POPSIM-A as an operating model for testing North Pacific chub mackerel stock assessment models[1]

Dr. Larry Jacobson (Consultant)
LarryJacobson6@gmail.com

For North Pacific Fishery Commission

June 4, 2019

FINAL

[1] Distributed along with PsuedoData-10.R and PopSimFunctions-20.R where the numbers designate version, along with POPSIM-A input and output files for a chub mackerel demonstration listed in an appendix.

Summary

This report was written under contract to assist the North Pacific Fishery Commission (NPFC) and its Chub Mackerel Stock Assessment Working Group (WG) with work testing stock assessment models for chub mackerel (*Scomber japonicus*). Paraphrasing the consultancy agreement, I mean to attend a review meeting[2], participate in discussions and provide and demonstrate the following products:

a) Preliminary POPSIM-A input files for simulating a hypothetical chub mackerel population and fishery for one demonstration scenario based on a Japanese a Virtual Population Analysis (VPA) provided to me. *Done, as described in this report. Both will be distributed at or before the 2020 TWG CMSA03 meeting via Commission staff or as staff direct.*

b) Functions in R or POPSIM-A for saving simulated population variables and pseudo-data in one or more databases. *Same as a).*

c) A database with simulated "true" values and a database with pseudo-data from the demonstration scenario. *Same as a). POPSIM-A outputs a data structure with true values automatically. R software generates the database with pseudo-data from POPSIM-A output.*

d) R software for extracting POPSIM population variables and pseudo-data from database(s). *Same as a). Very little software is required because simple R commands are used. The subset() function, in particular, works well.*

e) Model performance results for demonstration scenario(s) and assessment models that can be called directly from POPSIM. *Not completed because it is no longer possible to call current versions of the assessment models from POPSIM-A (see this report). Members will have to write their own software for testing the assessment models they are interested in using. Such software will extract pseudo-data from the database building input files, running models, collecting model estimates and evaluating model performance. These tasks can all be done in R and the R software provided is designed to facilitate the work. Organization of WG efforts and the structure of a database for sharing assessment model test results are important topics to address. In an effort to enhance progress, I demonstrated R code at the review meeting for testing a "silly" assessment mackerel for chub mackerel either fitting the model entirely in R or by calling an AD-Model Builder program that runs in DOS (Appendices 9-11).*

f) A written report and detailed presentation at the March 2020 assessment meeting. The presentation will be interactive to demonstrate the use of software for simulation and testing and how scenarios can be modified. *Report same as a). Presentation to be completed at the meeting. I expect to modify the report, software and other materials at the meeting based on discussions.*

h) Complete documentation for running POPSIM, documentation useful in linking POPSIM to new assessment models, description of new software and any changes made to existing

[2] The meeting was rescheduled and then convened on 27 May 2020 as a teleconference due to the COVID-19 pandemic.

software, along with advice about using POPSIM for chub mackerel work. *Same as a). The documentation in this report supplements help files distributed with POPSIM-A that members will still need to read.*

i) New POPSIM code if any modifications are made. *No changes were made to POPSIM-A (see text for advice re future modifications).*

Introduction

Chub mackerel (*Scomber japonicus*) is a widely distributed pelagic species found in both hemispheres of the Pacific Ocean. One stock in the northwest Pacific supports a large and valuable commercial fishery involving NPFC members in national and international waters. NPFC is developing stock assessment models and considering management approaches for chub mackerel (e.g. NPFC-2019-TWG CMSA02-WP02). Following Deroba et al. (2015), NPFC intends to test candidate stock assessment models and modeling approaches by simulation using the POPSIM-A fishery simulator as an operating model (NPFC-2019-TWG CMSA02-Final Report). POPSIM-A was developed by the National Oceanographic and Atmospheric Administration (NOAA), Northeast Fisheries Science Center (NEFSC) in Woods Hold, MA USA. It is currently available in the NOAA Fisheries Integrated Toolbox https://noaa-fisheries-integrated-toolbox.github.io/POPSIM-A/. Use fisheries.toolbox@noaa.gov for correspondence and questions.

Biology and distribution patterns (FAO 2000) make stock assessments difficult for chub mackerel and similar species and model testing useful. Chub mackerel are pelagic found generally along coasts but also offshore as adults, fast growing and moderately short-lived (reported maximum age usually 8-12 y) with first maturity at 2-4 y. Chub mackerel are batch spawners who likely reproduce whenever conditions are suitable (February-June,Kanamori et al. 2019 and at other times elsewhere (FAO 2000). As with other pelagic species, spatial distribution varies with body size, season, sea surface temperature and interannually depending on environmental conditions and climate. They can be difficult to catch in trawls during research (survey) or commercial fishing with trawl gear. Dip nets with lights or encircling gear such as purse seines and lampara nets are used in commercial fishing. Such gear have catch rates that are strongly influenced by distribution patterns, the environment and search time. Commercial catch-per-unit-effort (CPUE) data are often available but typically nonlinear and difficult to interpret. Egg and larval surveys are used to evaluate spawning stock size or recruitment because ichthyoplankton density is easier to estimate.

The hypothetical chub mackerel stock in this report is based on VPA results in addition to advice and other information provided by Dr. Shota Nishijima (Japanese National Research Institute of Fisheries Science) a few literature studies and general knowledge about such species. These assumptions should be refined immediately by subject experts if NPFC conducts further testing. The work and results described in this report are all for demonstration purposes only and meant to facilitate the WG's future efforts.

Materials and methods

I used POPSIM-A (version 8.2, release date 12/12/2014 with C code compiled 12/6/2013) and wrote custom R software for post-processing on a Windows 10 laptop. Users should read the help files distributed by NOAA fisheries with the POPSIM-A GUI. I provide POPSIM-A documentation not readily available elsewhere, fully document the R software and provide a few ideas and suggestions. See Appendix 1 for short-cut instructions and Appendices 2-3 for a more detailed explanation of the POPSIM-A menu bar and input screens.

POPSIM-A's core is a general age-structured fishery simulation model written in Microsoft C using commercially available IMSL Numerical Library functions (https://www.roguewave.com/products-services/imsl-numerical-libraries). The C code was graciously provided to me by Dr. Chris Legault (Northeast Fisheries Science Center, Chris.Legault@noaa.gov) for use in checking the structure of internal calculations and documenting the program. It was not recompiled, however, because there was no need and the IMSL software license was too expensive. Fortunately, the existing code was stable, had most of the necessary features and worked well. One option possibly required for chub mackerel but not initially available (nonlinear CPUE data) was included in the R code. I will not distribute the code with this report but individuals who wish to see or use it should contact NOAA Fisheries directly at fisheries.toolbox@noaa.gov.

POPSIM-A is distributed with a useful GUI wrapper written in Visual Basic for Windows that includes help files, facilitates and checks data entry, documents program operation displays simulated fishery results (e.g. true simulated biomass) and summarizes pseudo-data. It was also intended to make data files and run a handful of stock assessment models used in the northeastern US, collect stock assessment model results and compare assessment model estimates to the simulated population. The GUI was useful for chub mackerel but was not used for running assessment models and collecting model estimates because the Visual Basic code has not been updated to work with the recent versions of the assessment models. This situation does not pose a problem to WG members who are interested in using their own stock assessment programs (see Discussion).

According to documentation in the GUI, POPSIM-A was intended to work with the following versions of stock assessment model programs mostly developed at NOAA's Northeast Fishery Science Center: VPA (Virtual Population Analysis, Version 3.2), CSA, (Catch-Survey Analysis, Version 4.0), ASAP (Age Structured Assessment Program, Version 2.0) and ASPIC (A Stock Production Model Incorporating Covariates, Version 5.34). The GUI appears to support the SS3 (Stock Synthesis 3) model as well but the version number is not specified. With the exception of ASPIC and CSA, the versions are obviously deprecated and no longer supported.

ASPIC Version 5.34.6 and CSA Version 4.3 are available but I was not able to make either work with POPSIM-A. WG members could contact the NOAA Fisheries Toolbox to see if deprecated program versions are still available or to seek advice on this topic. However, the use of deprecated models should be considered carefully.

Simulated true population dynamics

The user specifies the first and last years "in the model" (y_1, y_L). However, calculations are carried out through year y_{L+1} in POPSIM-A to accommodate simulated stock assessments conducted in the year after the last with catch data when additional survey data may be available (commonly done in the northeastern US and by Japanese scientists for chub mackerel). Fishing mortality is calculated for years with catch data but stock size (including recruitment) is calculated through the beginning of year y_{L+1}. The user can turn survey sampling off in y_{L+1} (see below) or ignore the last survey observation when making assessment model input files if desired. The POPSIM-A catch data for chub mackerel cover 1978-2018 but abundance is also calculated for 2019 as in the Japanese VPA.

The first age is always referred to as age 1 in simulations. However, it is easy to adjust data and input parameters (e.g. for the spawner-recruit relationship) to include young-of-year fish (age 0), as was done for chub mackerel. The user specifies the last age (a_{max}) in simulation calculations as well as the plus-group age ($a_+ \leq a_{max}$). The user must supply parameters (including mean body weights in the spawning stock, mean body weights in the population, maturity, natural mortality, fishery selectivity, etc.) for ages 1 to a_{max}. The plus-group age affects only the simulated pseudo-data. In processing pseudo-data, it may be important to remember that weight- and mortality at age are not necessarily the same for all ages in the simulated plus group even if they are assumed the same in stock assessment modeling (see below). The POPSIM-A simulation model for chub mackerel included ages 0-19 y identified as ages 1-20 y in the program because age 0 catch is appreciable in some years while the VPA used ages 0 to 6+ y. Extra age groups were used in POPSIM-A in case lightly fished or unfished populations were simulated and so that the simulation population could express its full growth potential if allowed to do so. Maturity and mean body weights were the same as in the VPA and the data for age 6 in the VPA were used for ages 6-19 in POPSIM-A. Fewer age groups can be used in future.

POPSIM-A uses instantaneous mortality rates and conventional fisheries equations in simulating the true chub mackerel population. In particular for $a \prec a_{max}$,

$N_{a+1,y+1} = N_{a,,y} e^{-Z_{a,y}} = N_{a,,y} e^{-(M_{a,y} + \sum_{f=1}^{n_f} F_{a,y,f})}$ and $C_{a,y,f} = \frac{F_{a,y,f}}{Z_{a,y}}(1 - e^{Z_{a,y}}) N_{a,y} w_{a,y,f}$ where $N_{a,y}$ is abundance at age *a* in year *y*, $M_{a,y}$ is the natural mortality, $F_{a,y,f}$ is the fishing mortality rate for fleet *f*, $\sum_{f=1}^{n_f} F_{a,y,f} = F_{a,y}$ is the total fishing mortality rate summed over fleets, $Z_{a,y} = M_{a,y} + F_{a,y}$ is the total mortality rate, $C_{a,y,f}$ (upper case) is catch weight and $w_{a,y,f}$ is mean weight at age for the fleet. Fishing mortality for a fleet, age and year is $F_{a,y,f} = F_{y,f} v_{a,y,f}$ where $v_{,fa,y}$ is fishery selectivity at age (likely scaled to a maximum of one, but see below) and $F_{y,f}$ is the fully recruited fishing mortality (at the age where $v_{a,y,f}$ = 1). As explained under the Fishing Mortality input screen in Appendix 3, the user should take care in specifying true fishing mortality rate and fishery selectivity parameters so they are well defined. POPSIM-A will use the values provided by the user, even if they are poorly defined. The last age group is always treated as a plus group in POPSIM-A so $N_{a_{max},y+1} = N_{a_{max},y} e^{-Z_{a_{max},y}} + N_{a_{max-1},y} e^{-Z_{a_{max-1},y}}$. Only one fleet was assumed for chub mackerel $(n_f = 1)$ but it and the R software was tested using multiple fleets for another species.

The user enters the assumed initial age composition vector $\overline{N}_{a,1}$ which, for chub mackerel, were taken from the VPA with some adjustments to account for the differences in plus-groups. The total number of fish in the first year was the same in both models but chub mackerel aged 6+ in the VPA were spread across ages 6-19+ y in POPSIM-A. The natural mortality rate M=0.4 in the VPA was used for all ages and all years in POPSIM-A. As mentioned later, the initial age composition vector includes recruitment in the first year of the model. Recruitments in later years are specified separately.

Each fleet requires the user to supply a vector of fishery selectivity at age ($v_{a,y,f}$) for all of the years in the model. In addition, the user must supply the fully recruitment fishing mortality rate ($F_{y,f}$) for each fleet and year or the true catch ($C_{y,f}$) or landings weight ($L_{y,f} = C_{y,f} - D_{y,f}$ where $D_{y,f}$ is weight discarded). If landings or catch are specified, POPSIM-A tries to find the full recruit fishing mortality that results in the desired harvest given stock size, fishery selectivity and discard assumptions. However, the target catch and landings options may cause the program to terminate with a "`Maximum Iterations Exceeded in Harvest for Fleet:`" error if the catch is near zero or too high and stock size becomes too low to provide the requested harvest. It is safer to specify fishing mortality rates unless the user is prepared to potentially tune input parameters to avoid very high and low biomass levels relative to catch.

Discarded weight is $D_{y,f} = \sum_{a=1}^{a_{max}} C_{a,y,f} \delta_{a,y,f}$ where $\delta_{a,y,f}$ is a discard fraction specified by the user. This option provides a means of simulating incomplete catch data (catch data consistently less than the true catch) and is one of two ways for simulating errors in the data (see below). The discard fractions can vary over time and among fleets so that the extent of under reported catch can vary. Some discard of young chub mackerel is assumed for demonstration purposes (discard mortality about $0.1F_{1,y}$ for age 0 decreasing to almost zero by age 3 y in POPSIM-A).

Stock biomass and spawning biomass are the sum of numbers at age times mean weight at age with an additional adjustment to spawning biomass for mortality. In particular, spawning stock biomass is $S_y = \sum_{a=1}^{a_{max}} f_{y,a} N_a m_{a,y} w_{a,y}$ where $m_{a,y}$ is maturity and $w_{a,y}$ is weight at age for spawners. The adjustment term for spawning biomass $f_{y,a}$ accounts for natural and fishing mortality that may occur before spawning. In particular, $f_{y,a} = e^{-(\tau_y M_{a,y} + \gamma_y F_{a,y,f})}$ where τ_y and γ_y are the fraction of natural and fishing mortality that occur before spawning specified by the user for each year on the Biological input screen (Appendix 3). For chub mackerel, $\tau_y = \gamma_y = 0$ in all years to mimic assumptions in the VPA. These parameters have a strong effect on spawning biomass calculations when mortality rates are high and may be important in comparing spawning biomass estimates from different models or in spawning biomass calculations.

Recruitment $N_{1,y}$ at the beginning of the first year is entered by the user as the number of fish in the first age group as part of the initial age composition, used as is, and does not include process errors. Recruitment in each year after the first (years $y_2 \ to \ y_{L+1}$) can be user specified (option 1). Options 2-4 are for the Beverton-Holt $(R' = \frac{\alpha S'}{\beta + S'})$, Ricker $(R' = S' e^{\alpha + \beta S'})$ or Shepherd $(R' = \alpha S' / \left[1 + \left(\frac{S'}{\kappa} \right)^{\beta} \right])$ spawner-recruit models where $\alpha, \ \beta, \ \kappa$ are parameters, $S' = S/\delta$ and $R = \eta R'$. The purpose of the scaling parameters $(\delta, \ \eta)$ is not clear but they could be used to reduce the size of intermediate numbers in exponential calculations if suitable values were chosen and spawner-recruit parameters were adjusted accordingly ($\delta = \eta = 1.0$ were used for chub mackerel and recommended). They may also be useful if spawner-recruit models are taken from previous studies. Options 5-6 are empirical bootstrap approaches. Option 5 draws recruitment from a single user specified set. Option 6 draws recruitment from one of two sets based on spawning biomass (i.e. draw from set 1 if biomass less than a user-specified threshold and draw from set 2 otherwise). Options 1-4 include lognormal variability in the true simulated values based on a user specified CV ($s = \sqrt{(1 + CV^2}$ where *s* is the standard deviation of log recruitment and CV is the coefficient of variation for arithmetic recruitment). Thus, even under Option 1, random process errors can be applied to recruitment for each year. Lognormal variability is applied with a correction for bias $\left(e^{-s^2/2} \right)$ so that the lognormal errors do not change mean recruitment. I used option 1 for chub mackerel (user specified recruitments from the VPA) with no variability (s=0) to match the VPA population dynamics.

Simulated population/fishery

As mentioned in the program help files, POPSIM-A can save an R list containing the true population dynamics and most of the input parameters as an RDATA file (Appendix 4). It is important to run the “Export PopSim data to R" command under the File menu in the GUI after POPSIM-A has completed to save the file for use later. Otherwise, the user may inadvertently use an RDAT file from a previous run. Use the dget() command in R to retrieve the list as shown in (Appendix 5).

Pseudo-data

POPSIM-A inputs that control pseudo-data for model testing have no effect simulation of true population dynamics. POPSIM-A generates two types of pseudo-data in addition to simulating the population. "Fishery samples" are catch-at-age and/or landings-at-age (to accomodate discard) from each fleet during each year and iteration while "Survey samples" are catch-at-age in each survey in each iteration during each year. One iteration generates a complete set of pseudo-data for all fleets, surveys and years. Pseudo-data include fishery and survey samples for ages 1 to a_+. Fleets and surveys are identified by a number (1 to n_f or n_s) and each is associated with the user specified set of age-specific selectivity parameters which may change over time. Survey data for recruits, for example, separated from combined survey data for the other age groups would be considered two surveys in POPSIM-A, even if the data are collected by the same research program in real life because the two surveys are defined by separate age selectivity parameters in the same year within POPSIM-A.

The R software calculates total catch and landings in numbers and weight in various forms (totals, at age, by fleet, etc.) in each iteration for use in testing assessment models. These calculations are based on catch-at-age by fleet and year with simulated errors, user specified discard fractions and fishery weight at age. Similarly, the R software calculates survey catch in numbers and weight in various forms (totals, at age, by survey, etc. with simulated errors) from survey samples and population weight at age for use in testing assessment models. It is important to ensure that the "Generate comma separated files for iterative results" box is checked in the Options input screen before running POPSIM-A so that pseudo-data files in the correct format are saved for use by R (Options screen, Appendix 3). Failure to check the control box may result in the accidental use of output files from a previous run. I attempted to provide pseudo-data in all of the forms that might be useful for chub mackerel. Discards are not tabulated by R but can be calculated by subtraction or easily added to database calculations later.

Total catch numbers as pseudo-data (with error) were calculated in POPSIM-A for each fleet and year as $\widehat{c}_{y,f} = c_{y,f}e^{s_{y,f}r_{y,f}}$ where $c_{y,f}$ (lower case) is true total catch number, "hats" (^) denote pseudo-data, $r_{y,f}$ is a random number from a standard normal distribution and the standard deviation $s_{y,f} = \sqrt{ln\left(1 + CV^2_{y,f}\right)}$ where the CV is specified by the user. If $s_{y,f} = 0$, then catch pseudo-data are perfectly accurate. Catch at age pseudo-data are $\widehat{c}_{a,y,f} = \widehat{c}_{y,f}\widehat{p}_{a,y,f}$ where the $\widehat{p}_{a,y,f}$ are drawn from a multinomial distribution with parameters $\pi_{a,y,f} = \frac{c_{a,y,f}}{c_{y,f}}$. The sample size $k_{y,f}$ is specified by the user and can vary over time and among fleets to vary the accuracy of the catch-at-age pseudo-data. Thus the catch-at-age data carry lognormal errors in total catch and multinomial sampling errors. If $k_{y,f} = 0$ Then fishing occurs but there is no sampling and no recorded catch (extra care is required in using POPSIM-A output files in this case, see R code). Conversely, if the sample size is large (e.g $k_{y,f}$=10,000), then the catch-at-age data will be nearly error free as assumed in VPA, Discard and landing pseudo-data are calculated based on discard fractions. Weights are based on fleet- and year specific mean weights at age. The CV for total chub mackerel catch was 0.10 and the assumed sample size was 100 in each year.

POPSIM-A does not simulate CPUE pseudo-data which is used to measure stock size in many fisheries. However, a very simple approach was used for chub mackerel and an additional more complicated approach is described below under “CPUE”. Both involve post-processing pseudo-data.

Pseudo-data for catch-at-age from survey samples are calculated in much the same way as fishery samples but data from different surveys are never combined and weight is based on true population weight at age. Pseudo-data for total catch numbers in survey *j* are: $\widehat{I}_{y,j} = q_{y,j}\left(\sum_{a=1}^{n_{a_{max}}} v_{a,y,j}N_{a,y,j}\right)e^{s_{y,j}r_{y,j}}$ where $q_{y,j}$ is catchability, $v_{a,y,j}$ is selectivity at age and the standard deviation and random number are analogous to the same quantities for fishery samples. Survey catch-at-age pseudo-data are calculated in the same way as for fishery samples and the sample size for each survey can vary over time and among surveys.

Users may need to use age-specific survey indices where a time series of survey data consists of a single age group. In such a case, there could be one index for each age group all parameterized as separate surveys in POPSIM-A to simulate indices for each age group derived from one survey program in real life (see recruitment indices below for chub mackerel). There are at least two ways to handle such a case in POPSIM-A. The easiest approach would be to define $n_{a_{max}}$ survey indices each with selectivity set to 1.0 for the target age and zero for all other ages. Each simulated age-specific survey would have a CV with this approach and measurement errors for different age groups in the same survey program and year would be uncorrelated, although errors in data for ages in the same year are correlated in real life. Such simulated survey data would carry no sampling error (the specified sample size would be irrelevant) although age specific indices are affected by sampling error in the real world. The user might choose to increase the CV for each age group to make up for lack of correlation and sampling error.

Another more realistic approach to simulating age-specific indices could be done while post-processing survey pseudo-data but would require additional R programming. Calculations in R would simulate age-structured indices as $\widehat{I}_{a,y,j} = \widehat{I}_{y,j}\widehat{p}_{a,y,j}$ where the total survey $\widehat{I}_{y,j}$ carries one measurement error affecting indices for each age and the age composition pseudo-data $\widehat{p}_{a,y,j}$ carries correlated sampling errors. Survey ID numbers and other book-keeping would be adjusted also during post-processing.

There were four surveys for chub mackerel. Surveys 1 and 2 were for recruits (age 0 identified in POPSIM-A as age 1) with selectivity = 1 for the first age group and zero selectivity for other ages. Surveys 3 and 4 are spawning biomass surveys (average maturity ogive same as survey age selectivity). Catchability coefficients for chub mackerel were $q_{y,s} = 1$ rather than the values estimated in the VPA to facilitate interpretation of testing results (models with $q_{y,s}$ estimates near 1.0 are “good”). The standard deviations measuring survey precision on the log scale were 0.91, 0.91, 0.73 and 0.67 based on the VPA so the arithmetic CVs used in POPSIM-A for chub mackerel were 1.14, 1.14, 0.83 and 0.75 in all years. The assumed multinomial sample size was 100 in all cases though the specification has no effect on the recruit surveys with all samples from the same age group.

