# An Introduction to the FLCore package

FLR Core Team

November 30, 2010

# Outline



## Introduction

# Outline



## Introduction

## Basic classes

# Outline



## Introduction

## Basic classes

## Composite classes

# Outline



## Introduction

## Basic classes

## Composite classes

## List classes

# Outline



## Introduction

## Basic classes

## Composite classes

## List classes

## Model classes

# FLCore structure



Follows S4 paradigm with structured data implemented as classes and several methods to apply on objects of the classes.

# FLCore structure

Follows S4 paradigm with structured data implemented as classes and several methods to apply on objects of the classes.

- Classes - empty definitions of data

# FLCore structure



Follows S4 paradigm with structured data implemented as classes and several methods to apply on objects of the classes.

- Classes - empty definitions of data
- Objects - instances of the classes which have data following the class definition

# FLCore structure

Follows S4 paradigm with structured data implemented as classes and several methods to apply on objects of the classes.

- Classes - empty definitions of data
- Objects - instances of the classes which have data following the class definition
- Methods - implementation of actions to be executed on objects depending on its class

# Inside FLCore

# Inside FLCore



- Basic classes:
  FLArray, FLQuant, FLCohort, FLQuantPoint, FLPar

# Inside FLCore



- Basic classes:
  FLArray, FLQuant, FLCohort, FLQuantPoint, FLPar
- Composite classes:
  FLComp, FLBiol, FLCatch, FLFleet, FLIndex, FLMetier, FLModel, FLStock

# Inside FLCore



- Basic classes:
  FLArray, FLQuant, FLCohort, FLQuantPoint, FLPar
- Composite classes:
  FLComp, FLBiol, FLCatch, FLFleet, FLIndex, FLMetier, FLModel, FLStock
- Lists of classes:
  FLLst, FLBiols, FLCatches, FLCohorts, FLFleets, FLIndices, FLMetiers, FLQuants, FLStocks

# Inside FLCore



- Basic classes:
  FLArray, FLQuant, FLCohort, FLQuantPoint, FLPar
- Composite classes:
  FLComp, FLBiol, FLCatch, FLFleet, FLIndex, FLMetier, FLModel, FLStock
- Lists of classes:
  FLLst, FLBiols, FLCatches, FLCohorts, FLFleets, FLIndices, FLMetiers, FLQuants, FLStocks
- Model class:
  FLModel, FLGrowth, FLSR

# Inside FLCore



- Basic classes:
  FLArray, FLQuant, FLCohort, FLQuantPoint, FLPar
- Composite classes:
  FLComp, FLBiol, FLCatch, FLFleet, FLIndex, FLMetier, FLModel, FLStock
- Lists of classes:
  FLLst, FLBiols, FLCatches, FLCohorts, FLFleets, FLIndices, FLMetiers, FLQuants, FLStocks
- Model class:
  FLModel, FLGrowth, FLSR
- Methods

# Basic classes

| | parent | nSlots | virtual | child | distance |
|---|---|---|---|---|---|
| FLCohort | FLArray | 2 | FALSE | | |
| FLQuant | FLArray | 2 | FALSE | FLQuantPoint | 1.00 |
| FLQuantPoint | FLQuant | 2 | FALSE | | |

# FLQuant

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

Dimensions are:

# FLQuant

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

Dimensions are:

1. User defined (age, length etc.)

# FLQuant

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

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

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

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

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



```
> data(ple4)
> flq <- window(landings.n(ple4), start = 1995, end = 2001)
> dimnames(flq)

$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"
```

# FLQuant methods



```
> getClassMethods("FLQuant", "package:FLCore")

  [1] "apply"           "areaMeans"       "areaSums"        "areaVars"
  [5] "as.data.frame"   "as.FLQuant"      "barchart"        "[<-"
  [9] "["               "bubbles"         "bwplot"          "capacity<-"
 [13] "catch<-"         "catch.n<-"       "catch.q<-"       "catch.wt<-"
 [17] "coerce"          "crewshare<-"     "cv"              "dimMeans"
 [21] "dimnames<-"      "dims"            "dimSums"         "dimVars"
 [25] "discards<-"      "discards.n<-"    "discards.sel<-"  "discards.wt<-"
 [29] "dotplot"         "effort<-"        "effshare<-"      "E"
 [33] "fcost<-"         "fec<-"           "FLBiol"          "FLCatch"
 [37] "FLCohort"        "FLIndex"         "FLMetier"        "FLQuant"
 [41] "FLQuantPoint"    "FLStock"         "harvest<-"       "harvest.spwn<-"
 [45] "histogram"       "index<-"         "index.q<-"       "index.var<-"
 [49] "iter<-"          "iterMeans"       "iters"           "iterVars"
 [53] "jacknife"        "landings<-"      "landings.n<-"    "landings.sel<-"
 [57] "landings.wt<-"   "loglAR1"         "mat<-"           "m<-"
 [61] "m.spwn<-"        "names"           "n<-"             "plot"
 [65] "price<-"         "print"           "propagate"       "pv"
 [69] "quant<-"         "quant"           "quantile"        "quantMeans"
 [73] "quantSums"       "quantTotals"     "quantVars"       "rec<-"
 [77] "r"               "rlnorm"          "rnorm"           "rpois"
 [81] "rSq"             "seasonMeans"     "seasonSums"      "seasonVars"
 [85] "sel.pattern<-"   "setPlusGroup"    "sp"              "spr0"
 [89] "spwn<-"          "ssb<-"           "stock<-"         "stock.n<-"
 [93] "stock.wt<-"      "stripplot"       "sweep"           "unitMeans"
 [97] "units<-"         "units"           "unitSums"        "unitVars"
[101] "vcost<-"         "window"          "wt<-"            "xyplot"
[105] "yearMeans"       "yearSums"        "yearTotals"      "yearVars"
```

