Subject: Re: performing multiple histograms without loops
Posted by Jeremy Bailin on Wed, 03 Feb 2010 14:12:47 GMT
View Forum Message <> Reply to Message

> Did you compare the speed of your approach to the loop-approach? For
> reasonably sized datasets I expect the loop to be faster. This was
> also true for the chunk index generation you mentioned.

Here's some test code (just does the histogram, since the reverse_indices come out in different forms and it will depend what exactly you want to do with them). The scaling depends on ndatasets, datalen, and datarange. I had Ed's problem in mind, where he said each data set corresponded to a pixel in a large image, so there could be millions of data sets, and datalen seemed low since he was willing to append them all by hand. First, the punch line:

ndatasets	datalen	datarange	Loop-speed	Vectorized-speed
10	10	4	5.8e-5	3.0e-5
10000	10	4	0.014	0.0031
10000000	10	4	12	3.5
10	10000	4	0.0010	0.0029
10	10000000	4	0.81	2.8
1000	1000	4	0.0044	0.031
10	10	100	5.4e-5	4.2e-5
10000	10	100	0.014	0.0086
10000000	10	100	11.5	9.1
10	10000	100	0.0011	0.0030
10	10000000	100	0.80	2.8
10000	10	10000	0.39	0.40
10	10000	10000	0.0016	0.0035

So, if datarange is low and ndatasets is larger than datalen, the vectorized version wins. If datalen is larger than ndatasets, the loop wins no matter what. If datarange is large, the loop usually wins or comes close (but note that datarange never needs to be larger than the product of ndatasets and datalen - if it is, use uniq+value_locate to remap the values into a smaller range). The best approach definitely depends on your problem!

Here's the code:

```
ndatasets=10l
datalen=10l
datarange=4l

datasets = round(randomu(seed,datalen,ndatasets)*datarange)
```

```
; needed for both approaches, so leave it out of timing
minval=min(datasets, max=maxval)

; loop version
t0=systime(/sec)
for i=0l,ndatasets-1 do h=histogram(datasets[*,i], min=minval,
max=maxval)
t1=systime(/sec)
print, 'Loop approach: ',t1-t0

; vectorized version
t0=systime(/sec)
dataspan=maxval-minval
datasets -= minval
h = reform(histogram(datasets + rebin(transpose(ul64indgen(ndatasets)
*dataspan), $
  size(datasets,/dimen)), min=0, max=ulong64(ndatasets)*dataspan-1),
dataspan, $
  ndatasets)
t1=systime(/sec)
print, 'Vectorized approach: ',t1-t0
```

-Jeremy.