R software

The R software was organized into two R command files which are documented in headers and in the code. All of the functions are in the first command file named POPSIMFunctions.R which is too long to include as an appendix but was distributed with this report. It includes two important functions called getCatchPseudoData() and getSurveyPseudoData(). Both of the main functions automatically generate graphs that are helpful in checking simulations and pseudo-data. Both remove existing survey or fishery pseudo-data before adding new data to existing databases.

The first R command file also includes some utility functions. Of particular interest, the utility function makeDB() in POPSIMFunctions.R creates or appends pseudo-data or other information (such as a serial number and time-stamp) to an output database for assessment model testing. The function lookDB() provides useful information about the contents of a database.

The second R command file named PseudoData.R loads libraries and sets global variables such as the plus-group specification, path and file names (Appendix 6). It reads the RDAT database generated by POPSIM-A that contains most of the input data used to run POPSIM-A and the true simulated population variables such as spawning biomass and fishing mortality. The true values other than weights at age (see below) are hardly used at this step but will be key in evaluating model performance after testing. It then builds a pseudo-database containing fishery and survey data for each of the iterations requested by the user (one set of pseudo-data per iteration) by calling getCatchPseudoData() and getSurveyPseudoData(). There were 1000 iterations for chub mackerel.

The pseudo-database is a large dataframe with a simple relational style structure (Appendix 7). It is automatically saved as an RDAT and ASCII comma delimited files for distribution. There are two key variables (columns) used to identify information in the database. Source (“sourc” in the datatable) identifies the datum on the record as pertaining to either a fishery or survey while name (“nam”) carries a descriptive identifier for the type of data. The other variables on each record are year, age group (“age grp”), fleet, iteration (“iter”) and value. Fleet holds the ID number that identifies commercial fishing fleets and surveys. Fleets and surveys are numbered sequentially starting from 1. The chub mackerel case, for example has just one fleet (ID number 1) and four surveys (ID numbers 1-4) All variables except nam can be missing (NA).

The pseudo-database for model testing includes the information normally available to a stock assessment biologist (Appendix 8). Such data include catch weight and numbers by age, all ages combined, fleet and all fleets combined, along with fleet specific true weight at age by fleet. Weight at age data with errors are not available in POPSIM-A and I did not program them in R because they probably have small effects and because the error structure was unknown to me. Fleet level data are omitted when only one fleet is used. Survey pseudo-data include total catch numbers and weight (ages combined) and numbers at age along with the true population weight at age. The R software saves a unique serial number and timestamp in the database automatically for the fishery and surveys pseudo-data. Old data records, serial numbers and timestamps are deleted if a database is reused.

CPUE

POPSIM-A does not generate CPUE pseudo-data or, in particular, CPUE that are nonlinear and not proportional to stock size although CPUE are nonlinear and used for chub mackerel. To handle chub mackerel, the R software can generate nonlinear surveys like CPUE to use in model testing. Total catch from such a survey is $\widehat{CPUE}_{y,j} = \widehat{I}_{y,j}^{\eta}$ where η is an exponent provided by the user and $\widehat{I}_{y,j}$ is simulated survey data calculated as described above. CPUE catch-at-age is $\widehat{CPUE}_{a,y,j} = \widehat{\pi}_{a,y,j}\widehat{CPUE}_{y,j}$ where the proportions at age were calculated from the original untransformed survey. Thus, the nonlinear relationship changes the total survey catch but not the fractions at age. As with surveys, total CPUE simulated in this way carries a lognormal error and the age composition carries multinomial errors. The exponents (η) assumed for chub mackerel based on VPA results were 1.6, 1.6, 1 and 1. Thus, the two recruit surveys were assumed to decline and increase faster than stock size as in the VPA (hyper-depletion, Hilborn and Walters 1992).

Another approach to simulating CPUE in post-processing would calculate CPUE from fishery samples as $\widehat{CPUE}_{y,j} = \frac{K_{y,j}}{F_{y,j}}$ where $K_{y,j}$ is catch or landings (in weight or number) for one fishery. After calculation, the simulated CPUE could be treated as a survey or as fishery CPUE depending on the structure of the stock assessment model. Fishing mortality is theoretically proportional to fishing effort ($E_{y,j}$) so that $F_{y,j} = Q_{y,j}E_{y,j}$ where $Q_{y,j}$ is a catchability coefficient for fishing. Thus, the ratio of catch and fishing mortality would be proportional to CPUE as long as catchability $Q_{y,j}$ was reasonably constant over time. Differences in catchability among ages would be absorbed by the survey or fishery catchability coefficients estimated in the stock assessment model tested. CPUE simulated in this way would be a nonlinear measure of stock size because catch in Baranov's equation $C = \frac{QE}{Z}N(1 - e^{-(QE+M)})$ is a nonlinear function of fishing effort. However, it would not be affected by many factors such as search time and gear saturation that exacerbate nonlinearity in real life. The simulated CPUE would carry lognormal errors assumed in catch but no errors due to estimation of fishing effort, which are probably common.

Model testing

A simple “silly” stock assessment model (annual catch curves fit to data from one survey) was used to demonstrate how real stock assessment models might be tested with pseudo-data from POPSIM-A. The first example extracts pseudo-data, fits catch curves and displays test results entirely in R (Appendix 9). The second extracts data in R (Appendix 10) and fits catch curves using an AD-Model Builder program that runs in a DOS window (Appendix 11).

Results

The POPSIM-A demonstration for chub mackerel with 1000 iterations took about one minute on an inexpensive laptop computer running Windows 10. It took R about 5 minutes to generate the database and save it in both RDAT and csv formats.The pseudo-data base had 2.7 million rows and about 34% were from fishery samples.

The simulated chub mackerel results from POPSIM-A were nearly identical to VPA estimates (differences < 1%) as expected given their similar mathematical structure and the source of POPSIM-A input data (Appendix 12). Graphical output from R indicates that the pseudo-data are reasonable given the true underlying population and assumptions about errors (Appendix 13). All input and output files are listed in Appendix 14.

Discussion

The WG can use the GUI to make POPSIM-A input data files and call the C program to carry out simulations that generate pseudo-data, at least initially. After the simulations are complete, use the POPSIM-A and R software to extract and store the pseudo-data and simulated true values in databases for model testing. The pseudo-data can be distributed to assessment model testers who can build their own assessment model datafiles, run the models and collect model results for comparison to true values which may be distributed separately if desired.

Eventually, the Visual Basic portion of POPSIM-A will stop working due to upgrades in other software and it may not be updated. Some members may wish to run POPSIM-A in Linux or some other operating system that does not support Visual Basic. If the need arises, a member or members could obtain an IMSL license, obtain and modify the C code and recompile it, or work with NOAA staff to update the program. Input data files used by the C program are well documented internally and easily modified in a text editor (e.g. see CHUB-DEMO-1.DAT for chub mackerel). The POPSIM-A help files could be preserved in sine form although much of their content relates to use of the GUI. The original or revised programs can be run in Windows or Linux at the system prompt using a command like “AGEPOPSIM.exe datafilename.DAT”. The system() function can be used in R for this purpose.

I was able to modify, recompile using the free GNU gcc compiler, edit data files and run an earlier version of POPSIM for Windows in R before retiring from NOAA fisheries. Based on my experience with the earlier version, the C code is worth preserving, modifying and using for future model testing. The C (and Visual Basic) codes were written by an experienced professional scientific programmer and are not too difficult to follow. Moreover, the algorithms and scientific basis of the simulations seem logical and sound. Programmers have different styles and some bugs may remain but the POPSIM software has been extensively tested. These arrangements would put the chub mackerel working group in a typical position for testing models without having to write and extensively debug the simulation program.

WG members will inevitably develop a new population simulator, probably in R or its successor that does not involve languages like VisualBasic or C. Experience at my previous laboratory (Northeast Fisheries Science Center, Woods Hole, MA) suggests that a GUI interface, though handy, is difficult to maintain if it calls assessment models which seem to constantly evolve. If a GUI is required, then its function might be best restricted to the simulation program. Separate code can be written to read pseudo-data and call assessment models.

As decided by the NPFC Chub Mackerel Working Group (NPFC-2019-TWG CMSA02-Final Report), all of the work described here used the simple age-structured POPSIM-A version rather than the length- and age-structured version (POPSIM-L) which is more realistic and flexible but harder to set up. As a practical matter, it was probably better to start with the simpler approach. More complex and realistic testing can be done later if required. Much of the information in this report applies and the R software should also be useful with POPSIM-L. Much of the information in this report also applies to the POPSIM-L.

The WG will have to decide how to handle testing with different plus-groups. I recommend running POPSIM-A use a relatively large a_{max} (use an age where mean size approaches the von Bertalanffy maximum L_∞) and set $a_+ = a_{max}$ in POPSIM-A so that no information is lost in pseudo-data output. With this approach, the simulation program would not have to be rerun for different plus-group specifications and all testers could use datafiles derived from the same POPSIM-A run and output. The R software could then be used to adjust plus-group assumptions in pseudo-data. Alternatively, Working Group members testing models can do the plus-group adjustments themselves which require care but are not difficult.

Working group organization and project design

The R software was set up so that pseudo-data for all iterations are stored in a single database that can be distributed to model testers as an RDAT or comma-separated ASCII text file. Data from the true simulated population are stored separately in an R object produced by POPSIM-A that can be kept or distributed separately. Thus, if desired, model testers can work as they would in a real stock assessment without any direct information about the true population. Comparison of model results to true values would have to be done after the stock assessment models are fit, perhaps by different personnel or the Working Group as a whole if the work is organized in this manner.

As a priority, I recommend developing a common flexible database structure for storing key estimates from assessment models (e.g. biomass estimates) fit to each set of pseudo-data. Such a structure will facilitate analysis by the group as a whole or by members in their own laboratories.

Deroba et al. (2015) describe difficulties in keeping track of pseudo-data files used by various scientists who were testing models. To alleviate these problems, the software for chub mackerel was written to provide all of the data necessary to run anticipated types of stock assessment models, or at least the data that might be available in chub mackerel stock assessments and used to tabulate assessment model data. It may be best to assign the work of distributing pseudo-data to a single person or small group who can coordinate the distribution of test cases, ensure testers are using the same datasets and keep track of the serial numbers associated with each testing scenario.

Acknowledgements

Dr. Chris Legault (National Oceanographic and Atmospheric Administration, Northeast Fisheries Science Center, Woods Hole, MA, USA) provided access to POPSIM code so I could check internal calculations. Dr Shota Nishijima (National Research Institute of Fisheries Science, Fisheries Research and Education Agency, Yokohama, Japan) provided advice and VPA output and suggestions on this report. Mr. Alan Seaver (Northeast Fisheries Science Center, retired) wrote the C and VisualBasic programs. Mr. Alex Zavolokin (NPFC Secretariat) provided logistical support and other assistance.

Literature cited

Deroba, J., et al. Simulation testing the robustness of stock assessment models to error: some results from the ICES strategic initiative on stock assessment methods. ICES J. Mar. Sci. 72: 19–30.

Hernández C. and S. Ortega. 2000. Synopsis of biological data on the chub mackerel (Scomberjaponicus Houttuyn, 1782). FAO Fisheries Synopsis. No. 157. Rome, 77p.

Hilborn, R. and Walters, C.J. (1992) Quantitative fisheries stock assessment, choice, dynamics and uncertainty. Chapman and Hall, London.

Kanamori Y., Takasuka A., Nishijima S. and Okamura H. 2019. Climate change shifts the spawning ground northward and extends the spawning period of chub mackerel in the western North Pacific. Mar Ecol Prog Ser 624:155-166. https://doi.org/10.3354/meps13037

Nishijima, S., R. Yukami, M. Ichinokawa, and H. Okamura. 2019. Preliminary analysis of state-space stock assessment model for the chub mackerel in the Northwest Pacific. North Pacific Fishery Commission, Tokyo, JAPAN. NPFC-2019-TWG CMSA02-WP02.

TWG CMSA. 2019. Report of the 2nd TWG CMSA meeting. NPFC-2019-TWG CMSA02-Final Report.

Appendix 1. Shortcut instructions for running POPSIM-A and R software for chub mackerel.

POPSIM-A

1) Download and install the “Age Population Simulator” (POPSIM-A) from the NOAA Fisheries Toolbox download page https://nmfs-fish-tools.github.io/POPSIM-A/. It comes with an example data file if you want to experiment.
2) Put CHUB-DEMO-1.DAT into a data directory.
3) After installation, click on the POPSIM-A icon. Under “Options” on the menu bar, select your favorite text file editor using the “Select Program for Viewing Report File” (Appendix 3). The editor you choose will be used to look at a summary output file from the GUI if you choose to later. However, the file can also be viewed outside of POSIM-A in any editor of your choice.
4) Under “Options”, check “Command Prompt Stays Open When Running Population Simulator” so that the DOS window used to run the POPSIM-A C program stays open after the program finishes running. This is useful because it keeps any error messages in the open DOS window where you can see them. The disadvantage is that you have to close the window (click the upper right corner) when you are done to see POPSIM-A output screens. Otherwise, the DOS window will disappear as soon as the C program terminates.
5) From the File menu, open the CHUB-DEMO-1.DAT file using the “Open Existing Input Data File” and see the “General Data Input Screen”.
6) Reduce the number of iterations to some small number (e.g. 10). Push the “SET” button so that the change takes effect. NOTE-PUSHING THE SET BUTTON WILL ZERO OUT THE DISCARD FRACTION INPUT DATA DUE TO A BUG IN THE GUI. CHECK THE DISCARD FRACTION DATA BEFORE PROCEEDING IF YOU ARE ASSUMING DISCARDS! Data on other input screens seem unaffected.
7) Make sure the “Generate Comma Delimited Files for Iterative Results” is checked on the bottom of the “Options” input data screen (it should be already).
8) Under the “Run” menu, do “Run Population Simulator” which should take a second or two. Close the DOS window when you are ready. If the program ran well, you will see a dialog box (push “OK”) and you will see the “PopSIm Basic Results Page”. You can move back and forth between the output and input screens by using the “Windows” commands on the menu bar.
9) At some point, before or after looking at results, do “Export PopSim Data to R” under the File menu.

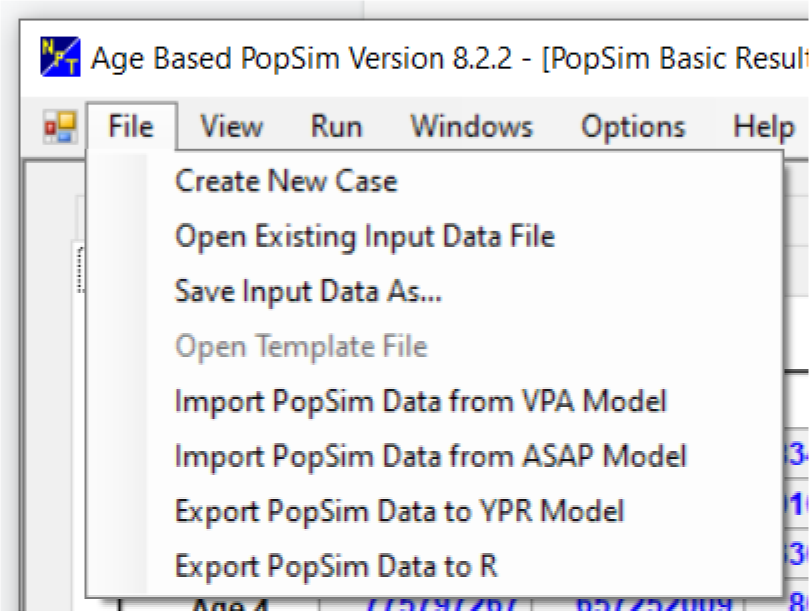
R

1) Put PopSimFunctions.R and PseudoData.R into a directory. Include the version numbers in the file names (e.g. PopSimFunctions-15.R and PseudoData-5.R).
2) Edit PseudoData.R so that R variables “adir” and “afnm” point to the data directory and base name for the data file (e.g. afnm="CHUB-DEMO-1”).

3) Edit PseudoData.R so that the R variables “codeDir” and “codeFile” point to PopSimFunctions.R
4) Source PseudoData.R in R to make a data.frame (pseudo-database) called “db” which will be saved in the data directory and named automatically (e.g. as “CHUB-DEMO-1-PseudoData.RDAT” and “CHUB-DEMO-1-PseudoData.csv”.

Appendix 2. POPSIM-A menu bar. See POPSIM-A help files and notes below each screen for additional explanations. Some commands are not active (grey font) until simulations are complete.

MENU BAR - FILE


Age Based PopSim Version 8.2.2 - [PopSim Basic Result
File
View
Run
Windows
Options
Help
Create New Case
Open Existing Input Data File
Save Input Data As...
Open Template File
Import PopSim Data from VPA Model
Import PopSim Data from ASAP Model
Export PopSim Data to YPR Model
Export PopSim Data to R

Create New Case: Start from the absolute beginning. The next steps are to set the parameters on the General Data Input Screen, push the SET button at the bottom and Save Input Data As… before adding data to other screens.

Open Existing Input Data File: Will open an existing file that may or may not be complete and workable.

Save Input Data As: Self explanatory.

Open Template File: Opens a template data file for a stock assessment model (no longer used).

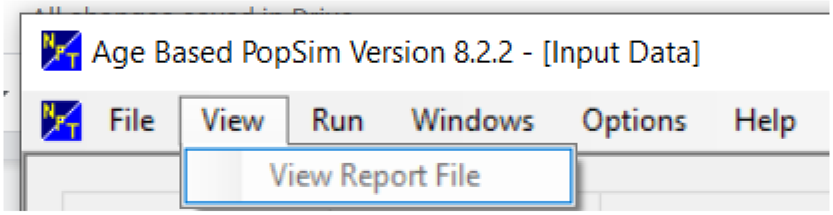
Import PopSim Data from VPA/ASAP Model: Creates a POPSIM-A data file from NEFSC VPA or ASAP assessment model output (not tested).

Export PopSim Data to YPR Model: Creates a NOAA Fisheries yield-per-recruit (YPR) model data file (not tested).

Export PopSim Data to R: Writes simulated true population results and most input controls to an RDATA file for use later in R software (e.g. for chub mackerel). Do this after simulations are complete.

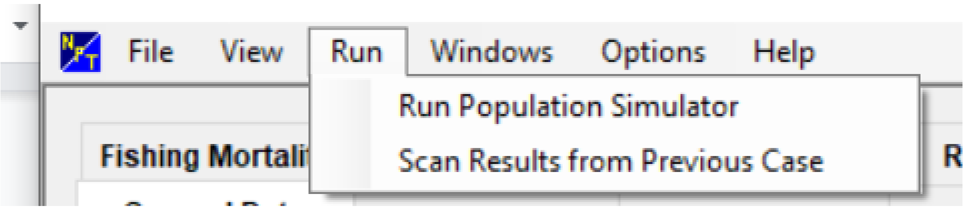
.

MENU BAR - VIEW



View Report File: Opens a text editor (specified by user under Options) to view the main POPSIM-A output file (e.g. CHUB-DEMO-1.out) after simulations are complete. Command activates after POPSIM-A has run.

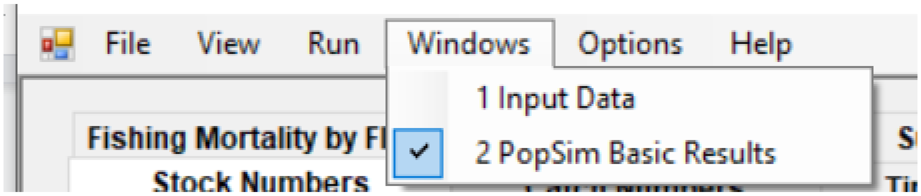
MENU BAR - RUN



Run Population Simulator: Checks input data and runs POPSIM-A.

Scan Results from Previous Case: Reads results from previous runs and reconstructs information on output screens. This function is meant to save the user from rerunning scenarios with thousands of iterations and possibly different random number seeds.

MENU BAR - WINDOWS

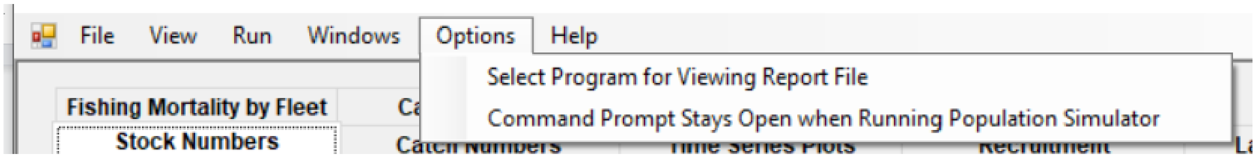


Input Data: View input data screens

PopSim Basic Results: View simulation output after simulation results are complete.

This window is not active until simulations are complete or previous results have been scanned in.

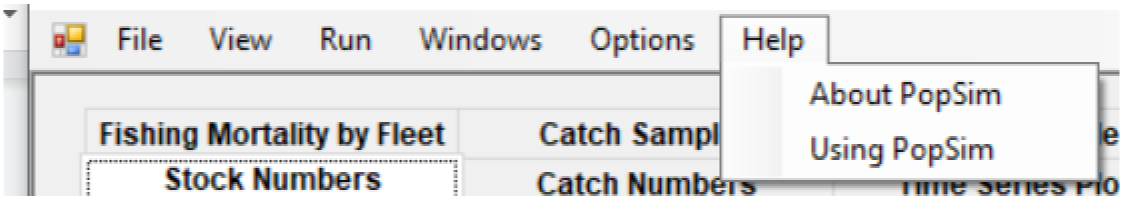
MENU BAR - OPTIONS


File
View
Run
Windows
Options
Help
Select Program for Viewing Report File
Command Prompt Stays Open when Running Population Simulator
Fishing Mortality by Fleet
Stock Numbers
Catch Numbers
Time Series Plots
Recruitment

Select Program for Viewing Report File: Opens a dialog box used to select a text file editor for viewing report files. You can select an editor by navigating either to its Windows shortcut file or, in some cases, the executable file.

Command Prompt Stays Open when Running Population Simulator: Select this option to see error messages written to the DOS window as the POPSIM-A simulation program is run from the GUI . The DOS window will then stay open when the underlying simulation program written in C terminates normally or because of an error. This is useful particularly in building and debugging data files because the error messages (e.g. unable to open a file) are helpful in fixing problems. POPSIM-A will stop completely and the output screen will not be visible until you close the command prompt by clicking in the upper right hand corner.

