**State-space assessment project notes**

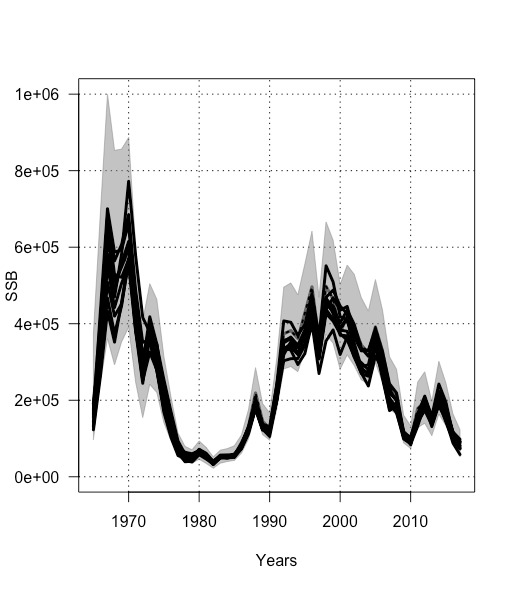
**10/2**

Project started. This notebook tracks progress and decisions made.

**10/10**

Herring run 13 model is up and running. There was a problem with trying to run it without all of the data files that was causing R to crash. My solution was to generate some default data files and go from there. Might be an operating system issue but seems easy enough to bypass for now.

I did some quick simulation self-tests using the SAM "simulate" feature which allows you to simulate from a fitted sam model (see end of herring\_run13.R).



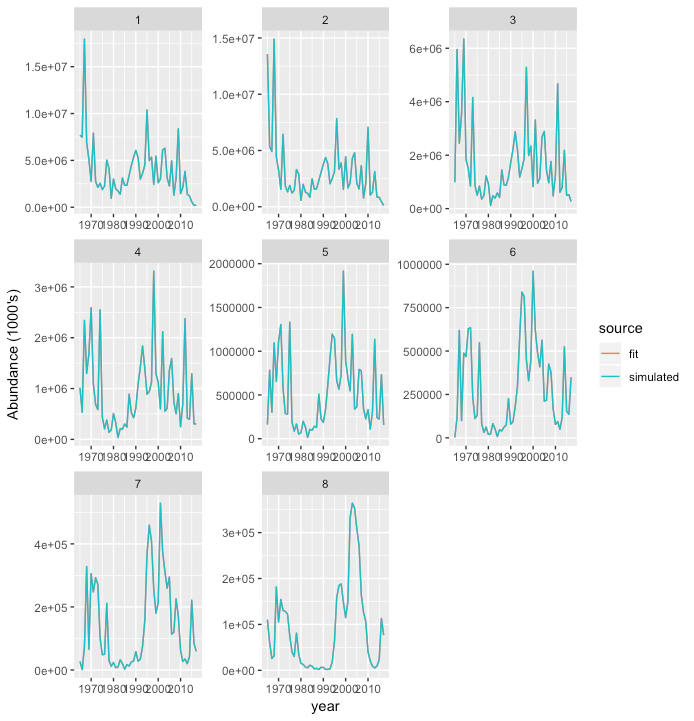
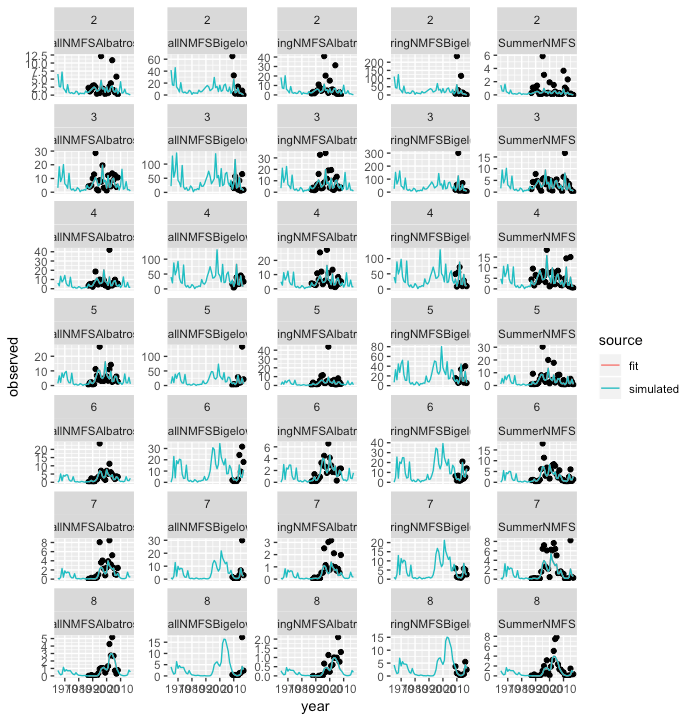
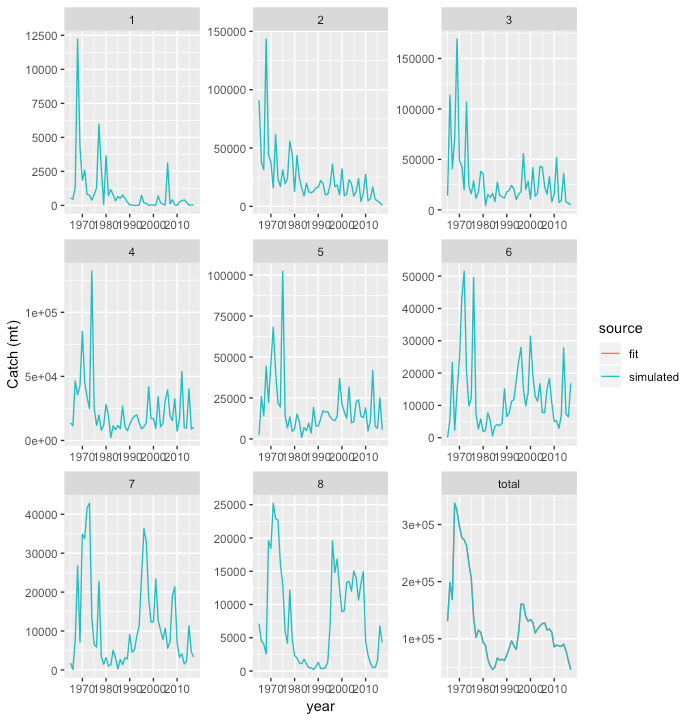
**10/17**

