LH_Sampling

Description: Functions for simulating fish populations and estimating life history parameters.
URL: https://github.com/evaschemmel/LHsampling.git

Authors: Erin Bohaboy (erin.bohaboy@noaa.gov)
Eva Schemmel (eva.schemmel@noaa.gov)

Updated: 11June 2021
References: Schemmel E., Bohaboy E., Kinney M., O'Malley J. An assessment of sampling strategies for estimating fish growth from fishery-dependent samples. *In prep*.
Introduction: An individual-based model (IBM) incorporating within-population variability in von Bertalanffy growth, size-dependent natural mortality, and a size-selective fishery to simulate an exploited fish population and catch (harvest). A bootstrap algorithm allows the user to investigate various sampling approaches including sampling strategy (proportional or fixed otolith sampling, POS or FOS, respectively), sample size, supplementation with fishery-independent sampling, and assumptions regarding von Bertalanffy t_0 and the relationship between variance of length at age and age. A function to produce plots of the bootstrap sampling results is also provided.

Required packages: reshape, dplyr, ggplot2, magrittr, assertthat.

`simulate_population_harvest` IBM to simulate a fish population and catches.

Description:

An individual-based model (IBM) that simulates fish age and length values for a population and fishery harvest.

Usage:

```
simulate_population_harvest(Linf, Linf_sd, M, Lorenzen, F, mincat,
catsd, maxcat, maxcatsd, L0, L0_sd, k, k_sd, Amax, age_max, N)
```

Arguments:

`Linf`	Von Bertalanffy theoretical asymptotic length (cm)
`Linf_sd`	Population standard deviation of asymptotic length (cm)
`M`	Instantaneous natural mortality rate (yr^{-1})

`Lorenzen`	TRUE / FALSE specifying whether natural mortality is a function of individual length following Lorenzen (Lorenzen, 2000; Lorenzen, 2005)
`F`	Apical (fully selected) instantaneous fishing mortality rate (yr^{-1})
`mincat`	Minimum length at 50% fishery selectivity (cm)
`catsd`	Slope of the ascending region of selectivity at length (cm), see details
`maxcat`	Maximum length at 50% fishery selectivity (cm)
`maxcatsd`	Slope of the descending region of selectivity at length (cm), see details
`L0`	Von Bertalanffy length at age 0 (cm)
`L0_sd`	Population standard deviation of length at age 0 (cm)
`k`	Von Bertalanffy growth coefficient
`k_sd`	Population standard deviation of Von Bertalanffy growth coefficient
`Amax`	Maximum longevity (years)
`age_max`	An arbitrary age selected to represent "old" fish (years)
`N`	The number of age 0 fish in each simulated cohort, typical value = 100,000

Details:

The individual-based simulation model begins with a cohort of *N* age 0 fish which progress through ages 0 to A_{max} following the Von Bertalanffy growth function for length at age (L_a) with individual-specific L_∞ and L_0 (length at age 0) parameters drawn from a normal distribution of mean = μ and standard deviation = σ [~N(μ, σ)].

$$L_a = L_\infty(1 - e^{-k(a-a_0)})$$

and

$$a_0 = \frac{Ln(1 - L_0/L_\infty)}{k}$$

L_a = predicted length (cm) at the end of age *a* (years)
L_∞ = asymptotic length in cm
k = growth coefficient
a = age in years
a_0 = the theoretical age at which the fish would have zero length

