
Subject: Re: missing data

Posted by [velt](#) on Wed, 15 Jun 1994 20:36:27 GMT

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

In article lp3@ncar.ucar.edu, caron@acd.ucar.edu (John Caron) writes:

- > 1) efficiency of where():
- >
- > Can anyone comment on the efficiency of using:
- >
- > max = max(my_arr(where(my_arr ne missing_data)))
- >
- > to find the maximum value of an array, but excluding
- > missing values?

It seems fairly trivial, depending on how you code the missing data. As long as the missing data has a value less than the maximum value in your array (which is guaranteed if your missing data is less than all not-missing data), you can remove the where all together. Where is pretty efficient though.

- >
- > the array here has size 63000 elements, and only a few (0 - 30)
- > are "missing". I am imagining that the where() command returns
- > an array of indices, which then is used to grab just those indices
- > out of my_arr.
- >
- > In a higher level language I could make a single pass through my_arr
- > to find the max value, excluding the missing data as I go.

I guess you mean a *lower* level language

- > Is there a more efficient way to do this in IDL, or is this just
- > the price we pay for its generality?

I don't know if any of the following are more efficient (likely not), but there are more general ways of dealing with the problem. You can think of the histogram function, or the uniq function. Each of these will give you (after simple manipulation) a list of values present, which you can then test against not-allowed values. If you have only one not-allowed value (such as a missing value), you are best off with your line.

Robert Velthuisen.
