Subject: Re: rebin and !values.f_nan
Posted by James Kuyper on Mon, 16 Jul 2007 04:04:35 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> kuyper writes:
>
>> I haven't had to do this with IDL, so I didn't realize that IDL
>> handled it inconveniently. What I would normally want to have happen
>> when re-binning data with NaNs is that every element of the output
>> array whose calculation involved one of the NaN's in the input array
>> would itself contain a NaN, while all the other elements of the array
>> would be calculated normally. That's not something that could be
>> achieved by the approach you suggest. If that's a feature not already
>> provided as an option by rebin, it should be.
>
> In my little experiment, that is *exactly* how it
> worked. And how I would have expected it to work, too.
> How could IDL do anything else and not be accused of
> "created data where none exists."

I've been at home this weekend, without access to IDL, so I hadn't
gotten around to testing it. However, what Nick diescribed was not the
creation of data where none exists, but the destruction of usable data
by rebin():

> If I use Rebin and there is a NaN value, new array becomes also NaN.

I understood that to mean that the entire re-binned array was set to
NaN, not just isolated portions of it.