
Subject: Re: Histogram and bin sizes

Posted by [pgrigis](#) on Wed, 20 Feb 2008 19:43:35 GMT

[View Forum Message](#) <> [Reply to Message](#)

jeffnettles4...@gmail.com wrote:

> I've always wondered why you have to use a constant bin size with
> HISTOGRAM().
> To quote J.D.'s famous tutorial: "a histogram
> represents nothing more than a fancy way to count." Doesn't an
> imposed constant bin size imply that this is the only way it's ok to
> count? I can think of several reasons i wouldn't want to do this - I
> used logarithmic bin sizes in my dissertation, for example (now i'm
> hoping someone isn't going to answer this post saying i screwed up in
> my dissertation :-)).

I use logarithmic bins myself quite often, and the fact that a
logarithmic bin
size is the same as a constant bin size in log space, makes it is easy
to use histogram to get that. Less regulars binning don't work with
histogram, but nobody stops you from writing your own version to work
with them (it will not be as fast as histogram though).

Ciao,
Paolo

> And besides, Excel lets you use arbitrary bin
> sizes....and if Excel lets you do it, it has to be ok, right???? ;-)
>
> Jeff

Subject: Re: Histogram and bin sizes

Posted by [Conor](#) on Thu, 21 Feb 2008 14:05:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Feb 20, 2:43 pm, pgri...@gmail.com wrote:

> jeffnettles4...@gmail.com wrote:
>> I've always wondered why you have to use a constant bin size with
>> HISTOGRAM().
>> To quote J.D.'s famous tutorial: "a histogram
>> represents nothing more than a fancy way to count." Doesn't an
>> imposed constant bin size imply that this is the only way it's ok to
>> count? I can think of several reasons i wouldn't want to do this - I
>> used logarithmic bin sizes in my dissertation, for example (now i'm
>> hoping someone isn't going to answer this post saying i screwed up in
>> my dissertation :-)).
>
> I use logarithmic bins myself quite often, and the fact that a

> logarithmic bin
> size is the same as a constant bin size in log space, makes it is easy
> to use histogram to get that. Less regulars binning don't work with
> histogram, but nobody stops you from writing your own version to work
> with them (it will not be as fast as histogram though).
>
> Ciao,
> Paolo
>
>> And besides, Excel lets you use arbitrary bin
>> sizes....and if Excel lets you do it, it has to be ok, right???? ;-)
>
>> Jeff

You can always do whatever binning you want, you just have to transform your data to the new space and then bin it constantly. Why doesn't histogram let you use arbitrary binsizes? Not being an IDL developer I don't know for sure, but I would guess it's a speed issue. The simpler a program is the faster it is. I use histogram all the time because it's one of the speedier programs in IDL. It would make me very sad if in order to make histogram more flexible, it also became much slower, especially since by transforming my data set I can use arbitrary bin sizes for histogram.

Subject: Re: Histogram and bin sizes

Posted by [Kenneth P. Bowman](#) on Thu, 21 Feb 2008 14:54:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

In article

<3a7ec87f-f7e6-40b8-a6de-8328810e0e41@62g2000hsn.googlegroups.com>,
Conor <cmancone@gmail.com> wrote:

> You can always do whatever binning you want, you just have to
> transform your data to the new space and then bin it constantly. Why
> doesn't histogram let you use arbitrary binsizes? Not being an IDL
> developer I don't know for sure, but I would guess it's a speed
> issue. The simpler a program is the faster it is. I use histogram
> all the time because it's one of the speedier programs in IDL. It
> would make me very sad if in order to make histogram more flexible, it
> also became much slower, especially since by transforming my data set
> I can use arbitrary bin sizes for histogram.

If your bins are of uniform width, or if you can transform your data such that your bins are of uniform width, then it is possible to *compute* the bin index for each value with a simple linear transformation (and conversion to LONG if necessary). This will run at 1 value per clock cycle on some machines.

If the bins are of irregular width, such that no simple transformation will serve, then it is necessary to *search* the list of bins to determine the bin index. This is necessarily much slower than an arithmetic calculation, as it involves unpredictable branching.

In this example, binning 10^7 numbers into irregular bins is about 10 times slower than using HISTOGRAM with regular bins due to the time required to do a binary search.

