An Introduction to FLR

FLR Core Team

November 18, 2010

Outline

Outline

FLCore

What's inside FLCore?

FLCore

What's inside FLCore?

- FLQuant - basic class

FLCore

What's inside FLCore?

- FLQuant - basic class
- FLStock - fish stock

FLCore

What's inside FLCore?

- FLQuant - basic class
- FLStock - fish stock
- FLBiol - biological population

FLCore

What's inside FLCore?

- FLQuant - basic class
- FLStock - fish stock
- FLBiol - biological population
- FLModel - model class

FLCore

What's inside FLCore?

- FLQuant - basic class
- FLStock - fish stock
- FLBiol - biological population
- FLModel - model class
- FLSR - stock-recruitment relationships

FLCore

What's inside FLCore?

- FLQuant - basic class
- FLStock - fish stock
- FLBiol - biological population
- FLModel - model class
- FLSR - stock-recruitment relationships
- FLIndex - survey indices

FLCore

What's inside FLCore?

- FLQuant - basic class
- FLStock - fish stock
- FLBiol - biological population
- FLModel - model class
- FLSR - stock-recruitment relationships
- FLIndex - survey indices
- FLFleet - fleet (includes FLCatch and FLMetier)

FLCore

What's inside FLCore?

- FLQuant - basic class
- FLStock - fish stock
- FLBiol - biological population
- FLModel - model class
- FLSR - stock-recruitment relationships
- FLIndex - survey indices
- FLFleet - fleet (includes FLCatch and FLMetier)
- FLIst - list classes (FLFleets, FLStocks, FLIndices etc.)

FLQuant

Essentially a six dimensional array used to store data of a particular type (e.g. catch numbers).

Dimensions are:

FLQuant

Essentially a six dimensional array used to store data of a particular type (e.g. catch numbers).

Dimensions are:

1. User defined (age, length etc.)

FLQuant

Essentially a six dimensional array used to store data of a particular type (e.g. catch numbers).

Dimensions are:

1. User defined (age, length etc.)
2. Year

FLQuant

Essentially a six dimensional array used to store data of a particular type (e.g. catch numbers).

Dimensions are:

1. User defined (age, length etc.)
2. Year
3. Unit (substocks, male/female)

FLQuant

Essentially a six dimensional array used to store data of a particular type (e.g. catch numbers).

Dimensions are:

1. User defined (age, length etc.)
2. Year
3. Unit (substocks, male/female)
4. Season

FLQuant

Essentially a six dimensional array used to store data of a particular type (e.g. catch numbers).

Dimensions are:

1. User defined (age, length etc.)
2. Year
3. Unit (substocks, male/female)
4. Season
5. Area

FLQuant

Essentially a six dimensional array used to store data of a particular type (e.g. catch numbers).

Dimensions are:

1. User defined (age, length etc.)
2. Year
3. Unit (substocks, male/female)
4. Season
5. Area
6. Iter

FLQuant - example

```
> test.quant

An object of class "FLQuant"
, , unit = unique, season = all, area = unique

    year
age  1995   1996   1997   1998   1999   2000   2001
  1    7751   1104    892    196    549   2634   4509
  2   36575  42496  42855  30401   8689  15819  35886
  3   81398  64382  86948  68920 155971  39550  52480
  4   78370  46359  43669  56329  39857 164330  48238
  5   36499  32130  22541  16713  24112  14993  89949
  6   17953  14460  13518   6432   6829   9343   6836
  7    9772  10605   6362   4986   2783   2130   4418
  8    4366   4528   3632   2506   2246   1030   1127
  9    2336   2624   2179   1761   1521    940    637
  10   3753   4892   4181   3119   3093   2097   2309

units:  thousands

> dim(test.quant)

[1] 10  7  1  1  1  1
```

FLQuant - example contd.

```
> dimnames(test.quant)

$age
 [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10"

$year
[1] "1995" "1996" "1997" "1998" "1999" "2000" "2001"

$unit
[1] "unique"

$season
[1] "all"

$area
[1] "unique"

$iter
[1] "1"
```