MENU BAR - HELP

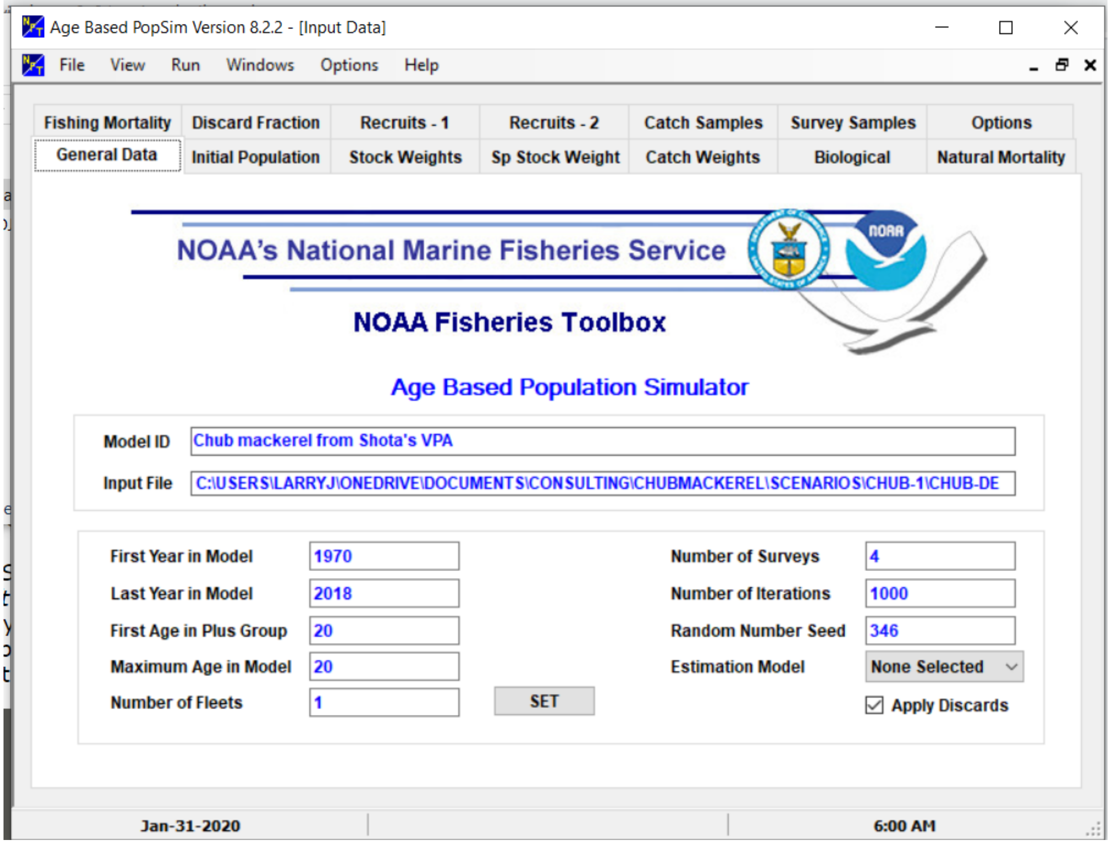


About PopSim: Shows NOAA’s copyright, compile and build dates.

Using PopSim: Opens useful help files that the user should read.

Appendix 3. POPSIM-A data input screens. See POPSIM-A help files and notes below each screen for explanations. Right click on any input grid to see options for cutting, pasting, filling right, filling down, etc. but be careful as these commands may work differently than in other applications. Be sure to scroll around large input screens to check that all cells have been filled. The information manipulated in the boxes is text and it is possible to enter characters where you intend a numeric value. For example, “10o00” instead of “10000”. Format numbers to the desired precision before pasting them into an input screen,

GENERAL DATA INPUT SCREEN



Model ID: Chub mackerel from Shota's VPA

Input File: C:\USERS\LARRYJ\ONEDRIVE\DOCUMENTS\CONSULTING\CHUBMACKEREL\SCENARIOS\CHUB-1\CHUB-DE

First Year in Model: 1970

Last Year in Model: 2018

First Age in Plus Group: 20

Maximum Age in Model: 20

Number of Fleets: 1

Number of Surveys: 4

Number of Iterations: 1000

Random Number Seed: 346

Estimation Model: None Selected

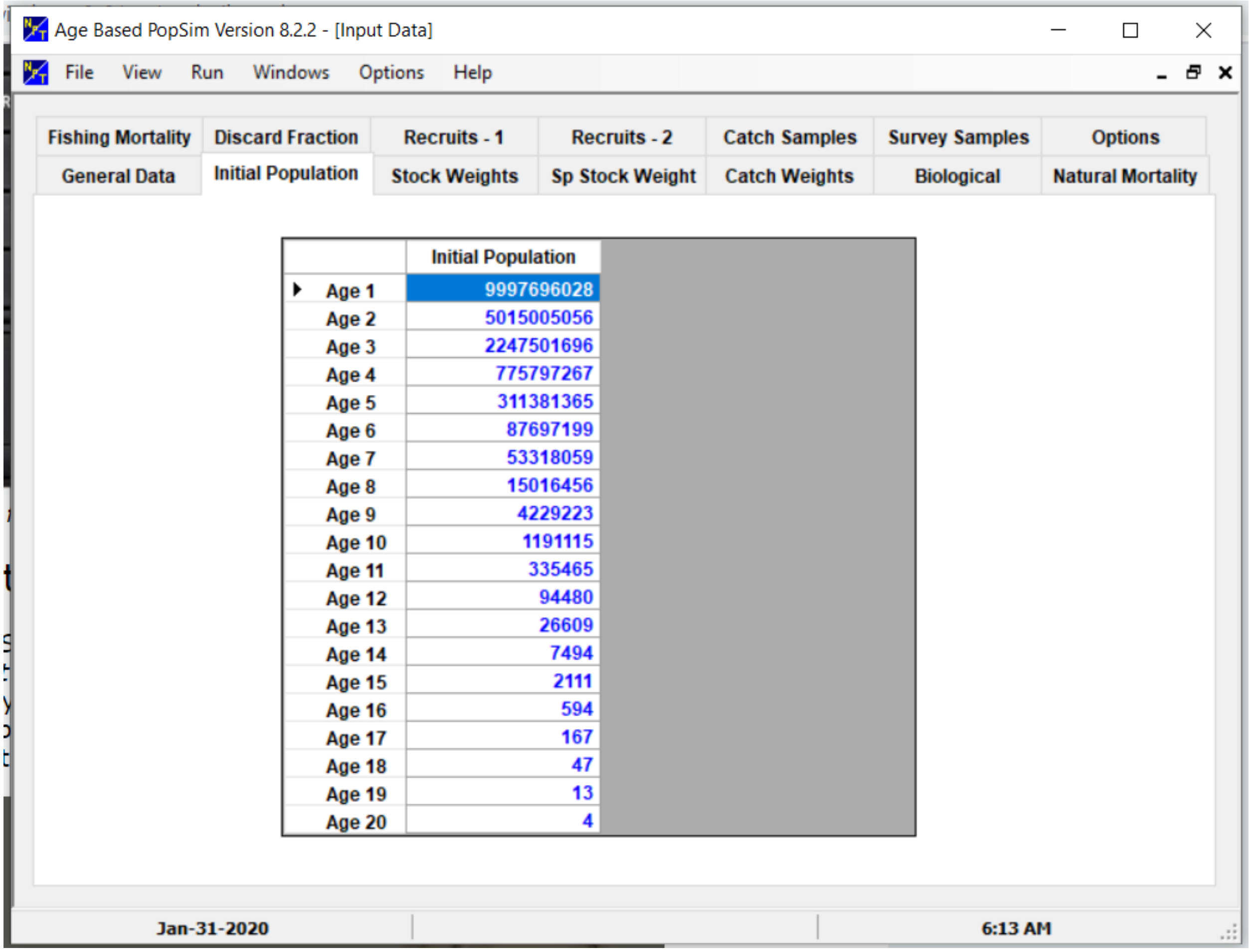
☑ Apply Discards

Notes:

1) First and last year for catch data. POPSIM-A carries out population dynamics calculations through the last year of catch data + 1.
2) "Estimation Model" can be one of four stock assessment models available in the NOAA toolbox. However, this feature has not been maintained and no longer functions with updated assessment models.
3) The "Apply Discards" checkbox affects pseudo-data only. If checked, the "Discard Fraction" input screen is activated and pseudo-data for landings (catch minus discard) are produced.
4) The "SET" button is the most important control on this screen because it causes all of the specifications and changes to inputs on the screen to be incorporated into other screens and arrays used for data and calculations. For example, a change to the "Number of Surveys" has no effect until the SET button is pushed. After the button is pushed, input screens on the "Survey Samples" screen will be dimensioned correctly. Push SET early and often!
5) NOTE-PUSHING THE SET BUTTON WILL ZERO OUT THE DISCARD FRACTION INPUT DATA DUE TO A BUG IN THE GUI. CHECK THE DISCARD DATA AFTER

PUSHING SET IF YOU ASSUME DISCARDING OCCURS. Data on other input screens seem unaffected.

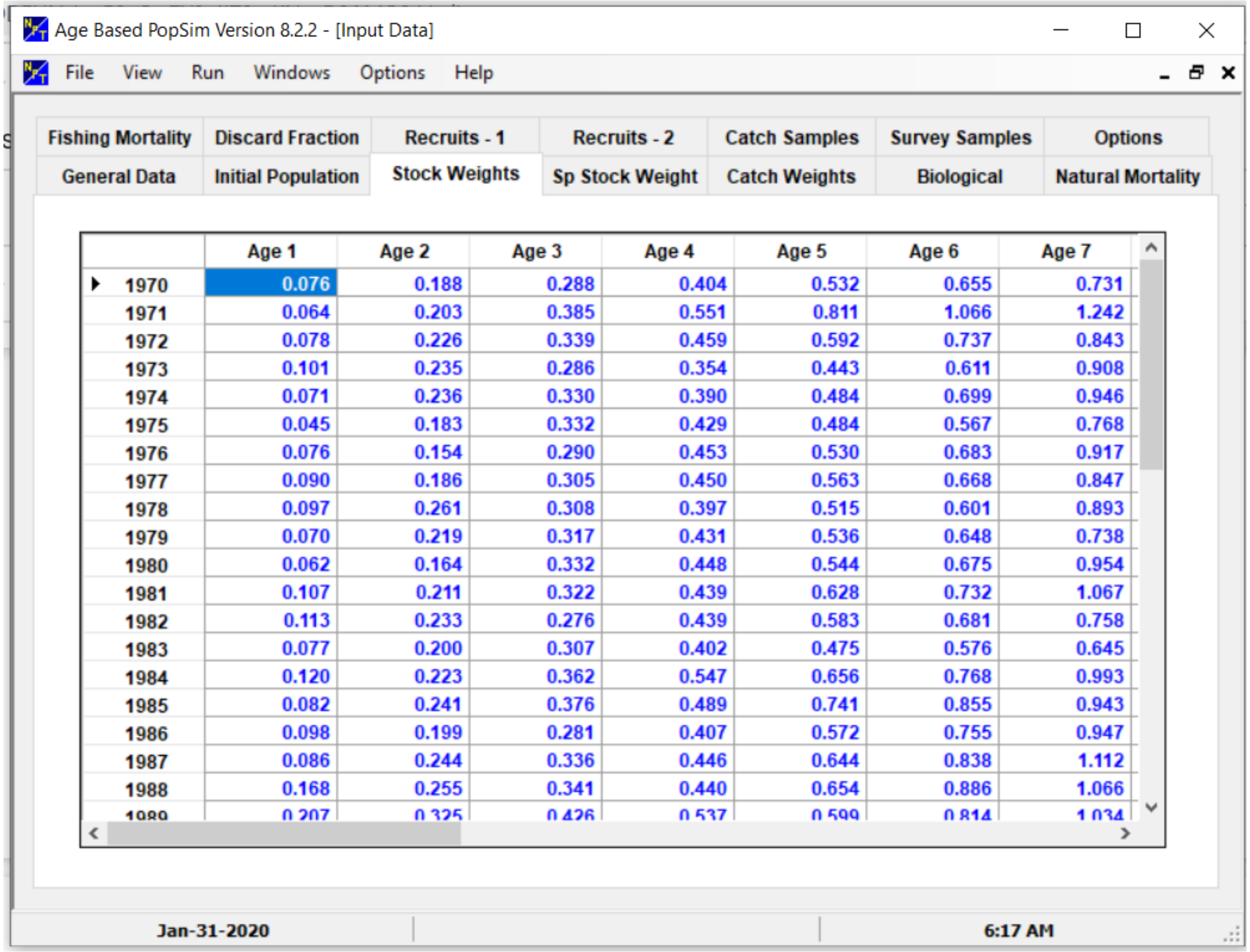
INITIAL POPULATION INPUT SCREEN



	Initial Population
Age 1	9997696028
Age 2	5015005056
Age 3	2247501696
Age 4	775797267
Age 5	311381365
Age 6	87697199
Age 7	53318059
Age 8	15016456
Age 9	4229223
Age 10	1191115
Age 11	335465
Age 12	94480
Age 13	26609
Age 14	7494
Age 15	2111
Age 16	594
Age 17	167
Age 18	47
Age 19	13
Age 20	4

1) The number of age 1 fish on this screen is always recruitment in the first year, even if a spawner recruit model is used for subsequent years. Individual recruitments may be entered for each year in a later screen but they will be for the second year with catch data to the last year with catch data + 1.
2) Units are numbers of fish for chub mackerel and units for mean weights (below) are kg so that spawning biomass, catch biomass, etc. will be in metric tons. POPSIM-A assumes this convention but other units can be used if care is exercised.

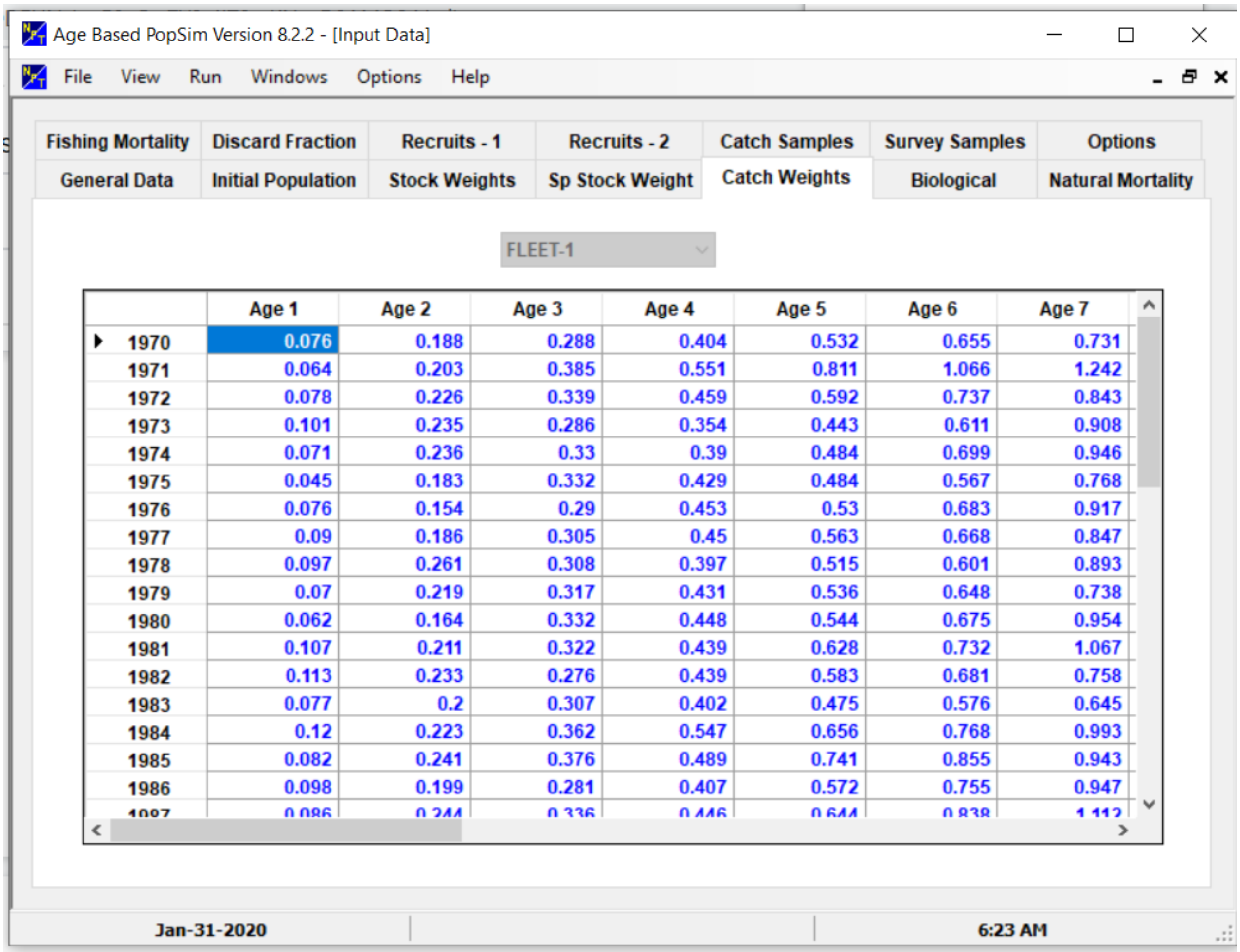
STOCK WEIGHTS AND SPAWNING STOCK WEIGHTS INPUT SCREENS



	Age 1	Age 2	Age 3	Age 4	Age 5	Age 6	Age 7
1970	0.076	0.188	0.288	0.404	0.532	0.655	0.731
1971	0.064	0.203	0.385	0.551	0.811	1.066	1.242
1972	0.078	0.226	0.339	0.459	0.592	0.737	0.843
1973	0.101	0.235	0.286	0.354	0.443	0.611	0.908
1974	0.071	0.236	0.330	0.390	0.484	0.699	0.946
1975	0.045	0.183	0.332	0.429	0.484	0.567	0.768
1976	0.076	0.154	0.290	0.453	0.530	0.683	0.917
1977	0.090	0.186	0.305	0.450	0.563	0.668	0.847
1978	0.097	0.261	0.308	0.397	0.515	0.601	0.893
1979	0.070	0.219	0.317	0.431	0.536	0.648	0.738
1980	0.062	0.164	0.332	0.448	0.544	0.675	0.954
1981	0.107	0.211	0.322	0.439	0.628	0.732	1.067
1982	0.113	0.233	0.276	0.439	0.583	0.681	0.758
1983	0.077	0.200	0.307	0.402	0.475	0.576	0.645
1984	0.120	0.223	0.362	0.547	0.656	0.768	0.993
1985	0.082	0.241	0.376	0.489	0.741	0.855	0.943
1986	0.098	0.199	0.281	0.407	0.572	0.755	0.947
1987	0.086	0.244	0.336	0.446	0.644	0.838	1.112
1988	0.168	0.255	0.341	0.440	0.654	0.886	1.066
1989	0.207	0.325	0.426	0.537	0.599	0.814	1.034

1) Units are KG for chub mackerel so that biomass estimates are MT (POPSIM-A convention).
2) These affect total stock biomass in POPSIM-A and survey catch weights in R calculations.
3) The "Sp Stock Weights" input screen (not shown) on the next tab is similar but used only to compute spawning biomass.

CATCH WEIGHTS INPUT SCREEN



FLEET-1

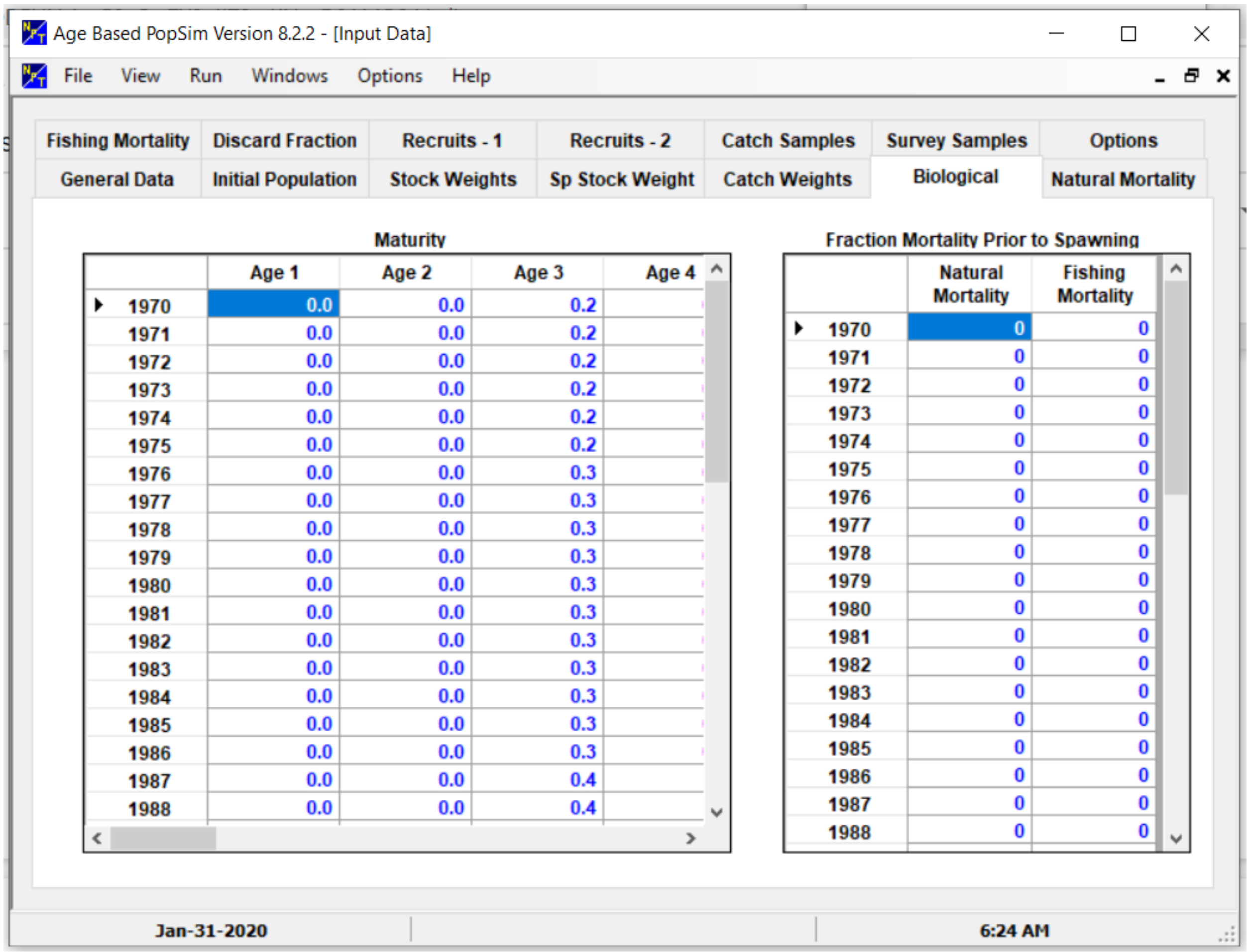
	Age 1	Age 2	Age 3	Age 4	Age 5	Age 6	Age 7
1970	0.076	0.188	0.288	0.404	0.532	0.655	0.731
1971	0.064	0.203	0.385	0.551	0.811	1.066	1.242
1972	0.078	0.226	0.339	0.459	0.592	0.737	0.843
1973	0.101	0.235	0.286	0.354	0.443	0.611	0.908
1974	0.071	0.236	0.33	0.39	0.484	0.699	0.946
1975	0.045	0.183	0.332	0.429	0.484	0.567	0.768
1976	0.076	0.154	0.29	0.453	0.53	0.683	0.917
1977	0.09	0.186	0.305	0.45	0.563	0.668	0.847
1978	0.097	0.261	0.308	0.397	0.515	0.601	0.893
1979	0.07	0.219	0.317	0.431	0.536	0.648	0.738
1980	0.062	0.164	0.332	0.448	0.544	0.675	0.954
1981	0.107	0.211	0.322	0.439	0.628	0.732	1.067
1982	0.113	0.233	0.276	0.439	0.583	0.681	0.758
1983	0.077	0.2	0.307	0.402	0.475	0.576	0.645
1984	0.12	0.223	0.362	0.547	0.656	0.768	0.993
1985	0.082	0.241	0.376	0.489	0.741	0.855	0.943
1986	0.098	0.199	0.281	0.407	0.572	0.755	0.947
1987	0.086	0.244	0.336	0.446	0.644	0.838	1.112

Jan-31-2020

6:23 AM

1) Catch weights used in fishery calculations (KG) are for each fleet year and age.

