
Subject: Re: histogram, how to trasfer from linear bins to logarithmic bin?

Posted by [JDS](#) on Wed, 30 Sep 2009 18:58:00 GMT

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

On Sep 29, 5:51 pm, David Fanning <n...@dfanning.com> wrote:

> JDS writes:

>> This would better be described as a way to create a histogram using
>> any arbitrary bins of your own devising; pretty cool indeed, since you
>> can design those bins in whatever way is useful. It does require you
>> to sort your array beforehand, and so in this case would be less
>> efficient than just taking the histogram of the log of your data.

>

> Actually, as I realized a couple of weeks ago
> in Australia when I was teaching this example,
> the array does NOT need to be sorted in this
> example. The cutoff vector needs to be monotonically
> increasing, but the array you are partitioning does
> not need to be.

>

> I've been using this method (without sorting) to
> process quite a lot of data recently, and I am
> *extremely* pleased with how darn fast it is!

Very good point. The sort is over the bin vector, which can be (and usually is) much shorter than the data vector. And you will likely setup your bin boundary vector sorted to begin with. That said, for me HISTOGRAM(ALOG10) is still faster than HISTOGRAM(VALUE_LOCATE) (see below). You'll also note some "sky is falling" razors-edge differences between bins if you look closely.

JD

+++++

```
n=100000000L
```

```
a=randomu(sd,n)*1.e8
```

```
t=systime(1)
```

```
h=histogram(a,alog10(a),BINSIZE=1)
```

```
print,'Hist(log)',systime(1)-t
```

```
t=systime(1)
```

```
mn=min(a,MAX=mx)
```

```
mn=alog10(mn) & mx=alog10(mx)
```

```
nbin=ceil(mx-mn)
```

```
bin=10.^(mn + findgen(nbin))
```

```
h2=histogram(value_locate(bin,a))
```

```
print,'Hist(value_locate)',systemtime(1)-t
```

```
print,h,h2
```

```
END
```

```
Hist(log))          1.9417701
```

```
Hist(value_locate)  3.7843559
```

```
      7      60      646      6203      63126
```

```
629722
```

```
  6286765  62867178  30146293
```

```
      7      60      647      6202      63126
```

```
629724
```

```
  6286771  62867205  30146258
```