;Use HISTOGRAM to bin 10^7 numbers with evenly-spaced bins

```
r = 100.0*RANDOMU(seed, 10^7)
t0 = SYSTIME(/SECONDS)
h1 = HISTOGRAM(r, MIN = 0.0, BINSIZE = 1.0)
PRINT, 'Time for evenly-spaced bins : ', SYSTIME(/SECONDS) - t0
```

;Use VALUE_LOCATE and HISTOGRAM to bin 10^7 numbers
;with unevenly-spaced bins

```
bins = [0.0, FINDGEN(99) + 0.1*RANDOMU(seed, 99)]
t0 = SYSTIME(/SECONDS)
i = VALUE_LOCATE(bins, r)
h2 = HISTOGRAM(i, MIN = 0, BINSIZE = 1)
PRINT, 'Time for unevenly-spaced bins : ', SYSTIME(/SECONDS) - t0
```

IDL> @hist_bins

```
Time for evenly-spaced bins :    0.076565981
Time for unevenly-spaced bins :    0.79658723
```

Ken Bowman

Subject: Re: Histogram and bin sizes

Posted by [jeffnetles4870](#) on Thu, 21 Feb 2008 16:29:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Feb 21, 9:05 am, Conor <cmanc...@gmail.com> wrote:

> On Feb 20, 2:43 pm, pgri...@gmail.com wrote:

>

>

```

>
>> jeffnettles4...@gmail.com wrote:
>>> I've always wondered why you have to use a constant bin size with
>>> HISTOGRAM().
>>> To quote J.D.'s famous tutorial: "a histogram
>>> represents nothing more than a fancy way to count." Doesn't an
>>> imposed constant bin size imply that this is the only way it's ok to
>>> count? I can think of several reasons i wouldn't want to do this - I
>>> used logarithmic bin sizes in my dissertation, for example (now i'm
>>> hoping someone isn't going to answer this post saying i screwed up in
>>> my dissertation :-) ).
>
>> I use logarithmic bins myself quite often, and the fact that a
>> logarithmic bin
>> size is the same as a constant bin size in log space, makes it is easy
>> to use histogram to get that. Less regulars binning don't work with
>> histogram, but nobody stops you from writing your own version to work
>> with them (it will not be as fast as histogram though).
>
>> Ciao,
>> Paolo
>
>>> And besides, Excel lets you use arbitrary bin
>>> sizes....and if Excel lets you do it, it has to be ok, right???? ;-)
>
>>> Jeff
>
> You can always do whatever binning you want, you just have to
> transform your data to the new space and then bin it constantly. Why
> doesn't histogram let you use arbitrary binsizes? Not being an IDL
> developer I don't know for sure, but I would guess it's a speed
> issue. The simpler a program is the faster it is. I use histogram
> all the time because it's one of the speedier programs in IDL. It
> would make me very sad if in order to make histogram more flexible, it
> also became much slower, especially since by transforming my data set
> I can use arbitrary bin sizes for histogram.

```

That actually sounds like what i've done in the past. For my dissertation i needed two kinds of histograms: logarithmic bins (which was fine, no trouble there) and bins that had arbitrary sizes. For the latter, i would either do the histograms in Excel (yuck) or compute two or three histograms in IDL using histogram() with different bin sizes and sort of do some "mixing and matching" of the resulting arrays to get what i wanted. Of course, David hadn't written his awesome histoplot routine yet then either :(Anyway, i'm up against the arbitrary bin sizes problem again for a project i'm doing for someone, and it got me wondering whether this situation is just so rare it wasn't worth supporting in histogram(). I wouldn't

want to lose histogram's speed either though.

Jeff

Subject: Re: Histogram and bin sizes

Posted by [Conor](#) on Thu, 21 Feb 2008 17:19:29 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Feb 21, 11:29 am, jeffnettles4...@gmail.com wrote:

> On Feb 21, 9:05 am, Conor <cmanc...@gmail.com> wrote:

>

>

>

>> On Feb 20, 2:43 pm, pgri...@gmail.com wrote:

>

>>> jeffnettles4...@gmail.com wrote:

>>>> I've always wondered why you have to use a constant bin size with

>>>> HISTOGRAM().

>>>> To quote J.D.'s famous tutorial: "a histogram

>>>> represents nothing more than a fancy way to count." Doesn't an

>>>> imposed constant bin size imply that this is the only way it's ok to

>>>> count? I can think of several reasons i wouldn't want to do this - I

>>>> used logarithmic bin sizes in my dissertation, for example (now i'm

>>>> hoping someone isn't going to answer this post saying i screwed up in

>>>> my dissertation :-)).

>

>>> I use logarithmic bins myself quite often, and the fact that a

>>> logarithmic bin

>>> size is the same as a constant bin size in log space, makes it is easy

>>> to use histogram to get that. Less regulars binning don't work with

>>> histogram, but nobody stops you from writing your own version to work

>>> with them (it will not be as fast as histogram though).

