

# Package: FLAssess (via r-universe)

August 16, 2024

**Title** Generic Classes and Methods for Stock Assessment Models

**Version** 2.6.3

**Date** 2018-10-02

**Description** A generic set of classes for stock assessment models are provided here. Individual assessment packages should extend the basic classes.

**Collate** classFLAssess.R accessors.R methodsFLAssess.R FLSP.R SepVPA.R VPA.R handyfuncs.R

**Depends** methods, FLCore(>= 2.5.0)

**Imports** lattice

**Additional\_repositories** <http://flr-project.org/R>

**License** GPL-2

**LazyLoad** Yes

**LazyData** No

**RoxygenNote** 7.2.1

**Repository** <https://flr.r-universe.dev>

**RemoteUrl** <https://github.com/flr/FLAssess>

**RemoteRef** HEAD

**RemoteSha** 03d018dbad8d4a4538e57259da5eee9780ff5948

## Contents

assess-methods . . . . .	2
data . . . . .	2
FLAssess . . . . .	2
FLSepVPA.control-class . . . . .	4
no.discards-methods . . . . .	5
plot . . . . .	5
SepVPA-methods . . . . .	6
VPA-methods . . . . .	6
Z-methods . . . . .	7

**Index****8**


---

assess-methods	<i>Call an stock assessment model</i>
----------------	---------------------------------------

---

**Description**

The assess methods calls the stock assessment function according to the class of the control object given. Stock assessment methods in FLR generally require a control class that contains all the options the model and its fitting algorithm provide. Classes are specific to each method, so see the relevant help pages for detailed information.

The assess method allows different stock assessment procedures to be carried out in a simulation procedure, or in any other function, by simply altering the control class in the argument list.

This is a generic method to be extended for individual stock assessment models.

**Methods**

**object = "FLAssess"** The generic, to be extended for each specific assessment model

---

data	<i>FLAssess dataset</i>
------	-------------------------

---

**Description**

**ass** Results of performing an XSA on ple4 data

Datasets can be loaded by issuing data(ass).

---

FLAssess	<i>FLAssess class and methods</i>
----------	-----------------------------------

---

**Description**

FLAssess is the basic structure for age-based stock assessment. It provides a standard class for data input, diagnostic inspection and stock status estimation; either for use within a working group setting or as part of a formal Management Strategy Evaluation (MSE).

The FLAssess class can be extended to create specific implementations of assessment methods e.g. FLICA, FLSURBA, FLXSA, providing a common interface for all assessment methods. For example, within ICES there are two main stock assessment methods, ICA for pelagic and XSA for demersal stocks. However, differences between the methods are mainly artefacts of how they were independently developed rather than methodological. By incorporating such methods in a common class this problem will hopefully be avoided in the future.

FLAssess also incorporates methods for performing virtual population analysis (VPA) and stock projection.

In common with other FLR classes, all of the slots of the FLAssess class have accessor and replacement methods. Users are encouraged to use these instead of the @ slot accessor.

## Objects from the Class

Objects can be created by calls to the methods VPA or SepVPA. They also can be created by calls of the form `new("FLAssess", ...)`.

## Slots

**name** A character string give name of stock.  
**desc** Whatever you want.  
**range** A named vector specifying the range of the object.  
**call** Call that generated the object.  
**catch.n** An FLQuant that contains estimated catch numbers-at-age.  
**stock.n** An FLQuant that contains estimated stock numbers-at-age.  
**harvest** An FLQuant that contains estimated fishing mortality-at-age.  
**index.name** A vector containing names for the FLIndexes  
**index.range** A list containing ranges for the FLIndexes  
**index** A list of FLQuants, corresponding to the index values used when fitting.  
**index.hat** A list of FLQuants, corresponding to the fitted index values.  
**index.res** A list of FLQuants, corresponding to the index residuals.  
**index.var** A list of FLQuants, corresponding to the variances of index values in fitting.

## Methods

**summary:** Returns a list, with a vector for each dimension in the object, that contains the levels of each dimension.  
**show:** Prints the given object.  
**plot:** Plots stock.n against index  
**+:** Updates a FLStock with stock.n and harvest estimates from an FLAssess object  
**merge:** Updates a FLStock with stock.n and harvest estimates from an FLAssess object  
**update:** Updates a FLStock with stock.n, catch.n & harvest estimates  
**SepVPA:** Separable VPA  
**VPA:** Virtual Population Analysis

## Author(s)

FLR Team

## See Also

FLIndices-class, FLStock-class

## Examples

```
# see the Methods help pages for specific examples
```

---

FLSepVPA.control-class

*Control class for FLSepVPA*

---

### Description

Objects of this class contain all the necessary settings for the Separable VPA model available through function SepVPA.

