

Hi,

A couple of weeks ago it was pointed out that the ++ operator used in a vectorised form would work on repeated indices meaning that it's possible to do this:

```
IDL> a=[1,2]
IDL> ++a[[0,0,0,1,0,0]]
IDL> print,a
      6      3
```

here a[0] has been incremented five times because the index 0 appears five times.

It was pointed out that this meant that David's mode calculation example of:

```
array = [1, 1, 2 , 4, 1, 3, 3, 2, 4, 5, 3, 2, 2, 1, 2, 6, -3]
distfreq = Histogram(array, MIN=Min(array))
maxfreq= Max(distfreq)
mode = Where(distfreq EQ maxfreq) + Min(array)
Print, mode
```

could be re-written as:

```
array = [1, 1, 2 , 4, 1, 3, 3, 2, 4, 5, 3, 2, 2, 1, 2, 6,-3]
f=intarr(max(array)-min(array)+1)
f[array-min(array)]++
junk=max(f,idx)
mode=idx + min(array)
print,mode
```

(in fact I see David's article on this has already been updated to include the ++ solution). It was also shown that this solution wasn't any slower than using histogram. It's not necessarily better but in certain situations could be easier to read.

Anyway, it was pointed out that this meant that a++ behaves differently from a+=1 which some people didn't like and others didn't mind. Some of us (ok, mainly me) worried that using this behaviour of the ++ operator leading to problems later if it turned out that the behaviour wasn't intended by ITTVIS and they would try to rationalise ++ with +=.

I volunteered to contact them to query if the behaviour was intended and if it could be considered future-proof. The response included:

"the output of the following commands seem to be the expected one (and should not change in the future)

```
a = [1, 2]
a [[0, 0, 1, 0]]++
print,a"
```

so, as promised, this is me sharing the result of the query with the rest of the group. Everyone can feel safe about relying on the ++ operator to work on repeated indices.

Thanks,

Allan