BIOLOGICAL INPUT SCREEN



Maturity

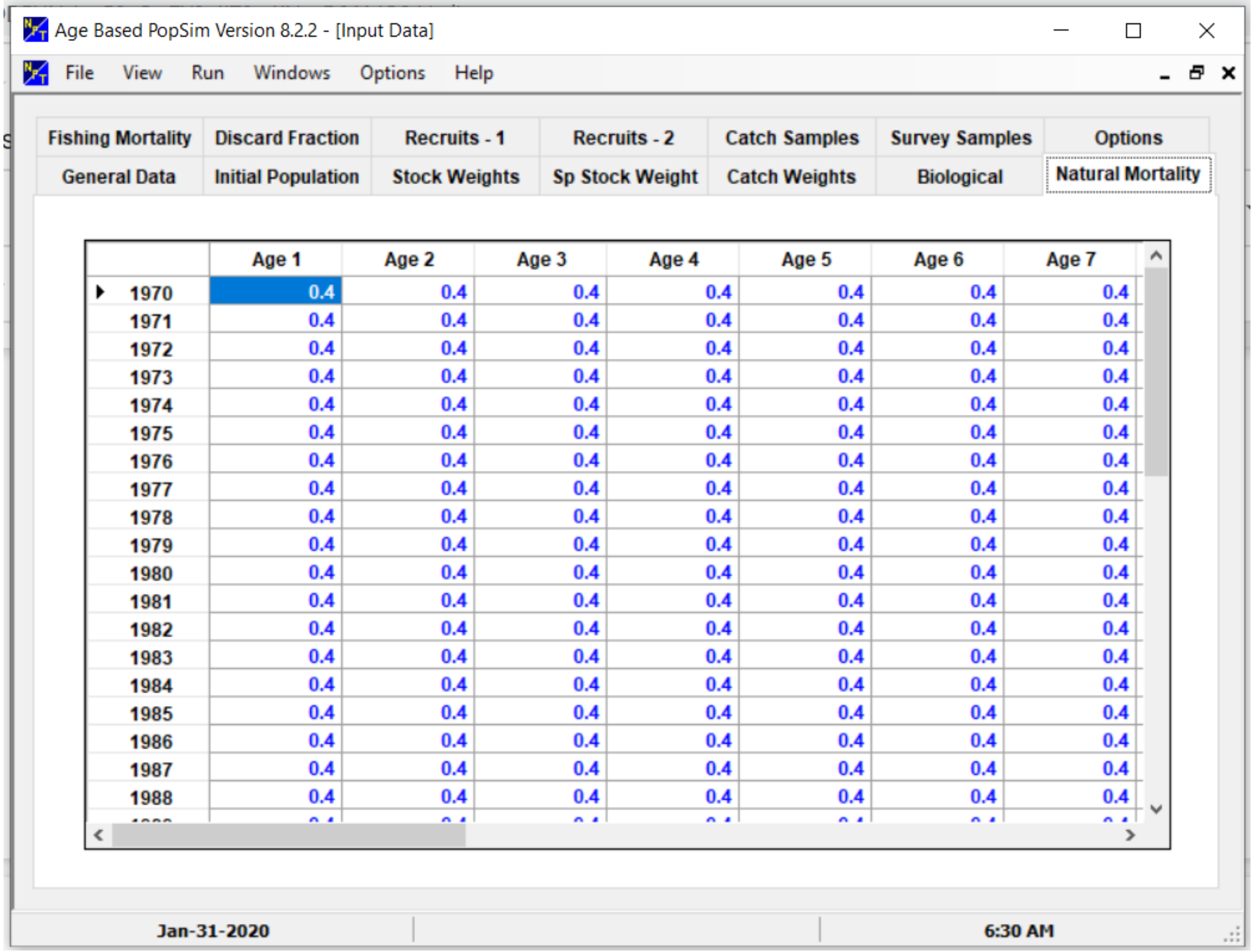
	Age 1	Age 2	Age 3	Age 4
1970	0.0	0.0	0.2	
1971	0.0	0.0	0.2	
1972	0.0	0.0	0.2	
1973	0.0	0.0	0.2	
1974	0.0	0.0	0.2	
1975	0.0	0.0	0.2	
1976	0.0	0.0	0.3	
1977	0.0	0.0	0.3	
1978	0.0	0.0	0.3	
1979	0.0	0.0	0.3	
1980	0.0	0.0	0.3	
1981	0.0	0.0	0.3	
1982	0.0	0.0	0.3	
1983	0.0	0.0	0.3	
1984	0.0	0.0	0.3	
1985	0.0	0.0	0.3	
1986	0.0	0.0	0.3	
1987	0.0	0.0	0.4	
1988	0.0	0.0	0.4	

Fraction Mortality Prior to Spawning

	Natural Mortality	Fishing Mortality
1970	0	0
1971	0	0
1972	0	0
1973	0	0
1974	0	0
1975	0	0
1976	0	0
1977	0	0
1978	0	0
1979	0	0
1980	0	0
1981	0	0
1982	0	0
1983	0	0
1984	0	0
1985	0	0
1986	0	0
1987	0	0
1988	0	0

1) The proportion of each age group that is sexually mature must be specified for each age and year.
2) The fraction of natural and fishing mortality before spawning might, for example, be both set at 0.5 for midyear spawning or at 0 for year round spawning. Alternatively, the fractions might differ, for example, if there is a pulse fishery that operates at the end of the year on spawning aggregations.
3) The fractions can have substantial effect on spawning biomass calculations and comparability of spawning biomass calculated by POPSIM-A and other programs (e.g. VPA), particularly when mortality rates are high even though there is no effect on calculated fishing mortality rates, January 1 stock size, etc.

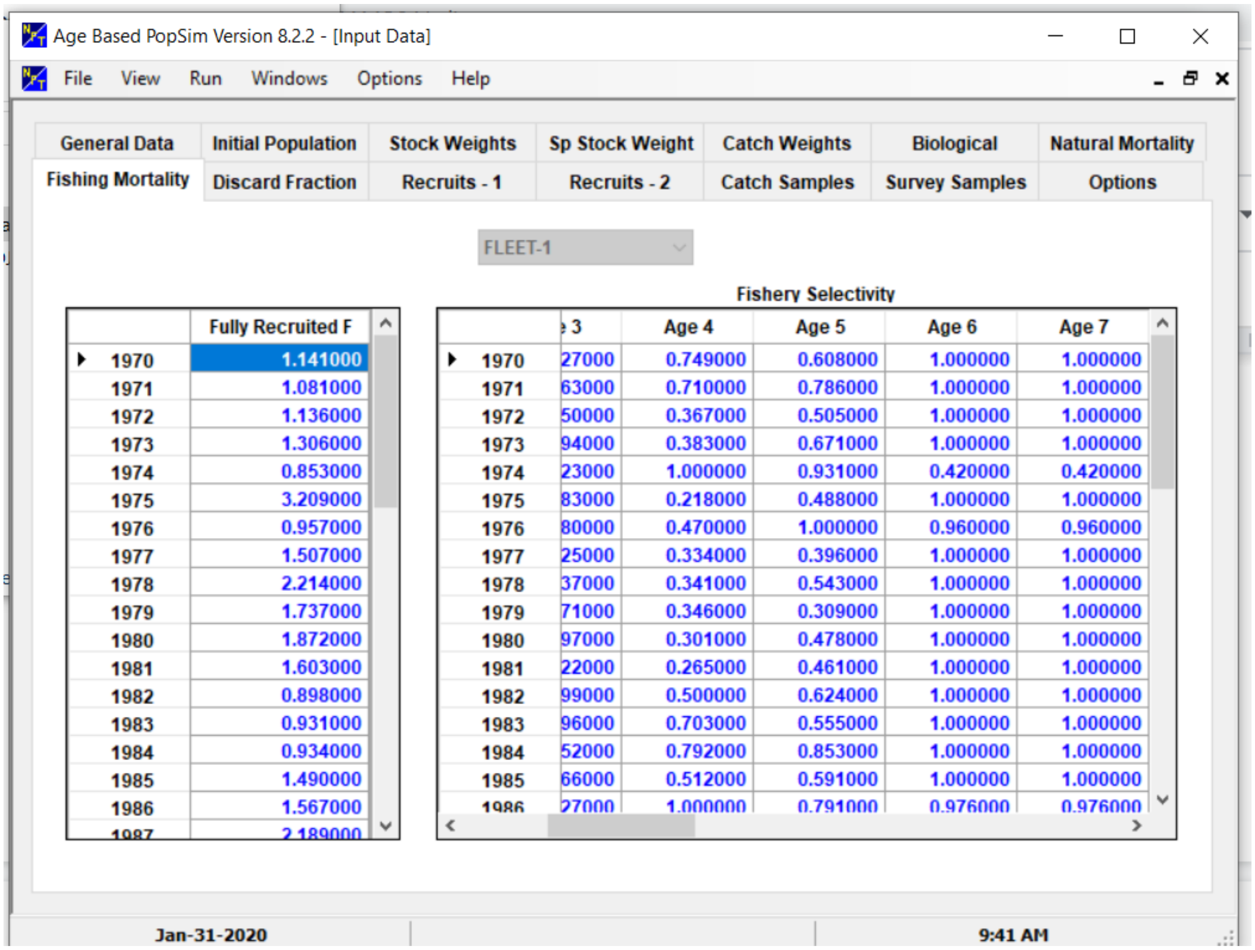
NATURAL MORTALITY INPUT SCREEN



	Age 1	Age 2	Age 3	Age 4	Age 5	Age 6	Age 7
1970	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1971	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1972	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1973	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1974	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1975	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1976	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1977	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1978	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1979	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1980	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1981	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1982	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1983	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1984	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1985	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1986	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1987	0.4	0.4	0.4	0.4	0.4	0.4	0.4
1988	0.4	0.4	0.4	0.4	0.4	0.4	0.4

1) Self-explanatory.

FISHING MORTALITY INPUT SCREEN



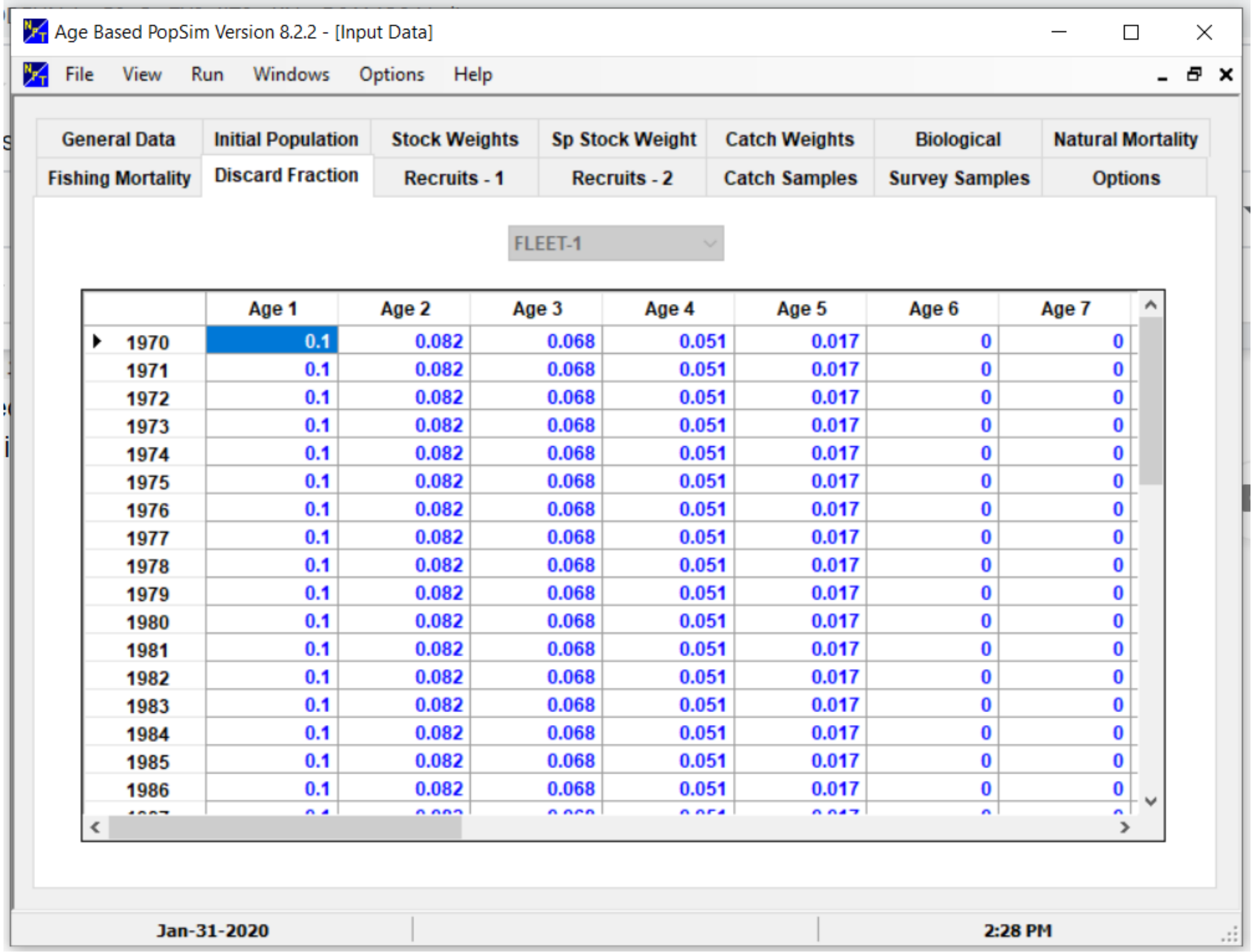
	Fully Recruited F
1970	1.141000
1971	1.081000
1972	1.136000
1973	1.306000
1974	0.853000
1975	3.209000
1976	0.957000
1977	1.507000
1978	2.214000
1979	1.737000
1980	1.872000
1981	1.603000
1982	0.898000
1983	0.931000
1984	0.934000
1985	1.490000
1986	1.567000
1987	2.189000

Fishery Selectivity

	3	Age 4	Age 5	Age 6	Age 7
1970	27000	0.749000	0.608000	1.000000	1.000000
1971	63000	0.710000	0.786000	1.000000	1.000000
1972	50000	0.367000	0.505000	1.000000	1.000000
1973	94000	0.383000	0.671000	1.000000	1.000000
1974	23000	1.000000	0.931000	0.420000	0.420000
1975	83000	0.218000	0.488000	1.000000	1.000000
1976	80000	0.470000	1.000000	0.960000	0.960000
1977	25000	0.334000	0.396000	1.000000	1.000000
1978	37000	0.341000	0.543000	1.000000	1.000000
1979	71000	0.346000	0.309000	1.000000	1.000000
1980	97000	0.301000	0.478000	1.000000	1.000000
1981	22000	0.265000	0.461000	1.000000	1.000000
1982	99000	0.500000	0.624000	1.000000	1.000000
1983	96000	0.703000	0.555000	1.000000	1.000000
1984	52000	0.792000	0.853000	1.000000	1.000000
1985	66000	0.512000	0.591000	1.000000	1.000000
1986	27000	1.000000	0.791000	0.976000	0.976000

1) Fishing mortality at age in POPSIM-A is computed as the product of the “Fully Recruited F” and “Fishery Selectivity” exactly as the two terms are input on the screen and without any adjustments. For example, the product of F = 2 and selectivity = (1, 1, 1, 1) will give the same fishing mortality at age in simulations as F=1 and selectivity = (2, 2, 2, 2). For the sake of definition and interpretation, the selectivity terms for each fleet in each year should be scaled so that the age with the maximum fishing mortality has selectivity = 1 and *F* is defined as the fishing mortality on the fully recruited age group (i.e. with maximum selectivity = 1), which may change from year to year, and is theoretically proportional to fishing effort. This convention seems to be assumed in POPSIM-A documentation, input and output screens but is not enforced in calculations (the program will use the values as entered). Another option is to scale the selectivity terms for each year and fleet to 1 at a particular age so that F is defined as the fishing mortality on that age.

DISCARD FRACTION INPUT SCREEN

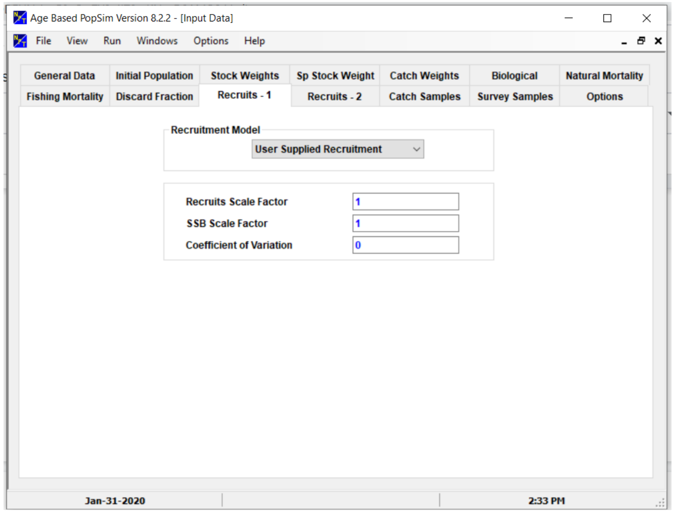


FLEET-1

	Age 1	Age 2	Age 3	Age 4	Age 5	Age 6	Age 7
1970	0.1	0.082	0.068	0.051	0.017	0	0
1971	0.1	0.082	0.068	0.051	0.017	0	0
1972	0.1	0.082	0.068	0.051	0.017	0	0
1973	0.1	0.082	0.068	0.051	0.017	0	0
1974	0.1	0.082	0.068	0.051	0.017	0	0
1975	0.1	0.082	0.068	0.051	0.017	0	0
1976	0.1	0.082	0.068	0.051	0.017	0	0
1977	0.1	0.082	0.068	0.051	0.017	0	0
1978	0.1	0.082	0.068	0.051	0.017	0	0
1979	0.1	0.082	0.068	0.051	0.017	0	0
1980	0.1	0.082	0.068	0.051	0.017	0	0
1981	0.1	0.082	0.068	0.051	0.017	0	0
1982	0.1	0.082	0.068	0.051	0.017	0	0
1983	0.1	0.082	0.068	0.051	0.017	0	0
1984	0.1	0.082	0.068	0.051	0.017	0	0
1985	0.1	0.082	0.068	0.051	0.017	0	0
1986	0.1	0.082	0.068	0.051	0.017	0	0

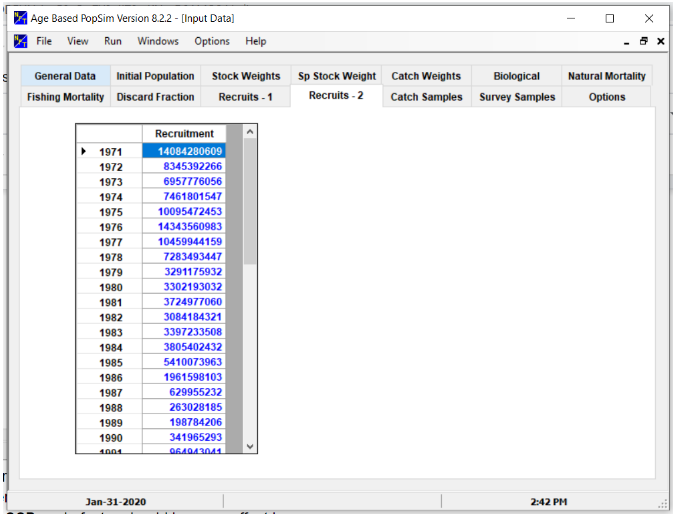
1) If the “Apply Discards” check box is marked on the first screen, then this screen will be active for each fleet. The numbers entered are the proportions of each age group in each year discarded dead (so called “mortal discard” and excluding discards that survive the capture/discard process).
2) NOTE-PUSHING THE SET BUTTON ON THE GENERAL DATA SCREEN WILL ZERO OUT THE DISCARD FRACTION INPUT DATA DUE TO A BUG IN THE GUI.