Been working on coding up the SAM model in R so we can then recreate the self-tests of the simulate feature. To get started I'm using the Atl herring run 13 fit as the parameter set of the simulation. You can check out the code here: https://github.com/perretti/state-space-tests/tree/master/simulation.

My goal right now is to verify I have the model equations right by recreating the fit time series exactly. So far I've be able to recreate the N-at-age fit time series by back-calculating the fit process errors and simulating forward from the fit initial conditions. And I've been able to do the same for total catch. Now I'm working on recreating the survey fits.

**10/24**

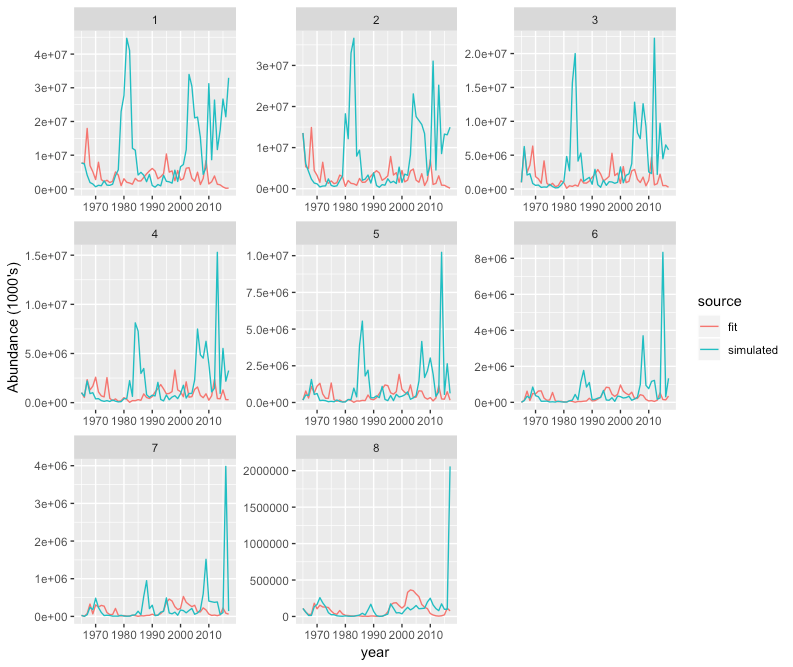
Finished coding the deterministic SAM equations and checked them against the SAM fit output. The two lines are identical so I'm convinced I have the deterministic component of the model correct.