FLStock

Represents a fish stock and comprises a number of slots.

```
> summary(ple4)

An object of class "FLStock"

Name: Plaice in IV
Description: Imported from a VPA file. ( N:\Projecten\ICES WG\Demersale werkgroep WGNSSK\2009\stock\ple-ns
Range:         min        max        pgroup        minyear        maxyear        minfbar        maxfbar
        1         10         10         1957         2008         2         6
Quant: age

catch         : [ 1 52 1 1 1 1 ], units =  tonnes
catch.n       : [ 10 52 1 1 1 1 ], units =  thousands
catch.wt      : [ 10 52 1 1 1 1 ], units =  kg
discards      : [ 1 52 1 1 1 1 ], units =  tonnes
discards.n    : [ 10 52 1 1 1 1 ], units =  thousands
discards.wt   : [ 10 52 1 1 1 1 ], units =  kg
landings      : [ 1 52 1 1 1 1 ], units =  tonnes
landings.n    : [ 10 52 1 1 1 1 ], units =  thousands
landings.wt   : [ 10 52 1 1 1 1 ], units =  kg
stock         : [ 1 52 1 1 1 1 ], units =  tonnes
stock.n       : [ 10 52 1 1 1 1 ], units =  thousands
stock.wt      : [ 10 52 1 1 1 1 ], units =  kg
m             : [ 10 52 1 1 1 1 ], units =  NA
mat           : [ 10 52 1 1 1 1 ], units =  NA
harvest       : [ 10 52 1 1 1 1 ], units =  f
harvest.spwn  : [ 10 52 1 1 1 1 ], units =  NA
m.spwn        : [ 10 52 1 1 1 1 ], units =  NA
```

FLBiol

Represents a biological population

```
> summary(test.biol)

An object of class "FLBiol"

Name: Plaice in IV
Description: Imported from a VPA file. ( N:\Projecten\ICES WG\Demersale werkgroep WGNSSK\2009\stock\ple-ns
Range:         min        max        pgroup        minyear        maxyear        minfbar        maxfbar
        1         10         10         1957         2008         2         6
Quant: age

n             : [ 10 52 1 1 1 1 ], units =  thousands
m             : [ 10 52 1 1 1 1 ], units =  NA
wt            : [ 10 52 1 1 1 1 ], units =  kg
fec           : [ 10 52 1 1 1 1 ], units =  NA
spwn          : [ 10 52 1 1 1 1 ], units =  NA
```