RECRUITS-1 AND RECRUITS-2 INPUT SCREENS FOR USER SUPPLIED RECRUITMENT (OPTION 1)


Age Based PopSim Version 8.2.2 - [Input Data]
File View Run Windows Options Help
General Data | Initial Population | Stock Weights | Sp Stock Weight | Catch Weights | Biological | Natural Mortality
Fishing Mortality | Discard Fraction | Recruits - 1 | Recruits - 2 | Catch Samples | Survey Samples | Options
Jan-31-2020
2:33 PM

Recruitment Model: User Supplied Recruitment

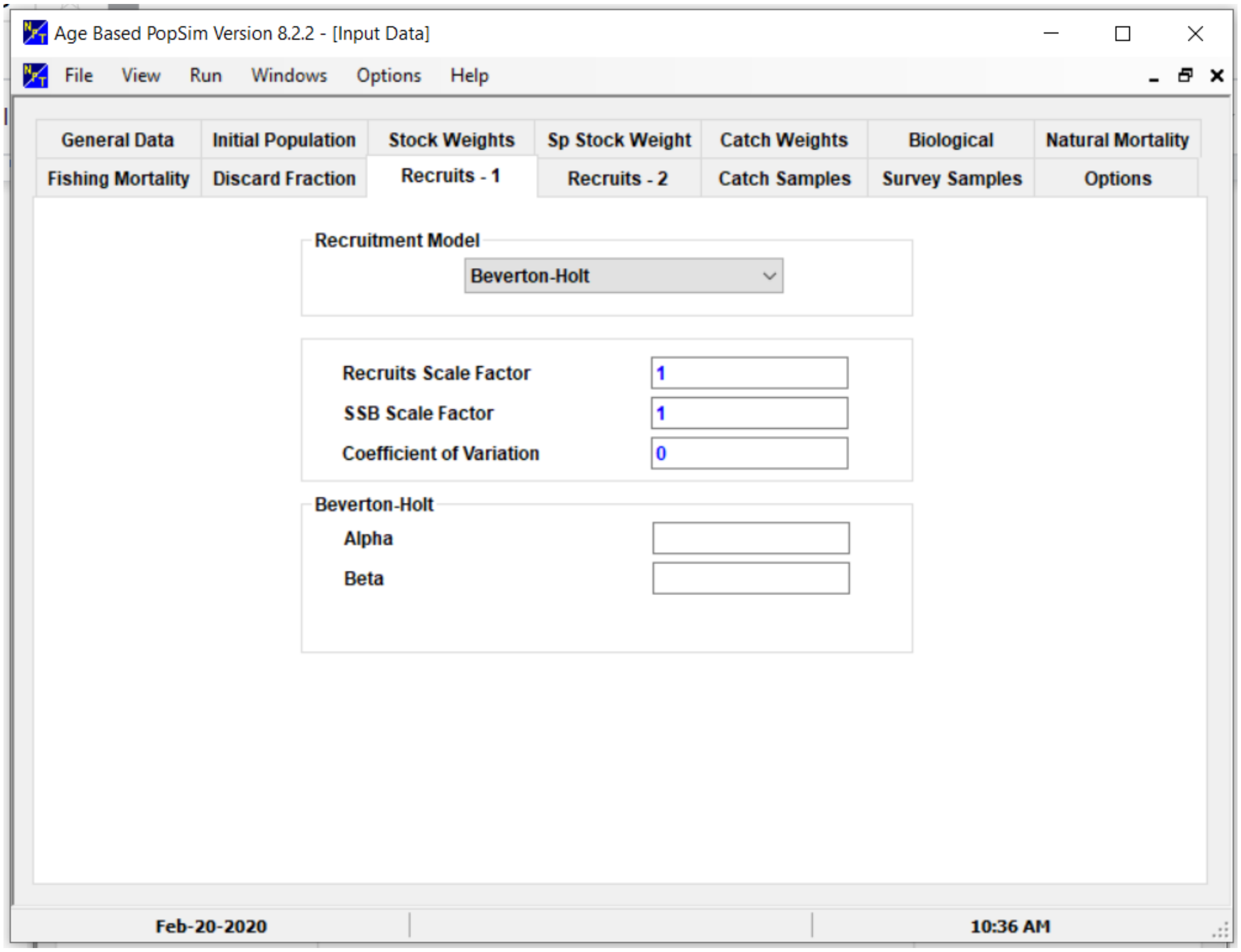
Recruits Scale Factor	1
SSB Scale Factor	1
Coefficient of Variation	0


Age Based PopSim Version 8.2.2 - [Input Data]
File View Run Windows Options Help
General Data | Initial Population | Stock Weights | Sp Stock Weight | Catch Weights | Biological | Natural Mortality
Fishing Mortality | Discard Fraction | Recruits - 1 | Recruits - 2 | Catch Samples | Survey Samples | Options
Jan-31-2020
2:42 PM

	Recruitment
1971	14084280609
1972	8345392266
1973	6957776056
1974	7461801547
1975	10095472453
1976	14343560983
1977	10459944159
1978	7283493447
1979	3291175932
1980	3302193032
1981	3724977060
1982	3084184321
1983	3397233508
1984	3805402432
1985	5410073963
1986	1961598103
1987	629955232
1988	263028185
1989	198784206
1990	341965293
1991	964943041

1) Option 1 was used for chub mackerel and the values shown are numbers of age zero fish from the Japanese VPA. POPSIM-A will use these values for the year indicated. Note - they start in the year after the first year in the model.
2) Recruits as entered are multiplied by the “Recruits Scale Factor” before calculations. If you enter recruits in thousands on the next page, then use 1000 here for the scaling factor.
3) The SSB scale factor should have no effect using this Option.
4) Recruitments will include lognormal process errors if the “Coefficient of Variation” > 0. The results are multiplied by a bias correction factor $e^{-0.5s^2}$ before use in calculations where s^2 is the log scale variance corresponding to the arithmetic CV.

RECRUITS-1 INPUT SCREEN FOR THE BEVERTON-HOLT RECRUITMENT MODEL (OPTION 2)



Recruitment Model: Beverton-Holt

Recruits Scale Factor	1
SSB Scale Factor	1
Coefficient of Variation	0

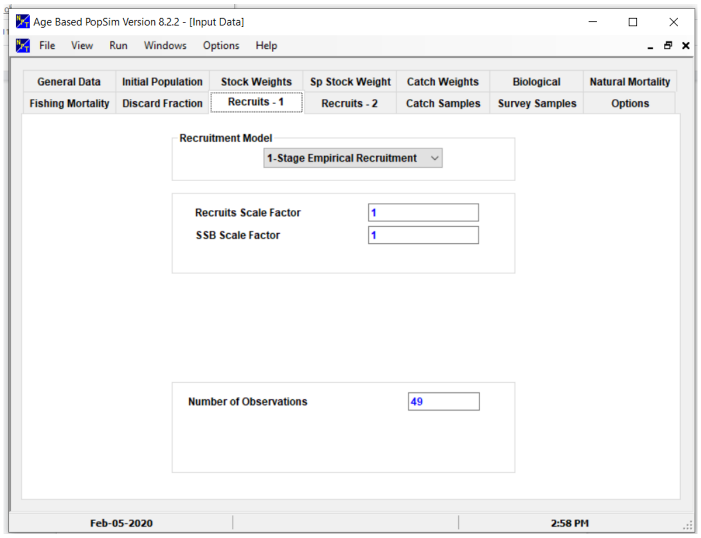
Beverton-Holt

Alpha	
Beta	

Feb-20-2020 10:36 AM

1) The Recruits-2 screen is not necessary and turned off.
2) Self explanatory but be careful that the units implied by the parameters alpha and beta are the same as used in POPSIM-A or that the Recruits and SSB scale factors are set appropriately.
3) Input screens for the Ricker (Option 3) and Shepherd (Option 4) models are similar and not shown.

RECRUITS-1 AND RECRUITS-2 INPUT SCREENS FOR THE 1-STAGE EMPIRICAL RECRUITMENT MODEL (OPTION 5).

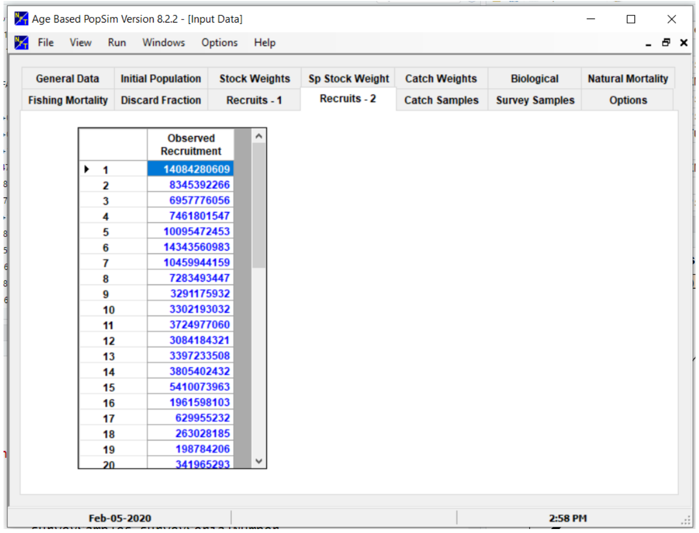


Recruitment Model: 1-Stage Empirical Recruitment

Recruits Scale Factor: 1

SSB Scale Factor: 1

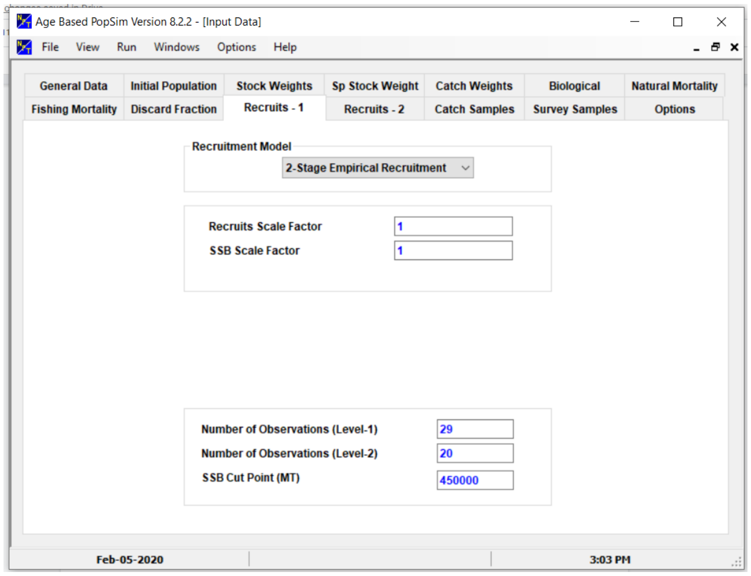
Number of Observations: 49



	Observed Recruitment
1	14084280609
2	8345392266
3	6957776056
4	7461801547
5	10095472453
6	14343560983
7	10459944159
8	7283493447
9	3291175932
10	3302193032
11	3724977060
12	3084184321
13	3397233508
14	3805402432
15	5410073963
16	1961598103
17	629955232
18	263028185
19	198784206
20	341965293

1) The same set of VPA recruitments are shown here but they will be applied in random order (by bootstrapping), rather than used in the year for which they were originally supplied.
2) No additional lognormal process errors can not be included with this recruitment option.

RECRUITS-1 AND RECRUITS-2 INPUT SCREENS FOR THE 2-STAGE EMPIRICAL RECRUITMENT MODEL (OPTION 6).



Recruitment Model: 2-Stage Empirical Recruitment

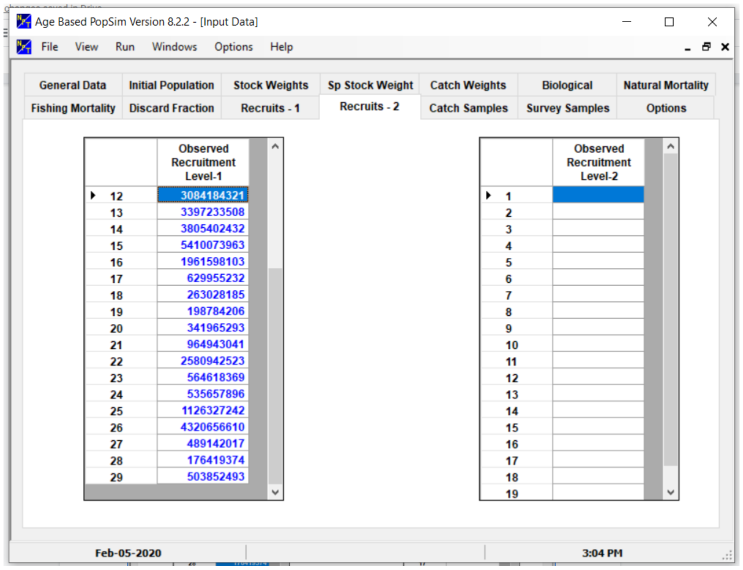
Recruits Scale Factor: 1

SSB Scale Factor: 1

Number of Observations (Level-1): 29

Number of Observations (Level-2): 20

SSB Cut Point (MT): 450000

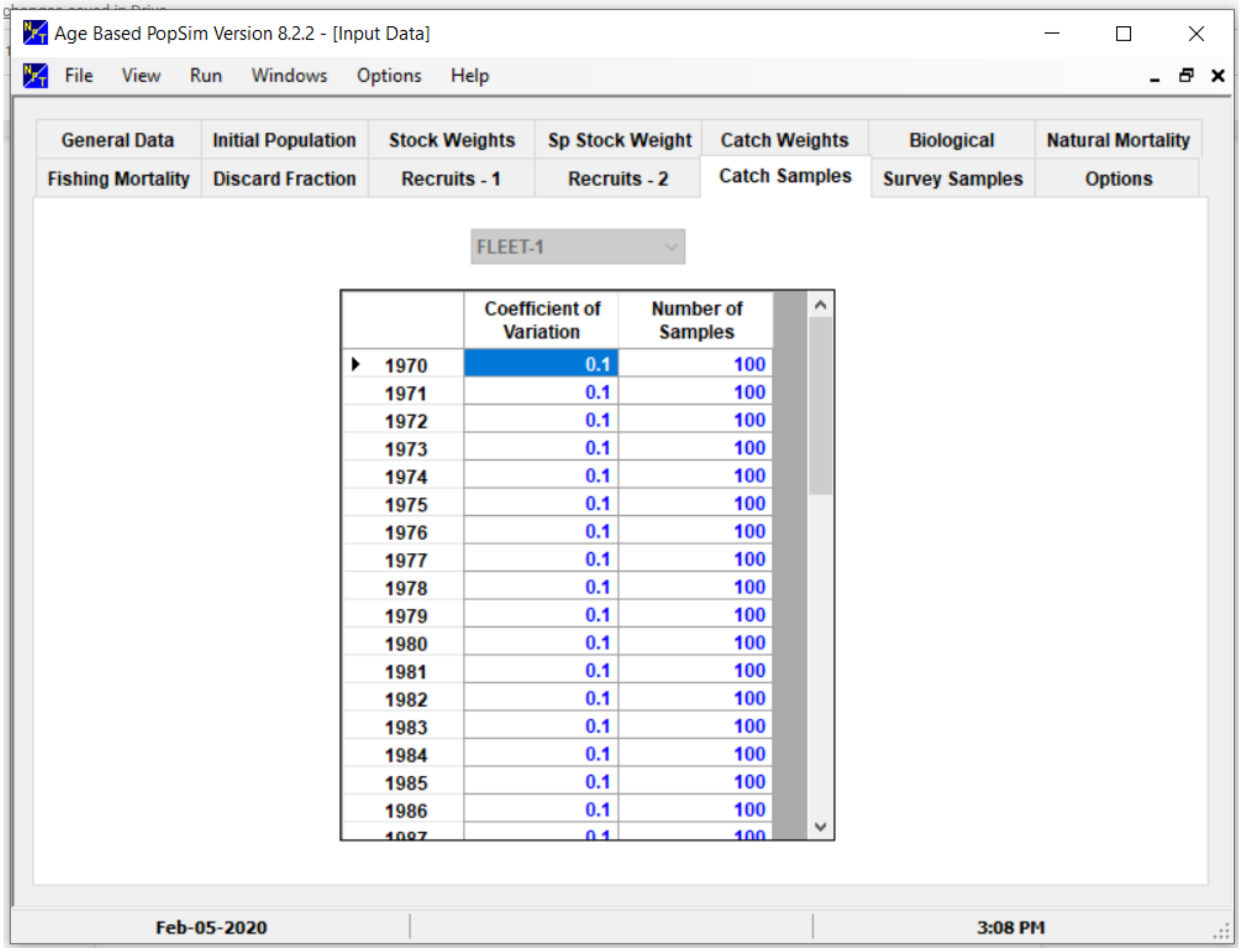


	Observed Recruitment Level-1
12	3084184321
13	3397233508
14	3805402432
15	5410073963
16	1961598103
17	629955232
18	263028185
19	198784206
20	341965293
21	964943041
22	2580942523
23	564618369
24	535657896
25	1126327242
26	4320656610
27	489142017
28	176419374
29	503852493

	Observed Recruitment Level-2
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	

1) The 2-Stage Empirical Recruitment Option example shown here will draw a recruitment from the set of 29 values on the left if SSB < 450,000 mt and from the set of 20 on the right if SSB is higher.
2) This a bootstrapping approach with some density dependent effects. It may be appropriate if the recruitments when spawning stock biomass is low tend to be higher or lower on average than when spawning biomass is higher.
3) No additional lognormal process errors can not be included with this recruitment option.

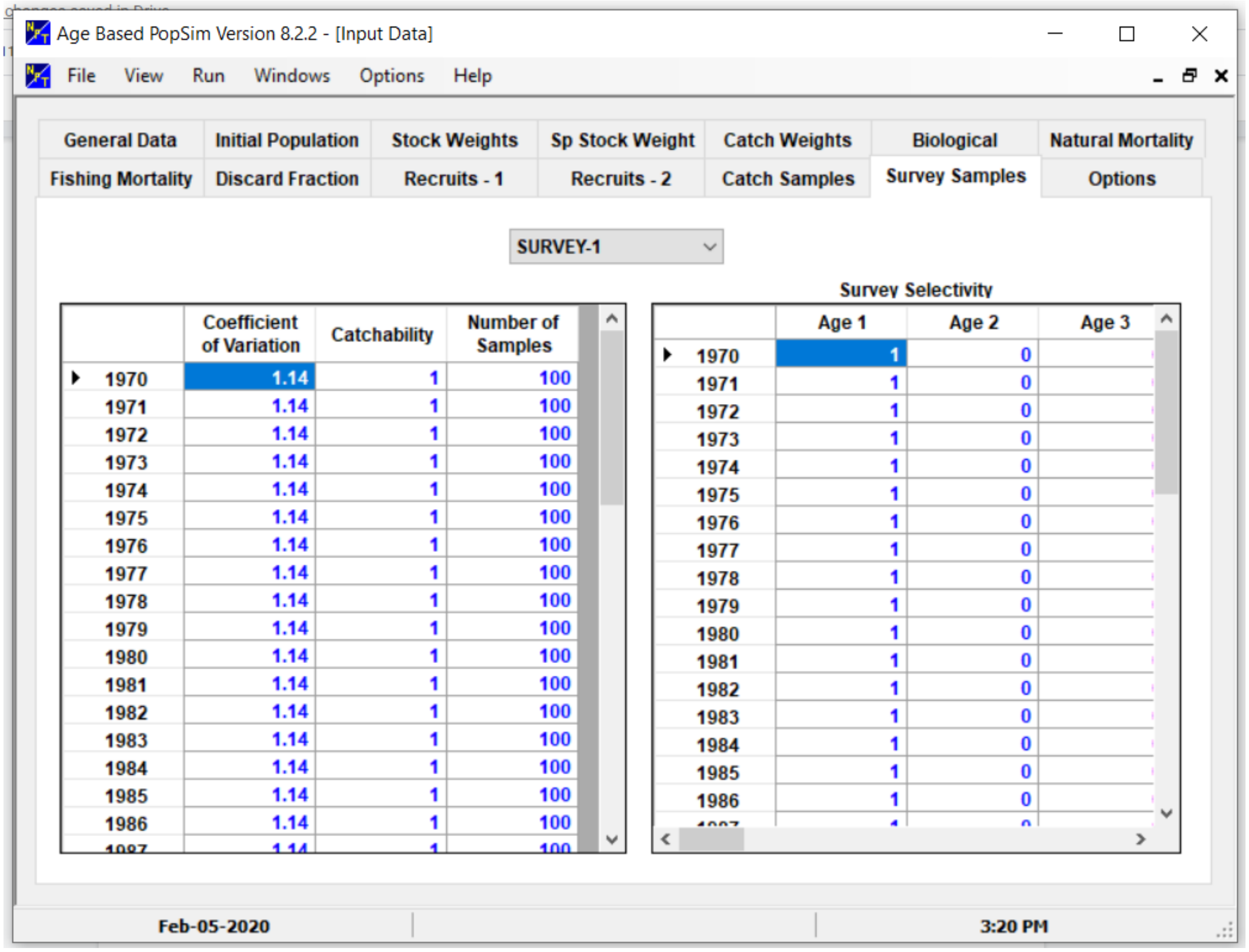
CATCH SAMPLES INPUT SCREEN



	Coefficient of Variation	Number of Samples
1970	0.1	100
1971	0.1	100
1972	0.1	100
1973	0.1	100
1974	0.1	100
1975	0.1	100
1976	0.1	100
1977	0.1	100
1978	0.1	100
1979	0.1	100
1980	0.1	100
1981	0.1	100
1982	0.1	100
1983	0.1	100
1984	0.1	100
1985	0.1	100
1986	0.1	100
1987	0.1	100

1) The catch pseudo-data (total numbers and total numbers at age) for Fleet-1 will have lognormal errors with log scale variance determined by the arithmetic CV specified here for each year.
2) Catch-at-age proportions for each year will be drawn from a multinomial distribution with the true catch proportions as parameters and using “Number of samples” as the effective sample size.
3) Thus, total numbers at age will have multinomial errors which are correlated for ages in the same year and fleet plus lognormal errors that are independent among years and fleets but correlated for age groups in the same year and fleet. The lognormal errors for catch-at-age will disappear if the catch-at-age pseudo-data are converted to proportions later by the user for use in modeling (e.g. for the Stock Synthesis model).
4) The effective sample size is generally lower than the number of fish or vessels sampled because samples are usually not independent.
5) Specify for each fleet.