At each age, individual fish are subject to death by natural mortality followed by death from harvest (fishing mortality) in a Bernoulli random process with probability p_M or p_F, for natural death or harvest, respectively. According to the Baranov catch equation (Quinn and Deriso, 1999), the probability of natural mortality at age *a* for fish *i* ($p_{M,a,i}$) is

$$\frac{M_{a,i}}{F_{a,i}+M_{a,i}}(1-e^{-F_{a,i}-M_{a,i}})$$

where *M_{a,i}* is natural mortality at age *a* for fish *i* and is a function of individual length at age ($L_{a,i}$, calculated from the VBGF, below), following Lorenzen (Lorenzen, 2000; Lorenzen, 2005):

$$M_{a,i}=\frac{M1}{L_{a,i}}$$

Parameter *M1* describes the relationship between length and natural mortality in the overall population. *M1* is estimated within the model by simultaneously solving the following set of equations:

$$\left\{\begin{array}{c} M_a=\frac{M1}{L_a} \\ M_{overall}=\frac{-Ln(Survivorship)_{Amax}}{A_{max}} \\ Survivorship_{Amax}=\frac{N_{Amax}}{N_0}=\prod_{a=0}^{A_{max}}e^{-M_a}=f(M_a,A_{max},L_{\infty},L_0,k) \end{array}\right\}$$

where:
M_a = population expected natural mortality at age
L_a = population mean length at age
$M_{overall}$ = overall natural mortality
N_{Amax} and N_0 = expected number of fish in the population at ages A_{max} and 0.
$Survivorship_{Amax}$ = individual probability of an individual surviving natural mortality to reach A_{max}.

The probability of fishing mortality (being harvested) at age *a* for fish *i* ($p_{F,a,i}$) is

$$\frac{F_{a,i}}{F_{a,i}+M_{a,i}}(1-e^{-F_{a,i}-M_{a,i}})$$

Where $F_{a,i}$ is the fishing mortality at age *a* for fish *i* and is the product of apical (fully selected) fishing mortality (*F*) and selectivity at age conditioned on length ($Selex_L$):

$$F_{a,i}=F*Selex_L$$

$Selex_L$ is modeled as the cumulative normal probability density of mean = *mincat* and standard deviation = *mincatsd*.

At each time step, individual fish that experience death by natural mortality are removed from the simulated cohort. Of the fish that survived natural mortality, the individuals that die by fishing mortality are removed from the simulated cohort and set aside as harvested fish. The fish that survived both natural and fishing mortality undergo Von Bertalanffy growth and advance to the next age. A_{max} + 1 cohorts are created and then 1 age (from $a = 0$ to A_{max}) is taken from the survivors and harvest of each cohort to form the simulated population and catch.

Value:

A named list:

$population (dataframe: $age, $length): the simulated population

$harvest (dataframe: $age, $length): the simulated harvest

$Avg_age (dataframe: $Ages, $L_age, $M_age, and $Selex): characteristics of the simulated population at age

$parameters named list of 19 elements including all input parameters used in the simulation and the simulated population coefficient of variation of length at age_max and age_0

Example:

```
S1_Auric_lowF <- simulate_population_harvest(Linf = 32.5, Linf_sd =
2.5, M = 0.18, F = 0.09, Lorenzen = TRUE, mincat = 10, catsd = 2.5,
maxcat = 200, maxcatsd = 0, L0 = 10, L0_sd = 2.5, k = 0.6, k_sd = 0,
Amax = 32, age_max = 15, N = 100000)
```

`LH_sample` Bootstrap sampling routine to estimate life history parameters

Description:

A bootstrap sampling routine to estimate life history parameters from fishery catches simulated by `simulate_population_harvest()`.

Usage:

```
##Example for Pristipomoides auricilla using life history parameters
from O'Malley et al. 2019
## S1: Yelloweye, small fish in catch, F = 0.5M

LH_sample <- function(sim_output, n_boots, samp_size, sample_type,
supp_large = FALSE, supp_large_n_per_bin = 3, supp_small = FALSE,
supp_small_n_per_bin = 3, supp_min_length = 2, constrained = FALSE,
t0 = 0, SD_L_const = TRUE, save_bootstraps = FALSE, Amax = NULL,
age_max = NULL, Lbin_width = 2)
```

Arguments:

`sim_output` Output from `simulate_population_harvest()`

`n_boots` Number of bootstrap samples to perform

`samp_size` Total sample size for each bootstrap

`sample_type` The sampling strategy to be used, either proportional otolith sampling ('POS') or fixed otolith sampling ('FOS')

