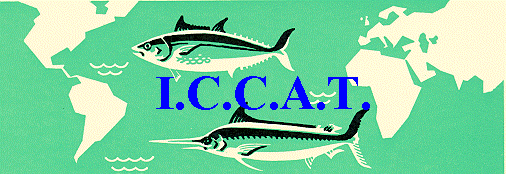
**INTERNATIONAL COMMISSION FOR THE CONSERVATION OF ATLANTIC TUNAS**

**COMMISSION INTERNATIONALE POUR LA CONSERVATION DES THONIDES DE L´ATLANTIQUE**

**COMISION INTERNACIONAL PARA LA CONSERVACION**

**DEL ATUN ATLANTICO**



**ASSESSMENT PROGRAM DOCUMENTATION**

**Program: PRODFIT (ver. U1)**

Fits a generalized stock production model to catch and effort data making an equilibrium approximation.

**Current Catalog Entry: October 2000 First Cataloged by ICCAT: October 2000**

**Catalogue Committee**

**External:** David Die (U. Miami, USA) and Jaime Mejuto (IEO, Spain)

**ICCAT Secretariat:** Victor Restrepo

NOTE: As part of its efforts to carry out Quality Management, ICCAT´s Standing Committee on Research and Statistics is developing a catalog of stock assessment applications. The purpose of the catalog is not to evaluate the relative merits of various assessment methods, but rather whether the software implementing the method works as intended and is adequately documented.

1. **PROGRAM NAME**

PRODFIT

1. **VERSION (DATE)**

Version U1, dated October, 1993

1. **LANGUAGE**

Fortran 77

1. **PROGRAMMER / CONTACT PERSON**

Originally programmed in Fortran IV by William W. Fox, Jr. NMFS, USA Adapted for PCs by

Alain Fonteneau IRD

B.P. 570

Victoria, Seychelles [irdsey@seychelles.net](mailto:irdsey@seychelles.net)

1. **DISTRIBUTION LIMITATIONS**

None.

1. **COMPILER NEEDS / STAND-ALONE**

Does not require other software, except operating system. Catalogued version compiled for use in MS DOS / Windows systems.

Users must be able to print and edit ASCII files and enter commands at a command prompt to use PRODFIT.

1. **PURPOSE**

Fits the generalized stock production model of Pella and Tomlinson (1969) by least-squares and equilibrium approximation.

1. **DESCRIPTION**

The following description was taken from Fox (1975b). The generalized stock production model is:

*t*

*dP/dt =*

*HPm*

*- KPt*

*- qf t Pt*

(1)

where, *P* is the population size (usually in terms of weight), *f* is effective fishing effort, i.e., standardized from nominal fishing effort to be proportional to the instantaneous fishing mortality coefficient, *q* is the constant of proportionality (the catchability coefficient), and *H*, *K*, and *m* are constant parameters. At equilibrium (i.e., *dP*/*dt* = 0)

*Pm*-1 = (*K*/*H*) + (*q*/*H*)*f*

or

and

*Um*-1 = (*Kqm*-1/*H*) + (*qm*/*H*)*f*

1

*U =* (*a + bf* )*m-*1

(2)

where *U* is the catch per unit effort.

The management implications of the generalized stock production are computed as:

1

*U* max

= *a m*-1 ,

1

*U opt*

*f opt*

(*a* / *m*) *m* 1 ,

= (*a* / *b*)(1/ *m* 1) ,

1

*Y max*

(*a* / *b*)(1/ *m* 1)(*a* / *m*) *m* 1

where *U*max is the relative density of the population before exploitation; *Uopt* is the relative population density providing the maximum sustainable yield; *fopt* is the amount of fishing effort to obtain the maximum sustainable yield; and *Ymax* is the maximum sustainable yield.

**Estimation**

Since catch and fishing effort data usually do not represent equilibrium conditions as required by equation (2), the fishing effort must be adjusted to approximate equilibrium conditions. This is done by computing a weighted average of fishing effort for year *i* over some previous number of years, *k*, which corresponds to the number of year classes making a significant contribution to the catch in year *i*, i.e:

*f i =* (*k f i + (k - l) f i-*1 *+ + f i-k +*1)*/* (*k* + (*k* -1) + ... + 1) . (3)

The data set of *(Ui , f i )* pairs are then utilized to estimate the parameters in equation (2). Note that *k* - 1 data points at the beginning of the set are lost unless some information about those *k* - 1 years prior to the data set can be entered. Note that *k* can be different each year.

PRODFIT provides least-squares estimates of the parameters *a*, *b*, and *m* in equation (2) by minimizing

*S = W*

*i*

*i* (*U i*

*- U*ˆ )2

(4)

where *Wi* are statistical weights for specifying a multiplicative error structure. An iterative pattern search optimization routine is utilized to locate the least-squares parameter estimates. In order to facilitate termination of the searching procedure, however, the sum-of-squares space is searched with *m*, *Umax*, and *Ymax*. The catchability coefficient, *q*, is estimated after estimating *a*, *b* and *m* by utilizing the integral of equation (1) to compute a *q* for each year, then the yearly *q*-values are averaged using arithmetic and geometric means.

*i*

Variability indices, *V*(*X*), of all the parameters are computed by the "delta", or propagation of error, method. These are not actual variances, but are useful for judging the fit of the model in a quantitative manner. An error index is computed for convenience as

*Ex =* (*100*

*V* ( *X* ))*/X*ˆ

where *X* is the estimated parameter.

**Summary of major assumptions:**

1. Single, closed population that follows the dynamics of equation (1).
2. The fishable population is constant (selectivity remains constant through time, or there are no age- structured effects).
3. Constant catchability through time.
4. No time lags in recruitment or in density-dependent growth, natural mortality and reproduction.
5. Equilibrium conditions are achieved at a constant rate of fishing.
6. The equilibrium approximation approach is sufficient to account for transient changes in population size.
7. **REQUIRED INPUTS**
   1. One catch and effort series, {*Ci, fi*}.
   2. Optionally, if equation (3) is used: A vector of significant year class numbers, *ki*.
8. **PROGRAM OUTPUTS**

An ASCII file containing the following information:

* 1. The raw data.
  2. The values of (*U i , f i*) used for fitting.
  3. Starting values and final estimates of the parameters, and the variability indices.
  4. The fitted *U*ˆ values and residuals

*i*

* 1. Estimates of the management-related parameters.
  2. Time-specific catchability estimates
  3. Vectors of equilibrium *f*, *C* and *U* for plotting (NB: Added by A. Fonteneau):

1. **DIAGNOSTICS**
   1. Plots of estimated vs. observed *U*.
   2. PRODFIT tests for incompatible inputs

(i) for *m*>=1, *a*>0.0, and *b*<0.0, or

(ii) for *m*<1, *a*>0.0, and *b*>0.0 has failed.

1. **OTHER FEATURES**

None.

1. **HISTORY OF METHOD PEER REVIEW**

The generalized production model underlying PRODFIT was described by Pella and Tomlinson (1969). Fox (1975a), published on using the equilibrium-approximation method for fitting the generalized model. The equilibrium approximation method was devised by Gulland (1969). Pella and Tomlinson (1969) and Fox (1975a) are both peer reviewed publications.

1. **STEPS TAKEN BY PROGRAMMER FOR VALIDATION**

Fox (1975a) reports on deterministic and stochastic simulation studies using PRODFIT to evaluate the equilibrium approximation method.

1. **TESTS CONDUCTED BY OTHERS**

In the process of completing the first version of this catalogue entry, Dr. David Die (University of Miami, ddie@rsmas.miami.edu) conducted the following test:

TEST: Data were from a Morgan (1978) lobster study; it provides the data and is peer- reviewed. Data were entered as provided in Morgan (1978) and PRODFIT(U1) options were set, to the best of tester´s ability as they were in the paper. (Note: articles dod not always provide details on all the options used. For instance the stopping criteria for the search algorithms where never provided).

PRODFIT (U1) gave similar results to Morgan (1978) when fitted to the lobster data, but not exactly the same values:

|  |  |  |
| --- | --- | --- |
| Parameter | Morgan (1978) | PRODFIT\* |
| a | 3.3108 | 3.26665 |
| b | 0.3087 | 0.300026 |
| m | 2.1 | 2.12 |
| MSY | 8,466,610 | 8,462,040 |
| Fopt | 5,642,140 | 5,752,100 |
| q (unweighted geometric mean) | 0.1433 x 10-7 | 0.765576 x 10-7 |

The one exception is the estimate of *q* which greatly differs between Morgan (1978) and the output of PRODFIT(U1). NOTE: PRODFIT(U1) fails to run completely with the lobster data and the program stops before it outputs the unweighted arithmetic mean and the predicted catch and cpue vectors. No tests have been conducted to determine why this happens.

