Subject: Comparing structures
Posted by cab581 on Fri, 21 Feb 2014 21:52:53 GMT
View Forum Message <> Reply to Message

Hi everyone,

I have two structures that are made up of several fields. The last field (5) is the array which contains the measurement itself and the others contain information about the measurement (much like the star example on the Exelis website http://www.exelisvis.com/docs/Creating_and_Defining_St.html).

The two structures that I have need to be compared, but I'll use the star example from Exelis to explain what I want to do. Say I have two structures, one contains the masses of the star and the other contains the volumes and I want to work out the density of each part of the star at altitude i (OK, the astronomy in the example is a bit off, but it works), but the list of stars is not the same (the measurements may have been incomplete), so the star id (given by letters) fields may be

strucA.(1) = [a, b, c, f, g, i, m, n, p, q, r, s, w, x, y, z]
strucB.(1) = [b, d, e, f, h, i, j, l, m, q, r, s, t, v, w, x, y, z]

I tried to use something like,

```
for i=0,50
match = where(strucA.(1) eq strucB.(1), count)
if count le 0.0 then strucA[match].(5)[i] = !values.f_nan
density[i]=strucA/strucB
endfor
```

What I'm trying to say is where I have a, b, c, etc in the first row, find the corresponding letter in the second row and do the calculation.

However when I do this all it does is compare the nth letter in the first row to the nth letter in the second row, so the only results I'm getting in the example above are from f, q, r and s so I'm missing out on b, m, w, x, y, z.

Any ideas would be greatly appreciated. If I haven't explained myself properly then I'd be glad to make it clearer.

Subject: Re: Comparing structures
Posted by David Fanning on Fri, 21 Feb 2014 22:13:20 GMT
View Forum Message <> Reply to Message

cab581 writes:

> I have two structures that are made up of several fields. The last field (5) is the array which contains the measurement itself and the others contain information about the measurement (much like the star example on the Exelis website

http://www.exelisvis.com/docs/Creating_and_Defining_St.html).
>
> The two structures that I have need to be compared, but I'll use the star example from Exelis to explain what I want to do. Say I have two structures, one contains the masses of the star and the other contains the volumes and I want to work out the density of each part of the star at altitude i (OK, the astronomy in the example is a bit off, but it works), but the list of stars is not the same (the measurements may have been incomplete), so the star id (given by
letters) fields may be
>
> strucA.(1) = [a, b, c, f, g, i, m, n, p, q, r, s, w, x, y, z]
> strucB.(1) = [b, d, e, f, h, i, j, l, m, q, r, s, t, v, w, x, y, z]
>
> I tried to use something like,
>
> for i=0,50
> match = where(strucA.(1) eq strucB.(1), count)
> if count le 0.0 then strucA[match].(5)[i] = !values.f_nan
> density[i]=strucA/strucB
> endfor
>
> What I'm trying to say is where I have a, b, c, etc in the first row, find the corresponding letter in the second row and do the calculation.
>
> However when I do this all it does is compare the nth letter in the first row to the nth letter in the second row, so the only results I'm getting in the example above are from f, q, r and s so I'm missing out on b, m, w, x, y, z.
>
> Any ideas would be greatly appreciated. If I haven't explained myself properly then I'd be glad to make it clearer.

It might take a week or so to sort out the problems with this code, but here is a place to start. There is a famous Where function gotcha that occurs when there are two arrays on either side of the EQ operator in the Where function.

http://www.idlcoyote.com/misc_tips/noidea.html

In this case, the Where function seldom (if ever) does what you think it is doing. I would walk though this code with a very fine sieve and see if you can discover the places where your logic is leaking out.

Cheers,

David

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

---

Subject: Re: Comparing structures
Posted by cab581 on Fri, 21 Feb 2014 22:49:03 GMT
View Forum Message <> Reply to Message

On Friday, February 21, 2014 5:13:20 PM UTC-5, David Fanning wrote:
> cab581 writes:
>
>
>
>> I have two structures that are made up of several fields. The last field (5) is the array which contains the measurement itself and the others contain information about the measurement (much like the star example on the Exelis website http://www.exelisvis.com/docs/Creating_and_Defining_St.html).
>
>>
>
>> The two structures that I have need to be compared, but I'll use the star example from Exelis to explain what I want to do. Say I have two structures, one contains the masses of the star and the other contains the volumes and I want to work out the density of each part of the star at altitude i (OK, the astronomy in the example is a bit off, but it works), but the list of stars is not the same (the measurements may have been incomplete), so the star id (given by
>
> letters) fields may be
>
>>
>
>> strucA.(1) = [a, b, c, f, g, i, m, n, p, q, r, s, w, x, y, z]
>
>> strucB.(1) = [b, d, e, f, h, i, j, l, m, q, r, s, t, v, w, x, y, z]
>
>>
>
>> I tried to use something like,
>
>>
>
>> for i=0,50
>

>> match = where(strucA.(1) eq strucB.(1), count)
>
>> if count le 0.0 then strucA[match].(5)[i] = !values.f_nan
>
>> density[i]=strucA/strucB
>
>> endfor
>
>>
>
>> What I'm trying to say is where I have a, b, c, etc in the first row, find the corresponding letter in the second row and do the calculation.
>
>>
>
>> However when I do this all it does is compare the nth letter in the first row to the nth letter in the second row, so the only results I'm getting in the example above are from f, q, r and s so I'm missing out on b, m, w, x, y, z.
>
>>
>
>> Any ideas would be greatly appreciated. If I haven't explained myself properly then I'd be glad to make it clearer.
>
>
>
> It might take a week or so to sort out the problems with this code, but
>
> here is a place to start. There is a famous Where function gotcha that
>
> occurs when there are two arrays on either side of the EQ operator in
>
> the Where function.
>
>
>
>  http://www.idlcoyote.com/misc_tips/noidea.html
>
>
>
> In this case, the Where function seldom (if ever) does what you think it
>
> is doing. I would walk though this code with a very fine sieve and see
>
> if you can discover the places where your logic is leaking out.
>
>
>

> Cheers,
>
>
>
> David
>
>
>
> Cheers,
>
>
>
> David
>
>
>
>
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Thanks for this, I'm going to spend the weekend trying to solve this using the where function, just for my own satisfaction. If anyone can spot a glaring mistake or some other method of doing the same thing that I may be overlooking then I'd me most appreciative.

I have a few other lines of code in my program that utilized 'where' and I have come across that function's rather confusing results before.

Thanks again

---

Subject: Re: Comparing structures
Posted by Matthew Argall on Sat, 22 Feb 2014 02:56:04 GMT
View Forum Message <> Reply to Message

I think you want to use Value_Locate() instead of Where() in this case (remember to sort).

http://www.idlcoyote.com/code_tips/valuelocate.html