Simulations results in large swings in abundance. The reason the simulate.sam() feature gives such well-constrained simulations is it simulates, "conditional on estimated values of F and N, rather than also simulating F and N forward from their initial values". In other words, it doesn't simulate N or F, but rather just uses the fit values.

In order to generate a simulation that actually simulates N and F you have to call sam.fit with sim.condRE = FALSE, and then pass that fit to simulate.sam().

Now, if you try to fit a sam model to these simulations, about half the time it crashes with a gradient error. I think that's because it can't fit to the wild swings implied by the fit.



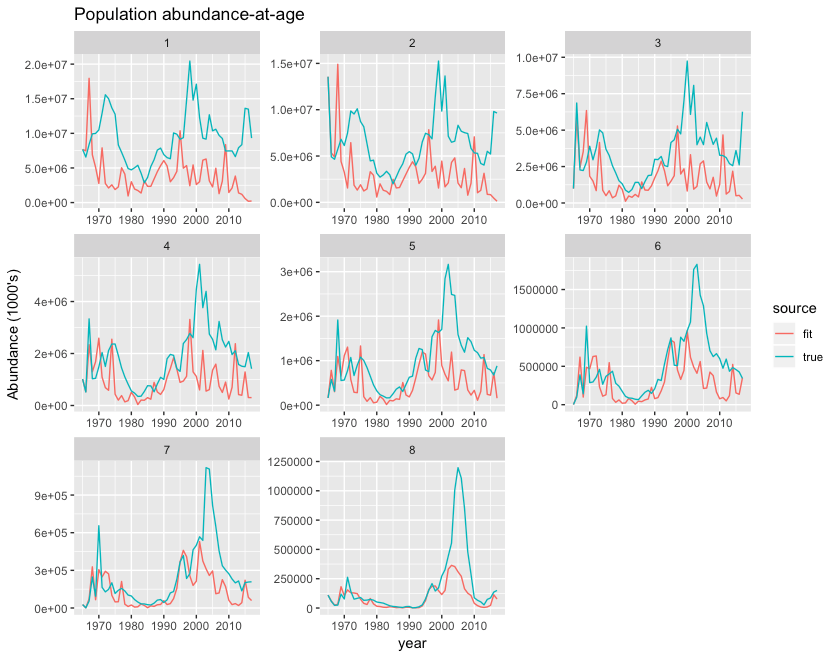
Here is my best guess for what's going on with the wild swings--

