Subject: Re: IDL 4.0.1, best way to deal with missing/bad data
Posted by rfinch on Wed, 14 Feb 1996 08:00:00 GMT
View Forum Message <> Reply to Message

>> The question comes as to the best way to handle missing/bad data
>> within IDL. By handle I mean don't use the data in computations, and
>> don't plot it. I can think of three ways:

Mark> There is another way - use the IDL WHERE() function and the capability to
Mark> subscript and array with another.

Mark> read, data, flag
Mark> good = where(flag eq 0)
Mark> data = data(good)
Mark> plot, data

This doesn't fully do what I need, because the resulting plot doesn't
show gaps where missing data was, as we need here.

I've talked to RSI about this problem; they think that the next
release, all computational routines will recognize missing data with
the /NAN keyword, so all you have to do is replace your missing values
with NANs. For now I guess I will use the following construct:

```
a = [1, 2, 3, !VALUES.F_NAN]
result = computation(where(a(finite(a) eq 1)))
```

Of course, the where command doesn't fail gracefully so I'll have to
have a check above to make sure there are any good values.

Note to RSI: how about introducing the null matrix (like MATLAB), so
that we don't have to have a separate WHERE test? In other words,
instead of:

```
good_ndx=where(a eq good_val, count)
if count gt 0 then result=computation(a(good_ndx))
```

just use:

```
result=computation(a(where(a eq good_val)))
```

and if the WHERE returns null, it fails silently, and result is simply
a null vector or matrix too.
--
"Nada burra la chamaca." A.G.
Opinions expressed are mine, not my employer or news host.
rfinch@toe.cs.berkeley.edu