### Objects from the Class

Objects can be created by calls of the form `new("FLSepVPA.control", ...)` or by calling the [FLSepVPA.control](#) function.

### Slots

`sep.nyr`: Object of class "integer". Number of years for separable model.

`sep.age`: Object of class "integer". Reference age for fitting the separable model. Default value = 4

`sep.sel`: Object of class "numeric". Default value = 1.0

### Methods

No methods defined with class "FLSepVPA.control" in the signature.

### Author(s)

Iago Mosqueira

### References

JG Shepherd, SM Stevens. 1983. Separable VPA: User's guide - Int. Rep., MAFF Direct. Fish. Res.

### See Also

FLSepVPA

### Examples

```
sep.vpa.control <- FLSepVPA.control(sep.nyr=5, sep.age=5)
```

---

no.discards-methods     *Sets catch data if no discards*

---

### Description

This method sets the slots `discards.n` and `discards.wt` to 0, and sets the slots `catch`, `catch.n` and `catch.wt` to their landings equivalents.

### Methods

`no.discards<-(obj)` Takes in an object of type `FLStock` and carries out the above actions. Returns an object of type `FLStock`

---

`plot`     *Plot method for FLAssess class*

---

### Description

Standard plot methods for the `FLAssess` class. FLR plot methods are based on [lattice](#), and attempt to show a general view of the object contents.

Users are encouraged to write their own plotting code make use of the overloaded [lattice](#) methods, for example [xyplot](#) or [bwplot](#). See also [lattice-FLCore](#).

### Methods

**signature(x=FLAssess,y=missing)** : Plot of an *FLAssess* object.

### Author(s)

The FLR Team

### See Also

[plot](#)

**Description**

Performs Separable Virtual Population Analysis. Methods for function SepVPA in package **in Package ‘FLAssess’**. Requires an object of class `FLSepVPA.control` to be created.

**Methods**

`SepVPA` <- function(stock, control=FLSepVPA.control(), ref.harvest="missing", fratio="missing", fit.plusgroup, where stock is of type `FLStock`, control is of type `FLSepVPA.control()`, ref.harvest and fratio are numeric and fit.plusgroup is Boolean.

**Examples**

```
# Example based on ple4 dataset
data(ple4)
# Set up stock with correct dimensions
my.stock <- FLStock(catch.n(ple4))
my.stock@range["plusgroup"] <- 15
#load catch data and mortality
my.stock@catch.n <- ple4@catch.n
my.stock@catch.n[my.stock@catch.n==0] <- 1
my.stock@m <- ple4@m
my.control <- FLSepVPA.control(sep.age = 5)
# Set up in final year
my.stock@stock.n[, "2001"] <- ple4@stock.n[, "2001"]
# Run SepVPA
my.stock.SepVPA <- SepVPA(my.stock, my.control, fit.plusgroup=TRUE)
```

**Description**

Implements Pope’s Virtual Population Analysis (VPA).

The method returns an object of class `FLVPA` class. This extends the `FLAssess` class although the `FLVPA` class currently has the same slots as the `FLAssess` class and does not add new ones. Objects can be created by calls of the form `new("FLVPA", ...)` or by calling the `VPA` function.

The other arguments to `VPA` are

**fratio** A numeric with default "missing"

**fit.plusgroup** A Boolean with default TRUE

**desc** A character string which appended to the desc slot of the object

For the sake of speed, most of the calculations are carried out in C.

**Generic function**

VPA(stock, ...)

**Methods**

**signature(object=FLStock)** : Performs a VPA on the FLStock object. Other arguments are `fratio`, `fit.plusgroup` and `desc`, which are described above.

**Author(s)**

Laurence Kell

**See Also**

FLSepVPA

**Examples**

```
# use the ple4 data set
data(ple4)
ple4.test <- ple4
# Remove 0s and set as 1s
catch.n(ple4.test)[catch.n(ple4.test)==0] <- 1
# Remove harvest and stock.n values
stock.n(ple4.test)[] <- NA
harvest(ple4.test)[] <- NA
# Set Fs in final year and final ages
harvest(ple4.test)[,"2001"] <- harvest(ple4)[,"2001"]
harvest(ple4.test)[,"10",] <- harvest(ple4)[,"10",]
# Run the VPA
ple4.vpa <- VPA(ple4.test)
# Take a look at the harvest
plot(harvest(ple4.vpa))
```

---

Z-methods

*Calculates total mortality*

---

**Description**

This method calculates total mortality from the simple addition of `m` and `harvest`.