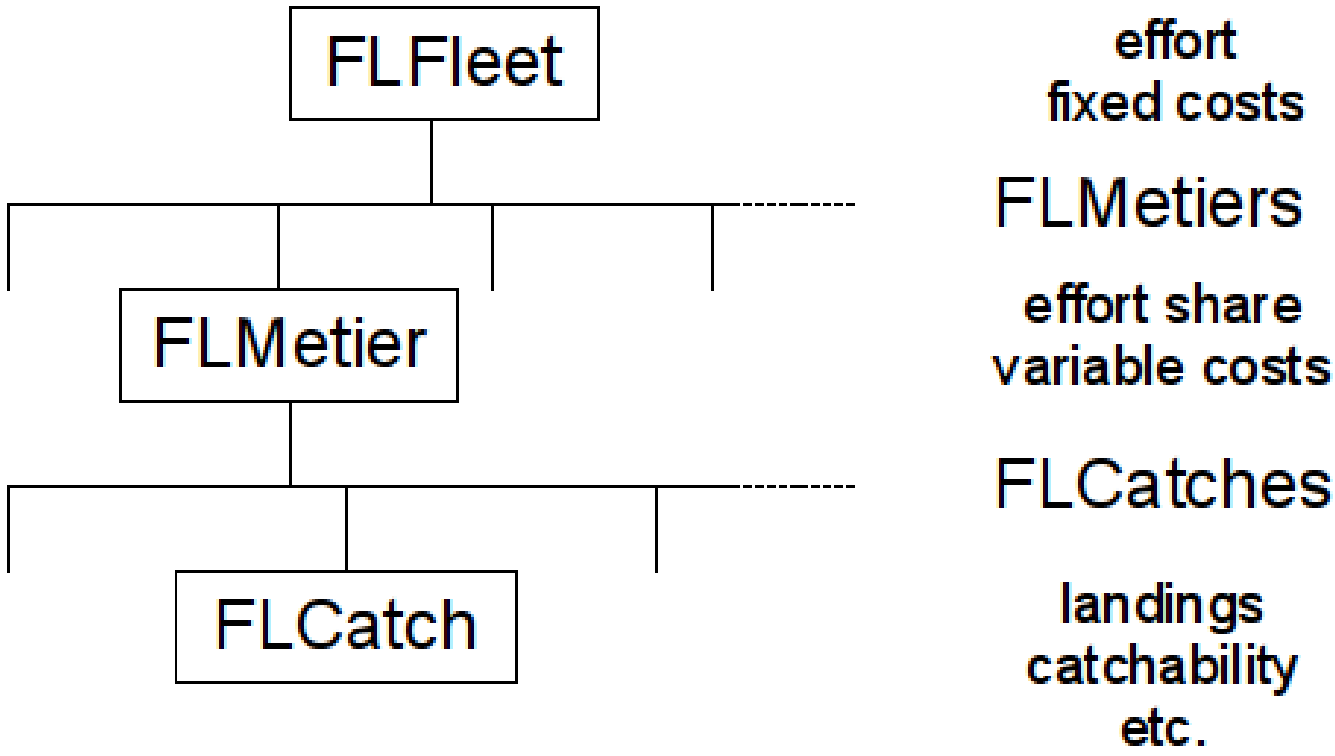
FLIndex

Represents a index (e.g. index of abundance from a survey)

```
> data(ple4.index)
> summary(ple4.index)

An object of class "FLIndex"

Name: BTS-Isis
Description: Plaice in IV . Imported from VPA file.
Range:         min         max        pgroup        minyear        maxyear        startf         endf
        1         8          NA        1985           2008           0.66           0.75
Type :
Distribution :
Quant: age

index          : [ 8 24 1 1 1 1 ], units =  NA
index.var      : [ 8 24 1 1 1 1 ], units =  NA
catch.n        : [ 8 24 1 1 1 1 ], units =  NA
catch.wt       : [ 8 24 1 1 1 1 ], units =  NA
effort         : [ 1 24 1 1 1 1 ], units =  NA
sel.pattern    : [ 8 24 1 1 1 1 ], units =  NA
index.q        : [ 8 24 1 1 1 1 ], units =  NA
```

