iSAM

Process Description:

Tag\_by\_age

Author:

*Scott Rasmussen*

*Zaita*

*scott.rasmussen@zaita.com*

# Process Overview

What is the point of this process? What does it offer the that the other processes don't?

This process is based on the transition\_by\_age process, whereby fish are moved from untagged categories to categories representing tag cohorts. Added functionalities compared with the transition\_by\_age process are the options to set initial mortality and tag loss rates through time. Ongoing mortality of tagged fish due to the tagging event can be achieved through a mortality rate process, but beware of setting the time steps.

# Example Configuration File Syntax

Please put in a list of all parameters you expect to be able to use in the configuration file, including the type of parameter, is it a list or single value, is it optional or have a default value etc. The more information here the better.

Even adding a description is helpful as this is automatically picked up by the documentation generator and put in to the manual.

@process <label ; string>  
type tag\_by\_age

years Define the years where the category transition is applied, with the option of following single cohorts or having multiple tagging events together.  
Type: Vector of integers or integer range  
Default: No default  
Value: Valid model years

from Define the categories that are the source of the transition process  
Type: String vector  
Default: No default  
Value: A valid list of categories from @model.categories

selectivities Define the selectivities applied to the source categories, will determine the proportion of each of the categories to move  
Type: String vector, of length from  
Default: No default  
Value: A valid list of selectivity labels defined by @selectivity

to Define the categories that are the sink of the transition process (the tagged categories)  
Type: String vector, of length from  
Default: No default  
Value: A valid list of categories from @model.categories

U\_max Define the maximum proportion of individuals that can be moved  
Type: Constant  
Default: 0.99  
Value: Must be > 0 and < 1

penalty Define the penalty to encourage models parameter values away from those which result in not enough individuals to move  
Type: String  
Default: No default  
Value: Valid penalty label defined by @penalty

min\_age Define the minimum age for the process  
Type: Integer  
Default: No default  
Value: A valid age in the range @model.min age and @model.max age

max\_age Define the maximum age for the process  
Type: Integer  
Default: No default  
Value: A valid age in the range @model.min age and @model.max age

n Define the number of individuals to move in each year for all age classes combined  
Type: Estimable vector, of length years  
Default: No default, only required if table is proportions {note it should error out if table is numbers and there is a n given}  
Value: A vector of values giving the numbers in each year

plus\_group Use age plus group  
Type: bool  
Default: true

minus\_group Use age minus group  
Type: bool  
Default: true

Table: Define the start of the table of values and the type of table provided  
Type: string, of value “numbers” or “proportions”  
Default: No default  
Value: If numbers, the table will contain numbers at age to be moved, otherwise it will contain proportions at age.

[label] Define the following data as the number or proportion of individuals to move in each age class for the year specified by the label  
Type: Estimable vectors, of length max\_age – min\_age +1  
Default: No default  
Value: The label is a valid value from years. It is followed by a vector of values giving the numbers or proportions in each age class. This subcommand is repeated for each unique year value. There needs to be one label per value of years. If table type is proportions, then the sum of proportions in each line needs to equal one.

end\_table: Defines the end of the table, with no value associated with it.

initial\_mortality Define the mortality rate as a proportion (not an M) to be applied immediately after the transition, not part of the mortality block.

Type: Estimable, of length one or of length *from*

Default: No default

Value: A positive real number applied to all categories, or a vector of positive real numbers with one value per transition category

initial\_mortality\_selectivity Define the age selectivity to be applied to the mortality rate

Type: Estimable, of length *initial\_mortality*

Default: None (i.e., selectivity one)

Value: A function applied to all categories, or a vector of functions with one value per transition category with labels defined by @selectivity (I don’t think as a function is implemented yet), to pro-rata the total mortality between ages

loss\_rate: Defines the rate of tag loss, i.e. of fish to be annually transitioned back (from the *to* categories to the *from* categories), starting from the years the fish are tagged only  
Type: Estimable real number or vector of real numbers of length *from*  
Default: 0  
Value: Must be >= 0 and <= 1. If a vector, each will be applied to each category to be transitioned

loss\_rate\_selectivity Define the age selectivity to be applied to the loss rate

Type: Estimable, of length *loss\_rate*

Default: None (i.e., selectivity one)

Value: A function applied to all categories, or a vector of functions with one value per transition category with labels defined by @selectivity (I don’t think as a function is implemented yet), to pro-rata the total loss rate between ages

# Step-by-step

A detailed list of steps the process goes **through in the following order (steps 1-3)** to achieve its purpose.

At the appropriate time step where the process is called:

1. If the partitions defined by parameter *to* already exists:
   1. Apply the loss rate: transfer a proportion of the *to* categories back to the *from* categories by applying the *loss\_rate* corresponding to each category.
2. If the partitions defined by parameter *to* does not exist yet:
   1. If model year is less than the first parameter of *years*, do nothing
   2. If model year equals the first parameter of *years*, create new partitions as defined by the parameter *to*
3. If the current model year is one of the parameters *years*
   1. If [year] doesn’t exists in the *table proportions*, error out “year specified in tagging years but no data is present in the table”
   2. Apply the *selectivities* to the *from* categories to get the proportion of each category to be moved, as well as maximum numbers that can be moved
   3. Remove the number of fish by age from the combined *n* and *table proportions* or *table numbers*, from the *from* categories, proportionally (between categories) to the selected numbers calculated, and subject to the maximum exploitation rate *U\_max*, and adding *penalty* if necessary
   4. Before adding these numbers to the *to* categories, apply *initial\_mortality* to these fish only (not those already in the category).

**Scott's Implementation**

Before first execution (Build):

1. If we're using the numbers table
   1. Build list of how many fish to remove each year at each age with number in year at age taken from the number's table.
2. If we're using the proportions table
   1. Build list of how many fish to remove each year at each age using the N parameter and multiplying that by the proportions provided in the proportion table.

During actual execution (Execute):

1. Apply the tag loss rate to any fish that have previous been tagged.
   1. Tagged fish = tagged fish – (tagged fish \* loss\_rate)
   2. Untagged fish = untagged fish + (tagged fish \* loss\_rate)
2. Calculate the proportion of each category of fish to be moved
   1. For each category and each age, calculate the selected numbers:  
      selected numbers at age of category i = numbers at age in the population \* selectivity of category i
   2. For each category and each age, calculate the maximum number of fish to be moved:  
      numbers to be moved at age of category i = number at aged to be moved (from build 2.1) \* selected numbers at age of category i / sum over all categories (selected numbers at age of category i)
3. Calculate how many fish we want could move (exploitation rate)
   1. number of fish is calculated above
   2. exploitation = number of fish to remove / total number of fish (proportion)
      1. if exploitation is higher than u\_max make it u\_max and trigger the penalty with the difference in amount of fish between total\_stock \* u\_max and number to remove
4. Migrate the exploitation amount of fish while doing current mortality on that amount being moved
   1. take the exploitation amount from the current population
   2. remove the initial mortality amount (current = current – (current \* initial mortality) \* initial mortality selectivity at age. from the fish we're migrating (not the population).
   3. add the fish with mortality removed to the 'to' category