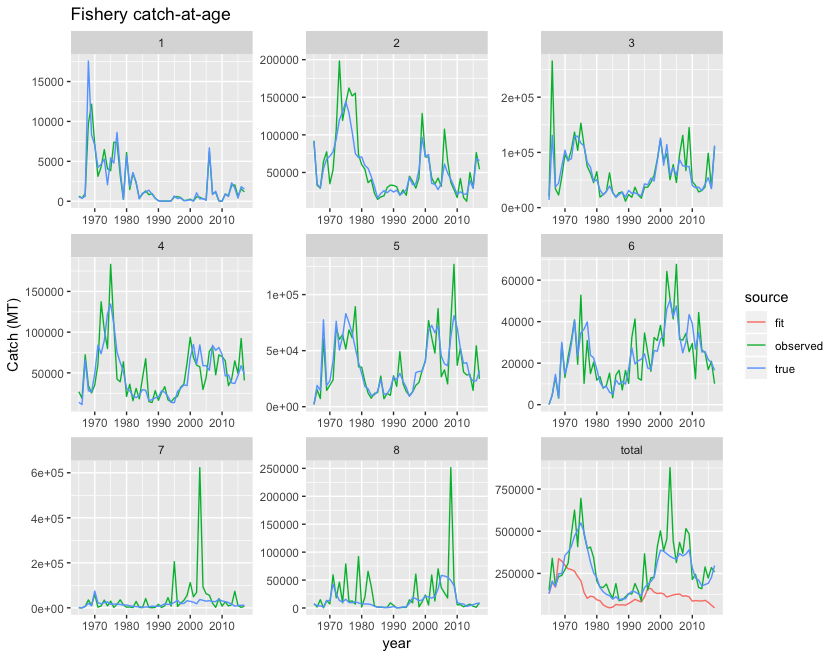
https://github.com/perretti/state-space-tests/blob/master/random\_walk\_case\_study/random\_walk.R

Estimates on the natural scale are skewed toward higher values due to the log-transformation.

Plan is to run some self-tests next.

**11/6**

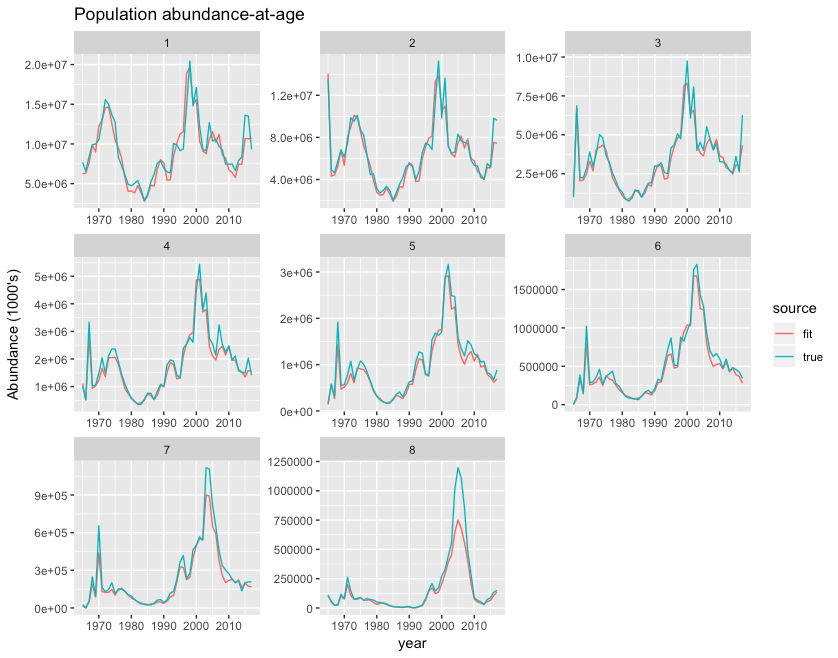
To give you a sense of how a typical simulation replicate compares to the original herring 13 fit, here is one example vs the original (so the fit is \*not\* to the simulation here): 