FLFleet

A more complicated class with three levels: Fleet, Metier and Catch

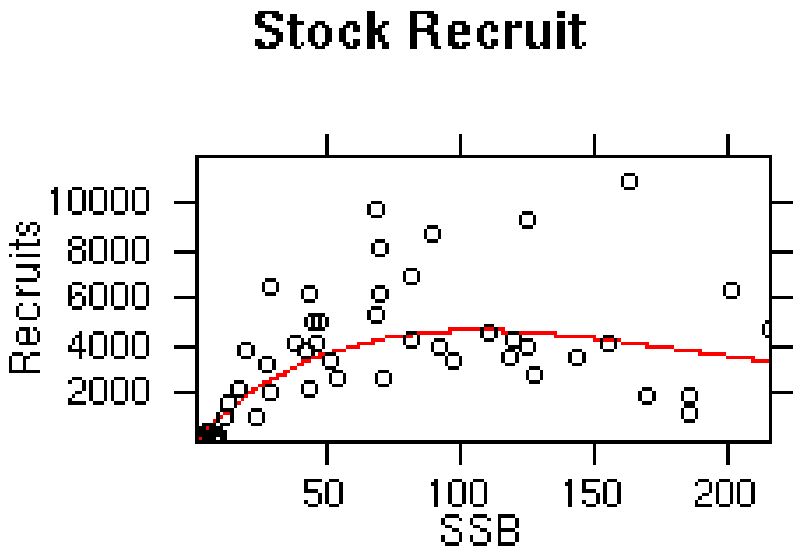


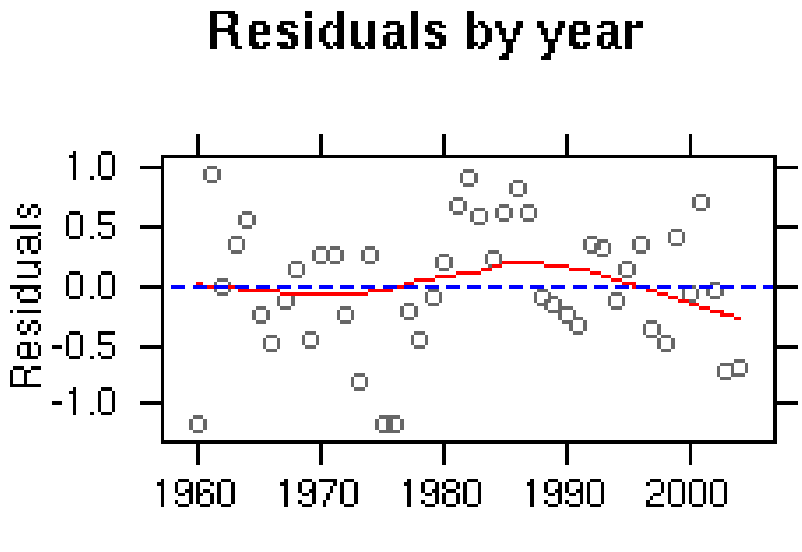
FLSR

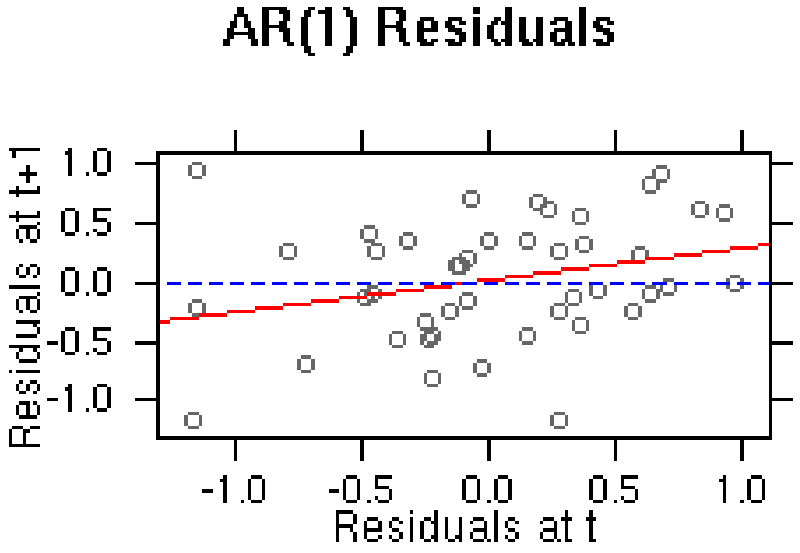
Class for fitting stock-recruitment relationships. Extends FLModel.