1. **NOTES BY ICCAT**
2. (10/2000). The ways inputs/outputs are handled is not very flexible:

For instance if the user forgets to close all input and output files when running the program, DOS will give an error message indicating it does not have access to the file, or it will warn that the file is being used. Unfortunately there is no way to go back from such an error and closing the files and running PRODFIT(U1) again does not solve the problem. The user needs to restart the PC to be able to run it again.

PRODFIT(U1) requires the input file to be called PRODFIPC.PAR (unless the user modifies the FORTRAN program and recompiles it). When preparing this input file with NOTEPAD don’t forget to use SAVE AS and the option type of file “all files” otherwise it will save the .par as part of the root of the filename and give it a \*.txt extension.

1. (10/2000). Limited testing suggests that PRODFIT(U1) gives the results that it should give.

However, questions remain about the calculation of catchability coefficients (*q*).

1. **SOURCES CITED**

Fox, W.W., Jr. 1975a. Fitting the generalized stock production model by least-squares and equilibrium approximation. Fishery Bulletin, U.S. 73(1):23-37.

Fox, W.W., Jr. 1975b. PRODFIT user's manual. NMFS Southwest Fisheries Center, La Jolla, California.

Gulland, J.A. 1969. Manual of methods for fish stock assessment. Part 1. Fish population analysis.

FAO Man. Fish. Sci. 4, 154 p.

Morgan,G.R. 1978. Assessment of the stocks of the western rock lobster Panulirus cygnus using surplus yield models. Aust. J. Mar. Freshwater Res., 30(3), 355-363.

Pella, J. J. and P. K. Tomlinson. 1969. A generalized stock production model. Bull. Inter-Am. Trop.

Tuna Comm. 13:419-496.

1. **AUTHOR**===**S NOTES**

None in addition to Fox (1975a) and Fox (1975b).

**APPENDIX 1. ALGORITHM**

* 1. Input or compute average effort values (equation 3).
  2. Input or compute starting values for *a*, *b* and *m*.
  3. To facilitate the search, transform *a* and *b* into *Umax* and *Ymax*, which are not too sensitive to small changes in *m*.
  4. Compute the objective function, equation (4)
  5. Monitor the objective function for convergence. If achieved, end. Otherwise, modify the values of

*m*, *Umax* and *Ymax* according to the search algorithm until convergence is achieved.

Note: The search algorithm is based on the routine MIN that appeared in Pella and Tomlinson (1969), modified by Fox (1975a and 1975b).

The computation of other quantities, such as time-specific *q* values, is explained in Fox (1975a).

**APPENDIX 2. USER´S GUIDE**

The following was taken from Fox (1975b).

**Data Input**

Option 1. -- A catch and fishing effort history, {*Ci,fi*}, of *i*=1...*n* years length and a vector of significant year class numbers {*ki*} are read in. There may be embedded zeros, if they are true zeros and do not simply reflect a lack of information. The only real problem with unreal zeros, however, occurs in the estimation of

*q*. The catch per unit effort vector is computed internally and the averaged fishing effort vector is computed by equation (3) with SUBROUTINE AVEFF.

Option 2. -- If one wishes to compute the averaged fishing effort vector by another method or if data are obtained which represent equilibrium conditions, then this option is selected and the vectors of catch per

unit effort and averaged (or equilibrium) fishing effort *{Ui ,*

be made, however.

**Starting Values**

*f i }* are read in directly. No estimate of *q* can

Option 1. -- Initial estimates of the parameters are computed in SUBROUTINE INEST and the user provides the starting estimate for *m*, either 0, 1 or 2.

Option 2. -- Occasionally the data are so variable that INEST does not provide compatible starting values for the parameters. In this case, or in any case, the user may opt to enter directly all the initial parameter estimates.

**Model Option**

The user may allow PRODFIT to estimate *m* to any desired precision. Frequently, however, the data are so variable that no significant reduction in the residual sum of squares is obtained by varying *m*. The user then has the option to fix *m* at 2, the logistic model; at 1, Gompertz model; or at 0, the asymptotic yield model.

**Weighting Option**

The user may select statistical weights assuming a multiplicative error structure or may choose to not weight the observations, i.e., *Wi* = 1 for all *i*.

**Input File Setup**

NB1: Input file must be called “prodfitpc.par”; output file is “prodfitpc.lis”. Run Controls

NB2: All of the next 10 inputs are left-justified and 3 characters wide.

1. Title (80 characters max.)
2. NC= number of data points entered.
3. NDP = Data preparation option. Enter 0 for option 0 or 1 for option 1. See above.
4. NST = Starting values option. Enter 0 if starting values are to be computed by the program or 1 if they are to be read in (i.e.entered by user).
5. KK = Number of significant digits to which the parameters are searched. Suggest 5.
6. NPM = Number of digits past the decimal point to which m is searched. Suggest 2.
7. VL = Fraction for determining parameters upper and lower limits during search. Suggest 0.25.
8. XM = Starting value of m . This number can only be 0.0, 1.0, or 2.0.
9. XS = Model option. Enter 0 if m is to be estimated or 1 if it is fixed at the starting value (XM above).
10. XW = Weighting option. Enter 0 for the additive error model (unweighted) or 1 for the multiplicative error model (weighted). Suggest 1.

Data

NB3: The next 3 variables are entered as 3 columns, and NC rows. The format of each line is (2X,2F6.0,F10.0). This means: Two spaces (or two year digits to index time which are ignored by the program), followed by 6 characters for C, followed by six characters for XE, followed by 10 characters for XK.

C(i) = Catch records from 1 to NC (see line 2). If the value of NDP (above) is 1 then enter Catch per unit Effort records.

XE(I) = Fishing effort records from 1 to NC.

XK(I) = Number of significant year-classes contributing to the catches in line number 3. This line is NOT entered if the value of NDP (above, line 1) is 1.

Parameters

A = Starting value for *a*. B = Starting value for *b*.

XM = Starting value for *m*.

SE2 = Starting value for residual sum of squares. This line is NOT entered if the value of NST (above line no. 1) is 0.

**APPENDIX 3. WORKED EXAMPLE**

**Example input data file**

BET ATL TOTAL SCRS 1997

36 NB YEARS 0

0

5

2

.25

1.0 M 1

1

|  |  |  |  |
| --- | --- | --- | --- |
| 61 | 17.0 | 14.3 | 6. |
| 62 | 23.1 | 18.9 | 6. |
| 63 | 26.0 | 19.8 | 6. |
| 64 | 23.5 | 19.9 | 6. |
| 65 | 39.2 | 35.1 | 6. |
| 66 | 25.1 | 23.6 | 6. |
| 67 | 25.0 | 21.5 | 6. |
| 68 | 23.7 | 17.2 | 6. |
| 69 | 36.7 | 28.6 | 6. |
| 70 | 42.3 | 33.2 | 6. |
| 71 | 55.8 | 44.9 | 6. |
| 72 | 47.2 | 42.6 | 6. |
| 73 | 57.0 | 42.4 | 6. |
| 74 | 64.1 | 45.5 | 6. |
| 75 | 61.3 | 59.0 | 6. |
| 76 | 45.3 | 45.3 | 6. |
| 77 | 54.9 | 32.3 | 6. |
| 78 | 52.7 | 37.8 | 6. |
| 79 | 46.0 | 36.3 | 6. |
| 80 | 63.8 | 50.8 | 6. |
| 81 | 68.2 | 63.4 | 6. |
| 82 | 73.7 | 75.7 | 6. |
| 83 | 59.3 | 53.8 | 6. |
| 84 | 69.3 | 65.0 | 6. |
| 85 | 74.2 | 72.2 | 6. |
| 86 | 59.8 | 52.8 | 6. |
| 87 | 49.3 | 38.4 | 6. |
| 88 | 59.1 | 47.0 | 6. |
| 89 | 69.6 | 72.3 | 6. |
| 90 | 72.4 | 85.9 | 6. |
| 91 | 84.8 | 98.9 | 6. |
| 92 | 86.6 | 99.5 | 6. |
| 93 | 101.9 | 126.6 | 6. |
| 94 | 110.4 | 150.2 | 6. |
| 95 | 104.0 | 153.7 | 6. |
| 96 | 107.3 | 176.5 | 6. |

**Example output data file**

1))

LEAST-SQUARES FIT TO THE GENERALIZED STOCK PRODUCTION MODEL -- U=(A+B\*F)\*\*(1/(M- WITH THE METHODO F EQUILIBRIUM APPROXIMATION

BY WILLIAM W. FOX. JR.

