**All classes**

Create a class from the class\_template folder

Need to

Problems:

Define precise rules for all the standard classes, templates, so no ambiguities (e.g. Horace objects, when is S=0, ERR=0 if npix=0 is a good example)

~~Create functions to automatically create binary operations, and to update generic routines from class template: problem is that currently have to make changes in sigvar and in class operator if change binary operations and class template.~~ DONE

**SIGVAR**

Needed by binary arithmetic:

sigvar

sigvar\_size

sigvar\_set

Needed by other functions:

sigvar\_get needed by multifit

sigvar\_getx needed by multifit if do not have a mask\_points constructor

needed by xye constructor

Problems:

* Need to ensure that sigvar\_get and sigvar\_getx are fully synchronised. How ensure? How about a single function that returns x,y,e,m, which in turn is called by sigvar\_get and sigvar\_getx. Probably only used by multifit, and xye, so can retro-adjust and remove sigvar\_get and sigvar\_getx. Just retain them in multifit for backward compatibility.
* Not clear if sigvar\_get returns mask array with 1 meaning keep or remove
* sigvar\_set can be used to make an invalid object. Should we force a check with isvalid, or just declare that sigvar is a special case and demand that care is taken.
* Creeate xye, xyem classes? Might make multifit neater.

**Multifit**

Problems:

* No documentation on form of fit function if x-y-e triple
* If nested function e.g. resolution function, then cannot vary resolution function parameters, I think
* Need to be able to apply functional constraints, not simply ratio binding. Do we want bounds too?