`supp_large` TRUE / FALSE specifying whether supplemental samples will be collected from large length bins

`supp_large_n_per_bin` The number of samples per length bin to be collected from large bins (ignored if *supp_large* = FALSE)

`supp_small` TRUE / FALSE specifying whether supplemental samples will be collected from small length bins

`supp_small_n_per_bin` The number of samples per length bin to be collected from small bins (ignored if *supp_small* = FALSE)

`supp_min_length` The minimum length fish that could be collected from the wild fish population

`constrained` TRUE / FALSE specifying whether theoretical time at length zero (t_0) should be estimated

`t0` If *constrained* = TRUE, the fixed value for t_0 (typically 0)

`SD_L_const` TRUE / FALSE describing assumptions of population variance in length at age. If TRUE, then standard deviation ($\sqrt{\sigma^2}$) of length at age is assumed a linear function of age. If FALSE, then the coefficient of variation of length at age is assumed a linear function of age.

`save_bootstraps` TRUE / FALSE specifying whether all bootstrap samples will be included in the function output

`Amax` Maximum longevity (years). If not specified, this value is taken from sim_output.

`age_max` An arbitrary age selected to represent "old" fish (years). If not specified, this value is taken from *sim_output*.

`Lbin_width` The width of each length bin (cm).

Details:

This function will take *n_boots* samples (without replacement) from the harvested individuals following either a fixed otolith sampling (FOS) or proportional otolith sampling (POS) strategy.

The function then parameterizes the von Bertalanffy growth function and estimates the population coefficient of variation of length at age for each bootstrap sample.

To produce each FOS bootstrap, *n* individuals (calculated as total sample size, *samp_size*, divided by the number of length bins represented in the harvest, rounded up to the next integer) are randomly sampled, without replacement, from the harvest for each length bin. If less than *n* individuals are available in a harvest length bin, then additional samples are randomly drawn from the remaining length bins in the harvest until the total prescribed number of samples (*n* × number of length bins represented in the harvest) are attained. If greater than *samp_size* individuals have been collected, individuals within fully filled length bins will be randomly discarded from the sample until *samp_size* is attained. If supplemental sampling is being implemented (*supp_large* or *supp_small* = TRUE) then the specified number of samples per large (*supp_large_n_per_bin*) or small (*supp_small_n_per_bin*) are taken randomly (without replacement) from the population. For this step, 'small' and 'large' length bins are defined as any bins containing individuals in the population where fewer than *supp_large_n_per_bin* or *supp_small_n_per_bin* individuals in the FOS sample. After supplemental samples are added, individuals from the most populous length bins will be randomly discarded from the sample until *samp_size* is attained.

Each POS bootstrap is produced in much the same way as for FOS. The primary difference is the 'target' number *n* of individuals per length bin *L* (n_L) is calculated based on the proportional distribution of lengths within the entire harvest, where n_L is rounded to the nearest integer. If $\sum n_L \neq samp\_size$ then n_L will be updated by adding or subtracting from the most populous length bin. Making adjustments to the number of samples targeted in the most populous length bin minimizes effects on the POS scheme from adjusting for the desired *samp_size*. For each POS bootstrap, n_L individuals are randomly sampled, without replacement, from the harvest for each length bin. If supplemental sampling is being implemented (*supp_large* or *supp_small* = TRUE) then the specified number of samples per large (*supp_large_n_per_bin*) or small (*supp_small_n_per_bin*) are taken randomly (without replacement) from the population. For this step, 'small' and 'large' length bins are defined as any bins containing individuals in the population with fewer than *supp_large_n_per_bin* or *supp_small_n_per_bin* in the POS sample. After supplemental samples are added, individuals from the non-supplemental length bins will be randomly discarded from the sample until *samp_size* is attained.