BET ATL TOTAL SCRS 1997

RAW DATA

CATCH EFFORT NO. YEAR CLASSES

.170000E+02 .143000E+02 .600000E+01

.231000E+02 .189000E+02 .600000E+01

.260000E+02 .198000E+02 .600000E+01

.235000E+02 .199000E+02 .600000E+01

.392000E+02 .351000E+02 .600000E+01

.251000E+02 .236000E+02 .600000E+01

.250000E+02 .215000E+02 .600000E+01

.237000E+02 .172000E+02 .600000E+01

.367000E+02 .286000E+02 .600000E+01

.423000E+02 .332000E+02 .600000E+01

.558000E+02 .449000E+02 .600000E+01

.472000E+02 .426000E+02 .600000E+01

.570000E+02 .424000E+02 .600000E+01

|  |  |  |
| --- | --- | --- |
| .641000E+02 | .455000E+02 | .600000E+01 |
| .613000E+02 | .590000E+02 | .600000E+01 |
| .453000E+02 | .453000E+02 | .600000E+01 |
| .549000E+02 | .323000E+02 | .600000E+01 |
| .527000E+02 | .378000E+02 | .600000E+01 |
| .460000E+02 | .363000E+02 | .600000E+01 |
| .638000E+02 | .508000E+02 | .600000E+01 |
| .682000E+02 | .634000E+02 | .600000E+01 |
| .737000E+02 | .757000E+02 | .600000E+01 |
| .593000E+02 | .538000E+02 | .600000E+01 |
| .693000E+02 | .650000E+02 | .600000E+01 |
| .742000E+02 | .722000E+02 | .600000E+01 |
| .598000E+02 | .528000E+02 | .600000E+01 |
| .493000E+02 | .384000E+02 | .600000E+01 |
| .591000E+02 | .470000E+02 | .600000E+01 |
| .696000E+02 | .723000E+02 | .600000E+01 |
| .724000E+02 | .859000E+02 | .600000E+01 |
| .848000E+02 | .989000E+02 | .600000E+01 |
| .866000E+02 | .995000E+02 | .600000E+01 |
| .101900E+03 | .126600E+03 | .600000E+01 |
| .110400E+03 | .150200E+03 | .600000E+01 |
| .104000E+03 | .153700E+03 | .600000E+01 |
| .107300E+03 | .176500E+03 | .600000E+01 |

DATA FOR FITTING

CATCH/EFFORT AVERAGE EFFORT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| .106356E+01 | .242000E+02 |  | | | |
| .116279E+01 | .240762E+02 |  |  |  |  |
| .137791E+01 | .223810E+02 |  |  |  |  |
| .128322E+01 | .240238E+02 |  |  |  |  |
| .127410E+01 | .265619E+02 |  |  |  |  |
| .124276E+01 | .318095E+02 |  |  |  |  |
| .110798E+01 | .359333E+02 |  |  |  |  |
| .134434E+01 | .390952E+02 |  |  |  |  |
| .140879E+01 | .421476E+02 |  |  |  |  |
| .103898E+01 | .477095E+02 |  |  |  |  |
| .100000E+01 | .479095E+02 |  |  |  |  |
| .169969E+01 | .438190E+02 |  |  |  |  |
| .139418E+01 | .419000E+02 |  |  |  |  |
| .126722E+01 | .397810E+02 |  |  |  |  |
| .125591E+01 | .420952E+02 |  |  |  |  |
| .107571E+01 | .477571E+02 |  |  |  |  |
| .973580E+00 | .567238E+02 |  |  |  |  |
| .110223E+01 | .579857E+02 |  |  |  |  |
| .106615E+01 | .614238E+02 |  |  |  |  |
| .102770E+01 | .656238E+02 |  |  |  |  |
| .113258E+01 | .625714E+02 |  |  |  |  |
| .128385E+01 | .553095E+02 |  |  |  |  |
| .125745E+01 | .516952E+02 |  |  |  |  |
| .962656E+00 | .566762E+02 |  |  |  |  |
| .842840E+00 | .646619E+02 |  |  |  |  |
| .857432E+00 | .753667E+02 |  |  |  |  |
| .870352E+00 | .849714E+02 |  |  |  |  |
| .804897E+00 | .100095E+03 |  |  |  |  |
| .735020E+00 | .117762E+03 |  |  |  |  |
| .676643E+00 | .131514E+03 |  |  |  |  |
| .607932E+00 | .147905E+03 |  |  |  |  |
| STARTING VALUES | A = .100044E+01 | B = -.634602E-05 | M = | .100100E+01 | RESIDUAL SUM OF |

SQUARES = .100000E+39

RE-PARAMETERIZED STARTING VALUES AND LIMITS

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | VALUE | LOWER | UPPER |
| UMAX = | .154662E+01 | .115996E+01 | .193327E+01 |
| YMAX = | .896562E+02 | .672421E+02 | .112070E+03 |

M = .100100E+01 UMAX = .157662E+01 YMAX = .935341E+02 S?Q = .495075E+00

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

|  |  |  |  |
| --- | --- | --- | --- |
| .143000E+02 | .189000E+02 | .198000E+02 | .199000E+02 |
| .351000E+02 | .236000E+02 | .215000E+02 | .170000E+02 |
| .231000E+02 | .260000E+02 | .235000E+02 | .392000E+02 |
| .251000E+02 | .250000E+02 |  |  |
|  |  | \*\*\* FINAL ESTIMATES \*\*\* |  |

0RESIDUAL SUM OF SQUARES MINIMIZED WITHIN PARAMETER PRECISION OF

NO. DECIMAL PLACES FOR M = 2

0 NO. DIGITS FOR UMAX AND YMAX = 5 WEIGHTED ESTIMATES

FIXED M

A = .100046E+01 VAR. INDEX A = .248237E-08 B = -.620084E-05 VAR. INDEX B = .591248E-12 M = .100100E+01 VAR. INDEX M = .000000E+00 RESIDUAL SUM OF SQUARES = .495075E+00

DEGREES OF FREEDOM = .290000E+02 RESIDUAL VAR. INDEX = .170716E-01

0 DEGREE OF FIT INDEX = .656606E+00 VARIABILITY INDEX MATRIX

|  |  |  |
| --- | --- | --- |
| .248237E-08 | -.337935E-10 | .000000E+00 |
| -.337935E-10 | .591248E-12 | .000000E+00 |
| .000000E+00 | .000000E+00 | .000000E+00 |

AVERAGE EFFORT CATCH/EFFORT PRED. C/E ERROR TERM

|  |  |  |  |
| --- | --- | --- | --- |
| .242000E+02 | .106356E+01 | .135694E+01 | -.216209E+00 |
| .240762E+02 | .116279E+01 | .135808E+01 | -.143795E+00 |
| .223810E+02 | .137791E+01 | .137239E+01 | .401869E-02 |
| .240238E+02 | .128322E+01 | .135840E+01 | -.553460E-01 |
| .265619E+02 | .127410E+01 | .133720E+01 | -.471895E-01 |
| .318095E+02 | .124276E+01 | .129439E+01 | -.398893E-01 |
| .359333E+02 | .110798E+01 | .126180E+01 | -.121904E+00 |
| .390952E+02 | .134434E+01 | .123723E+01 | .865719E-01 |
| .421476E+02 | .140879E+01 | .121415E+01 | .160310E+00 |
| .477095E+02 | .103898E+01 | .117290E+01 | -.114178E+00 |
| .479095E+02 | .100000E+01 | .117151E+01 | -.146398E+00 |
| .438190E+02 | .169969E+01 | .120163E+01 | .414491E+00 |
| .419000E+02 | .139418E+01 | .121603E+01 | .146498E+00 |
| .397810E+02 | .126722E+01 | .123208E+01 | .285188E-01 |
| .420952E+02 | .125591E+01 | .121444E+01 | .341436E-01 |
| .477571E+02 | .107571E+01 | .117262E+01 | -.826469E-01 |
| .567238E+02 | .973580E+00 | .110914E+01 | -.122219E+00 |
| .579857E+02 | .110223E+01 | .110058E+01 | .150168E-02 |
| .614238E+02 | .106615E+01 | .107734E+01 | -.103876E-01 |
| .656238E+02 | .102770E+01 | .104959E+01 | -.208510E-01 |
| .625714E+02 | .113258E+01 | .106967E+01 | .588111E-01 |
| .553095E+02 | .128385E+01 | .111896E+01 | .147360E+00 |
| .516952E+02 | .125745E+01 | .114432E+01 | .988597E-01 |
| .566762E+02 | .962656E+00 | .110953E+01 | -.132379E+00 |
| .646619E+02 | .842840E+00 | .105599E+01 | -.201845E+00 |
| .753667E+02 | .857432E+00 | .988091E+00 | -.132234E+00 |
| .849714E+02 | .870352E+00 | .930973E+00 | -.651162E-01 |
| .100095E+03 | .804897E+00 | .847648E+00 | -.504347E-01 |
| .117762E+03 | .735020E+00 | .759677E+00 | -.324572E-01 |
| .131514E+03 | .676643E+00 | .697551E+00 | -.299740E-01 |
| .147905E+03 | .607932E+00 | .630122E+00 | -.352149E-01 |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* MANAGEMENT IMPLICATIONS OF THE FITTED MODEL \*\*\*