SURVEY SAMPLES INPUT SCREEN



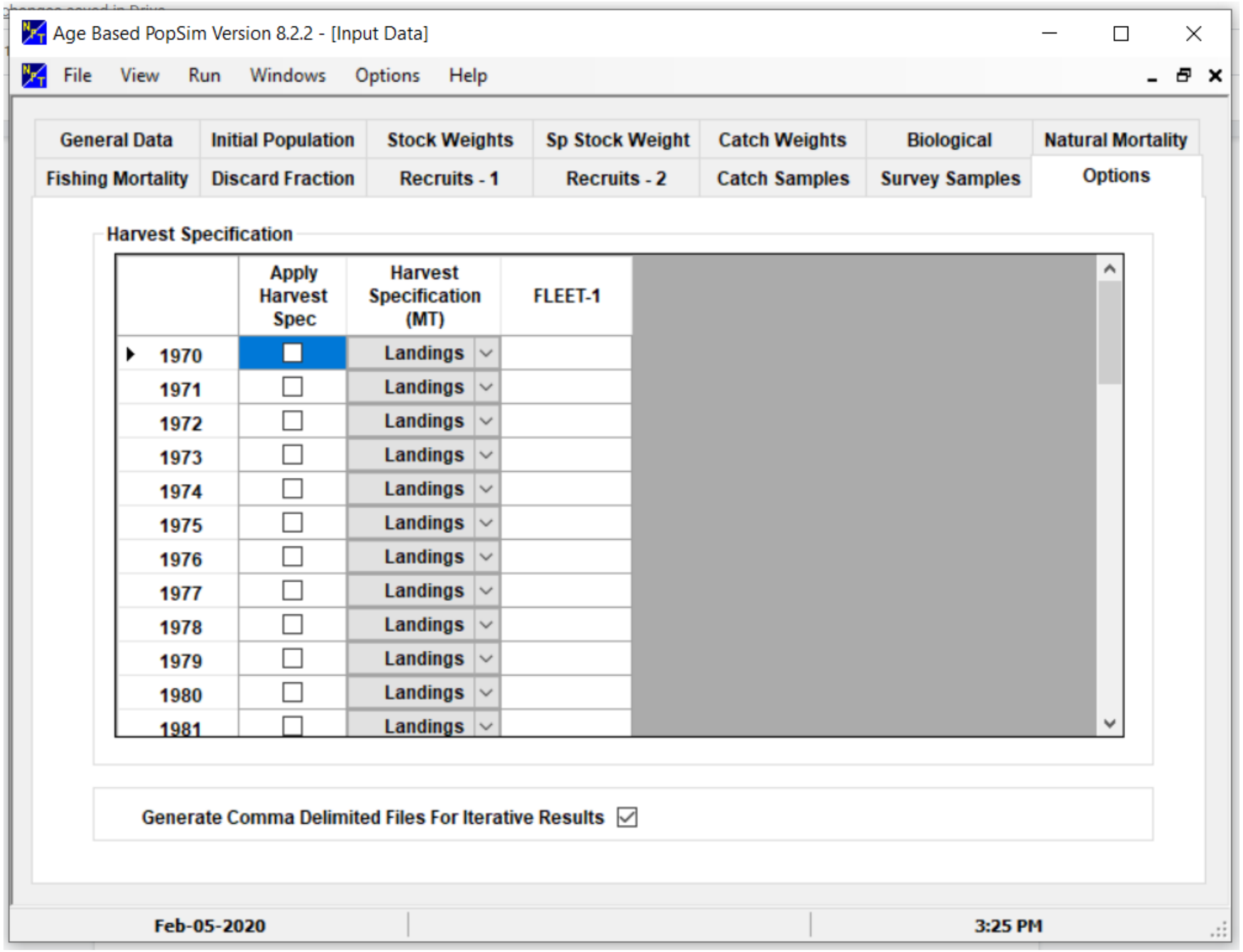
	Coefficient of Variation	Catchability	Number of Samples
1970	1.14	1	100
1971	1.14	1	100
1972	1.14	1	100
1973	1.14	1	100
1974	1.14	1	100
1975	1.14	1	100
1976	1.14	1	100
1977	1.14	1	100
1978	1.14	1	100
1979	1.14	1	100
1980	1.14	1	100
1981	1.14	1	100
1982	1.14	1	100
1983	1.14	1	100
1984	1.14	1	100
1985	1.14	1	100
1986	1.14	1	100
1987	1.14	1	100

Survey Selectivity

	Age 1	Age 2	Age 3
1970	1	0	
1971	1	0	
1972	1	0	
1973	1	0	
1974	1	0	
1975	1	0	
1976	1	0	
1977	1	0	
1978	1	0	
1979	1	0	
1980	1	0	
1981	1	0	
1982	1	0	
1983	1	0	
1984	1	0	
1985	1	0	
1986	1	0	
1987	1	0	

1) The left hand side for survey samples is like the previous screen for catch samples. Survey pseudo-data will be catch rates proportional to stock size with lognormal variability determined by the CV for each year and survey.
2) Survey age composition data will be drawn from multinomial distributions with the true proportions as parameters and the effective sample sizes on the screen.
3) Thus, total numbers at age will have multinomial errors which are correlated for ages in the same year and survey plus lognormal errors that are independent among years and surveys but correlated for age groups in the same year and survey. The lognormal errors for catch-at-age will disappear if the catch-at-age pseudo-data are converted to proportions later by the user for use in modeling (e.g. for the Stock Synthesis model).
4) Catchability affects the magnitude and may change every year.
5) The right hand side gives the selectivity at age in each year for the survey.
6) Each survey requires the same types of information.

OPTIONS INPUT SCREEN



Harvest Specification

	Apply Harvest Spec	Harvest Specification (MT)	FLEET-1
▸ 1970	□	Landings	
1971	□	Landings	
1972	□	Landings	
1973	□	Landings	
1974	□	Landings	
1975	□	Landings	
1976	□	Landings	
1977	□	Landings	
1978	□	Landings	
1979	□	Landings	
1980	□	Landings	
1981	□	Landings	

Generate Comma Delimited Files For Iterative Results ☑

Feb-05-2020 3:25 PM

1) Check the "Apply Harvest Spec" box if you want to specify mortality in terms of a landings or catch quota in the given year. Otherwise, the program will use the fully recruited fishing mortality entered earlier. Selectivity parameters specified on the previous screen are used in either case.
2) You can use quotas for some years and not others.
3) Choose "Landings" or "Total catch" under "Harvest Specification (MT)".
4) Enter the landings or catch quota for each fleet under the fleets name.
5) ***IMPORTANT:*** **CHECK THE "Generate Comma Delimited Files for Iterative Results" to automatically make catch and survey sample files used by R to populate the pseudo-database.**

Appendix 4. This is a description of the information in a POPSIM-A output RDAT file that contains true values and most of the input data from a population simulation (but no pseudo-data). Use the “Export POPSIM data to R” option under the File menu after the POPSIM run is completed to generate this output. The file is named automatically using the left side of the main input file name (e.g. “TEST 0AGE”) and the suffix “.RDAT”. In R, use the dget() command (see Appendix 5) to retrieve the R list shown below. The material below includes extra annotation (starting with **# BOLD AND IN ALL CAPS)** provided by L. Jacobson. The data in the list are stored as scalars, vectors, matrices or arrays and dataframes with descriptive row and column names (often year, fleet or survey) that can be accessed directly (e.g. trueDat$catchweight is an array with dimensions age, year and fleets). This example is from a non-chub mackerel example but the annotations in **bold-face** and structure are similar for chub mackerel. Ellipses (“. . . .“) are for additional data not shown. The name of each element in the list is given just before annotations,

```
> #load the R object with true values and simulation details
> trueName=paste(baseNam, ".RDAT", sep = "")
> paste(trueName)
[1] "C:/Users/LarryJ/OneDrive/Documents/Consulting/ChubMackerel/Scenarios/Fluke/FlukeAge/TEST
0AGE.RDAT"
> trueDat = dget(trueName)
> print(trueDat)
$genparms #GENPARMS ARE ENTERED ON THE POPSIM "GENERAL DATA" INPUT SCREEN
$genparms$minyear #FIRST YEAR IN MODE AND DATA
[1] 1982
$genparms$maxyear #LAST YEAR IN CATCH DATA, SURVEYS CAN BE CALCULATED THROUGH MAXYR+1
[1] 2005
$genparms$nyears #N YEARS IN CATCH DATA
[1] 24
$genparms$nages #NUMBER OF AGE GROUPS IN SIMULATION (>= PLUS-GROUP AGE).  AGES IN THIS EXAMPLE
WILL BE CODED 1 TO 15.
[1] 15
$genparms$plusage #LAST AGE IN PSEUDO-DATA OUTPUT (<= NAGES)
[1] 8
$genparms$nfleets #NUMBER OF FISHING FLEETS
[1] 2
$genparms$nsurveys #NUMBER OF SURVEYS OR CPUE TIME SERIES, SEE NOTES RE DEFINITION OF A SURVEY
[1] 2
$genparms$niter #NUMBER OF PSEUDO-DATA SETS TO GENERATE
```

```
[1] 1000
$genparms$randomseed #INITIAL RANDOM NUMBER SEED
[1] 776953
$genparms$discards #IF TRUE, CATCH PSEUDO-DATA DO NOT INCLUDE DISCARDS (OBSERVED CATCH < TRUE
CATCH DEPENDING ON DISCARD FRACTIONS)
[1] TRUE
$genparms$modelflag
[1] 0 #IDENTIFIES ASSESSMENT MODEL FIT TO PSEUDO-DATA, 0 MEANS NO ASSESSMENT MODEL
$stockweight #TRUE MEAN BODY WEIGHTS AT AGE FOR POPULATION (KG HERE, USED IN R TO CALCULATE
SURVEY WEIGHT), "STOCK WEIGHTS" INPUT SCREEN
year
age 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994
1 0.0299 0.0299 0.0299 0.0299 0.0299 0.0299 0.0299 0.0299 0.0299 0.0299 0.0299 0.0299 0.0299
2 0.1619 0.1582 0.1521 0.1566 0.1560 0.1548 0.1576 0.1525 0.1533 0.1546 0.1549 0.1557 0.1580
....
$spstockweight #TRUE MEAN BODY WEIGHTS FOR SPAWNERS
year
age 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994
1 0.0978 0.0919 0.0963 0.0957 0.0946 0.0972 0.0923 0.0931 0.0943 0.0947 0.0954 0.0976 0.0959
2 0.2363 0.2195 0.2227 0.2265 0.2232 0.2279 0.2197 0.2161 0.2197 0.2218 0.2240 0.2298 0.2285
....
$catchweight #MEAN BODY WEIGHTS FOR EACH FLEET, CATCH WEIGHTS SCREEN
, , fleet = 1 #FIRST FLEET
year
age 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994
1 0.1053 0.1025 0.1046 0.1043 0.1037 0.1050 0.1026 0.1030 0.1036 0.1038 0.1041 0.1052 0.1044
2 0.2421 0.2319 0.2289 0.2341 0.2320 0.2335 0.2316 0.2259 0.2283 0.2302 0.2316 0.2352 0.2361
....
$maturity #FRACTION MATURE EACH AGE AT TIME OF SPAWNING IN EACH YEAR FOR SPAWNING BIOMASS,
"BIOLOGICAL" INPUT SCREEN
year
age 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994
1 0.0191 0.0191 0.0191 0.0191 0.0191 0.0191 0.0191 0.0191 0.0191 0.0191 0.0191 0.0191 0.0191
2 0.3188 0.3188 0.3188 0.3188 0.3188 0.3188 0.3188 0.3188 0.3188 0.3188 0.3188 0.3188 0.3188
....
$fracpriorspawn #FRACTION OF NATURAL (1ST LINES) AND FISHING (2ND) LINE MORTALITY BEFORE
SPAWNING, "BIOLOGICAL" INPUT SCREEN
year
1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998
natmort 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83
```

fmort 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83 0.83
....
$natmort #INSTANTANEOUS NATURAL MORTALITY RATES BY AGE AND YEAR, "NATURAL MORTALITY" INPUT SCREEN
year
age 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999
1 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
....
$fselec #FISHERY SELECTIVITY BY FLEET, YEAR AND AGE (MAX VALUE EACH FLEET AND YEAR SHOULD BE 1, "FISHING MORTALITY" INPUT SCREEN
, , fleet = 1 **#FIRST FLEET**
year
age 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994
1 0.0754 0.0735 0.0749 0.0747 0.0744 0.0752 0.0737 0.0739 0.0743 0.0744 0.0746 0.0753 0.0748
2 0.2957 0.2781 0.2786 0.2836 0.2805 0.2842 0.2781 0.2729 0.2764 0.2788 0.2808 0.2863 0.2859
....
$discfrac #FRACTION DISCARDED BY FLEET, YEAR AND AGE, "DISCARD FRACTION" INPUT SCREEN
, , fleet = 1 **#FIRST FLEET**
year
age 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996
1 0.296 0.296 0.296 0.296 0.296 0.296 0.296 0.296 0.296 0.296 0.296 0.296 0.296 0.296 0.296
2 0.292 0.292 0.292 0.292 0.292 0.292 0.292 0.292 0.292 0.292 0.292 0.292 0.292 0.292 0.292
....
$rec_model #METHOD FOR SIMULATING RECRUITMENT, VPA ESTIMATES FOR MACKEREL, "RECRUIT-1" INPUT SCREEN
[1] "user_vector"
$rec_scale #MULTIPLIED TIMES RECRUITMENT MODEL TO MATCH POPULATION ABUNDANCE UNITS , "RECRUIT-1" INPUT SCREEN
[1] 1
$rec_ssb_scale #MULTIPLIED TIMES SPAWNING BIOMASS PRIOR TO CALCULATING RECRUITMENT MODEL, "RECRUIT-1" INPUT SCREEN
[1] 1
$rec_cv #CV FOR LOGNORMAL VARIANCE IN RECRUITMENT FROM ALL RECRUITMENT MODELS, "RECRUIT-1" INPUT SCREEN
[1] 0.4
$rec_user #RECRUITMENT ESTIMATES FOR "USER-VECTOR" MODEL, "RECRUIT-2" INPUT SCREEN
1983 1984 1985 1986 1987 1988 1989 1990 1991 1992
78883000 47542000 47670000 52295000 43018000 12833000 26706000 29764000 28185000 31780000
....
$rec_alpha #SPAWNER-RECRUIT PARAMETER, NOT USED FOR MACKEREL, "RECRUIT-1" SCREEN INPUT IF

```
REQUIRED
[1] NA
$rec_beta #SPAWNER-RECRUIT PARAMETER, NOT USED FOR MACKEREL, "RECRUIT-1" INPUT SCREEN IF REQUIRED
[1] NA
$rec_kparm #SHEPHERD SPAWNER-RECRUIT PARAMETER NOT USED FOR MACKEREL, "RECRUIT-1" INPUT SCREEN IF REQUIRED
[1] NA
$rec_nobs1 #NUMBER OF EMPIRICAL RECRUITMENTS IN FIRST SET FOR EMPIRICAL RECRUITMENT MODEL IF USED, NOT USED FOR MACKEREL, "RECRUIT-1" INPUT SCREEN
[1] NA
$rec_obs1 #FIRST SET OF EMPIRICAL RECRUITMENT OBSERVATIONS , REC_NOBS1 VALUES, NOT USED FOR MACKEREL, "RECRUITS-2" INPUT SCREEN
[1] NA
$rec_ssb_cut #SSB CUTOFF PARAMETER FOR TWO-STAGE EMPIRICAL RECRUITMENT MODEL, NOT USED FOR MACKEREL, "RECRUIT-1" INPUT SCREEN IF REQUIRED
[1] NA
$rec_nobs2 #NUMBER OF EMPIRICAL RECRUITMENT ESTIMATES IN SECOND SET FOR EMPIRICAL RECRUITMENT MODEL, NOT USED FOR MACKEREL, "RECRUIT-1" INPUT SCREEN
[1] NA
$rec_obs2 #SECOND SET OF EMPIRICAL RECRUITMENT OBSERVATIONS, REC_NOBS2 VALUES, NOT USED FOR MACKEREL, "RECRUITS-2" INPUT SCREEN
[1] NA
$catch_sample_size #EFFECTIVE SAMPLE SIZE FOR MULTINOMIAL AGE SAMPLES BY FEET AND YEAR, "CATCH SAMPLES" INPUT SCREEN
fleet #FLEETS ARE IN COLUMNS, ONE LINE FOR EACH YEAR
year 1 #COLUMN FOR YEAR AND FLEET 1
1982 100
1983 100
....
$catch_sample_cv #CV FOR LOGNORMAL ERRORS IN TOTAL CATCH WEIGHT BY YEAR AND FLEET, "CATCH SAMPLES" INPUT SCREEN
fleet #FLEETS ARE IN COLUMNS, ONE LINE FOR EACH YEAR
year 1 #COLUMN FOR YEAR AND FLEET 1
1982 0.1 0.2
1983 0.1 0.2
....
$survey_sample_size #EFFECTIVE SAMPLE SIZE FOR MULTINOMIAL AGE SAMPLES BY SURVEY AND YEAR, "SURVEY SAMPLES" INPUT SCREEN
survey #SURVEYS ARE IN COLUMNS, ONE LINE FOR EACH YEAR
year 1 #COLUMN FOR YEAR AND SURVEY 1
1982 100
```

```
1983 100
....
1986 0 #NO SAMPLING SO NO SURVEY
....
$survey_sample_cv #CV FOR LOGNORMAL ERRORS IN TOTAL SURVEY CATCH NUMBERS BY YEAR AND SURVEY,
"SURVEY SAMPLES" INPUT SCREEN
survey #SURVEYS ARE IN COLUMNS, ONE LINE FOR EACH YEAR
year 1 #COLUMN FOR YEAR AND SURVEY 1
1982 0.1
1983 0.1
....
$survey_catchability #CATCHABILITY EACH SURVEY BY YEAR, "SURVEY SAMPLES" INPUT SCREEN
survey #SURVEYS ARE IN COLUMNS, ONE LINE FOR EACH YEAR
year 1 #COLUMN FOR YEAR AND SURVEY 1
1982 0.001
1983 0.001
....
$survey_selec #SELECTIVITY AT AGE BY YEAR FOR EACH SURVEY, "SURVEY SAMPLES" INPUT SCREEN
, , survey = 1
year
age 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
....
$harvest_spec #SPECIFY IF A QUOTA IS IMPOSED IN EACH YEAR AND WHETHER SPECIFIED IN TERMS OF
LANDINGS OR CATCH
Flag Type
1982 "FALSE" "Landings"
1983 "FALSE" "Landings"
....
$harvest_value #QUOTA FOR EACH FLEET BY YEAR
fleet #FLEETS ARE IN COLUMNS, ONE LINE FOR EACH YEAR
year 1 #COLUMN FOR YEAR AND FLEET 1
1982 0 0
1983 0 0
$stocknumbers #TRUE STOCK SIZE BY AGE AND YEAR IN SIMULATION, "STOCK NUMBERS" OUTPUT SCREEN
year
age 1982 1983 1984 1985 1986 1987 1988
1 72766000 8.029408e+07 5.071521e+07 5.159729e+07 3.429769e+07 3.493983e+07 7.847649e+06
```

```
2 42166000 5.912825e+07 6.525782e+07 4.121226e+07 4.192990e+07 2.787243e+07 2.839200e+07
....
$catchnumbers #TRUE TOTAL CATCH FOR EACH AGE AND YEAR, "CATCH NUMBERS" OUTPUT SCREEN
year
age 1982 1983 1984 1985 1986 1987 1988
1 495464.1770 532994.9370 343038.5700 348076.4140 230447.0910 2.372768e+05 5.223422e+04
2 1114080.3100 1470504.6800 1625824.2700 1044932.4300 1051663.8800 7.081772e+05 7.061018e+05
....
$totalstock #TOTAL ABUNDANCE IN EACH YEAR, "TIME SERIES PLOTS" OUTPUT SCREEN
1982 1983 1984 1985 1986 1987 1988 1989 1990
134166400 187555093 199730632 208410760 194883961 180754204 139142355 119942305 108155705
....
$jan1bio #TOTAL BIOMASS AT BEGINNING OF YEAR, "TIME SERIES PLOTS" OUTPUT SCREEN
1982 1983 1984 1985 1986 1987 1988 1989 1990
19890.69 39223.66 61583.41 88638.13 110013.09 119807.84 118681.65 103011.80 86982.16
....
$catchbio #TOTAL CATCH BIOMASS (ALL FLEETS) BY YEAR, "TIME SERIES PLOTS" OUTPUT SCREEN
1982 1983 1984 1985 1986 1987 1988 1989 1990
1398.724 2766.136 4913.466 10554.459 18205.579 25654.014 30568.883 27651.068 32076.659
....
$ssb #SPAWNING STOCK BIOMASS AT TIME OF SPAWNING "TIME SERIES PLOTS" OUTPUT SCREEN
1982 1983 1984 1985 1986 1987 1988 1989 1990 1991
....
$recruits #RECRUIT ABUNDANCE (NUMBER AGE 1 FISH BEGINNING OF YEAR), "TIME SERIES PLOTS" OUTPUT
SCREEN
1982 1983 1984 1985 1986 1987 1988 1989 1990 1991
....
$landings #LANDED WEIGHT IN EACH YEAR ("TIME SERIES PLOTS" OUTPUT SCREEN
1982 1983 1984 1985 1986 1987 1988 1989 1990
1082.496 2116.316 3790.547 8625.967 15427.578 22232.669 26882.461 24448.926 28748.315
....
$discards #DISCARDED WEIGHT IN EACH YEAR, "TIME SERIES PLOTS" OUTPUT SCREEN
1982 1983 1984 1985 1986 1987 1988 1989 1990
316.2278 649.8203 1122.9187 1928.4925 2778.0010 3421.3451 3686.4221 3202.1422 3328.3443
....
$Fmult #FULLY RECRUITED FISHING MORTALITY RATE (AT AGE WITH MAXIMUM F, FLEETS COMBINED, "TIME
SERIES PLOTS" OUTPUT SCREEN
1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000
0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
```

```
....
$fleetland #LANDINGS BY FLEET
fleet #FLEETS ARE IN COLUMNS, ONE LINE FOR EACH YEAR
year 1 #COLUMN FOR YEAR AND FLEET 1
1982 752.4402
1983 1575.8757
....
$F_Fleet #FULLY RECRUITED FISHING MORTALITY BY FLEET, "FISHING MORTALITY BY FLEET" OUTPUT SCREEN
fleet #FLEETS ARE IN COLUMNS, ONE LINE FOR EACH YEAR
year 1 #COLUMN FOR YEAR AND FLEET 1
1982 0.1
1983 0.1
....

>
```

Appendix 6. R code in PseudoData-10.R.

```
################################
#PseudoData-10.R - 3 May 2020 - Larry Jacobson, final code distributed
# to NPFC.
##########
# This R command file reads POPSIM-A output and generates a database
# containing data that might be used to test stock assessment models.
# It was set up to work with chub mackerel originally but is generic
# and could be used for any other species.  It was also tested with
# datafiles for "fluke" (summer flounder) distributed with POPSIM-A.
# Note that some code for the fluke example is included below.
#
# The user must specify the plus group age for pseudo-data (set variable
# plusAge to a value less than the maximum age in POPSIM-A or leave
# null).  Also, specify an exponent for each survey in POPSIM-A
# that can be used to make the survey nonlinear like CPUE (set variable
# expon to 1 for linear or some other value for nonlinear).  The directories
# for input data (variable adir) and R code (file name codeDir) must be specified.
# The basename for POPSIM-A input and output files must be specified
# (variable basenanme) and is used to name the output files from this
# R program.  The database is automatically written to an ASCI csv
# file and to an RDAT file.
#
# In addition to the database, some graphics are generated if the variable
# diagnosis is TRUE or left NULL.
#
# Check the headers for all of the functions in PopSimFunctions.R for other
# variables which have automatic defaults that can be set by the user.
# There are many control variables with default values whichm do not have
# to be specified.
#
#                               Larry Jacobson
#                               24 February 2010
# for chub mackerel.
require(lattice)
require(reshape)
require(memisc)
require(tictoc)

rm(list = ls())
graphics.off()

#calculate elapsed time
tic()

#########
#impose a plus group on pseudo-data written to database which must
# be <= to the plus-group used to make pseudo-data in POPSIM-A.
# (use NULL for no change)
```

```
plusAge = NULL
#plusAge=15
#plusAge=8

###########
#code directory and file names
codeDir = paste("C:/Users/LarryJ/OneDrive/Documents/Consulting/ChubMackerel/Rpopsim/",
                sep = "")
codeFile = "PopSimFunctions-20.R"
#load functions
source(paste(codeDir, codeFile, sep = ""))

###########
#Chub mackerel base file name for files from PopSim run and for output from this
program
adir = "C:/Users/LarryJ/OneDrive/Documents/Consulting/ChubMackerel/Scenarios/Chub-1/"
afnm = "CHUB-1"

###########
#Fluke case distributed with POPSIM (has multiple = 2 fleets and 2 survetys), make
sure to set
# expon and plusAge to NULL or some reasonable value
#
adir="C:/Users/LarryJ/OneDrive/Documents/Consulting/ChubMackerel/Scenarios/Fluke/Fluke
Age/"
# afnm="test0AGE"


#exponents used to make some surveys into cpue (one value for each survey)
# can also set entire vector to NULL for all linear
expon = c(1.6, 1.6, 1, 1) #for chub mackerel
# expon=NULL #use for fluke

#put the directory and base file name together
baseNam = paste(adir, afnm, sep = "")

###########
#
#calculations start here
#
###########
#
#load the R object with true values and simulation details
#use dget() because object is of dput type
trueName = paste(baseNam, ".RDAT", sep = "")
paste(trueName)
trueDat = dget(trueName)
#View(trueDat)

#########
```

```r
#get catch, landings, discards at age, totals, by fleet, etc.
db = getCatchPseudoData(baseNam = baseNam,
                        trueDat = trueDat,
                        plusAge = plusAge)
lookDb(db)

#########
#get survey data at age, total, in number, weight, etc.
db = getSurveyPseudoData(
  baseNam = baseNam,
  trueDat = trueDat,
  assessData = db,
  #append to existing file
  plusAge = plusAge,
  # plusAge,
  expon = expon
)

#check results
lookDb(db)

#descriptive name
oName = paste(afnm, "-PseudoData", sep = "")

#save pseudo-data in R file (retrieve with dget)
dput(db, file = paste(adir, oName, ".RDAT", sep = ""))

#comma-separated value database version if needed
write.csv(
  db,
  file = paste(adir, oName, ".csv", sep = ""),
  row.names = FALSE,
  quote = FALSE
)
```

```
#one dataframe with most of the truth for my use
ny=trueDat$genparms$nyears
truth = as.data.frame(with(
  trueDat,
  cbind(
    yr = with(trueDat$genparms, minyear:(maxyear+1)),
    stockN = totalstock,
    recr = recruits,
    ssb = c(ssb,NA),
    F = c(Fmult,NA),
    catchbio = c(catchbio,NA),
    landings = c(landings,NA)
  )
))
print(truth, row.names = FALSE)


toc()
```

Appendix 7. Structure of pseudo-database for chub mackerel from R software. Any variable other than nam and sourc may be missing (NA in R).