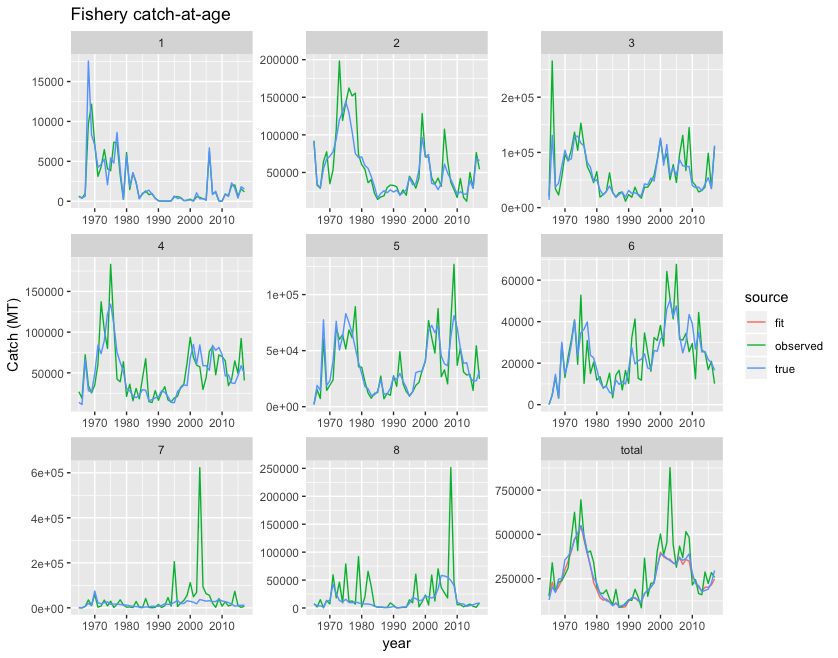


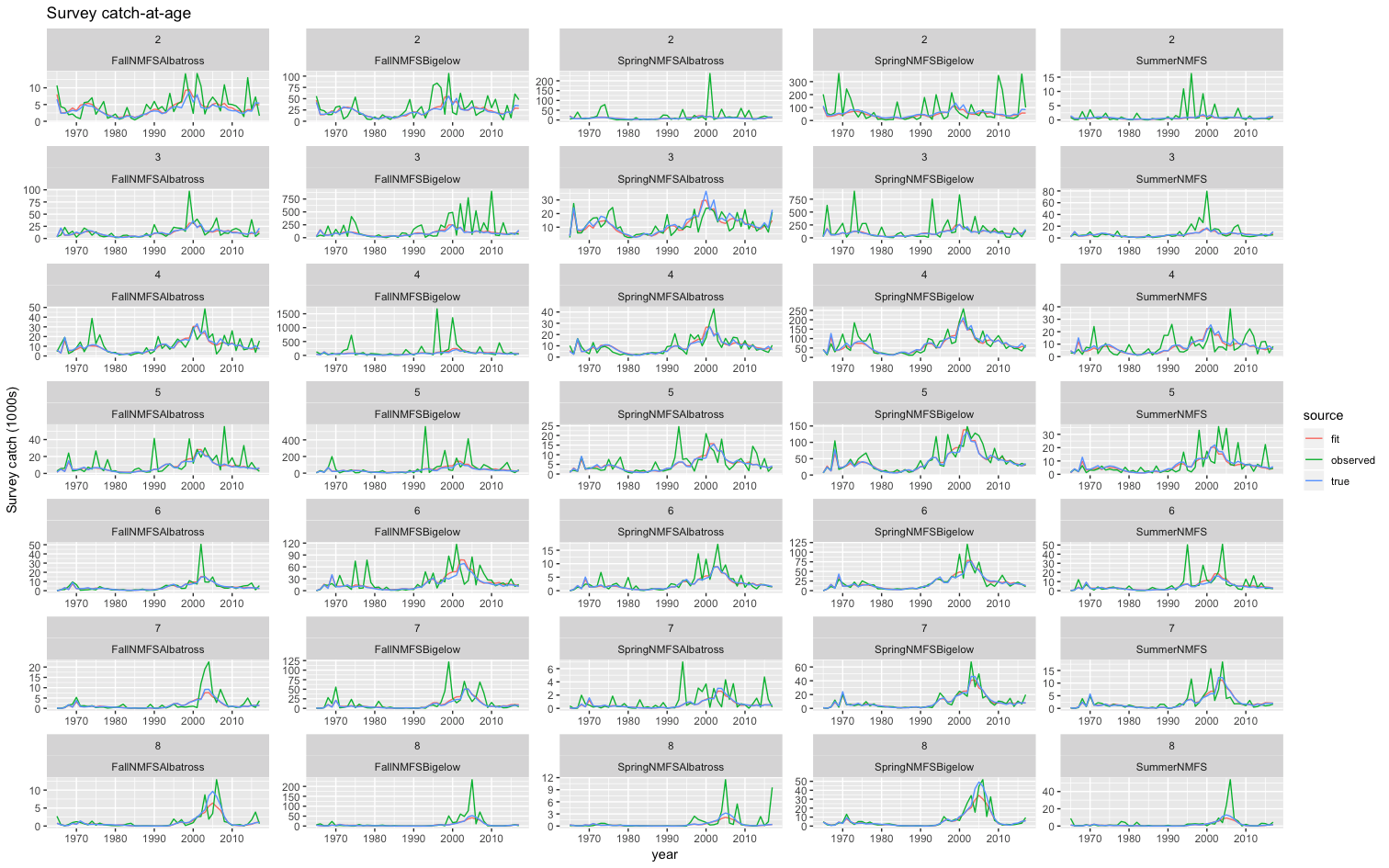


Notice that there are some pretty large observation errors. This is using the herring run 13 observation sd's. The true values sometimes go pretty high too, but that's just due to the process error issue we spoke about. We could also damp down the process error further.

Next let's look at the same replicate but this time with SAM fit to it.