|  |  |  |
| --- | --- | --- |
|  | VARIABILITY | ERROR INDEX |
| VALUE | INDEX | (PERCENT) |
| PRE-E PLOITATION CATCH/EFFORT = .157657E+01 | .616396E-02 | 4.979838 |
| OPTIMUM CATCH/EFFORT / .580279E+00 | .835035E-03 | 4.979838 |
| OPTIMUM FISHING EFFORT = .161188E+03 | .399233E+03 | 12.395960 |
| MAXIMUM SUSTAINA BLE YIELD = .935341E+02 | .608611E+02 | 8.340647 |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* ESTIMATES OF THE CATCHABILITY COEFFICIENT AN POPULATION SIZE \*\*

\*\* BY THE INTEGRAL METHOD \*\*

TIME CATCHABILITY COEFFICIENT

.143000E+02 .189000E+02 .198000E+02 .199000E+02

.351000E+02 .236000E+02 .215000E+02 .170000E+02

.231000E+02 .260000E+02 .235000E+02 .392000E+02

.251000E+02 .250000E+02

.170000E+02 .231000E+02 .260000E+02 .235000E+02 .392000E+02 .251000E+02 .250000E+02

.143000E+02 .189000E+02 .198000E+02 .199000E+02 .351000E+02 .236000E+02 .215000E+02

.118881E+01 .122222E+01 .131313E+01 .118090E+01 .111681E+01 .106356E+01 .116279E+01

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

-.620084E-05

( 1, 2) .104064E-02

( 2, 3) .472077E-02

( 3, 4) .633011E-02

( 4, 5) .239197E-02

( 5, 6) .162697E-02

( 6, 7) .268346E-02

( 7, 8) .156824E-01

( 8, 9) .134540E-01

( 9,10) .250672E-02

(10,11) .121047E-01

(11,12) .555565E-02

(12,13) .951557E-03

(13,14) .214769E-02

(14,15) .508079E-02

(15,16) .212200E-02

(16,17) .238608E-02

(17,18) .701574E-02

(18,19) .139688E-01

(19,20) .118584E-02

(20,21) .929826E-02

(21,22) .254212E-03

(22,23) .383175E-02

(23,24) .494185E-02

(24,25) .159323E-01

(25,26) .208468E-02

(26,27) .295094E-02

(27,28) .267425E-02

(28,29) .935242E-03

(29,30) .238728E-01

(30,31) .240717E-02

(31,32) .778541E-02

(32,33) .817195E-02

(33,34) .415916E-02

(34,35) .384592E-02

-.620084E-05

(35,36) .571067E-02

UNWEIGHTED GEOMETRIC MEAN

Q = .383017E-02

0 COND. VARIANCE Q = .397307E-06

VIRGIN POPL. SIZE = .411620E+03 OPTIMUM POPL, SIZE = .151502E+03

UNWEIGHTED ARITHMETIC MEAN

Q = .576607E-02

0 COND. VARIANCE Q = .824867E-06

VIRGIN POPL. SIZE = .273423E+03 OPTIMUM POPL, SIZE = .100637E+03

EFFORT CATCH CPUE

|  |  |  |
| --- | --- | --- |
| 1 | 1.57 | 1.567 |
| 10 | 14.82 | 1.482 |
| 19 | 26.63 | 1.401 |
| 28 | 37.11 | 1.325 |
| 37 | 46.38 | 1.253 |
| 46 | 54.53 | 1.185 |
| 55 | 61.66 | 1.121 |
| 64 | 67.86 | 1.060 |
| 73 | 73.20 | 1.003 |
| 82 | 77.76 | .948 |
| 91 | 81.61 | .897 |
| 100 | 84.81 | .848 |
| 109 | 87.43 | .802 |
| 118 | 89.51 | .759 |
| 127 | 91.11 | .717 |
| 136 | 92.26 | .678 |
| 145 | 93.03 | .642 |
| 154 | 93.44 | .607 |
| 163 | 93.53 | .574 |
| 172 | 93.33 | .543 |
| 181 | 92.88 | .513 |
| 190 | 92.21 | .485 |
| 199 | 91.33 | .459 |
| 208 | 90.27 | .434 |
| 217 | 89.06 | .410 |
| 226 | 87.72 | .388 |
| 235 | 86.26 | .367 |
| 244 | 84.69 | .347 |
| 253 | 83.05 | .328 |
| 262 | 81.33 | .310 |
| 271 | 79.55 | .294 |
| 280 | 77.72 | .278 |
| 289 | 75.86 | .263 |

## APPENDIX 4. SOURCE CODE

C \*\* PROGRAMME PRODFIT \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* PRO00010 C VERSION MODIFIED OCTOBER 1993 - ADAPTED TO PC PRO00020 C INPUT FILE 'PRODFIPC.PAR'; OUTPUT FILE 'PRODFIPC.LIS'

DIMENSION C(100),XE(100),E(100),UE(100),XUE(100),XK(100),Z(3,3), PRO00040 1G(2),D(2),V(4),X(2),TITLE(20) PRO00050

CHARACTER\*32 FILE5

COMMON/VALC/C,E,UE,XK,XE,XUE,XW PRO00060

COMMON/VALV/Z PRO00070

COMMON/VALS/NC,XM PRO00080

OPEN(5,FILE='PRODFIPC.PAR') OPEN(6,FILE='PRODFIPC.LIS')

400 READ(5,41,END=40000)TITLE PRO00090

PO=0 PRO00100

XC=0 PRO00110

UEV=0 PRO00120

SE2=0 PRO00130

DO 852 I=1,100 PRO00140

852 UE(I)=0 PRO00150

41 FORMAT(20A4) PRO00160

WRITE(6,7) PRO00170

7 FORMAT(1H ,22X,85HLEAST-SQUARES FIT TO THE GENERALIZED STOCK PRODUPRO00180 1CTION MODEL -- U=(A+B\*F)\*\*(1/(M-1)) /1H ,41X,44HWITH THE METHODO PRO00190 2F EQUILIBRIUM APPROXIMATION/1H ,54X,23H BY WILLIAM W. FOX. JR./) PRO00200 WRITE(6,12) TITLE PRO00210

12 FORMAT(1H ,25X,20A4) PRO00220

READ(5,2)NC,NDP,NST,KK,NPM,VL,XM,XS,XW PRO00230

2 FORMAT(I3/I3/I3/I3/I3/F4.0/F4.0/F4.0/F4.0) PRO00240

DO 1 I=1,NC

1 READ(5,3) C(I) ,XE(I),XK(I) RO00250

3 FORMAT(2X,2F6.0,F10.0) PRO00290

WRITE(6,52) PRO00300

52 FORMAT(1H ,8HRAW DATA//1H ,9X,5HCATCH,17X,6HEFFORT,10X,16HNO. YEARPRO00310

1 CLASSES/) PRO00320

WRITE(6,13)(C(I),XE(I),XK(I),I=1,NC) PRO00330

13 FORMAT( 5X,E13.6,10X,E13.6,10X,E13.6) PRO00340 NRC=NC PRO00350

GO TO 6 PRO00360

4 DO 5 I=1,NC PRO00370

UE(I)=C(I) PRO00380

5 E(I)=XE(I) PRO00390

GO TO 9 PRO00400

6 CALL AVEFXE(NC) PRO00410

9 WRITE(6,8) PRO00420

8 FORMAT(1H ,16HDATA FOR FITTING/1H ,6X,12HCATCH/EFFORT,10X,14HAVERAPRO00430 1GE EFFORT/) PRO00440

WRITE(6,10)(UE(I),E(I),I=1,NC) PRO00450

10 FORMAT(1H ,5X,E13.6,10X,E13.6) PRO00460 XNC=NC PRO00470

IF(NST)61,61,57 PRO00480

61 CALL INEST(NC,XM,A,B,SE2) PRO00490

GO TO 63 PRO00500

57 READ(5,58)A,B,XM,SE2 PRO00510

58 FORMAT(4E13.6) PRO00520

63 WRITE(6,26)A,B,XM,SE2 PRO00530

26 FORMAT(1H ,16H STARTING VALUES,5X,3HA =,E14.6,5X,3HB =,E14.6, PRO00540

1 5X,3HM =,E14.6, 5X,25HRESIDUAL SUM OF SQUARES =,E14.6/) PRO00550 IF(A)999,999,990 PRO00560

990 IF(B\*(XM-.99))991,999,999 PRO00570

999 WRITE(6,998) PRO00580

998 FORMAT(1H ,62H\*\*\*\* STARTING VALUES INCOMPATIBLE -- EXECUTION TERMIPRO00590

1NATED \*\*\*\*) PRO00600

GO TO 400 PRO00610

991 CONTINUE PRO00620

WRITE(6,107) PRO00630

107 FORMAT(1H ,43HRE-PARAMETERIZED STARTING VALUES AND LIMITS/1H0,11X,PRO00640 15HVALUE,10X,5HLOWER,10X,5HUPPER ) PRO00650