Variable name	Type	Required?	Notes
nam	Character	Yes	Descriptive name or timestamp
fleet	Numeric	NA if missing	Fleet or survey ID number (fleets and surveys with the same ID do not conflict, e.g. you can have fleets with IDs 1-4 and surveys with IDs 1-8).
sourc	Character	NA if missing	Sourc: catchSerialNumber, fisherySamples, surveySamples, surveySerialNumber
iter	Numeric	NA if missing	POPSIM-A iteration
year	Numeric	NA if missing	Year
ageGrp	Numeric	NA if missing	Age
value	Numeric	NA if missing	Any value

Appendix 8. Identifiers for “sourc” (source) and “nam” (name) fields used to identify pseudo-data and associated information for testing chub mackerel stock assessment models. For example, “landNumTot” is fishery pseudo-data for landings (catch minus discard, with error) aggregated over all fleets and ages. All pseudo-data are separated by year and simulation iteration.

Name (nam)	**Source (sourc)**	**Description**	**Separated by**	**Aggregated over**	**Units**
catMtTot	Fishery	Catch	Year	Fleet, age	MT
catNumByAgeTot	Fishery	Catch	Year, age	Fleet	Number
catNumTot	Fishery	Catch	Year	Fleet, age	Number
landMtTot	Fishery	Landings	Year	Fleet, age	MT
landNumByAgeTot	Fishery	Landings	Year, age	Fleet	Number
landNumTot	Fishery	Landings	Year	Fleet, age	Number
fishWtAtAgeTrue	Fishery (POPSIM-A input)	Mean weight at age	Year, fleet, age		KG
Timestamp (e.g. "2/24/2020 10:03:50")	Fishery (R)	Timestamp & serial number			
MtTotal	Survey	Mean catch rate	Year, survey	Age	Proportional to MT
NumByAge	Survey	Mean catch rate	Year, survey, age		Proportional to abundance
NumTotal	Survey	Mean catch rate	Year, survey	Age	Proportional to abundance
popWtAtAgeTrue	Survey (POPSIM-A input)	Mean weight at age	Year, age	Same values used for all surveys	Mean weight at age
Timestamp (e.g. "2/24/2020 10:05:12")	Survey (R)	Timestamp & serial number	n/a	n/a	Timestamp & serial number

Appendix 9. R code that demonstrates how a “silly” stock assessment model (annual catch curves) programmed entirely in R can be tested using demonstration pseudo-data for chub mackerel. The pseudo-data are survey catch-at-age from one of the surveys that tracks spawning biomass. The code is crude because it was developed quickly for use at a meeting. A real world example would probably be more complex and have to handle additional model controls, potential lack of convergence and other common problems and situations. The R dataframe db is a database containing demonstration pseudo-data for chub mackerel.

```
#reload pseudo-data
load(file = paste(adir, oName, ".RDAT", sep = ""))
#use one of the spawning biomass surveys (survey 3 or 4)
dbTest = subset(db, nam == "NumByAge" & fleet == 3)
lookDb(dbTest)
dim(dbTest)

#choose age range for catch curves (start at age 4 = full maturity)
startAge = 4
#need plus group age too
plusAge = max(dbTest$ageGrp)

##########
#now test the silly model to annual data by fitting in R
##########
#catch curve for each iteration and year
dbTest = subset(dbTest, ageGrp >= startAge)
dbTest = subset(dbTest, ageGrp < plusAge)

#drop zeroes then take logs
dbTest = subset(dbTest, value > 0)
dbTest$logvalue = log(dbTest$value)

#use loops (faster to program for me)
myIts = sort(unique(dbTest$iter))
rez = data.frame()
#fit model to data for one iteration at a time
for (its in myIts) {
  useDat = subset(dbTest, iter == its)
  myYrs = sort(unique(useDat$year))
  for (yrs in myYrs) {
    adat = subset(useDat, year == yrs)
    if (nrow(adat) > 2) {
      alm = lm(log(value) ~ ageGrp, data = adat)
      Z = -alm$coefficients[2]
      rez = rbind(rez, c(
        iter = its,
        year = yrs,
        Z = Z
      ))
    }
  }
}
names(rez) = c("iter", "year", "Z")
```

Appendix 10. R code that runs an AD-Model Builder program in DOS to test a “silly” stock assessment model for chub mackerel. The dataframe dbTest is a subset of the demonstration pseudo-database for chub mackerel which is also used in Appendix 9. A real world example would probably be more complex and have to handle additional model controls, potential lack of convergence and other common problems and situations.

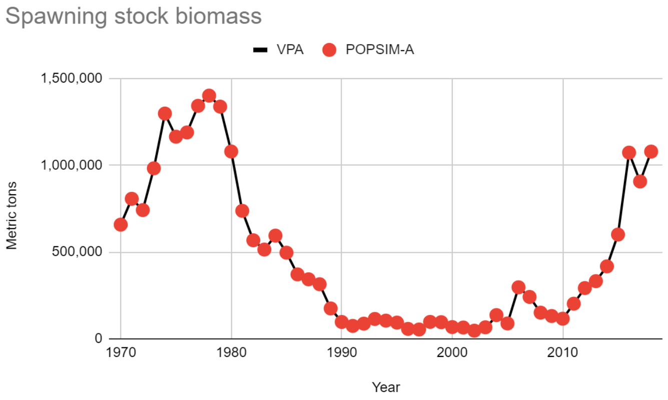
```
##########
#Now fit same model in ADMB
#Use linear regression code in
#     http://www.admb-project.org/courses/previous/august-2009/adintro.pdf
##########
library(fs)

#so system commands will run in same directory as rest of demo
#note=must put admb executable in adir
setwd(adir)

#sort data
dbTest = subset(dbTest, value > 0)
dbTest = dbTest[with(dbTest, order(iter, year, ageGrp)), ]
nObs = nrow(dbTest)

#bookkeeping
myIts = sort(unique(dbTest$iter))
nIter = length(myIts)
myYrs = sort(unique(dbTest$year))
nYear = length(myYrs)

#create data file for ADMB using data used for R test
adNameStart = "silly"
adName = paste(adir, adNameStart, ".dat", sep = "")

#a place to store catch curve results
allRez = c()

for (it in myIts) {
  cat(paste("Iteration", it, "\n"))

  useDat = subset(dbTest, iter == it)
  nObs = nrow(useDat)

  #delete dat admb file if it exists
  if (file.exists(adName)) {
    file_delete(adName)
  }
  fileConn = file(adName)
```

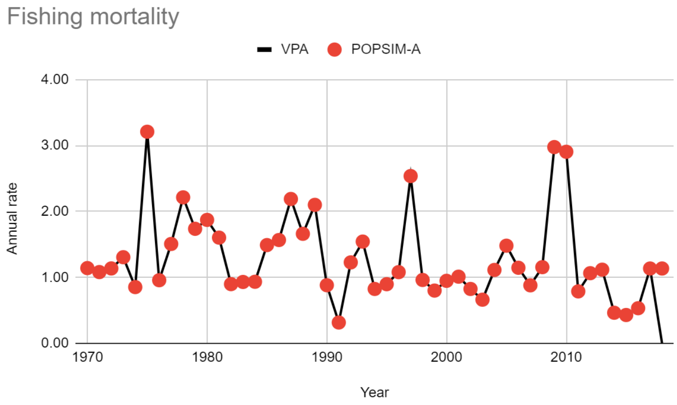
```
    #write data
    writeLines(
      c(
        "#Iter",
        paste(it),
        "#minYear",
        paste(min(myYrs)),
        "#maxYear",
        paste(max(myYrs)),
        "#nObs",
        paste(nObs),
        "#usedat$ageGrp",
        paste(useDat$ageGrp),
        "#usedat$year",
        paste(useDat$year),
        "#useDat$logvalue",
        useDat$logvalue,
        "#done"
      ),
      fileConn
    )
    close(fileConn)

    system("silly -nox", invisible = FALSE)
    oneRez = -scan("silly.rep")
    allRez = c(allRez, oneRez)
  }

  admbRez = as.data.frame(cbind(
    iter = rep(myIts, each = nYear),
    year = rep(myYrs, nIter),
    Z = allRez
  ))

summary(admbRez$Z)
```

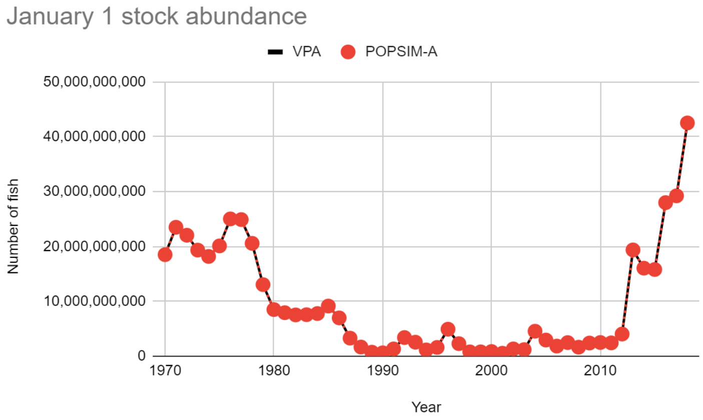
Appendix 11. AD-Model Builder code used for testing a “silly” stock assessment model (annual catch curves) for chub mackerel.

```
//Silly demo assessment model for chub mackerel (annual catch curves)
//Code originally from
http://www.admb-project.org/courses/previous/august-2009/adintro.pdf
DATA_SECTION
 init_int iter  //not used
 init_int minYear
 init_int maxYear
 init_int nObs
 init_vector ageGrp(1,nObs)
 init_ivector year(1,nObs)
 init_vector lncat(1,nObs);
INITIALIZATION_SECTION
 a 26.5;
 b -1.5;
PARAMETER_SECTION
 init_vector a(minYear,maxYear,1)
 init_vector b(minYear,maxYear,2)
 objective_function_value nll
 vector yhat(1,nObs)
PROCEDURE_SECTION
 yhat=a(year)+elem_prod(b(year),ageGrp);
 nll=sum(square(lncat-yhat));
REPORT_SECTION
 report << b << endl;
```

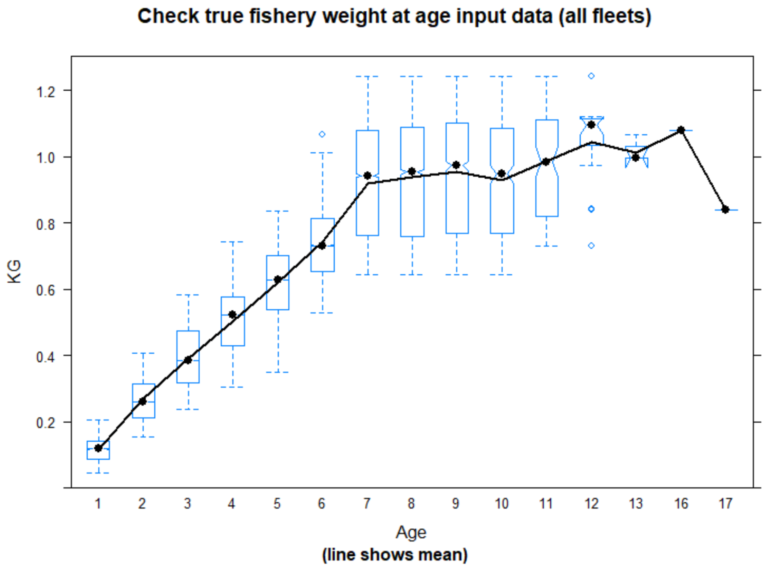
Appendix 12. Stock estimates from a Japanese VPA and simulated true values calculated for a hypothetical chub mackerel population by POPSIM-A.



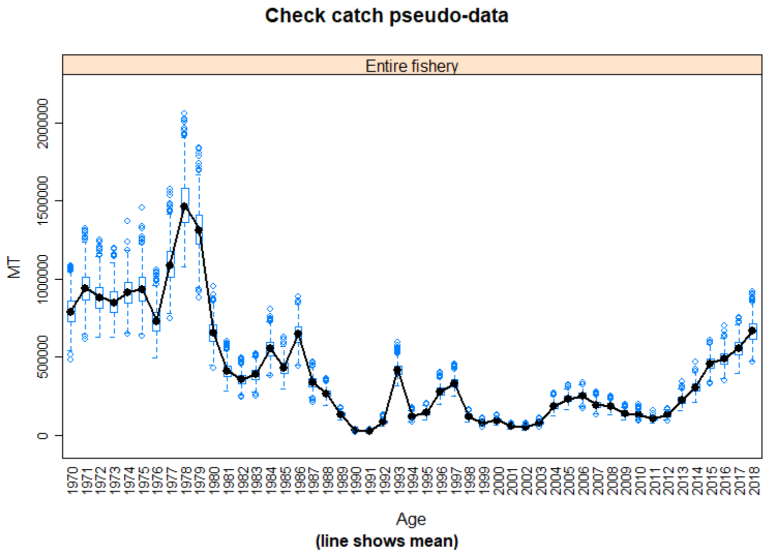




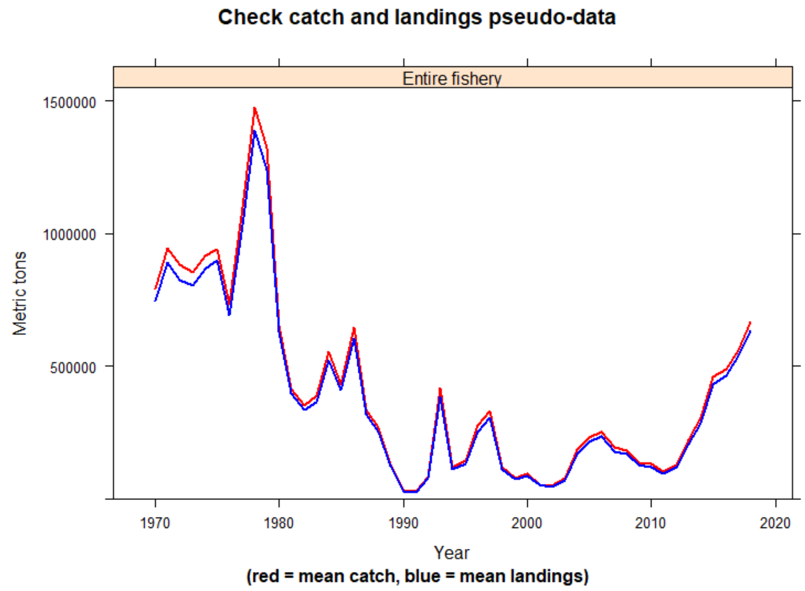
Appendix 12. Graphics made automatically in R for checking chub mackerel pseudo-data from POPSIM-A..



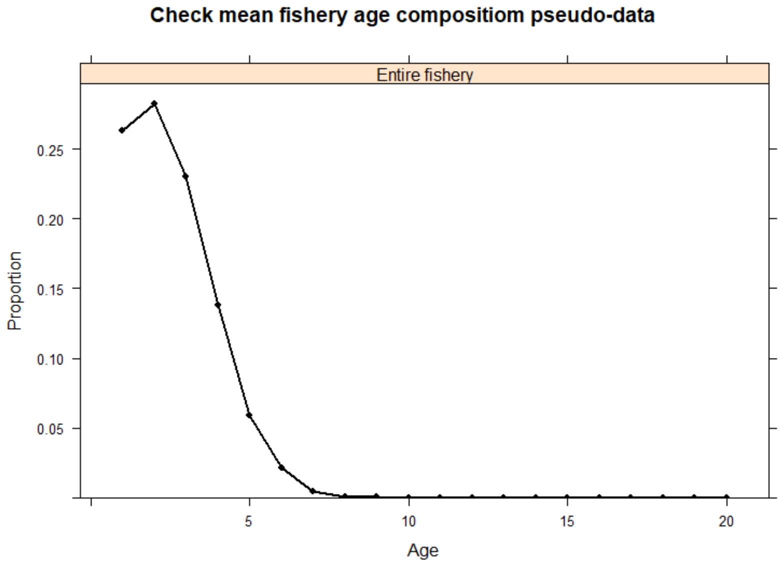
The box plots show distributions of fishery weight at age (all fleets) used as POPSIM-A input data for a hypothetical stock of chub mackerel. The line shows mean values. These are used to calculate catch weight pseudo-data.



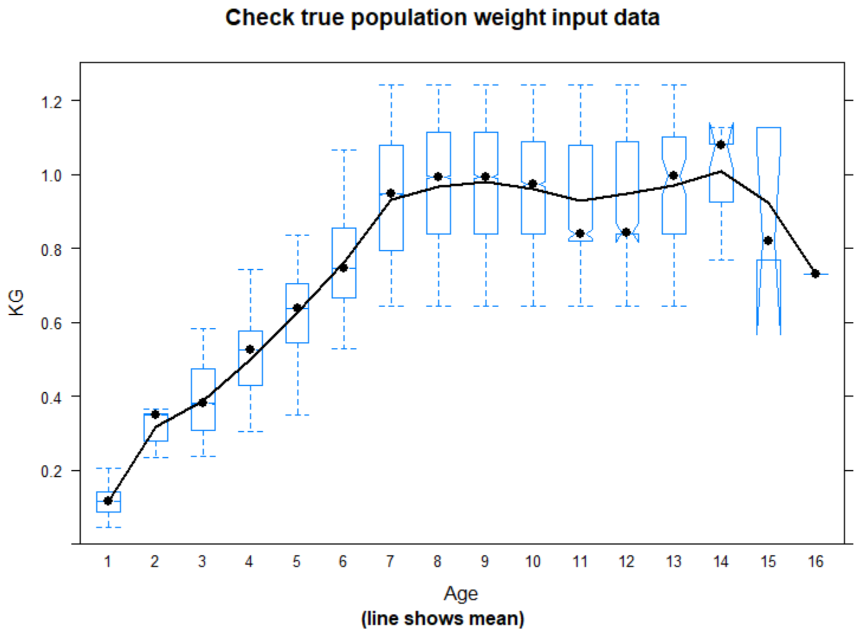
The box plots show the distribution of total catch pseudo-data in each year from POPSIM-A (fleets combined, 1000 iterations). The line shows mean values. Runs with multiple fleets will show a graph for each.



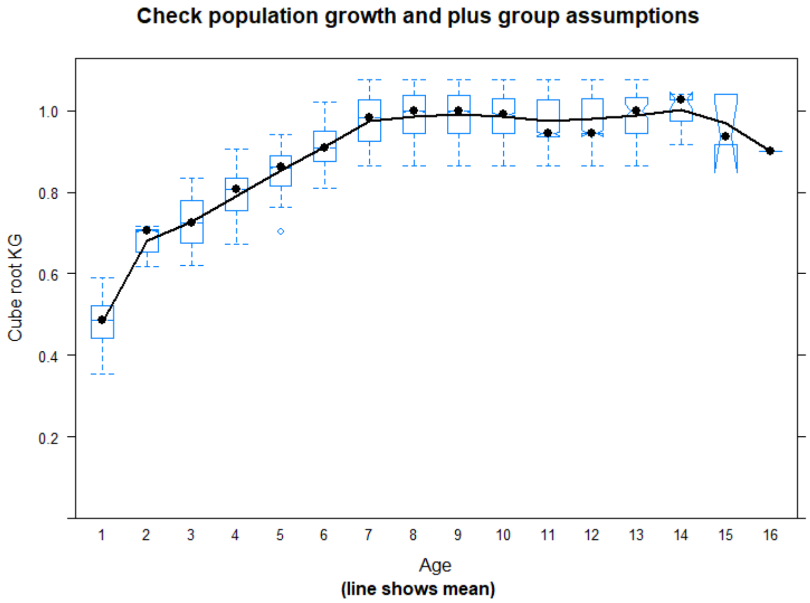
The red line shows mean total catch and the blue line shows mean total landings pseudo-data for a hypothetical chub mackerel stock from POPSIM-A with some discard of small fish (fleets combined, 1000 iterations). Runs with multiple fleets will show a graph for each.



