
Subject: Newbie question concerning summations/loops in IDL

Posted by [mbweller](#) on Tue, 29 Jul 2008 23:12:49 GMT

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

Hello,

I have need of some experienced users with sort of a newbie question.

I am writing a code that needs a summation in it, this is what I have thus far:

```
v=    ; volume of region
a=    ; area of region
o= 60*pi/180    ; fault dip angle
g=    ; scaling factor
t= 150    ; elastic lithosphere thickness
h=    ; depth of faulting
```

```
ind_small = where(thaext[1,*] lt t)
ind_large = where(thaext[1,*] ge t)
thaext_small = thaext[:,ind_small]
thaext_large = thaext[:,ind_large]
```

```
ens=(sin(o)*cos(o)/v)*    ; horizonatal normal strain for small faults
enl=(cos(o)/a)*    ; horizonatal normal strain for
large faults
evs=(-sin(o)*cos(o)/v)*    ; vertical normal strain for small faults
evl=(-cos(o)/a)*    ; vertical normal strain for large faults
```

The summation needs to be after * in the ens, enl, evs and evl fields.

It must be of the form:

summation N, i=0 [Di Li Hi] for small faults, where N = ind_small, Hi= T/sin(o) and

summation N, i=0 [Di Li] for large faults, where N=ind_large

Could anyone provide any insight/guidance?

Thanks,
~Matt

Subject: Re: Newbie question concerning summations/loops in IDL

Posted by [Chris\[6\]](#) on Thu, 31 Jul 2008 00:52:34 GMT

On Jul 30, 1:55 pm, mbwel...@gmail.com wrote:

> On Jul 30, 3:57 am, Jeremy Bailin <astroco...@gmail.com> wrote:

>

>

>

>> On Jul 30, 4:33 am, Wox <nom...@hotmail.com> wrote:

>

>>> On Tue, 29 Jul 2008 23:19:19 -0700 (PDT), mbwel...@gmail.com wrote:

>>>> On Jul 29, 7:27 pm, Chris <beaum...@ifa.hawaii.edu> wrote:

>>>> > On Jul 29, 1:12 pm, mbwel...@gmail.com wrote:

>

>>>> > > Hello,

>

>>>> > > I have need of some experienced users with sort of a newbie question.

>

>>>> > > I am writing a code that needs a summation in it, this is what I have

>>>> > > thus far:

>

>>>> > > v= ; volume of region

>>>> > > a= ; area of region

>>>> > > o= 60*pi/180 ; fault dip angle

>>>> > > g= ; scaling factor

>>>> > > t= 150 ; elastic lithosphere thickness

>>>> > > h= ; depth of faulting

>

>>>> > > ind_small = where(thaext[1,*] lt t)

>>>> > > ind_large = where(thaext[1,*] ge t)

>>>> > > thaext_small = thaext[:,ind_small]

>>>> > > thaext_large = thaext[:,ind_large]

>

>>>> > > ens=(sin(o)*cos(o)/v)* ; horizonatal normal strain for small faults

>>>> > > enl=(cos(o)/a)* ; horizonatal normal strain

for

>>>> > > large faults

>>>> > > evs=(-sin(o)*cos(o)/v)* ; vertical normal strain for small faults

>>>> > > evl=(-cos(o)/a)* ; vertical normal strain for

large faults

>

>>>> > > The summation needs to be after * in the ens, enl, evs and evl

>>>> > > fields.

>>>> > > It must be of the form:

>>>> > > summation N, i=0 [Di Li Hi] for small faults, where N = ind_small, Hi=

>>>> > > T/sin(o) and

>>>> > > summation N, i=0 [Di Li] for large faults, where N=ind_large

>

>>>> > > Could anyone provide any insight/guidance?

