Package: FLXSA (via r-universe)

September 7, 2024

Title eXtended Survivor Analysis for FLR

Version 2.6.6

Description Calculates stock numbers and fishing mortality at age from commercial catch data and one or more indices of abundance suing the method in Darby and Flatman (1994) and Shepherd (1999).

License GPL-2

Depends R(>= 3.5.0), methods, FLCore(>= 2.6), FLAssess(>= 2.5)

Suggests testthat

Additional_repositories http://flr-project.org/R

RoxygenNote 7.2.1

ByteCompile TRUE

Repository https://flr.r-universe.dev

RemoteUrl https://github.com/flr/FLXSA

RemoteRef HEAD

RemoteSha d5cfce1cfdb0b1471a9035460be31091949b3148

Contents

1

---

data   *FLXSA data sets*

Description

Data sets for testing and validation of the FLXSA package

Details

The three datasets available in this package (cod4, her4 and ple7a) each consist of four objects.

*.indices An FLIndices object holding various indices of abundance for each stock.

*.stock An stock object with the necessary catch estimates, biological parameters and previous stock estimates.

*.xsa.control The FLXSA.control object for the reference run.

*.xsa.ref The FLXSA object obtained from the reference run.

Author(s)

Laurie Kell

See Also

FLXSA-class,FLXSA.control-class

---

diagnostics-method   *XSA diagnostics*

Description

Provides the diagnostics table used in ICES WG to analyse the results of the XSA run.

Usage

```
## S4 method for signature 'FLXSA'
diagnostics(object, sections = rep(T, 8), ...)
```

Generic Function

call diagnostics(object, ...

object An object with the results of a VPA method.

FLXSA   *Create a new FLXSA object -run an XSA analysis-*

Description

This function runs an XSA (extended survivor analysis) and creates an FLXSA object used to analyse its results.

Usage

```
FLXSA(stock, indices, ...)
```

Arguments

stock   An FLStock object to be used for the analysis

indices   An FLIndices object holding the indices of abundance to consider in the model

control   An FLXSA.control object giving parameters of the model (see FLXSA.control)

desc   A short description of this analysis

diag.flag   If TRUE returns all diagnostics, if FALSEonly returns stock.n, harvest and control

Details

Virtual population analysis and cohort analysis are essentially accountancy methods whereby a stock's historical population structure may be reconstructed from total catch data given a particular level of natural mortality. Firstly, however, numbers at age in the last year and age have to be found since both methods iterate backwards down a cohort. The main problem in many sequential age based assessment methods is therefore to estimate these terminal population numbers. In XSA these are found from the relationship between catch per unit effort (CPUE), abundance and year class strength.

Estimates of the catchability for the oldest age in an assessment, tuned by the ad hoc or XSA procedures, are directly dependent on the terminal population or F values used to initialise the underlying VPA. Catchability at the oldest age is therefore under-determined and cannot be utilised without additional information. Within the ad hoc tuning procedures the additional information is obtained by making the assumption that the exploitation pattern on the oldest ages was constant during the assessment time series. F on the oldest age in the final year is estimated as a proportion of an average of the F for preceding ages in the same year. XSA uses an alternative approach by making the assumption that fleet catchability is constant (independent of age) above a certain age. The age (constant for all fleets) is user-defined. For each fleet, the catchability value estimated at the specified age, is used to derive population abundance estimates for all subsequent ages in the fleet data set.

Value

An FLXSA object is returned, whith slots:

n	An FLQuant with the number of individuals at age
f	An FLQuant with the fishing mortality
swt	An FLQuant with the stock weight
mat	An FLQuant with the maturity indices
qres	A list with residuals for q
cpue	A list with the various cpues
wts	A list with the various weights
control	The FLXSA.control object that was used for this analysis
call	A copy of the call to run this analysis
desc	A description of the analysis

Note

See update to learn how to update stock data according to an XSA analysis

Author(s)

Laurence Kell and Philippe Grosjean

References

Darby, C. D., and Flatman, S. 1994. Virtual Population Analysis: version 3.1 (Windows/Dos) user guide. Info. Tech. Ser., MAFF Direct. Fish. Res., Lowestoft, (1): 85pp.

Shepherd, J.G. 1992. Extended survivors analysis: an improved method for the analysis of catch-at-age data and catch-per-unit-effort data. Working paper No. 11 ICES Multi-species Assessment Working Group, June 1992, Copenhagen, Denmark. 22pp. (mimeo).

Shepherd, J.G. 1994. Prediction of yearclass strength by calibration regression analysis of multiple recruit index series. ICES J. Mar. Sci. In Prep.

See Also

FLXSA.control, FLStock-class

Examples

```
data(ple4)
data(ple4.indices)

res <- FLXSA(ple4, ple4.indices)
```

FLXSA-class   *Class FLXSA*

Description

A class for the results of an XSA analysis.

Objects from the Class

Objects can be created by calls of the form new("FLXSA", ...) and are output by calls to FLXSA.

Author(s)

Laurence Kell

References

Darby, C. D., and Flatman, S. 1994. Virtual Population Analysis: version 3.1 (Windows/Dos) user guide. Info. Tech. Ser., MAFF Direct. Fish. Res., Lowestoft, (1): 85pp.