SIGN=1. PRO00660

KKT=KK PRO00670

TOL=0. PRO00680

IF(XS-1.) 321,322,322 PRO00690

321 KK=2 PRO00700

NP=1 PRO00710

GO TO 323 PRO00720

322 KK=KKT PRO00730

NP=NPM PRO00740

323 NV=2 PRO00750

T=1. PRO00760

LLTT=0 PRO00770

X(1)=A\*\*(1./(XM-1.)) PRO00780

IF(XM)317,316,317 PRO00790

316 X(2)=1./B PRO00800

GO TO 304 PRO00810

317 X(2)=A/B\*(1./XM-1.)\*(A/XM)\*\*(1./(XM-1.)) PRO00820

304 SE20=SE2 PRO00830

AO=A PRO00840

BO=B PRO00850

XO1=X(1) PRO00860

XO2=X(2) PRO00870

G(1)=X(1) PRO00880

G(2)=X(2) PRO00890

XD=ALOG(G(1))/ALOG(10.)-1. PRO00900

MD=XD PRO00910

D(1)=10.\*\*MD PRO00920

XD=ALOG(G(2))/ALOG(10.)-1. PRO00930

MD=XD PRO00940

D(2)=10.\*\*MD PRO00950

V(1)=G(1)\*(1.-VL) PRO00960

V(3)=G(1)\*(1.+VL) PRO00970

V(2)=G(2)\*(1.-VL) PRO00980

V(4)=G(2)\*(1.+VL) PRO00990

WRITE(6,108)G(1),V(1),V(3) PRO01000

108 FORMAT(1H ,6HUMAX =,3E15.6) PRO01010

WRITE(6,109)G(2),V(2),V(4) PRO01020

109 FORMAT(1H ,6HYMAX =,3E15.6) PRO01030

CALL MIN(NV,KK,D,V,G,X,SE2,TOL) PRO01040

WRITE(6,310)XM,X(1),X(2),SE2 PRO01050

310 FORMAT(1H ,3HM =,E14.6,5X,6HUMAX =,E14.6,5X,6HYMAX =,E14.6,5X,5HS?PRO01060

1Q =,E14.6) PRO01070

A=X(1)\*\*(XM-1.) PRO01080

IF(XM)319,318,319 PRO01090

318 B=1./X(2) PRO01100

GO TO 320 PRO01110

319 B=A/X(2)\*(1./XM-1.)\*(A/XM)\*\*(1./(XM-1.)) PRO01120

320 IF(XS-1.) 324,743,743 PRO01130

324 IF(LLTT)742,742,743 PRO01140

742 IF(SE2-SE20)301,305,305 PRO01150

301 XMO=XM PRO01160

XM=XM+SIGN\*0.1\*\*NP\*(1.-XS) PRO01170

IF(XM.LT.0.999.OR.XM.GE.1.001) GO TO 314 PRO01180

XM=XM+SIGN\*0.1\*\*NP\*(1.-XS) PRO01190

314 IF(XM)315,304,304 PRO01200

315 XM=0.0 PRO01210

GO TO 304 PRO01220

305 SIGN=SIGN\*(-1.) PRO01230

T=T\*(-1.) PRO01240

IF(T)308,306,306 PRO01250

306 NP=NP+1 PRO01260

KK=KK+1 PRO01270

IF(NP-NPM)308,311,308 PRO01280

311 KK=KKT PRO01290

308 A=AO PRO01300

B=BO PRO01310

XM=XMO PRO01320

X(1)=XO1 PRO01330

X(2)=XO2 PRO01340

SE2=SE20 PRO01350

IF(NP-NPM)301,301,309 PRO01360

309 IF(KK-KKT)312,312,313 PRO01370

312 KK=KKT PRO01380

NP=NPM PRO01390

GO TO 304 PRO01400

313 KK=KK-1 PRO01410

LLTT=1 PRO01420

GO TO 304 PRO01430

743 WRITE(6,77) PRO01440

WRITE (6,43) (XE(KZ),KZ=1,7) ,(C(KZ),KZ=1,7) PRO01450 77 FORMAT(1H ,120(1H\*)) PRO01460

WRITE(6,37)NPM,KK PRO01470

37 FORMAT(1H ,53X,23H\*\*\* FINAL ESTIMATES \*\*\*//1H0,63HRESIDUAL SUM OF PRO01480

1SQUARES MINIMIZED WITHIN PARAMETER PRECISION OF/1H ,5X, PRO01490

230HNO. DECIMAL PLACES FOR M =,I3/1H0,5X,30HNO. DIGITS FOR UMAXPRO01500

3 AND YMAX =,I3/) PRO01510

IF (XW.EQ.1) WRITE(6,91) PRO01520

1. FORMAT (1H ,18HWEIGHTED ESTIMATES//) PRO01530 IF(XW.EQ.0) WRITE(6,92) PRO01540
2. FORMAT (1H ,20HUNWEIGHTED ESTIMATES//) PRO01550 IF(XS.EQ.0) WRITE (6,93) PRO01560
3. FORMAT (1H ,11HESTIMATED M//) PRO01570 IF(XS.EQ.1) WRITE (6,94) PRO01580
4. FORMAT (1H ,7HFIXED M//) PRO01590