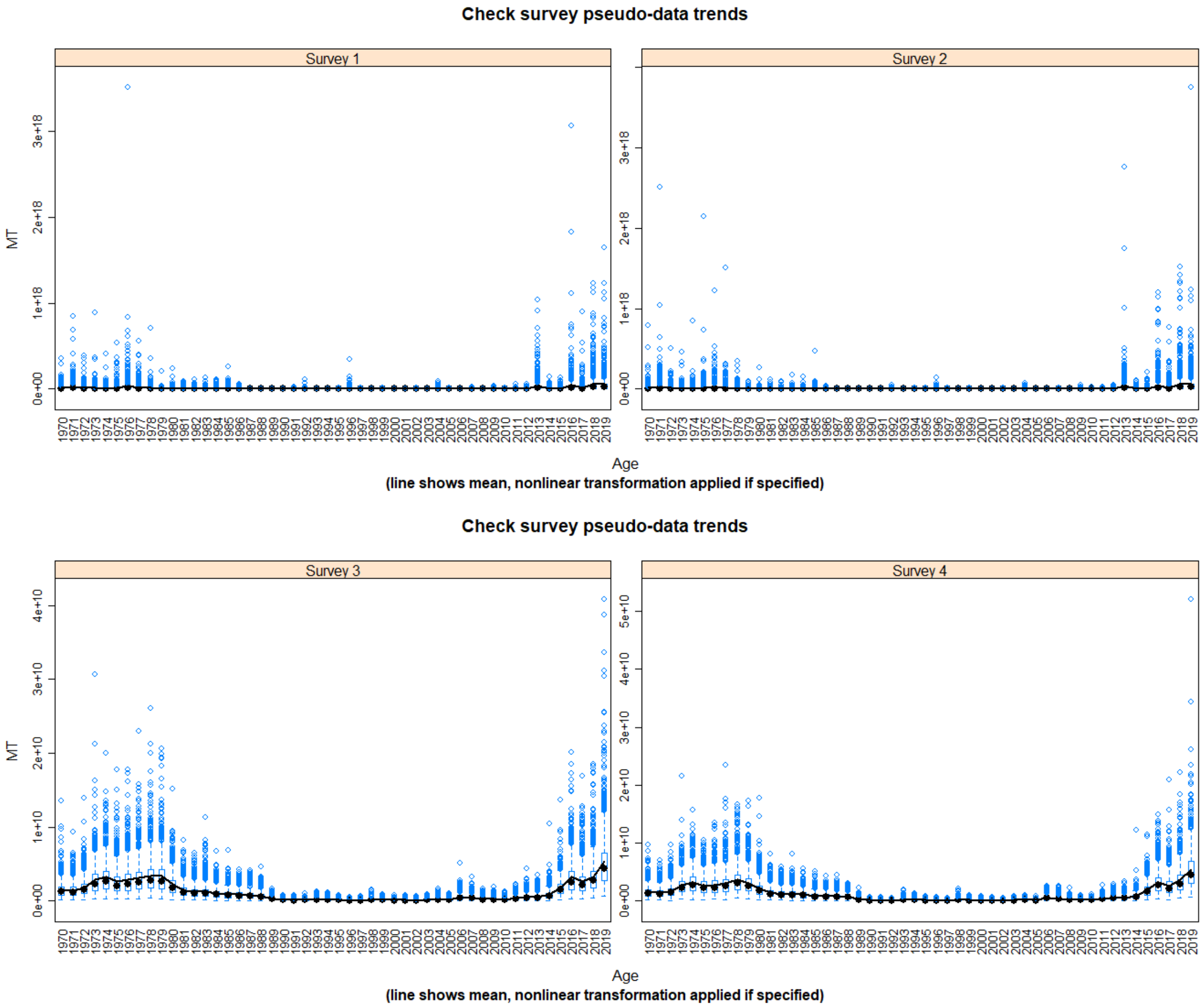
Average commercial age composition proportion pseudo-data. Runs with multiple fleets will show a graph for each.



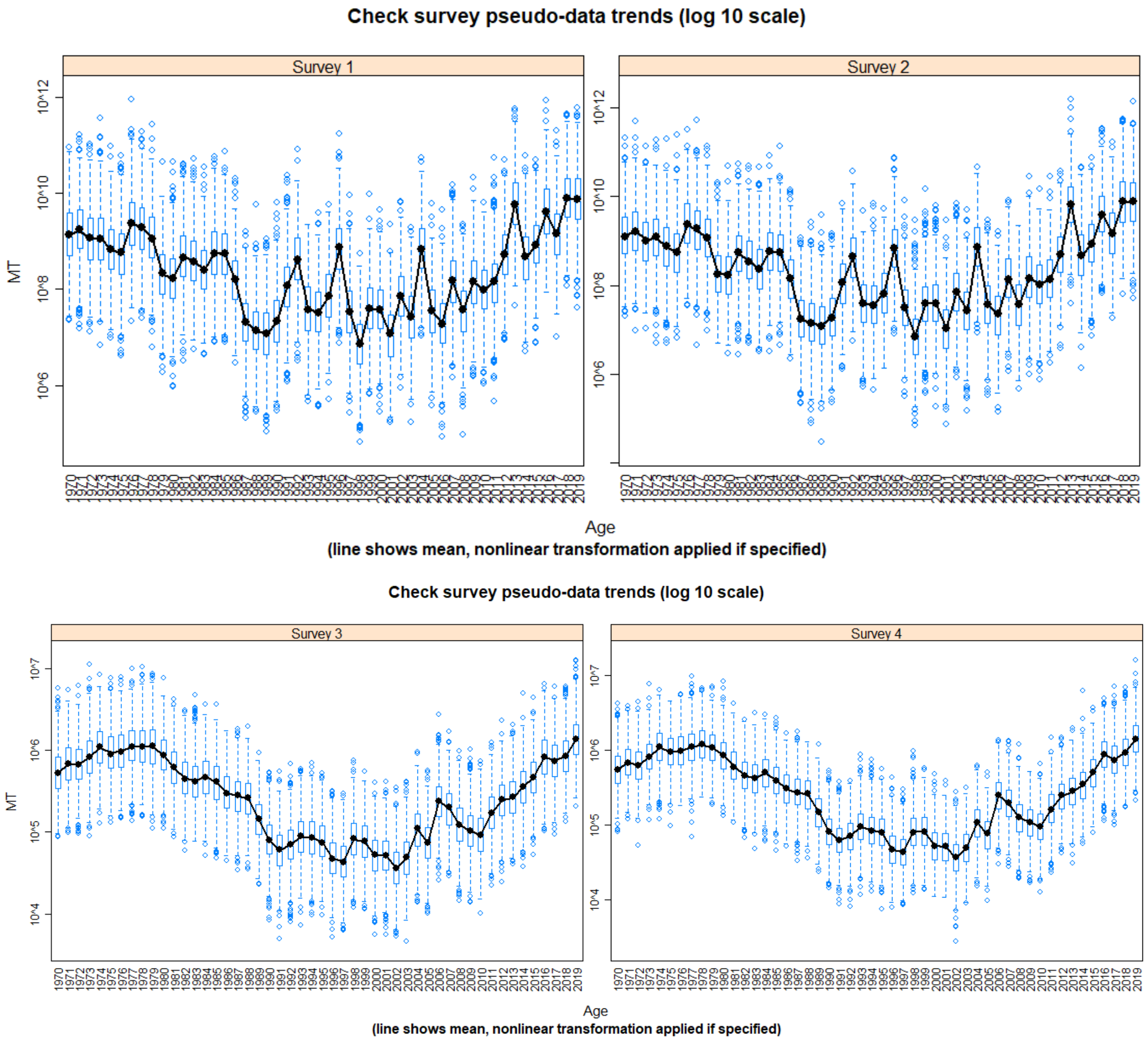
Distributions of population weight at age used as POPSIM-A input data for a hypothetical chub mackerel stock (same data used for all surveys). The line shows mean values. These are used to calculate survey weight pseudo-data.



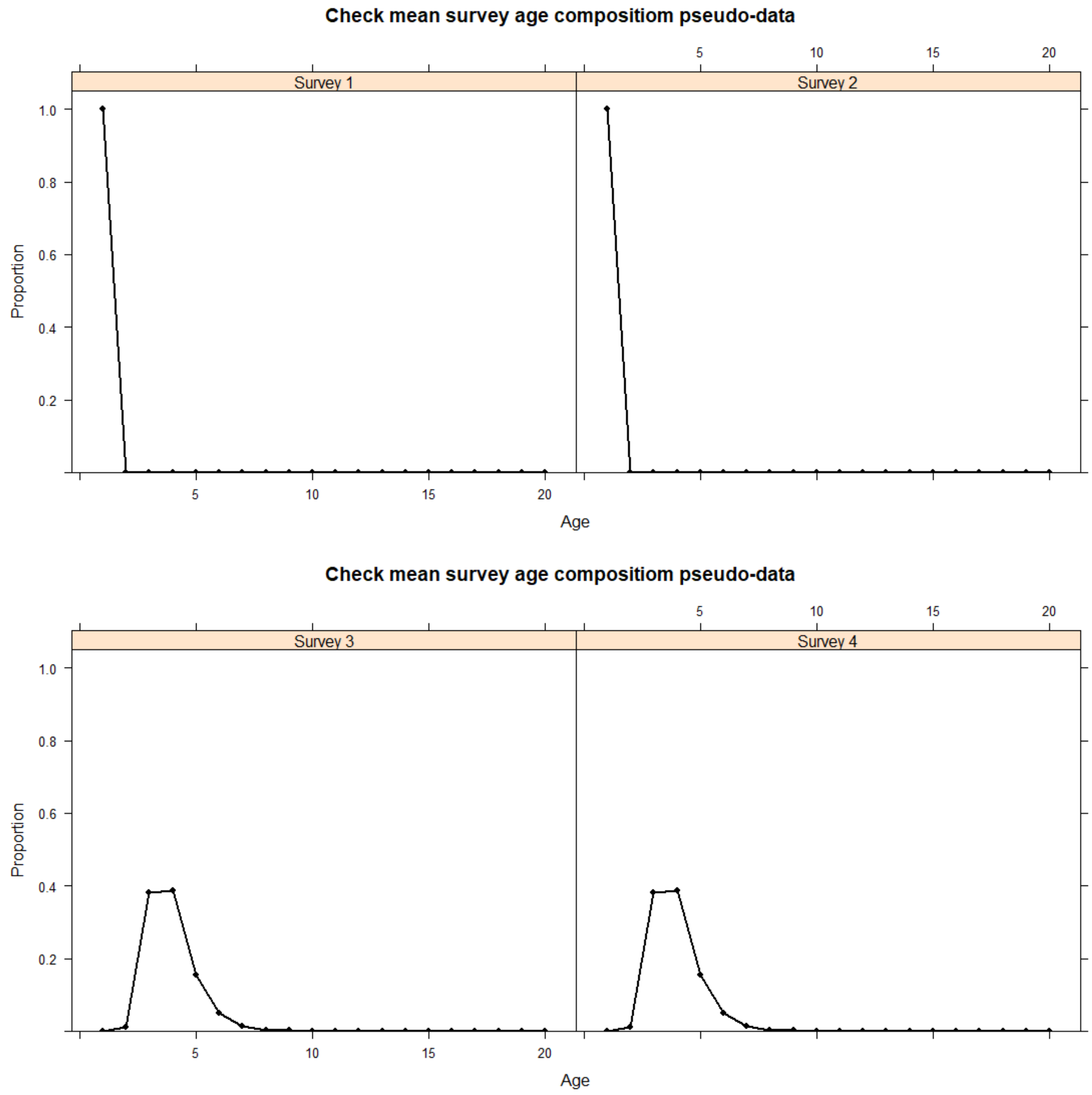
The cube root of mean body weight at age on the y-axis is roughly proportional to body length. This plot may help determine if growth and plus-group assumptions are compatible. In most cases, the plus-group used in calculations should be set to an age where this plot (or a plot of length vs. age) is relatively level.



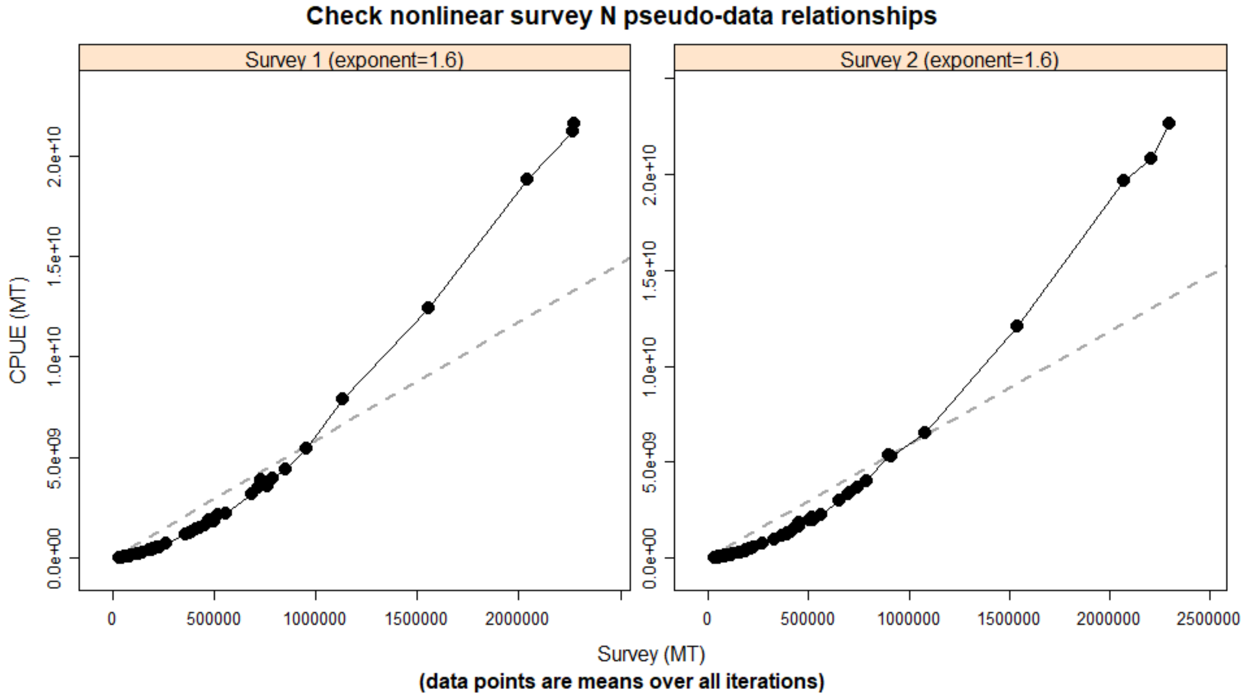
Survey catch weight pseudo-data for a hypothetical chub mackerel stock from POPSIM-A (ages combined, 1000 iterations). Surveys 1 and 2 include nonlinear effects (hyperdepletion). The line shows mean values. Figures are difficult to interpret because of high assumed CVs so log scale values are shown in the next set of plots.



Log 10 survey catch weight pseudo-data for a hypothetical chub mackerel stock from POPSIM-A (ages combined, 1000 iterations).   Surveys 1 and 2 include nonlinear effects (hyperdepletion). The line shows mean values. Note nonlinear y-axis.



Average survey age composition pseudo-data for a hypothetical chub mackerel stock from POPSIM-A (years combined, 1000 iterations). Surveys 1 and 2 are recruit surveys with a single age group.



Total survey catch weight by year for nonlinear CPUE on the y-axis from post-processing in R and as a conventional linear survey on the x-axis from POPSIM-A for a hypothetical stock of chub mackerel. The data points are annual averages (ages combined averaged over 1000 iterations). Bumpiness along the y-axis is due to calculating the averages of exponentiated survey pseudo-data that include simulated measurement errors.

Appendix 11. Files used to make example pseudo-data for chub mackerel with POPSIM-A and R. There are a number of files that contain similar data in different formats (e.g. catch sample information is found in `CHUB-DEMO-1.out, CHUB-DEMO-1.xx2, CHUB-DEMO-1_catch.csv, and CHUB-DEMO-1_catchdata.csv)`. Files actually utilized in this report are marked "Y" under the heading "Used". The "Export PopSim Data to R" command under the "File" menu in the GUI must be run after a POPSIM-A simulation is complete to generate `CHUB-DEMO-1.RDAT` (be careful!). Otherwise an old version of the file may persist or the file will be missing. The "Generate Comma Delimited Files for Iterative Results" on the "Options" input page in the GUI must be checked to generate files with pseudo-data like `CHUB-DEMO-1_catch.csv` and `CHUB-DEMO-1_catchdata.csv` automatically at the end of each run.

File name	Size (kilo bytes)	Used	Notes
CHUB-DEMO-1.DAT	143	Y	Basic POPSIM-A input file for chub mackerel C program made by GUI. Contents correspond to GUI input screens. File name used to name other output files.
CHUB-DEMO-1.out	784	Y	POPSIM-A output file with basic population results (true values) by age and year and statistical summary of pseudo-data. Used to automatic POPSIM-A output screens but not used by R.
CHUB-DEMO-1.RDAT	146	Y	POPSIM-A output with true values and most of the POPSIM-A input data in an R list. See Appendix 4.
CHUB-DEMO-1.xx1	141		POPSIM-A output with true population results for each age and year.
CHUB-DEMO-1.xx2	18,662		POPSIM-A output with catch samples (pseudo-data) by iteration, year, fleet and age
CHUB-DEMO-1.xx3	76,168		POPSIM-A output with survey samples (pseudo-data) by iteration, year, survey and age
CHUB-DEMO-1.xx4	0		POPSIM-A output with iterative results from assessment models (no longer used)
CHUB-DEMO-1_catch.csv	18,829		POPSIM-A output with catch samples (pseudo-data) by iteration, year, fleet and age - wide format
CHUB-DEMO-1_catchdata.csv	33,754	Y	POPSIM-A output with catch samples (pseudo-data) by iteration, year, fleet and age - long database format

CHUB-DEMO-1_landings.csv	937		POPSIM-A output with landings samples (pseudo-data) by iteration, year, fleet and age - wide format
CHUB-DEMO-1_landingsdata.csv	1,563	Y	POPSIM-A output with landings samples (pseudo-data) by iteration, year, fleet and age - long database format
CHUB-DEMO-1_PseudoData.csv	335,028	Y	R pseudo-database output, ascii csv text file format
CHUB-DEMO-1_PseudoData.RDAT	19,247	Y	R pseudo-database output, dataframe format compressed by R save() function
CHUB-DEMO-1_survey.csv	76,836		POPSIM-A output with survey samples (pseudo-data) by iteration, year, fleet and age - wide format
CHUB-DEMO-1_surveydata.csv	137,772	Y	POPSIM-A output with survey samples (pseudo-data) by iteration, year, fleet and age - long database format
PopSimFunctions.R	39	Y	R command file containing functions used to post-process pseudo-data in POPSIM-A output files and generate a database for testing assessment models
PseudoData.R	2	Y	R command file that calls functions in PopSimFunctions.R to post-process data in POPSIM-A output files and generate a database for testing assessment models
Total	720,051		

Appendix JJJ. R code to retrieve simulated true population results and input data from an R list stored by POPSIM-A as an RDATA file. The list and file are created after all iterations are completed when the “Export PopSim data to R” command under the File menu in the GUI is executed. The R list contains more information (e.g. age structured pseudo-data and POPSIM-A inputs) then used here.

```
#load the R object with true values and simulation details
trueName=paste(baseNam, ".RDAT", sep = "")
trueDat = dget(trueName)
View(trueDat)

#one dataframe with most of the truth for my use
ny=trueDat$genparms$nyears
truth = as.data.frame(with(
  trueDat,
  cbind(
    yr = with(trueDat$genparms, minyear:(maxyear+1)),
    stockN = totalstock,
    recr = recruits,
    ssb = c(ssb,NA),
    F = c(Fmult,NA),
    catchbio = c(catchbio,NA),
    landings = c(landings,NA)
  )
))
print(truth, row.names = FALSE)
```

Appendix 12. ADMB template code for “silly” (annual catch curve) assessment model used in chub mackerel demonstration.

```
//Silly demo assessment model for chub mackerel (annual catch curves)
//Code originally from
//http://www.admb-project.org/courses/previous/august-2009/adintro.pdf
DATA_SECTION
 init_int iter  //not used
 init_int minYear
 init_int maxYear
 init_int nObs
 init_vector ageGrp(1,nObs)
 init_ivector year(1,nObs)
 init_vector lncat(1,nObs);
INITIALIZATION_SECTION
 a 26.5;
 b -1.5;
PARAMETER_SECTION
 init_vector a(minYear,maxYear,1)
 init_vector b(minYear,maxYear,2)
 objective_function_value nll
 vector yhat(1,nObs)
PROCEDURE_SECTION
 yhat=a(year)+elem_prod(b(year),ageGrp);
 nll=sum(square(lncat-yhat));
REPORT_SECTION
 report << b << endl;
```

Appendix 13. R code used with ADMB program to fit silly (annual catch curves fit to spawning biomass survey data) stock assessment model for chub mackerel demonstration. In this example, db is a database generated by R for chub mackerel. Directory and file names were set in Pseudo-data.R.

```
library(fs)

#get data for a spawning biomass survey out of database
dbTest=subset(db,nam=="NumByAge" & fleet == 3 )
dbTest=subset(dbTest,ageGrp>=4 & ageGrp<plusAge)
dbTest=subset(dbTest,value > 0)
dbTest$logvalue=log(dbTest$value)
dim(dbTest)
[1] 54522     8 #number of rows and columns with 200 iterations

#so system commands will run in same directory as rest of demo
#note=must put admb executable in adir
setwd(adir)

#sort data
dbTest=subset(dbTest,value>0)
dbTest=dbTest[with(dbTest,order(iter,year,ageGrp)),]
nObs=nrow(dbTest)
#bookkeeping
myIts=sort(unique(dbTest$iter))
nIter=length(myIts)
myYrs=sort(unique(dbTest$year))
nYear=length(myYrs)
#create data file for ADMB using data used for R test
adNameStart="silly"
adName=paste(adir, adNameStart,".dat", sep = "")
#a place to store catch curve results
allRez=c()

for (it in myIts) {
  cat(paste("Iteration",it,"\n"))
  useDat=subset(dbTest,iter==it)
  nObs=nrow(useDat)
  #delete dat admb file if it exists
  if (file.exists(adName)) {
    file_delete(adName)
  }
  fileConn = file(adName)
  #write ADMB data
  writeLines(
    c(
      "#Iter",
      paste(it),
      "#minYear",
      paste(min(myYrs)),
      "#maxYear",
      paste(max(myYrs)),
      "#nObs",
      paste(nObs),
      "#usedat$ageGrp",
      paste(useDat$ageGrp),
      "#usedat$year",
      paste(useDat$year),
      "#useDat$logvalue",
      useDat$logvalue,
      "#done"
    ),
    fileConn
  )
  close(fileConn)
#call the admb program
  system("silly -nox",invisible=FALSE)
#the ADMB program wrote Z estimates for this iteration to silly.rpt
#convert from negative slopes to positive Z values before saving
  oneRez=-scan("silly.rep")
# add results for this iteration to larger set for analysis later
  allRez=c(allRez,oneRez)
}
```

#now allRez holds Z estimates for all years and iterations