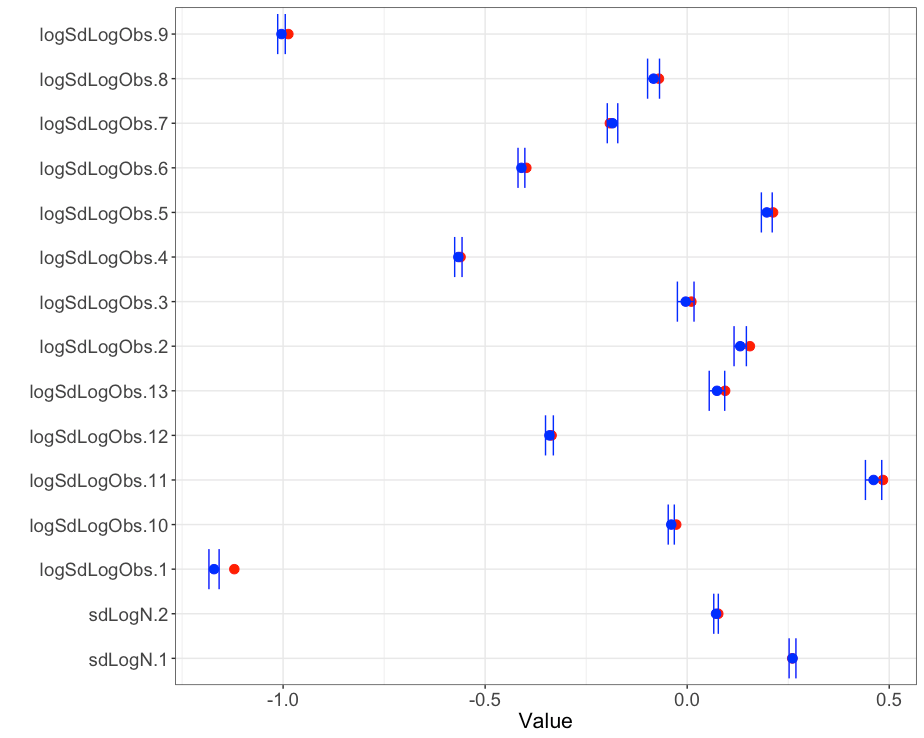






It seems to fit pretty well.

Next I did 100 replicates with this setup. Below is a plot of the true parameters (red) vs mean fit with 95% CI (blue). I just plotted the observation error variances and the log-N process error variance.



Overall it fits pretty well. Only one is significantly off (logSdLogObs.1), but there does seem to be a general pattern of slightly underestimating error.