DF=XNC-3.+XS PRO01600

IF(DF) 80,80,81 PRO01610

80 WRITE(6,83) PRO01620

83 FORMAT (1H ,100(1H\*)/1H ,30HERROR, ZERO DEGREES OF FREEDOM/1H ,100PRO01630

\*(1H\*)) PRO01640

GO TO 82 PRO01650

81 CALL COVAR(A,B,XM,XS,NC) PRO01660

DO 38 I=1,3 PRO01670

DO 38 J=1,3 PRO01680

38 Z(I,J)=Z(I,J)\*SE2/DF PRO01690

XC=NC PRO01700

DO 95 NJ=1,NC PRO01710

95 UEV=UE(NJ)+UEV PRO01720

IF(XW) 96,97,96 PRO01730

97 DO 86 NJ=1,NC PRO01740

86 PO=PO+(UE(NJ)\*\*2) PRO01750

PO=PO-((UEV)\*\*2/XC) PRO01760

GO TO 98 PRO01770

96 DO 87 NJ=1,NC PRO01780

87 PO=PO+(UE(NJ)/(UEV/XC)-1.)\*\*2 PRO01790

98 CC=(PO-SE2)/PO PRO01800

VA=Z(1,1) PRO01810

VB=Z(2,2) PRO01820

VM=Z(3,3)\*(1.-XS) PRO01830

RV=SE2/DF PRO01840

82 WRITE(6,39)A,VA,B,VB,XM,VM,SE2,DF,RV,CC PRO01850 WRITE(6,60) PRO01860

60 FORMAT(1H ,24HVARIABILITY INDEX MATRIX) PRO01870 WRITE(6,40)((Z(I,J),J=1,3),I=1,3) PRO01880

39 FORMAT(1H ,3HA =,E13.6,10X,15HVAR. INDEX A =,E13.6/1H ,3HB =,E13.PRO01890

16,10X,15HVAR. INDEX B =,E13.6/1H ,3HM =,E13.6,10X,15HVAR. INDEX PRO01900

2M =,E13.6/1H ,16X,25HRESIDUAL SUM OF SQUARES =,E13.6/1H ,21X,20HDEPRO01910

3GREES OF FREEDOM =,E13.6/1H ,20X,21HRESIDUAL VAR. INDEX =,E13.6/1HPRO01920

40,20X,21HDEGREE OF FIT INDEX =,E13.6//) PRO01930

40 FORMAT(1H ,3E16.6) PRO01940

WRITE(6,51) PRO01950

51 FORMAT(1H ,14HAVERAGE EFFORT,11X,14H CATCH/EFFORT,13X,12HPRED. C/PRO01960 1E ,11X,10HERROR TERM/) PRO01970

DO 42 I=1,NC PRO01980

SXU=1. PRO01990

XU=A+B\*E(I) PRO02000

IF(XU) 84,84,85 PRO02010

84 SXU=-1. PRO02020

XU=ABS(XU) PRO02030

85 PU=SXU\*(XU)\*\*(1./(XM-1.)) PRO02040

IF(XW)330,330,331 PRO02050

330 RE=UE(I)-PU PRO02060

GO TO 42 PRO02070

331 RE=UE(I)/PU-1. PRO02080

42 WRITE(6,43)E(I),UE(I),PU,RE PRO02090

43 FORMAT(1H ,1X,E13.6,12X,E13.6,12X,E13.6,12X,E13.6) PRO02100 UMAX=A\*\*(1./(XM-1.)) PRO02110

IF(XM)45,45,44 PRO02120

45 UOPT=0.0 PRO02130

VUOPT=0. PRO02140

SUOPT=0. PRO02150

FOPT=0.0 PRO02160

VFOPT=0. PRO02170

SFOPT=0. PRO02180

YEMAX=1./B PRO02190

VYMAX=1./B\*\*4\*VB PRO02200

GO TO 46 PRO02210

44 UOPT=UMAX\*XM\*\*(1./(1.-XM)) PRO02220

FOPT=A/B\*(1./XM-1.) PRO02230

YEMAX=UOPT\*FOPT PRO02240

PA=YEMAX\*(XM/(XM-1.))/A PRO02250

PB=YEMAX/B\*(-1) PRO02260

PM=YEMAX\*(ALOG(XM/A))/(XM-1.)\*\*2\*(1.-XS) PRO02270 VYMAX=PA\*\*2\*VA+PB\*\*2\*VB+PM\*\*2\*VM+2.\*PA\*PB\*Z(1,2)+2.\*PA\*PM\*Z(1,3)+2PRO02280 1.\*PB\*PM\*Z(2,3) PRO02290

PA=(1./XM-1.)/B PRO02300

PB=-A/B\*\*2\*(1./XM-1.) PRO02310

PM=-A/B/XM\*\*2\*(1.-XS) PRO02320

VFOPT=PA\*\*2\*VA+PB\*\*2\*VB+PM\*\*2\*VM+2.\*PA\*PB\*Z(1,2)+2.\*PA\*PM\*Z(1,3)+2PRO02330 1.\*PB\*PM\*Z(2,3) PRO02340

SFOPT=100.\*SQRT(VFOPT)/FOPT PRO02350

PA=UOPT/(A\*(XM-1.)) PRO02360

PM=-UOPT\*(XM\*ALOG(A/XM)+XM-1.)/(XM\*(XM-1.)\*\*2)\*(1.-XS) PRO02370 VUOPT=PA\*\*2\*VA+PM\*\*2\*VM+2.\*PA\*PM\*Z(1,3) PRO02380

SUOPT=100.\*SQRT(VUOPT)/UOPT PRO02390

46 PA=UMAX\*\*(2.-XM)/(XM-1.) PRO02400

PM=-UMAX\*ALOG(A)/(XM-1.)\*\*2\*(1.-XS) PRO02410

VUMAX=PA\*\*2\*VA+PM\*\*2\*VM+2.\*PA\*PM\*Z(1,3) PRO02420

SUMAX=100.\*SQRT(VUMAX)/UMAX PRO02430

SYMAX=100.\*SQRT(VYMAX)/YEMAX PRO02440

WRITE(6,77) PRO02450

WRITE(6,47)UMAX,VUMAX,SUMAX, UOPT,VUOPT,SUOPT, FOPT, PRO02460 1VFOPT,SFOPT, YEMAX,VYMAX,SYMAX PRO02470

47 FORMAT(1H ,39X,51H\*\*\* MANAGEMENT IMPLICATIONS OF THE FITTED MODEL PRO02480 1\*\*\*///1H ,54X,11HVARIABILITY,10X,11HERROR INDEX /1H ,39X, PRO02490 25HVALUE,11X,8H INDEX ,13X, 9H(PERCENT) /1H ,4X,31HPRE-E PRO02500 3PLOITATION CATCH/EFFORT =,E13.6,5X,E13.6,9X,F10.6 /1H , PRO02510 413X,22HOPTIMUM CATCH/EFFORT / PRO02520 5E13.6,5X,E13.6,9X,F10.6 /1H ,11X,24HOPTIMUM FISHING EFFORT = PRO02530 6 ,E13.6,5X,E13.6,9X,F10.6 /1H ,8X, 'MAXIMUM SUSTAINA PRO02540 7BLE YIELD =',E13.6,5X,E13.6,9X,F10.6) PRO02550

IF(NDP)33,33,50 PRO02560

33 WRITE(6,77) PRO02570

WRITE(6,71) PRO02580

1. FORMAT(1H ,31X,67H\*\* ESTIMATES OF THE CATCHABILITY COEFFICIENT AN PRO02590

1 POPULATION SIZE \*\*/1H ,50X,28H\*\* BY THE INTEGRAL METHOD \*\*////1H PRO02600 2,45X,4HTIME,10X,24HCATCHABILITY COEFFICIENT/) PRO02610

WRITE (6,43) (XE(KZ),KZ=1,7) ,(C(KZ),KZ=1,7) PRO02620

CALL QHAT(A,B,XM,NRC,FQ,FVQ,FWQ,FVWQ) PRO02630

WRITE(6,72) PRO02640

1. FORMAT (1H ///31X,25HUNWEIGHTED GEOMETRIC MEAN/) PRO02650 PMAX=UMAX/FQ PRO02660

POPT=UOPT/FQ PRO02670

WRITE(6,74)FQ,FVQ,PMAX,POPT PRO02680

74 FORMAT (1H ,60X,3HQ =,E14.6/1H0,44X,19H COND. VARIANCE Q =,E14.6/1PRO02690

1H ,44X,19HVIRGIN POPL. SIZE =,E14.6/1H ,43X,20HOPTIMUM POPL, SIZE PRO02700

2=,E14.6///) PRO02710

WRITE(6,75) PRO02720

75 FORMAT (1H ,31X,26HUNWEIGHTED ARITHMETIC MEAN/) PRO02730 PMAX=UMAX/FWQ PRO02740

POPT=UOPT/FWQ PRO02750

WRITE(6,74)FWQ,FVWQ,PMAX,POPT PRO02760

50 CONTINUE PRO02770

EFX=XE(NC) PRO02780

CALL CURV(EFX,A,B,XM) PRO02790

GO TO 400 PRO02800

40000 CONTINUE

CLOSE (5)