For each bootstrap sample, the coefficient of variation of length at age (CV_a) is calculated (given ≥ 2 individuals per age). The coefficient of variation of length at age 0 (CV_{a0}) and age_max (CV_{age_max}) are extrapolated assuming a linear relationship between age and CVa (SD_L_const = FALSE) or standard deviation of length at age (SD_L_const = TRUE). The slope and intercept are parameterized using *nls()*. If *nls()* fails to converge, then the average variance in length at age, over all ages, is used.

For each bootstrap sample, the von Bertalanffy growth function is parameterized using *nls()*, with t_0 (where $t_0 = a_0 + 1$) either estimated or fixed at the user specified value.

$$L_a = L_\infty(1 - e^{-k(a-a_0)})$$

and

$$a_0 = \frac{Ln(1 - L_0/L_\infty)}{k}$$

The total number of bootstrap samples for which *nls()* did not converge is reported out to the R console.

Value:

A named list:

$list_boot_samps: list containing a dataframe ($age, $length, $binL, $n, $nbin) for each bootstrap sample. Will only contain data if *save_bootstraps* == TRUE.

$list_boot_preds: list containing a dataframes of predicted length at age ($age, $length) calculated from the fitted von Bertalanffy growth model for each bootstrap sample

$list_boot_mods: list containing *nls*() fitted model objects for each bootstrap sample

$list_boot_CVs: list containing a dataframe of calculated CV of length at age (if defined, $age, $CV_age) for each bootstrap sample

$list_boot_SDs: list containing a dataframe of calculated standard deviation at length (if defined, $age, $SD_age) for each bootstrap sample

$parameter_outputs: dataframe ($Linf, $K1, $a0, $CV_L_a0, $CV_L_age_max) of *n_boots* rows of the estimated von Bertalanffy growth parameters

$params_input_output: list of 17 elements including all input parameters and $boots_nls_fail, which is the total number of bootstrap samples for which *nls*() failed to converge for the von Bertalanffy growth function.

$simulation_params: list of 19 elements including all input parameters used in `simulate_population_harvest()` and the simulated population coefficient of variation of length at age_max and age_0. This element is passed directly from `simulate_population_harvest()`.

$parameter_summary_all_boots: 95% confidence intervals, interquartile range, and arithmetic mean von Bertalanffy parameter estimates over all bootstraps (dataframe: $parm_name, $lower95, $lower50, $avg, $upper50, $upper95)

$list_some_boot_samps: list containing a dataframe for each of 9 randomly selected bootstrap samples ($age, $length, $binL, $n, $nbin)

Example:

```
S1_A9 <- LH_sample(sim_output = S1_Auric_lowF, n_boots = 1000,
samp_size = 300, sample_type = 'POS', supp_large = FALSE, supp_small =
FALSE, constrained = FALSE, save_bootstraps = TRUE,  Lbin_width = 2)
```

`LH_plot` Produce plots from an LH_sample object

Description:

A function that produces plots from the output of `LH_sample()`

Usage:

```
LH_plot <- function(sample_output, output_type = 'none')
```

Arguments:

`sample_output`	Output from `LH_sample()`
`output_type`	How plots are written and saved: ‘none’ displays in R graphics device only, ’pdf’ produces a single .pdf with all plots, and ‘png’ produces a separate .png for each plot.

Details:

This function outputs plots to the R graphics device, requiring the user press [enter] in the R console to advance to the next plot. Users are encouraged to examine each plot to confirm the characteristics of the simulated population, fishing fleet, and bootstrap samples match expectations.

Plots:

Selectivity at length as specified in the initial population simulation

Natural mortality at age

Scatterplot of population length at age, overlaid with average length at age as expected from the von Bertalanffy growth parameters passed from `simulate_population_harvest()`. The scatterplot is limited to a random sample of 50,000 individuals from the population.

Histograms of frequency distributions for population length, harvest length, population age, and harvest age.

Histograms of number of individuals per length bin for 9 randomly chosen bootstrap samples.

Value:

Will create files containing plots and save them to the working directory if *output_type* = ‘pdf’ or ‘png’. File names will include system time as [YYYYMMDDhhmmss].

Example:

```
LH_plot(S1_A9, output_type = ‘pdf’)
```