>

>>> Ciao,

>>> Paolo

>

>>>> And besides, Excel lets you use arbitrary bin

>>>> sizes....and if Excel lets you do it, it has to be ok, right???? ;-)

>

>>>> Jeff

>

>> You can always do whatever binning you want, you just have to

>> transform your data to the new space and then bin it constantly. Why

>> doesn't histogram let you use arbitrary binsizes? Not being an IDL

>> developer I don't know for sure, but I would guess it's a speed

>> issue. The simpler a program is the faster it is. I use histogram

>> all the time because it's one of the speedier programs in IDL. It

>> would make me very sad if in order to make histogram more flexible, it
>> also became much slower, especially since by transforming my data set
>> I can use arbitrary bin sizes for histogram.

>

> That actually sounds like what i've done in the past. For my
> dissertation i needed two kinds of histograms: logarithmic bins
> (which was fine, no trouble there) and bins that had arbitrary sizes.
> For the latter, i would either do the histograms in Excel (yuck) or
> compute two or three histograms in IDL using histogram() with
> different bin sizes and sort of do some "mixing and matching" of the
> resulting arrays to get what i wanted. Of course, David hadn't
> written his awesome histoplot routine yet then either :(Anyway, i'm
> up against the arbitrary bin sizes problem again for a project i'm
> doing for someone, and it got me wondering whether this situation is
> just so rare it wasn't worth supporting in histogram(). I wouldn't
> want to lose histogram's speed either though.

>

> Jeff

Arbitrary bin sizes should be pretty easy to program. You just need
to map your data points appropriately. For instance if you had the
data set:

```
x = randomu(seed,100)
```

and you wanted bins from:

```
[0-.1,.1-.3,.3-.35,.35-.8,.8-1]
```

you might do something like this:

```
x = randomu(seed,100)
bins = [ [0,.1], [.1,.3], [.3,.35], [.35,.8], [.8,1] ]
newx = fltarr(n_elements(x))
for i=0,n_elements(bins[0,*])-1 do begin
  w = where( x ge bins[0,i] and x lt bins[1,i], c )
  if c gt 0 then newx[w] = i+.5
endfor
```

```
hist = histogram(newx,binsize=1.0,min=0)
plothist,newx
```

Subject: Re: Histogram and bin sizes

Posted by [Conor](#) on Thu, 21 Feb 2008 17:20:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Feb 21, 12:19 pm, Conor <cmanc...@gmail.com> wrote:

> On Feb 21, 11:29 am, jeffnettles4...@gmail.com wrote:

>
>
>
>> On Feb 21, 9:05 am, Conor <cmanc...@gmail.com> wrote:
>
>>> On Feb 20, 2:43 pm, pgri...@gmail.com wrote:
>
>>>> jeffnettles4...@gmail.com wrote:
>>>> > I've always wondered why you have to use a constant bin size with
>>>> > HISTOGRAM().
>>>> > To quote J.D.'s famous tutorial: "a histogram
>>>> > represents nothing more than a fancy way to count." Doesn't an
>>>> > imposed constant bin size imply that this is the only way it's ok to
>>>> > count? I can think of several reasons i wouldn't want to do this - I
>>>> > used logarithmic bin sizes in my dissertation, for example (now i'm
>>>> > hoping someone isn't going to answer this post saying i screwed up in
>>>> > my dissertation :-)).
>
>>>> I use logarithmic bins myself quite often, and the fact that a
>>>> logarithmic bin
>>>> size is the same as a constant bin size in log space, makes it is easy
>>>> to use histogram to get that. Less regulars binning don't work with
>>>> histogram, but nobody stops you from writing your own version to work
>>>> with them (it will not be as fast as histogram though).
>
>>>> Ciao,
>>>> Paolo
>
>>>> > And besides, Excel lets you use arbitrary bin
>>>> > sizes....and if Excel lets you do it, it has to be ok, right???? ;-)
>
>>>> > Jeff
>
>>> You can always do whatever binning you want, you just have to
>>> transform your data to the new space and then bin it constantly. Why
>>> doesn't histogram let you use arbitrary binsizes? Not being an IDL
>>> developer I don't know for sure, but I would guess it's a speed
>>> issue. The simpler a program is the faster it is. I use histogram
>>> all the time because it's one of the speedier programs in IDL. It
>>> would make me very sad if in order to make histogram more flexible, it
>>> also became much slower, especially since by transforming my data set
>>> I can use arbitrary bin sizes for histogram.
>
>> That actually sounds like what i've done in the past. For my
>> dissertation i needed two kinds of histograms: logarithmic bins
>> (which was fine, no trouble there) and bins that had arbitrary sizes.
>> For the latter, i would either do the histograms in Excel (yuck) or
>> compute two or three histograms in IDL using histogram() with

>> different bin sizes and sort of do some "mixing and matching" of the
>> resulting arrays to get what i wanted. Of course, David hadn't
>> written his awesome histoplot routine yet then either :(Anyway, i'm
>> up against the arbitrary bin sizes problem again for a project i'm
>> doing for someone, and it got me wondering whether this situation is
>> just so rare it wasn't worth supporting in histogram(). I wouldn't
>> want to lose histogram's speed either though.