# FLQuantPoint

Six dimensional array used to summarize FLQuant objects.

```
> dimnames(FLQuantPoint(flq))

$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] "mean"   "median" "var"    "uppq"   "lowq"
```

# FLQuantPoint methods



```
> getClassMethods("FLQuantPoint", "package:FLCore")

 [1] "[<-"      "coerce"   "lowq<-"   "lowq"     "mean"     "mean<-"
 [7] "median<-" "median"   "plot"     "quantile" "rgamma"   "rlnorm"
[13] "rnorm"    "show"     "summary"  "uppq<-"   "uppq"     "var<-"
[19] "var"
```

# FLCohort

Six dimensional array used to store cohort data.

```
> dimnames(FLCohort(flq))

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

$cohort
 [1] "1985" "1986" "1987" "1988" "1989" "1990" "1991" "1992" "1993" "1994"
[11] "1995" "1996" "1997" "1998" "1999" "2000"

$unit
[1] "unique"

$season
[1] "all"

$area
[1] "unique"

$iter
[1] "1"
```

# FLCohort methods



```
> getClassMethods("FLCohort", "package:FLCore")

 [1] "bubbles"    "ccplot"     "coerce"     "dimnames<-" "dims"
 [6] "flc2flq"    "FLCohort"   "iter<-"     "plot"       "propagate"
[11] "xyplot"
```

# FLPar

A two dimensional array used to store parameter's data.

```
> dimnames(new("FLPar"))

$param
[1] ""

$iter
[1] "1"
```

# FLPar methods



```
> getClassMethods("FLPar", "package:FLCore")

 [1] "ab"            "Arith"         "as.data.frame" "[<-"
 [5] "["             "coerce"        "convertFLPar"  "densityplot"
 [9] "dims"          "FLPar"         "fmle"          "histogram"
[13] "iter<-"        "iter"          "mean"          "median"
[17] "names<-"       "names"         "params<-"      "plot"
[21] "propagate"     "show"          "splom"         "summary"
[25] "sv"            "sweep"         "units<-"       "units"
[29] "var"
```

# Composite classes

Classes that use FLQuant classes to define their slots.

| | parent | nSlots | virtual | child | distance |
|---|---|---|---|---|---|
| FLBiol | FLComp | 8 | FALSE | | |
| FLCatch | FLComp | 13 | FALSE | | |
| FLFleet | FLComp | 8 | FALSE | | |
| FLIndex | FLComp | 12 | FALSE | | |
| FLMetier | FLComp | 7 | FALSE | | |
| FLModel.1 | FLComp | 14 | FALSE | FLSR | 1.00 |
| FLModel.1.1 | FLComp | 14 | FALSE | FLGrowth | 1.00 |
| FLStock | FLComp | 20 | FALSE | | |

# FLStock

Represents a fish stock and comprises a number of slots.

```
> showClass("FLStock")

Class "FLStock" [package "FLCore"]

Slots:

Name:         catch      catch.n     catch.wt     discards   discards.n
Class:      FLQuant      FLQuant      FLQuant      FLQuant      FLQuant

Name:   discards.wt     landings   landings.n  landings.wt        stock
Class:      FLQuant      FLQuant      FLQuant      FLQuant      FLQuant

Name:       stock.n     stock.wt            m          mat      harvest
Class:      FLQuant      FLQuant      FLQuant      FLQuant      FLQuant

Name:  harvest.spwn       m.spwn         name         desc        range
Class:      FLQuant      FLQuant    character    character      numeric

Extends: "FLComp"
```