>

```

>>>> > > Thanks,
>>>> > > ~Matt
>
>>>> > I don't know what some of your variables are (Li? Di?), but you might
>>>> > want to look at TOTAL() to start- you can use that to do most
>>>> > summation tasks.
>
>>>> L and D are data from a ascii table that is already ready in, while i
>>>> is the indice of the summation. I've looked at total, but the examples
>>>> were sorely lacking. I was hoping that perhaps a useful example, given
>>>> my code and desire, could be supplied.
>
>>>> ~Matt
>
>>> I'm not sure what you mean with "summation N, i=0 [Di Li Hi] ... where
>>> N=ind_small". The index i goes from 0 to what? And what are you
>>> summing? D[i]*L[i]*H[i]?
>
>> Okay, if I understand it correctly, then what you're saying is that in
>> ind_small you multiply by an extra factor of t/sin(o) inside the sum,
>> but not in ind_large?
>
>> H = replicate(1., n_elements(D))
>> H[ind_small] = t/sin(o)
>> summation = total(D*L*H)
>
>> Is that what you're looking for?
>
>> (by the way, look up !RADEG).
>
>> -Jeremy.
>
> Thanks for the responses.
> I think that I did not adequately explain what I needed to do, Let me
> be more specific now. (this might be a little complicated)
>
> I have a .sav file which is a FLOAT array[2,7923] but may go as high
> as [2,18000] and the forms are as such: [id, Length].
>
> ind_small and ind_large are where I select the lengths to be smaller
> or larger, respectively, than t. Then place them back into the new
> matrices thaext_small and thaext_large. (not completely sure if this
> is necessary.)
>
> Now comes the part that I am a little confused on how to program.
>
> ens, enl, evs and evl fields are going to be a constant * a summation
> (which will be different for all four).

```

>
> The number of sums or (N) needs to be equal to the number of the
> faults down selected by Ind_small (or since ind_small =
> where(thaext[1,*] lt t), it needs to sum the number of the second
> column in the array). This number will be different for bot the large
> and small cases (eg. ind_large = where(thaext[1,*] ge t)). So, i then
> should be # of points in column 2 of ind_small/ind_large - 1 (I would
> think).
>
> The summation is $[D[i]*L[i]*H[i]]$ for small faults and the summation
> is $[D[i]*L[i]]$ for large faults, where:
> $D[i]=C[i]*L[i]$ for small faults and
> $D[i]=C[i]*H[i]$ for large faults,
> $L[i]=$ length (from column 2 of thaext_small/thaext_large) and
> $H[i]=(1/2 \text{ or } 1/3)*L[i]$ for small faults and
> $H[i] = t/\sin(o)$ for large faults and
> $C[i]$ may or may not be a constant
>
> This should now read as constant * summation $[C[i]*L[i]*L[i]*L[i]]$ for
> small faults and constant * summation $[C[i]*L[i]]$ for large
> faults.
>
> I think that's everything I need to be able to do, hopefully it's a
> bit clearer now.
>
> Thanks,
> ~Matt

Its still a bit confusing to me, but let me guess at what you're
trying to do (sorry if I'm still off)

You have some vector of ID's and lengths. You want to select out
certain rows of this array based on the value of lengths. You then
want to use these row numbers as reference to other arrays (like C and
L, which i assume are the same length as the ID/Length array). You
want to multiply the selected tables together and sum up the rows
which pass this criteria (some test on the length, like gt t). Is that
close? If so, do something like this

id ; [id, length] vector

a ; random vector 1

b ; random vector 2

want to sum the product of a and b for the rows for which the
corresponding id length is greater than, say 5

test=where(id[1,*] gt 5, count)

if count ge 0 then result=total(a[test]*b[test]) else print, 'No good
rows!'

return, result

am i getting close yet?

chris

Subject: Re: Newbie question concerning summations/loops in IDL

Posted by [Wox](#) on Thu, 31 Jul 2008 08:37:58 GMT

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

On Wed, 30 Jul 2008 16:55:35 -0700 (PDT), mbweller@gmail.com wrote:

<snip>

> The number of sums or (N) needs to be equal to the number of the
> faults down selected by ind_small (or since ind_small =
> where(thaext[1,*] lt t), it needs to sum the number of the second
> column in the array). This number will be different for bot the large
> and small cases (eg. ind_large = where(thaext[1,*] ge t)). So, i then
> should be # of points in column 2 of ind_small/ind_large - 1 (I would
> think).

>
> The summation is $[D[i]*L[i]*H[i]]$ for small faults and the summation
> is $[D[i]*L[i]]$ for large faults, where:
> $D[i]=C[i]*L[i]$ for small faults and
> $D[i]=C[i]*H[i]$ for large faults,
> $L[i]$ = length (from column 2 of thaext_small/thaext_large) and
> $H[i]=(1/2 \text{ or } 1/3)*L[i]$ for small faults and
> $H[i] = t/\sin(o)$ for large faults and
> $C[i]$ may or may not be a constant

>
> This should now read as constant * summation $[C[i]*L[i]*L[i]*L[i]]$ for
> small faults and constant * summation $[C[i]*L[i]]$ for large
> faults.

>
> I think that's everything I need to be able to do, hopefully it's a
> bit clearer now.