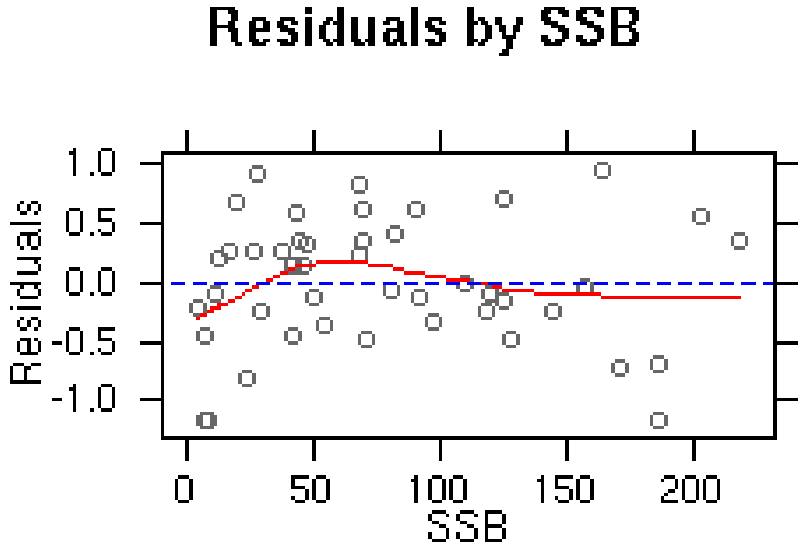
```
> data(nsher)
> summary(nsher)

An object of class "FLSR"

Name: Autumn spawning herring in IV, V  3/4/2005 14:46
Description: 'rec' and 'ssb' slots obtained from a 'FLStock' object
Range:

Quant: age

rec           : [ 1 45 1 1 1 1 ], units =  NA
ssb           : [ 1 45 1 1 1 1 ], units =  NA
residuals     : [ 1 45 1 1 1 1 ], units =  NA NA
fitted        : [ 1 45 1 1 1 1 ], units =  NA

Model:          rec ~ a * ssb * exp(-b * ssb)
<environment: 0x31944d0>
Parameters:
    params
iter     a        b
   1 119.4 0.009027

Log-likelihood:  16.352(0)
Variance-covariance:
               a            b
  a 258.66388793 1.838394e-02
  b   0.01838394 2.002586e-06
```

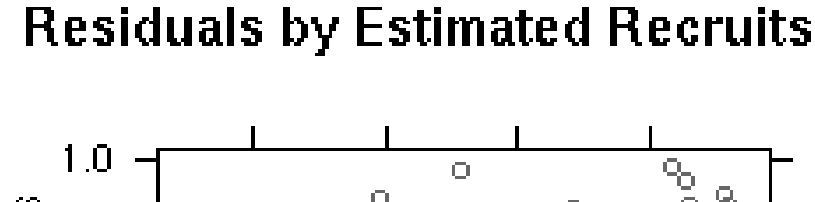
The FLSR plot

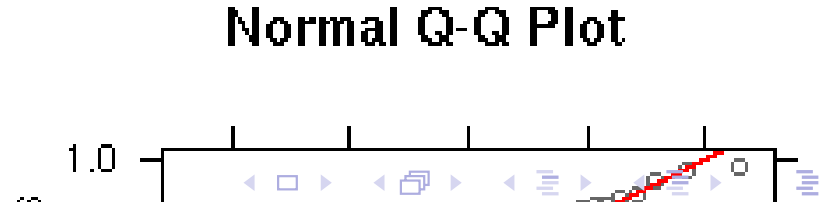












Slot accessors

Slot accessors

- Try to avoid using @ to access slots

Slot accessors

- Try to avoid using @ to access slots
- Use accessors instead

Slot accessors

- Try to avoid using @ to access slots
- Use accessors instead
- e.g. landings.n(stock) not stock@landings.n

Slot accessors

- Try to avoid using @ to access slots
- Use accessors instead
- e.g. landings.n(stock) not stock@landings.n
- Protects against internal changes

Slot accessors

- Try to avoid using @ to access slots
- Use accessors instead
- e.g. landings.n(stock) not stock@landings.n
- Protects against internal changes
- e.g. catch slots removed from FLCatch

Slot accessors

- Try to avoid using @ to access slots
- Use accessors instead
- e.g. landings.n(stock) not stock@landings.n
- Protects against internal changes
- e.g. catch slots removed from FLCatch
- But accessor catch() still works