>

>> Jeff

>

> Arbitrary bin sizes should be pretty easy to program. You just need
> to map your data points appropriately. For instance if you had the
> data set:

>

> x = randomu(seed,100)

>

> and you wanted bins from:

> [0-.1,.1-.3,.3-.35,.35-.8,.8-1]

>

> you might do something like this:

>

> x = randomu(seed,100)

> bins = [[0,.1], [.1,.3], [.3,.35], [.35,.8], [.8,1]]

> newx = fltarr(n_elements(x))

> for i=0,n_elements(bins[0,*])-1 do begin

> w = where(x ge bins[0,i] and x lt bins[1,i], c)

> if c gt 0 then newx[w] = i+.5

> endfor

>

> hist = histogram(newx,binsize=1.0,min=0)

> plothist,newx

Obviously you'll have to manually set the x-axis labels... This might
need a little tweaking but it should give the general idea.

Subject: Re: Histogram and bin sizes

Posted by [Kenneth P. Bowman](#) on Thu, 21 Feb 2008 22:54:19 GMT

[View Forum Message](#) <> [Reply to Message](#)

In article

<f6219865-59f4-4bf8-8718-67884c9df226@64g2000hsw.googlegroups.com>,

Conor <cmancone@gmail.com> wrote:

> Arbitrary bin sizes should be pretty easy to program. You just need
> to map your data points appropriately. For instance if you had the
> data set:
>


```

> x = randomu(seed,100)
>
> and you wanted bins from:
> [0-.1,.1-.3,.3-.35,.35-.8,.8-1]
>
> you might do something like this:
>
> x = randomu(seed,100)
> bins = [ [0,.1], [.1,.3], [.3,.35], [.35,.8], [.8,1] ]
> newx = fltarr(n_elements(x))
> for i=0,n_elements(bins[0,*])-1 do begin
>   w = where( x ge bins[0,i] and x lt bins[1,i], c )
>   if c gt 0 then newx[w] = i+.5
> endfor
>
> hist = histogram(newx,binsize=1.0,min=0)
> plothist,newx

```

This will work, but will be extremely slow because you test every value in the input array once for every bin.

The VALUE_LOCATE approach will be much faster, particularly for large numbers of bins, as it does a binary search.

Ken Bowman

Subject: Re: Histogram and bin sizes

Posted by [lyle_pakula](#) on Fri, 07 Mar 2008 00:14:13 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Feb 22, 1:54 am, "Kenneth P. Bowman" <k-bow...@null.edu> wrote:

```

>
>
> times slower than using HISTOGRAM with regular bins due to the time
> required to do a binary search.
>
> ;Use HISTOGRAM to bin 10^7 numbers with evenly-spaced bins
>
> r = 100.0 * RANDOMU(seed, 10^7)
> t0 = SYSTIME(/SECONDS)
> h1 = HISTOGRAM(r, MIN = 0.0, BINSIZE = 1.0)
> PRINT, 'Time for evenly-spaced bins  : ', SYSTIME(/SECONDS) - t0
>
> ;Use VALUE_LOCATE and HISTOGRAM to bin 10^7 numbers
> ;with unevenly-spaced bins
>

```

```
> bins = [0.0, FINDGEN(99) + 0.1*RANDOMU(seed, 99)]
> t0 = SYSTIME(/SECONDS)
> i = VALUE_LOCATE(bins, r)
> h2 = HISTOGRAM(i, MIN = 0, BINSIZE = 1)
> PRINT, 'Time for unevenly-spaced bins : ', SYSTIME(/SECONDS) - t0
```

My first comp.lang.idl post ..

I think you may have to add nbins to the above histogram call as if none of the elements lie in your desired bin range (i.e. value_locate returns an array of -1's) histogram will not return a mapping into your desired bin range.

e.g

```
function hist_ireg_bin, data, bin
; Calculate a histogram with irregular bin spacing
tmp = VALUE_LOCATE(bin, data)
return, HISTOGRAM(tmp, MIN = 0, BINSIZE = 1, nbins =
n_elements(bin))
end
```