Shepherd, J.G. 1992. Extended survivors analysis: an improved method for the analysis of catch-at-age data and catch-per-unit-effort data. Working paper No. 11 ICES Multi-species Assessment Working Group, June 1992, Copenhagen, Denmark. 22pp. (mimeo).

Shepherd, J.G. 1994. Prediction of yearclass strength by calibration regression analysis of multiple recruit index series. ICES J. Mar. Sci. In Prep.

See Also

FLXSA

FLXSA.control   *A class for FLXSA control options*

Description

Runs of the FLXSA method require a number of parameter to be set. Most of them change the behaviour of the solution-searching algorithm Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque eleifend odio ac rutrum luctus. Aenean placerat porttitor commodo. Pellentesque eget porta libero. Pellentesque molestie mi sed orci feugiat, non mollis enim tristique. Suspendisse eu sapien vitae arcu lobortis ultrices vitae ac velit. Curabitur id

Usage

```
FLXSA.control(
  x = NULL,
  tol = 1e-09,
  maxit = 30,
  min.nse = 0.3,
  fse = 0.5,
  rage = 0,
  qage = 10,
  shk.n = TRUE,
  shk.f = TRUE,
  shk.yrs = 5,
  shk.ages = 5,
  window = 100,
  tsrange = 20,
  tspower = 3,
  vpa = FALSE
)
```

Slots

x An object of class FLXSA. If provided, the 'FLXSA.control' is initialized with the corresponding values of an XSA analysis stored in the object. This is useful for getting the same initial sloteters for successive analyses. Specifying one or more of the other arguments supersedes default values, or values obtained from this FLXSA object

tol The covergence tolerance, i.e. difference between two successive iterations must be lower, to declare convergence of the model.

maxit The maximum number of iterations allowed

min.nse The minimum value of SE permitted in estimate of N hat

fse User set SE of F when shrinking to mean F

rage The oldest age for which the two parameter model is used for catchability at age. Note that this value should be one less than the value used in the executable version of XSA

qage The age after which catchability is no longer estimated. q at older ages set to the value at this age

shk.n If TRUE, shrinkage to mean N

shk.f If TRUE, shrinkage to mean F

shk.yrs The number of years to be used for shrinkage to F for terminal year

shk.ages The number of ages to be used for shrinkage to F for terminal age

window The time window to consider in the model

tsrange The number of years to be used in the time series weighting

tspower The power to be used in the time series taper weighting

vpa If FALSE use cohort analysis, otherwise, use VPA

Validity

VALIDITY Neque porro quisquam est qui dolorem ipsum.

You can inspect the class validity function by using getValidity(getClassDef('FLCatch'))

Accessors

All slots in the class have accessor and replacement methods defined that allow retrieving and substituting individual slots.

The values passed for replacement need to be of the class of that slot. A numeric vector can also be used when replacing FLQuant slots, and the vector will be used to substitute the values in the slot, but not its other attributes.

Constructor

A construction method exists for this class that can take named arguments for any of its slots. All slots are then created to match the requirements of the class validity.

Author(s)

Laurence Kell & Philippe Grosjean

See Also

[FLComp](#)

Examples

```
# To create a new FLXSA.control object with default parameters:
my.xsa.control <- FLXSA.control()
my.xsa.control
# Same, but changing values of some parameters
my.xsa.control <- FLXSA.control(maxit=50, shk.f=FALSE)
my.xsa.control
```

---

is.FLXSA   *is.FLXSA*

Description

These two functions return code if objects are of class FLXSA and FLXSA.control, respectively.

Usage

```
is.FLXSA(x)
```

Arguments

x   An object to be tested

Value

is.FLXSA returns TRUE if its argument is of class FLXSA (that is, has "FLXSA" amongst its classes) and FALSE otherwise. is.FLXSA.control returns TRUE if its argument is of class FLXSA.control (that is, has "FLXSA.control" amongst its classes) and FALSE otherwise.

Examples

```
xsa <- FLXSA.control()
is.FLXSA.control(xsa)
```

---

rand   *rand*

Description

Uses the internal standard errors to conduct a Monte Carlo simulation of the terminal N-at-age in the last year

Arguments

n   number of iters to generate

mean   an FLXSA object

Value

sd an FLStock object

Examples

```
## Not run:
data(ple4)
data(ple4.indices)
xsa =FLXSA(ple4,ple4.indices)
ple4=rand(100,ple4,xsa)
## End(Not run)
```

xsa.sa	*An FLXSA estimator module for the mse package*

Description

The 'mse' package can use different stock assessment methods as modules in the estimation step of a management procedure. This functions provides such a module for FLXSA. As a flag for convergence, the returned tracking FLQuant contains the number of maximum number of iterations (maxit).

Usage

```
xsa.sa(stk, idx, args, tracking, ...)
```

Arguments

stk	An FLStock.
idx	An FLIndices.

Value

A list containing the updated FLStock, and the tracking FLQuant.

Examples

```
data(ple4)
data(ple4.index)
xsa.sa(stk=ple4, idx=ple4.index, args=list(ay=2018), tracking=FLQuant())
```

Index