>
> Thanks,
> ~Matt

Still not clear to me! Check the code below for what Chris and I think you want. The step from ARRsmall to Ssmall is just a summation with the "total"-function. So somehow I think you want something else. Do you know LaTeX? Maybe you can write what you need in LaTeX. So the IDL code below does this (LaTeX):

```

\begin{eqnarray}
\mathrm{Ssmall} \quad \&= \sum_{i=0}^{\mathrm{nsmall}-1} \\
C_{\mathrm{indsmall}_{i}} \cdot L_{\mathrm{indsmall}_{i}} \cdot \\
\frac{t}{\sin o} \quad \&= \sum_{i=0}^{\mathrm{nlarge}-1} \\
C_{\mathrm{indlarge}_{i}} \cdot L_{\mathrm{indlarge}_{i}}^3 \cdot \\
\frac{1}{2} \\
\end{eqnarray}

```

So Ssmall is 1 number and Slarge is 1 number (you can bring the constants outside the sum off course). You say above there are (N) number of sums... I don't get it.

pro test

```

v= 1. ; volume of region
a= 1. ; area of region
o= 60*pi/180 ; fault dip angle
t= 150 ; elastic lithosphere thickness
n=100 ; number of points

; [id,length between 0 and 300]
thaext=[lindgen(1,n),reform(round(IMSL_RANDOM(n)*300),1,n)]

; some numbers
C=reform(round(IMSL_RANDOM(n)*10),1,n)

; indices for small and large lengths
ind_small=where(thaext[1,*] lt t,$
nsmall,comp=ind_large,ncomp=nlarge)

; arrays for summation
ARRsmall=C*thaext[1,ind_small]*(t/sin(o))
ARRlarge=C*(thaext[1,ind_large])^3./2

; summation
Ssmall=total(ARRsmall,/pres)
Slarge=total(ARRlarge,/pres)

; horizontal normal strain for small faults
ens=(sin(o)*cos(o)/v)*Ssmall

```

```
; horizontal normal strain for large faults
enl=(cos(o)/a)*Slarge
; vertical normal strain for small faults
evs=(-sin(o)*cos(o)/v)*Ssmall
; vertical normal strain for large faults
evl=(-cos(o)/a)*Slarge

end
```

Subject: Re: Newbie question concerning summations/loops in IDL

Posted by [Wox](#) on Thu, 31 Jul 2008 08:40:12 GMT

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

```
<snip>
> ; arrays for summation
> ARRsmall=C*thaext[1,ind_small]*(t/sin(o))
> ARRlarge=C*(thaext[1,ind_large])^3./2
<snip>
```

must be this:

```
; arrays for summation
ARRsmall=C[ind_small]*thaext[1,ind_small]*(t/sin(o))
ARRlarge=C[ind_large]*(thaext[1,ind_large])^3./2
```

Subject: Re: Newbie question concerning summations/loops in IDL

Posted by [mbweller](#) on Thu, 31 Jul 2008 18:49:38 GMT

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

On Jul 31, 1:40 am, Wox <nom...@hotmail.com> wrote:

```
> <snip>; arrays for summation
>> ARRsmall=C*thaext[1,ind_small]*(t/sin(o))
>> ARRlarge=C*(thaext[1,ind_large])^3./2
>
> <snip>
>
> must be this:
>
> ; arrays for summation
> ARRsmall=C[ind_small]*thaext[1,ind_small]*(t/sin(o))
> ARRlarge=C[ind_large]*(thaext[1,ind_large])^3./2
```

Alrighty,

After fiddling around for sometime I got it to work for me. I wanted

to thank everyone for the their help.

I did have another question though:

If I want to be able to allow the user to input their own variables and their own datafile after running the script: How might I go about doing that?

Thanks,
~Matt

Subject: Re: Newbie question concerning summations/loops in IDL
Posted by [Jean H.](#) on Tue, 05 Aug 2008 20:52:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

mbweller@gmail.com wrote:

```
> On Jul 31, 1:40 am, Wox <nom...@hotmail.com> wrote:
>> <snip>>; arrays for summation
>>> ARRsmall=C*thaext[1,ind_small]*(t/sin(o))
>>> ARRlarge=C*(thaext[1,ind_large])^3./2
>> <snip>
>>
>> must be this:
>>
>> ; arrays for summation
>> ARRsmall=C[ind_small]*thaext[1,ind_small]*(t/sin(o))
>> ARRlarge=C[ind_large]*(thaext[1,ind_large])^3./2
>
> Alrighty,
>
> After fiddling around for sometime I got it to work for me. I wanted
> to thank everyone for the their help.
>
> I did have another question though:
>
> If I want to be able to allow the user to input their own variables
> and their own datafile after running the script: How might I go about
> doing that?
>
> Thanks,
> ~Matt
Have a look at
READ and DIALOG_PICKFILE
```
