So ICM takes a few fishery related inputs and several life history traits and then uses some simple relationship to take get an estimate of the population abundance over time and can project out into the future. So first, let’s organize a list of the minimal inputs we are going to need for this.

| **Name** | **Type** | **Function used** | **Description** |
| --- | --- | --- | --- |
| N.options | Input | Por.sim() | An estimate of the catch/landings/removals from the fishery. |
| n.sims | User | Por.sim() | How many simulations do you want to run |
| q.ext | Value | Por.sim() | Population extinction threshold, basically a value after which the population is probably screwed |
| Sigma | Parameter | Por.sim() | Used if you are assuming autocorrelated values of r for the population. Set to 0 means r is random each year |
| u.long | User | Por.sim() | This will select more (but still not all how it is currently coded) of the exploitation data for the analysis |
| Repro.cycle | Input | Por.sim() | The number of times an individual reproduces in a year |
| Linf.mn | Input | Parms\_calc() | The mean estimate of the Von B asymptote. |
| k.mn | Input | Parms\_calc() | The mean estimate of the von B rate parameter. |
| t0.mn | Input | Parms\_calc() | The mean estimate of the von B age at size 0. |
| Linf.se | Input | Parms\_calc() | The standard error of the the Von B asymptote. |
| k.se | Input | Parms\_calc() | The standard error of the the Von B rate parameter. |
| t0.se | Input | Parms\_calc() | The standard error of the the Von B age at size 0. |
| mat.int.mn | Input | Parms\_calc() | The mean of the maturity model intercept. |
| mat.slop.mn | Input | Parms\_calc() | The mean of the maturity model slope |
| mat.int.se | Input | Parms\_calc() | The standard error of the maturity model intercept. |
| mat.slope.se | Input | Parms\_calc() | The standard error of the maturity model slope. |
| W.a | Input | Parms\_calc() | The Weight-length relationship intercept. |
| W.b | Input | Parms\_calc() | The Weight-length relationship exponent. |
| n.offspring | Input | Parms\_calc() | Average number of offspring which is transformed to number of female progeny per female per year. |
| gest.period | Input | Parms\_calc() | The gestation period in years. |
| max.age.lb | Input | Parms\_calc() | The lowest likely 'maximum age'. |
| repro.cycle | Input | Parms\_calc() | The number of times you reproduce in a year. |
| vb.cor.mat | Parameter | Parms\_calc() | The correlation matrix for the von B parameters to ensure that we don't get weird parameter combinations. |
| mat.cor.mat | Parameter | Parms\_calc() | The correlation matrix for the maturity parameters to ensure that we don't get weird parameter combinations. |
| age.mat | Output | F\_crit()  U\_calc()  Lotka\_r() | The age at maturity estimated in parms\_calc using life history information |
| max.age | Output | F\_crit()  U\_calc()  Lotka\_r() | The longevity of the species estimated in parms\_calc using life history information |
| mx | Output | F\_crit()  Lotka\_r() | The fecundity ogive, so this integrates age at maturity and weight information to sort out how many, estimated in parms\_calc using life history information |
| r.M | Output | F\_crit()  U\_calc()  Lotka\_r() | The natural mortality estimated in parms\_calc using life history information |
| sel | Input | F\_crit()  U\_calc()  Lotka\_r() | The selectivity of the stock. Currently this is set up as a single number and it is tweaked in the code, we'll need to make this more complex. |
| removals | Input | F.crit()  U\_calc() | Removals from the fishery in a given year. The can originate in the N.options object if doing the backwards simulation, but variable we input for future scenarios. |
| N | Output | U\_calc() | Population size in a given year, calculated |
| u | Input | Lotka\_r() | Exploitation rate (annual not instantaneous), currently set up to be 1 value. |
| r.cutoff | Parameter | Por\_sim()  Lotka\_r() | The maximum value we’ll allow for the Lotka\_r() calculations, do we want this for fish? |
| K | Parameter | Por\_sim() | Used in the forward projections where we assume a logistic growth model, going to be hard to estimate for fish me thinks? |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Questions on above table

1. u.long – I think it’d be useful to have this kinda term in there, but needs to be generalized so we can pick out the removal years we want to use.
2. Going to wrap all of the “Input” life history parameters into one big object, might be easiest to put them all into a ‘database’ (i.e. spreadsheet) so we can just call that in, though possible that we could use the Thorson FishLife package to pull right from Fishbase
3. n.offspring – So what do we want here for fish, any way we can get around this given how uncertain it would be I wonder?
4. vb.cor.mat & mat.cor.mat – Can we just stick with what we have for the matrix, or do we need to have a few options here? Seems like there would be some general ones

Notes in functions

* (Por\_sim) How should we handle removals, in the example we remove certain years, do we want to have that ability for our generalized model? See code around mean.removals <-
* (Por\_sim) The selectivity is just 1 number and is hard coded to be age 2 on wards we are going to need to generalize this and make a vector. See code around sel <- 1
* (Por\_sim) The carrying capacity…. Do we care since this is just used in the forward projections. If we do care, we’ll need to discuss how to parameterize that, maybe maximum Number ever observed.
* (Por\_sim) N.end, ideas on how we specify this in the absence of an assessment, if we use the last value of the assessment, then our result isn't independent of the assessment, that ok? Could we use a q corrected survey biomass value to kick it off if we didn’t have an assessment model? See code around N.end <- c(sample…
* (Por\_sim) There is a comment saying… 2 is a patch to get n.sims values for r that are <r.cutoff… not sure what that means.
* (Por\_sim) Would we ever want to grab the uncertainty around the B1.vec from the linear model… lm(log(Pop.vec)~years)$coef[2]
* (Por\_sim) If we care about the future simulations… We'll need to make an argument for the function so we can alter the future year removals scenarios, we'll probably want an option to use these, or use the above 'junk' code
* (Por\_sim) Question about using the u.long. We don’t use that right now, but we may want to if we want to use past removals to predict future removals, see code around if(u.long==T). Do we want to keep this u.long at all, or just change it so we can specify in the input what years we want to pull
* (Por\_sim) Question about whether we want to mess around with having autocorrelated r values for the simulation, see code around if(sigma>0)
* (Por\_sim) We use the last 3 years of removals to get estimate for year 1 of the future simulations. We may want to use something other than the removals in the final 3 years, so might make this an option. See code around R.ave
* (Por\_sim) In the forward projections we assume that the population grows and then the removals come after growth (so effectively at the end of the year), would this be useful to generalize so we could have removals then growth? See code around Forward.pop.vec[1]
* (Por\_sim) Do we want to put some uncertainty around r in the forward projections. Se code around xx <- as.vector(r.scenarios[[i]])

u.calc<-function(age.mat,max.age,r.M,sel,removals,N)