

---

Subject: Re: averaging over same index  
Posted by [btt](#) on Tue, 23 Aug 2005 16:45:12 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

andi.walther@web.de wrote:

> Hello,  
>  
> I have an array V(alues) and an array S(ubscripts of a target array)  
> and I want to  
> extract the mean of all values with the same index in order to put them  
> into the new array.  
>  
> As a simple example:  
> V = [ 3, 7, 99, 5, 2 , 10]  
> S = [ 1, 3, 3, 2, 0 ,1]  
>  
> new vector should be --> new = [ 2 , 6.5 , 5, 53 ]  
>  
> slow way would be: for n = 0 , max(S)-1 do new[n]=mean(v[where(S eq  
> n)])  
>  
> a bit faster without WHERE in the loop:  
>  
> VSorted = v[sort(S)]  
> SSorted = S[sort(S)]  
> uu = uniq(iSorted)  
> for n = 0 , n\_elements(uu)-1 do new[iSorted[uu[n]]] =  
> mean(vSorted[n:(n-1)>0])  
>  
> But is there a way to do what I want without resorting to  
> a loop? In my real-world problem I have vectors with 500000 elements  
> and the number of the occurence of indices  
> are quite irregular and can exceed 30 times.  
>  
> Thanks Andi  
>

Hello,

You might take a look at histogram. Something like the following quick knock-off might work.

PRO testmean

```
V = [ 3, 7, 99, 5, 2 , 10]
S = [ 1, 3, 3, 2, 0 ,1]
```

```
h = HISTOGRAM(s, rev = r, loc = loc)
nH = n_elements(h)
newV = FltArr(nH)
```

```
for i = 0, nH-1 Do begin
    if h[i] GT 0 then $
        newV[i] = MEAN(V[r[r[i]] : r[i+1]-1]))
endfor
```

```
print, newV
```

```
end
```

```
IDL> testmean
```

```
% Compiled module: TESTMEAN.
```

```
% Compiled module: MEAN.
```

```
% Compiled module: MOMENT.
```

```
    2.00000    6.50000    5.00000    53.0000
```

```
Cheers,  
Ben
```

---