CLOSE (6) STOP

END PRO02810

FUNCTION SSQ(X) PRO02820

COMMON/VALC/C,E,UE,XK,XE,XUE,XW PRO02830

COMMON/VALS/NC,XM PRO02840

DIMENSION C(100),E(100),UE(100),XK(100),XE(100),XUE(100),X(2) PRO02850 SSQ=0. PRO02860

ERR1=0. PRO02870

A=X(1)\*\*(XM-1.) PRO02880

IF(XM)7,6,7 PRO02890

6 B=1./X(2) PRO02900

GO TO 8 PRO02910

7 B=A/X(2)\*(1./XM-1.)\*(A/XM)\*\*(1./(XM-1.)) PRO02920

8 DO 1 I=1,NC PRO02930

XX=A+B\*E(I) PRO02940

IF(XX)2,3,3 PRO02950

2 SNXX=-1. PRO02960

|  |  |  |
| --- | --- | --- |
|  | GO TO 4 | PRO02970 |
| 3 | SNXX=1. | PRO02980 |
| 4 | XX=SNXX\*(ABS(XX))\*\*(1./(XM-1.)) | PRO02990 |
|  | IF(XW)10,10,11 | PRO03000 |
| 10 | SSQ=SSQ+(UE(I)-XX)\*\*2 | PRO03010 |
|  | GO TO 1 | PRO03020 |
| 11 | SSQ=SSQ+(UE(I)/XX-1.)\*\*2 | PRO03030 |
| 1 | CONTINUE | PRO03040 |
|  | RETURN | PRO03050 |
|  | END | PRO03060 |
|  | SUBROUTINE AVEFXE(NC) | PRO03070 |
|  | COMMON/VALC/C,E,UE,XK,XE,XUE,XW | PRO03080 |
|  | DIMENSION C(100),E(100),UE(100),XK(100),XE(100),XUE(100) | PRO03090 |
|  | IZ=0 | PRO03100 |
|  | DO 1 I=1,NC | PRO03110 |
|  | IF(XE(I))3,3,2 | PRO03120 |
| 2 | XUE(I)=C(I)/XE(I) | PRO03130 |
|  | GO TO 7 | PRO03140 |
| 3 | XUE(I)=0.0 | PRO03150 |
| 7 | K=XK(I) | PRO03160 |
|  | IF(I-K)4,5,5 | PRO03170 |
| 5 | SEFF=0.0 | PRO03180 |
|  | SXK=0. | PRO03190 |
|  | DO 6 J=1,K | PRO03200 |
|  | L=I-J+1 | PRO03210 |
|  | XJ=K+1-J | PRO03220 |
|  | SXK=SXK+XJ | PRO03230 |
| 6 | SEFF=SEFF+XE(L)\*XJ | PRO03240 |
|  | E(I)=SEFF/SXK | PRO03250 |
|  | GO TO 1 | PRO03260 |
| 4 | E(I)=0.0 | PRO03270 |
| 1 | CONTINUE | PRO03280 |
|  | DO 8 I=1,NC | PRO03290 |
| 11 | IF(E(I))9,9,12 | PRO03300 |
| 12 | IF(XUE(I))9,9,8 | PRO03310 |
| 9 | IZ=IZ+1 | PRO03320 |
|  | NCZ=NC-IZ | PRO03330 |
|  | DO 10 J=I,NCZ | PRO03340 |
|  | JJ=J+1 | PRO03350 |
|  | XUE(J)=XUE(JJ) | PRO03360 |
| 10 | E(J)=E(JJ) | PRO03370 |
|  | GO TO 11 | PRO03380 |
| 8 | UE(I)=XUE(I) | PRO03390 |
|  | NC=NC-IZ | PRO03400 |
|  | RETURN | PRO03410 |
|  | END | PRO03420 |
|  | SUBROUTINE INEST(NC,XM,A,B,SE2) | PRO03430 |
|  | COMMON/VALC/C,E,UE,XK,XE,XUE,XW | PRO03440 |
|  | DIMENSION C(100),E(100),UE(100),XK(100),XE(100),XUE(100),Y(100) | PRO03450 |
|  | IF(XM)6,2,1 | PRO03460 |
| 1 | IF(XM-1.)3,3,6 | PRO03470 |
| 2 | DO 4 I=1,NC | PRO03480 |
| 4 | Y(I)=1./UE(I) | PRO03490 |
|  | GO TO 8 | PRO03500 |
| 3 | DO 5 I=1,NC | PRO03510 |
| 5 | Y(I)=ALOG(UE(I)) | PRO03520 |
|  | GO TO 8 | PRO03530 |
| 6 | XM=2.0 | PRO03540 |
|  | DO 7 I=1,NC | PRO03550 |
| 7 | Y(I)=UE(I) | PRO03560 |
| 8 | SX=0. | PRO03570 |
|  | SX2=0. | PRO03580 |
|  | SY=0. | PRO03590 |
|  | SXY=0. | PRO03600 |
|  | SW=0. | PRO03610 |
|  | DO 10 I=1,NC | PRO03620 |
|  | SW=SW+1. | PRO03630 |
|  | SX=SX+E(I) | PRO03640 |
|  | SX2=SX2+E(I)\*\*2 | PRO03650 |
|  | SY=SY+Y(I) | PRO03660 |
| 10 | SXY=SXY+E(I)\*Y(I) | PRO03670 |
|  | B=(SXY-SX\*SY/SW)/(SX2-SX\*\*2/SW) | PRO03680 |
|  | A=SY/SW-B\*SX/SW | PRO03690 |
|  | IF(XM-1.)12,11,12 | PRO03700 |
| 11 | XM=1.001 | PRO03710 |
|  | A=(EXP(A))\*\*(XM-1.) | PRO03720 |

|  |  |  |
| --- | --- | --- |
|  | B=A\*B\*(1./XM-1.)\*(-1.) | PRO03730 |
| 12 | SE2=10.\*\*38. | PRO03740 |
|  | RETURN | PRO03750 |
|  | END | PRO03760 |
|  | SUBROUTINE MIN(NV,KK,DEL,A,GUESS,X,FOFX,CVAL) | PRO03770 |
|  | DIMENSION A(4),GUESS(2),XNEW(2),XNOW(2),X(2),DEL(2),DELTA(2) | PRO03780 |
|  | NK = KK | PRO03790 |
|  | NX = NV | PRO03800 |
|  | DO 5 I = 1, NX | PRO03810 |
|  | XNOW(I) = GUESS(I) | PRO03820 |
|  | XNEW(I) = XNOW(I) | PRO03830 |
| 5 | DELTA(I)=DEL(I) | PRO03840 |
|  | DO 14 J=1,NX | PRO03850 |
|  | IF(DELTA(J)) 11,11,14 | PRO03860 |
| 11 | DELTA(J)=GUESS(J) | PRO03870 |
| 14 | CONTINUE | PRO03880 |
| 20 | FNOW=SSQ(XNOW) | PRO03890 |
| 201 | FOLD = FNOW | PRO03900 |
| 200 | DO 40 I=1, NX | PRO03910 |
|  | XNEW(I) = XNEW(I) + DELTA(I) | PRO03920 |
|  | NA = NX+I | PRO03930 |
|  | IF(XNEW(I) - A(NA)) 22, 22, 21 | PRO03940 |
| 21 | XNEW(I) = A(NA) | PRO03950 |
| 22 | FNEW=SSQ(XNEW) | PRO03960 |
|  | IF(FNEW - FNOW) 25, 26, 26 | PRO03970 |
| 25 | FNOW = FNEW | PRO03980 |
|  | GO TO 40 | PRO03990 |
| 26 | XNEW(I) = XNOW(I) - DELTA(I) | PRO04000 |
|  | IF(XNEW(I) - A(I)) 28, 30, 30 | PRO04010 |
| 28 | XNEW(I) = A(I) | PRO04020 |
| 30 | FNEW=SSQ(XNEW) | PRO04030 |
|  | IF(FNEW - FNOW) 25, 34, 34 | PRO04040 |
| 34 | XNEW(I) = XNOW(I) | PRO04050 |
| 40 | CONTINUE | PRO04060 |
|  | IF(FNOW-FOLD)157,45,45 | PRO04070 |
| 157 | DIF=ABS (FNOW-FOLD) | PRO04080 |
|  | SML=AMIN1(FNOW,FOLD) | PRO04090 |
|  | DSS=DIF/SML\*100. | PRO04100 |
|  | IF(DSS.GT.CVAL) GO TO 50 | PRO04110 |
| 45 | NK = NK - 1 | PRO04120 |
|  | IF(NK) 46, 46, 47 | PRO04130 |
| 47 | DO 48 J=1,NX | PRO04140 |
| 48 | DELTA(J)=DELTA(J)\*.1 | PRO04150 |
|  | GO TO 200 | PRO04160 |
| 50 | DO 60 I = 1, NX | PRO04170 |
|  | T = XNOW(I) | PRO04180 |
|  | XNOW(I) = XNEW(I) | PRO04190 |
|  | XNEW(I) = 2.\*XNEW(I) - T | PRO04200 |
|  | NA = NX+I | PRO04210 |
|  | IF(XNEW(I) - A(NA)) 52, 60, 51 | PRO04220 |
| 51 | XNEW(I) = A(NA) | PRO04230 |
|  | GO TO 60 | PRO04240 |
| 52 | IF(XNEW(I) - A(I)) 53, 60, 60 | PRO04250 |
| 53 | XNEW(I) = A(I) | PRO04260 |
| 60 | CONTINUE | PRO04270 |
|  | FNEW=SSQ(XNEW) | PRO04280 |
|  | IF(FNEW - FNOW) 65, 70, 70 | PRO04290 |
| 65 | FNOW = FNEW | PRO04300 |
|  | GO TO 50 | PRO04310 |
| 70 | DO 71 I = 1, NX | PRO04320 |
| 71 | XNEW(I) = XNOW(I) | PRO04330 |
|  | GO TO 201 | PRO04340 |
| 46 | FOFX = FNOW | PRO04350 |
|  | DO 80 I = 1, NX | PRO04360 |
| 80 | X(I) = XNOW(I) | PRO04370 |
|  | RETURN | PRO04380 |
|  | END | PRO04390 |
|  | SUBROUTINE COVAR(A,B,XM,XS,NC) | PRO04400 |
|  | COMMON/VALC/C,E,UE,XK,XE,XUE,XW | PRO04410 |
|  | COMMON/VALV/Z | PRO04420 |
|  | DIMENSION C(100),E(100),UE(100),XK(100),XE(100),XUE(100),Z(3,3) | PRO04430 |
|  | DOUBLE PRECISION DM,C11,C21,C31,C22,C32,C33 | PRO04440 |
|  | DO 1 I=1,3 | PRO04450 |
|  | DO 1 J=1,3 | PRO04460 |
| 1 | Z(I,J)=0.0 | PRO04470 |
|  | DO 2 I=1,NC | PRO04480 |