**Methods**

`Z(obj)` Takes in an object of type `FLStock` and carries out the above actions. Returns an object of type `FLQuant`

**Examples**

```
data(ple4)
Z(ple4)
```

# Index

- \* **classes**
  - FLAssess, 2
  - FLSepVPA.control-class, 4
- \* **datasets**
  - data, 2
- \* **hplot**
  - plot, 5
- \* **manip**
  - no.discards-methods, 5
  - Z-methods, 7
- \* **methods**
  - assess-methods, 2
  - FLAssess, 2
  - no.discards-methods, 5
  - plot, 5
  - SepVPA-methods, 6
  - VPA-methods, 6
  - Z-methods, 7
- \* **models**
  - SepVPA-methods, 6
  - VPA-methods, 6
- +, FLAssess, FLStock-method (FLAssess), 2
- +, FLStock, FLAssess-method (FLAssess), 2
- ass (data), 2
- assess (assess-methods), 2
- assess, FLAssess-method
  - (assess-methods), 2
- assess, FLSepVPA.control-method
  - (SepVPA-methods), 6
- assess-methods, 2
- bwplot, 5
- data, 2
- FLAssess, 2
- FLAssess, FLQuant-method (FLAssess), 2
- FLAssess, missing-method (FLAssess), 2
- FLAssess-class (FLAssess), 2
- FLAssess-methods (FLAssess), 2
- FLSepVPA.control, 4, 6
- FLSepVPA.control
  - (FLSepVPA.control-class), 4
- FLSepVPA.control-class, 4
- FLVPA-class (VPA-methods), 6
- harvest, FLAssess, missing-method
  - (FLAssess), 2
- index.hat (FLAssess), 2
- index.hat, FLAssess-method (FLAssess), 2
- index.hat<- (FLAssess), 2
- index.hat<-, FLAssess, FLQuants-method
  - (FLAssess), 2
- index.name (FLAssess), 2
- index.name, FLAssess-method (FLAssess), 2
- index.name<- (FLAssess), 2
- index.name<-, FLAssess, character-method
  - (FLAssess), 2
- index.range (FLAssess), 2
- index.range, FLAssess-method (FLAssess), 2
- index.range<- (FLAssess), 2
- index.range<-, FLAssess, list-method
  - (FLAssess), 2
- index.res (FLAssess), 2
- index.res, FLAssess-method (FLAssess), 2
- index.res<- (FLAssess), 2
- index.res<-, FLAssess, FLQuants-method
  - (FLAssess), 2
- lattice, 5
- merge, FLStock, FLAssess-method
  - (FLAssess), 2
- no.discards (no.discards-methods), 5
- no.discards, FLStock-method
  - (no.discards-methods), 5
- no.discards-methods, 5



plot, [3](#), [5](#), [5](#)  
plot, FLAssess, missing-method  
    ([FLAssess](#)), [2](#)  
plot, FLAssess-methods (plot), [5](#)  
  
SepVPA, [3](#)  
SepVPA (SepVPA-methods), [6](#)  
SepVPA, FLStock-method (SepVPA-methods),  
    [6](#)  
SepVPA-methods, [6](#)  
show, [3](#)  
show, FLAssess-method ([FLAssess](#)), [2](#)  
summary, [3](#)  
summary, FLAssess-method ([FLAssess](#)), [2](#)  
  
units, FLAssess-method ([FLAssess](#)), [2](#)  
units<-, FLAssess, list-method  
    ([FLAssess](#)), [2](#)  
update, [3](#)  
update, FLAssess-method ([FLAssess](#)), [2](#)  
  
VPA, [3](#), [6](#)  
VPA (VPA-methods), [6](#)  
VPA, FLStock-method (VPA-methods), [6](#)  
VPA-methods, [6](#)  
  
window, FLAssess-method ([FLAssess](#)), [2](#)  
  
xyplot, [5](#)  
  
Z (Z-methods), [7](#)  
Z, FLStock-method (Z-methods), [7](#)  
Z-methods, [7](#)