDL> print, 0.05 le 4.0*0.01+0.01
>   0
>
> IDL> print, 0.05 le 0.05
>   1
>
> The first example *should* return 1 in a perfect world, but once you do mathematical operations (such as the multiplication and addition) you will introduce floating point arithmetic issues.

One way to reduce (but not eliminate) these types of issues is to use double precision for all calculations:

```
IDL> print, 0.05d le 4*0.01d + 0.01d
   1
```

Again, I can't stress enough that this is a problem with all languages. For example, in Python, if we use 32-bit floating-point numbers:

```
>>>  import numpy as np
>>>  f = np.float32
>>>  f(0.05) <= 4*f(0.01) + f(0.01)
False
```

Cheers,
Chris

Subject: Re: where function not finding value
Posted by Jahvasc on Wed, 11 Mar 2015 09:16:39 GMT
View Forum Message <> Reply to Message

Thank you all for your replies.

As I told you, I am aware of the problems caused by floating point precision and I also know that is not related to IDL itself. However, what has astonished me was the fact that different functions or ways to address the problem in IDL produced such discrepant results.

Because of your help, I realised that the problem was in the way I was defining the intervals. In fact, they were too large. Then, I re-wrote the program introducing an eps=1.e-3 value. The loop turned-out to be:

```
mmod=[findgen(10)*0.01+0.01,findgen(3)*0.1+0.2]
eps=1.d-3
for j=0,12 do begin
  a=where(massa ge mmod[j]-eps and massa le mmod[j]+eps,count)
  print,mmod(j),count
endfor
```

Now I get the right counts.
Thank you again!

Cheers,
Jaqueline

Subject: Re: where function not finding value
Posted by Craig Markwardt on Wed, 11 Mar 2015 14:06:43 GMT
View Forum Message <> Reply to Message

On Wednesday, March 11, 2015 at 5:16:47 AM UTC-4, Jahvasc wrote:
> Thank you all for your replies.

>
> As I told you, I am aware of the problems caused by floating point precision and I also know that is not related to IDL itself. However, what has astonished me was the fact that different functions or ways to address the problem in IDL produced such discrepant results.
>
> Because of your help, I realised that the problem was in the way I was defining the intervals. In fact, they were too large. Then, I re-wrote the program introducing an eps=1.e-3 value. The loop turned-out to be:
>
> ```
> mmod=[findgen(10)*0.01+0.01,findgen(3)*0.1+0.2]
> eps=1.d-3
> for j=0,12 do begin
>   a=where(massa ge mmod[j]-eps and massa le mmod[j]+eps,count)
>   print,mmod(j),count
> endfor
> ```
>
> Now I get the right counts.
> Thank you again!

Jacqueline, from your description of the problem, your samples are right at the edges of the bins. I guess you've found a way to do this, but it doesn't look pretty and using a WHERE() function inside a FOR loop, well, it's kind of wierd.

But one will always get strange results if one tries to histogram/bin data when the sample values fall exactly on bin edges. (believe me, I goofed on this for data from a multi-million dollar space mission)

There are lots of ways to solve this, but they all rely on moving the bin edges away from your data samples. These techniques will always work, not do not rely on having special knowledge of how the data is quantized.

```
;; With HISTOGRAM (bin edges start at 0.005)
hh = histogram(massa, min=0.005, max=0.405, binsize=0.010)

;; I guess these are where your values are sampled, right?
mmod = [0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.2, 0.3, 0.4]
;; Then put the edges just to the left, plus one edge on the right hand side
edges = [sample_vals - 0.005, 0.5]

;; With HISTOGRAM and VALUE_LOCATE
hh = histogram(value_locate(edges, massa), min=0, max=12)

;; With a loop, no WHERE
for i = 0, 12 do begin
  hh[i] = total(massa GE edges[i] AND massa LT edges[i+1])
endfor

;; OK, you really want a loop with WHERE?
```

```
for i = 0, 12 do begin
  wh = where(massa GE edges[i] AND massa LT edges[i+1], count)
  hh[i] = count
  print, mmod[i], hh[i]
endfor
```

The expression,

```
  massa GE edges[i] AND massa LT edges[i+1]
```

does two things. Since EDGES[i+1] will be used as the right hand edge of the i-th bin, and then used again as the left edge of the (i+1)-th bin, one can never miss counts. There are *never* any holes due to floating point arithmetic. By the way, VALUE_LOCATE() does all this behind the scenes, that's what it was designed to do.

Also, look at the strategic use of GE on one bin edge, and LT on another bin edge. This guarantees that one sample can't fall into two bins, i.e. prevents double-counting.

Craig

Subject: Re: where function not finding value
Posted by Phillip Bitzer on Wed, 11 Mar 2015 16:24:11 GMT
View Forum Message <> Reply to Message

Yes, this is all about floating point.

But, there's another issue that's worth noting: your test where statement is incorrect.

Histograms are inclusive on the low end.

Let's remove the floating point business, and look at an example:

```
x = [0, 3, 5, 10, 11, 20]
h=histogram(x, BINSIZE=5, MIN=0)
print, h
....          2          1          2          0          1
```

We can use reverse indices and locations to see that, for example, 10 and 11 fall in the 10-15 bin. So, your where statement should look like what Craig has.