

---

Subject: the NaN effect :-|

Posted by [rkombiyil](#) on Tue, 12 Jun 2007 15:33:59 GMT

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

---

This stumps me.. We had some discussions on NaN's earlier, but mostly wrt 'TOTAL'

Lets say,

```
IDL>a=[6.2,12.5,14.1,0.,22,!values.f_nan]
```

```
IDL> print,max(a)
```

```
22.0000
```

```
IDL> print,min(a)
```

```
0.00000
```

; Now I increase number of NaN's in the array :D

```
IDL> a=[6.2,12.5,14.1,0.,!values.f_nan,22,!values.f_nan]
```

```
IDL> print,max(a)
```

```
22.0000
```

```
IDL> print,min(a)
```

```
0.00000
```

; go on, repeat this (it is 00:23 where I am @, so CARPE NOCTEM! ) :-P

```
IDL> a=[!values.f_nan,6.2,12.5,14.1,0.,!values.f_nan,22,!values.f _nan]
```

```
IDL> print,min(a)
```

```
NaN
```

```
IDL> print,max(a)
```

```
NaN
```

```
=====
```

Same goes for MIN also. ??!!

IDL's docu says:

<snip from IDL ref guide: Page 1269/4090>

If the MAX function is run on an array containing NaN values and the NAN keyword is not set, an invalid result will occur.

</snip>

The same is said for MIN also.

So the result (OPS with MIN/MAX) is directly proportional to the number of NaN's we eat? er, add to the array? :P

So what is right and what is wrong? Enlighten, please.

/metachronist

---