X=A+B\*E(I) PRO04490

IF(X) 10,10,11 PRO04500

10 IF(XS) 12,12,13 PRO04510

13 X1=1. PRO04520

GO TO 14 PRO04530

12 WRITE (6,15) PRO04540

15 FORMAT (1H ,100(1H\*)/1H ,41HERROR, LOG. NEGATIVE VALUE = NO VARIANPRO04550

\*CES/1H ,100(1H\*)) PRO04560

DO 16 K=1,3 PRO04570

DO 16 J=1,3 PRO04580

16 Z(K,J)=0. PRO04590

GO TO 6 PRO04600

11 X1=1./(XM-1.)\*X\*\*((2.-XM)/(XM-1.)) PRO04610

X2= -X\*\*(1./(XM-1.))\*ALOG(X)/(XM-1.)\*\*2 PRO04620 14 IF(XW)7,7,8 PRO04630

|  |  |  |
| --- | --- | --- |
| 7 | W=1. | PRO04640 |
|  | GO TO 9 | PRO04650 |
| 8 | W=X\*\*(2./(1.-XM)) | PRO04660 |
| 9 | Z(1,1)=Z(1,1)+W\*X1\*\*2 | PRO04670 |
|  | Z(1,2)=Z(1,2)+W\*E(I)\*X1\*\*2 | PRO04680 |
|  | Z(1,3)=Z(1,3)+W\*X1\*X2\*(1.-XS) | PRO04690 |
|  | Z(2,2)=Z(2,2)+W\*(E(I)\*X1)\*\*2 | PRO04700 |
|  | Z(2,3)=Z(2,3)+W\*E(I)\*X1\*X2\*(1.-XS) | PRO04710 |
| 2 | Z(3,3)=Z(3,3)+W\*X2\*\*2\*(1.-XS) | PRO04720 |
|  | IF(XS-0.5)5,5,4 | PRO04730 |
| 4 | DM=Z(1,1)\*Z(2,2)-Z(1,2)\*\*2 | PRO04740 |
|  | TEMP=Z(1,1) | PRO04750 |
|  | Z(1,1)=Z(2,2)/DM | PRO04760 |
|  | Z(1,2)=-Z(1,2)/DM | PRO04770 |
|  | Z(2,2)=TEMP/DM | PRO04780 |
|  | Z(2,1)=Z(1,2) | PRO04790 |
|  | GO TO 6 | PRO04800 |
| 5 | Z(2,1)=Z(1,2) | PRO04810 |
|  | Z(3,1)=Z(1,3) | PRO04820 |
|  | Z(3,2)=Z(2,3) | PRO04830 |
|  | C11=Z(2,2)\*Z(3,3)-Z(2,3)\*Z(3,2) | PRO04840 |
|  | C21=-Z(1,2)\*Z(3,3)+Z(1,3)\*Z(3,2) | PRO04850 |
|  | C31=Z(1,2)\*Z(2,3)-Z(1,3)\*Z(2,2) | PRO04860 |
|  | C22=Z(1,1)\*Z(3,3)-Z(1,3)\*Z(3,1) | PRO04870 |
|  | C32=-Z(1,1)\*Z(2,3)+Z(1,3)\*Z(2,1) | PRO04880 |
|  | C33=Z(1,1)\*Z(2,2)-Z(1,2)\*Z(2,1) | PRO04890 |
|  | DM=Z(1,1)\*C11+Z(2,1)\*C21+Z(3,1)\*C31 | PRO04900 |
|  | Z(1,1)=C11/DM | PRO04910 |
|  | Z(1,2)=C21/DM | PRO04920 |
|  | Z(1,3)=C31/DM | PRO04930 |
|  | Z(2,1)=C21/DM | PRO04940 |
|  | Z(2,2)=C22/DM | PRO04950 |
|  | Z(2,3)=C32/DM | PRO04960 |
|  | Z(3,1)=C31/DM | PRO04970 |
|  | Z(3,2)=C32/DM | PRO04980 |
|  | Z(3,3)=C33/DM | PRO04990 |
| 6 | CONTINUE | PRO05000 |
|  | RETURN | PRO05010 |
|  | END | PRO05020 |
|  | SUBROUTINE QHAT(A,B,XM,NRC,FQ,FVQ,FWQ,FVWQ) | PRO05030 |
|  | COMMON/VALC/C,E,UE,XK,XE,XUE,XW | PRO05040 |
|  | DIMENSION C(100),E(100),UE(100),XK(100),XE(100),XUE(100),U(100) | PRO05050 |
|  | WRITE (6,9) (C(K),K=1,7),(XE(K),K=1,7) | PRO05060 |
|  | DO 1 I=1,NRC | PRO05070 |
|  | IF(XE(I)) 3,3,2 | PRO05080 |
| 2 | U(I)=C(I)/XE(I) | PRO05090 |
|  | GO TO 1 | PRO05100 |
| 3 | U(I)=0. | PRO05110 |
| 1 | CONTINUE | PRO05120 |
|  | XN=0. | PRO05130 |
|  | SQ=0. | PRO05140 |
|  | SQ2=0. | PRO05150 |
|  | SWQ=0. | PRO05160 |
|  | SWQ2=0. | PRO05170 |
|  | L=NRC-1 | PRO05180 |
| 9 | FORMAT (7E13.6) | PRO05190 |
|  | WRITE (6,9) (U(K),K=1,7) | PRO05200 |
|  | DO 4 I=1,L | PRO05210 |
|  | KKK=I+1 | PRO05220 |
|  | J=I+1 | PRO05230 |
|  | IF(U(I))4,4,5 | PRO05240 |

5 IF(U(J))4,4,6 PRO05250

6 X=-A/B-(XE(I)+XE(J))/2. PRO05260

Y=1./B PRO05270

Q=ABS(ALOG(ABS((X/U(I)\*\*(XM-1.)+Y)/(X/U(J)\*\*(XM-1.)+Y))))/(X\*XM-X)PRO05280

|  |  |  |
| --- | --- | --- |
|  | WRITE (6,9) B | PRO05290 |
| IF(Q) 4,4,7 | PRO05300 |
| 7 | WRITE(6,85) I,KKK,Q | PRO05310 |
| 85 | FORMAT (1H ,45X,1H(,I2,1H,,I2,1H),11X,E14.6) | PRO05320 |
|  | SWQ=SWQ+Q | PRO05330 |
|  | SWQ2=SWQ2+Q\*\*2 | PRO05340 |
|  | Q=ALOG(Q) | PRO05350 |
|  | SQ=SQ+Q | PRO05360 |
|  | SQ2=SQ2+Q\*\*2 | PRO05370 |
|  | XN=XN+1. | PRO05380 |
| 4 | CONTINUE | PRO05390 |
|  | FQ=SQ/XN | PRO05400 |
|  | FQ=EXP(FQ) | PRO05410 |
|  | FVQ=(SQ2-SQ\*\*2/XN)/(XN\*XN-XN)\*FQ\*\*2 | PRO05420 |
|  | FWQ=SWQ/XN | PRO05430 |
|  | FVWQ=(SWQ2-SWQ\*\*2/XN)/(XN\*XN-XN) | PRO05440 |
|  | RETURN | PRO05450 |
|  | END | PRO05460 |
|  | SUBROUTINE CURV(F,A,B,XM) | PRO05470 |
|  | IM=F\*3 | PRO05480 |
|  | IP=F/10 | PRO05490 |
|  | WRITE(6,4) | PRO05500 |
| 4 | FORMAT(1H ,' EFFORT CATCH',12X,'CPUE'//) | PRO05510 |
|  | DO 1 K=1,IM,IP | PRO05520 |
|  | U=(A+B\*K)\*\*(1/(XM-1)) | PRO05530 |
|  | P=U\*K | PRO05540 |
| 1 | WRITE(6,2)K,P,U | PRO05550 |
| 2 | FORMAT(I10,F10.2,F10.3) | PRO05560 |
|  | RETURN | PRO05570 |
|  | END | PRO05580 |