Subject: Histogram quickie
Posted by Christopher Thom on Fri, 08 Dec 2006 18:55:56 GMT
View Forum Message <> Reply to Message

Hi all,

I'm a long time where() fan, but trying to learn to wield this histogram beast. I'm working on an algorithm, and would like a way to divide an array of values into two bins, such that the sum of each bin is roughly equal. The values have no fixed distribution, so I expect the bin sizes to be non-uniform.

This sort of problem seems an ideal place to start earning my histogram badge, but I have to confess to only being able to think of brute-force-type solutions. Any suggestions?

cheers
chris

Subject: Re: Histogram quickie
Posted by JD Smith on Fri, 08 Dec 2006 22:38:07 GMT
View Forum Message <> Reply to Message

On Fri, 08 Dec 2006 12:55:56 -0600, Christopher Thom wrote:

> Hi all,
>
> I'm a long time where() fan, but trying to learn to wield this histogram
> beast. I'm working on an algorithm, and would like a way to divide an
> array of values into two bins, such that the sum of each bin is roughly
> equal. The values have no fixed distribution, so I expect the bin sizes to
> be non-uniform.
>
> This sort of problem seems an ideal place to start earning my histogram
> badge, but I have to confess to only being able to think of
> brute-force-type solutions. Any suggestions?

Probably WHERE will serve you well:

```
IDL> t=total(a,/CUMULATIVE)
IDL> bin1=where(t lt t[n_elements(a)-1]/2,COMPLEMENT=bin2)
```

Of course, there are many ways to divide values such that they fall into two roughly equal bins (n choose 2), some of which may be better than others.

JD

Subject: Re: Histogram quickie
Posted by Braedley on Sat, 09 Dec 2006 02:09:17 GMT
View Forum Message <> Reply to Message

I was thinking the same thing. This never struck me as a histogram problem. Sure, there are some novel uses for histogram, but this isn't one of them. If the array needs to be sorted, and you need to retain the original array, and you want one bin to have the low values and the other to have the highs, then I can conceivably see a use for histogram in this problem, but it would be so convoluted that it wouldn't be worth it (you would still have to go through JD's process anyways). The only difference I'm going to suggest is to allow for the situation that I commented about:

```
t=randomu(seed, x)
t=t[sort(t)]
u=total(t, /cumulative)
bin1=t[where(u le u[x-1]/2.0)]
bin2=t[where(u gt u[x-1[/2.0)]
```

Braedley

JD Smith wrote:
> On Fri, 08 Dec 2006 12:55:56 -0600, Christopher Thom wrote:
>
>> Hi all,
>>
>> I'm a long time where() fan, but trying to learn to wield this histogram
>> beast. I'm working on an algorithm, and would like a way to divide an
>> array of values into two bins, such that the sum of each bin is roughly
>> equal. The values have no fixed distribution, so I expect the bin sizes to
>> be non-uniform.
>>
>> This sort of problem seems an ideal place to start earning my histogram
>> badge, but I have to confess to only being able to think of
>> brute-force-type solutions. Any suggestions?
>
> Probably WHERE will serve you well:
>
> IDL> t=total(a,/CUMULATIVE)
> IDL> bin1=where(t lt t[n_elements(a)-1]/2,COMPLEMENT=bin2)
>
> Of course, there are many ways to divide values such that they fall into
> two roughly equal bins (n choose 2), some of which may be better than
> others.
>
> JD

Subject: Re: Histogram quickie
Posted by Christopher Thom on Mon, 11 Dec 2006 15:49:40 GMT
View Forum Message <> Reply to Message

Quoth JD Smith:

> On Fri, 08 Dec 2006 12:55:56 -0600, Christopher Thom wrote:
>
>> Hi all,
>>
>> I'm a long time where() fan, but trying to learn to wield this histogram
>> beast. I'm working on an algorithm, and would like a way to divide an
>> array of values into two bins, such that the sum of each bin is roughly
>> equal. The values have no fixed distribution, so I expect the bin sizes to
>> be non-uniform.
>>
>> This sort of problem seems an ideal place to start earning my histogram
>> badge, but I have to confess to only being able to think of
>> brute-force-type solutions. Any suggestions?
>
> Probably WHERE will serve you well:
>
> IDL> t=total(a,/CUMULATIVE)
> IDL> bin1=where(t lt t[n_elements(a)-1]/2,COMPLEMENT=bin2)
>
> Of course, there are many ways to divide values such that they fall into
> two roughly equal bins (n choose 2), some of which may be better than
> others.

aha! I wasn't aware of the /cumulative flag to total(), despite some
searching. Thanks for the pointer.

cheers
chris