Subject: Re: Unique combinations from a 1d array
Posted by David Fanning on Wed, 14 Jan 2004 22:38:16 GMT
View Forum Message <> Reply to Message

Darren writes:

> Does anyone know of a more efficient means to determine the set of all
> unique combinations of 2 from a 1d array? The following is an approach
> that works but for large arrays -say 3000 or more elements it is very
> slow. Part of the problem is due to memory because the number of

> total number of combinations is 4498500. Writing the paired difference
> results to a temporary file helped considerably, but is still far too
> slow. Any ideas would be much appreciated.
>
> Here is the code I have:
>

> n = n_elements(X)
> d = make_array(1, /float)
> for i=0, n-1 do for j=0, n-1 do begin
> if i le j then begin
> d = [d, X[i] - X[j]]
> endif
> endfor
> d = d[1:n-1]

Here is a method that gets the same answer as
your code. (Although I can't convince myself it
does what you *say* it does!)

```
x = RandomU(-3L, 10) * 10
```

Darren's method:

```
% Compiled module: $MAIN$.
IDL> .go
   0.000000     3.39667     1.30986     3.08815     8.37598
   0.751965     8.60027     6.79858     7.55522
```

My method:

```
IDL> Print, ( x[0] - Shift(x,1) )[1:*]
   0.000000     3.39667     1.30986     3.08815     8.37598
   0.751965     8.60027     6.79858     7.55522
```

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

---