# FLStock example

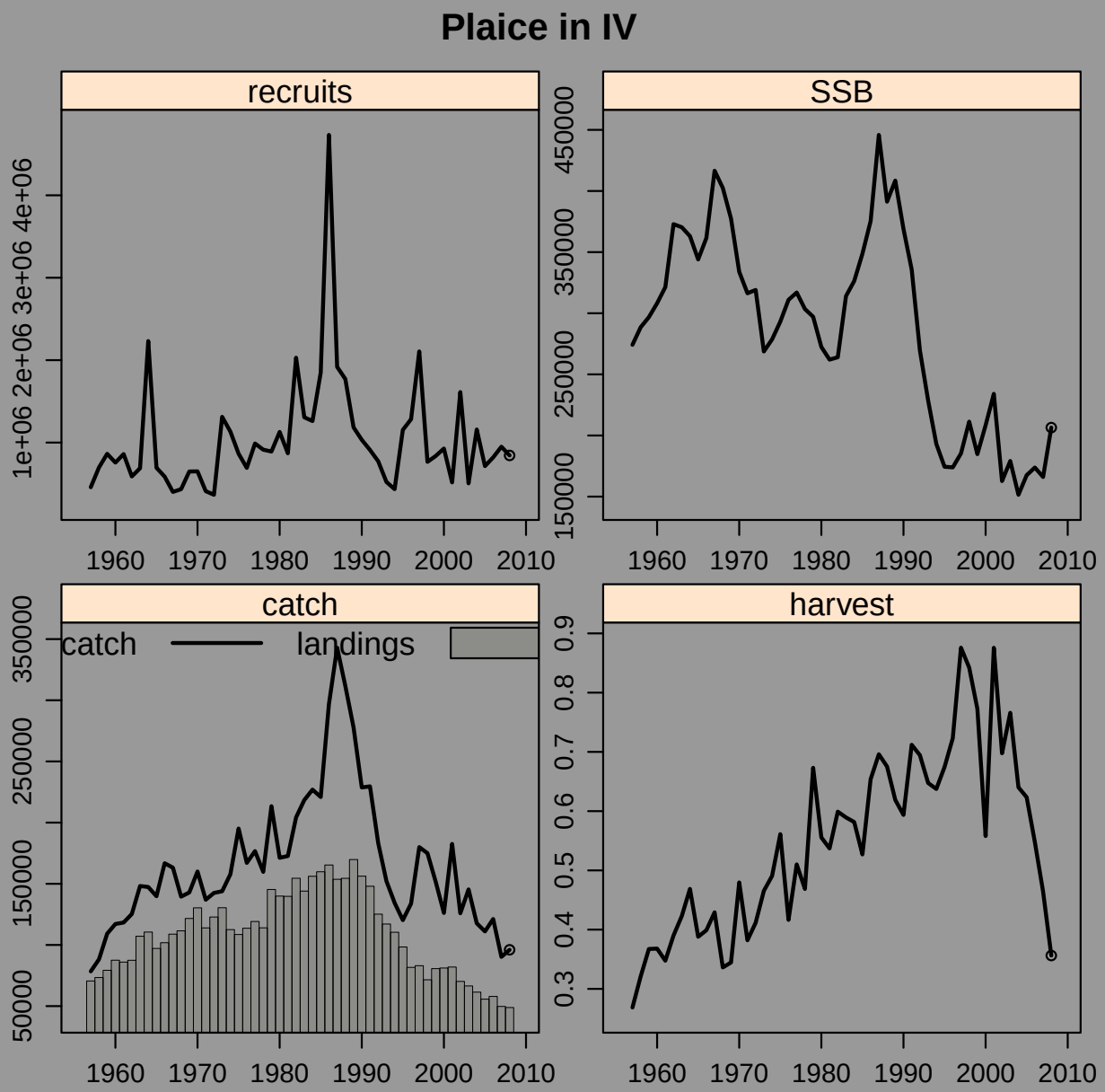


```
> summary(ple4)

An object of class "FLStock"

Name: Plaice in IV
Description: Imported from a VPA file. ( N:\Projecten\ICES WG\Demersale werkgroep WGNSSK\2009\st
Range:         min        max        pgroup        minyear        maxyear        minfbar
        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
```

# FLStock methods

```
> getClassMethods("FLStock", "package:FLCore")

 [1] "as.FLBiol"       "as.FLSR"         "[<-"             "["
 [5] "+"               "catch<-"         "catch"           "catch.n<-"
 [9] "catch.n"         "catch.wt<-"      "catch.wt"        "coerce"
[13] "computeCatch"    "computeDiscards" "computeLandings" "computeStock"
[17] "dimnames<-"      "dims"            "discards<-"      "discards"
[21] "discards.n<-"    "discards.n"      "discards.wt<-"   "discards.wt"
[25] "expand"          "fapex"           "fbar"            "harvest<-"
[29] "harvest"         "harvest.spwn<-"  "harvest.spwn"    "landings<-"
[33] "landings"        "landings.n<-"    "landings.n"      "landings.wt<-"
[37] "landings.wt"     "mat<-"           "mat"             "m<-"
[41] "m"               "m.spwn<-"        "m.spwn"          "name"
[45] "plot"            "qapply"          "range"           "rec"
[49] "r"               "setPlusGroup"    "sp"              "spr0"
[53] "ssb"             "ssbpurec"        "stock<-"         "stock"
[57] "stock.n<-"       "stock.n"         "stock.wt<-"      "stock.wt"
[61] "summary"         "survprob"        "trim"            "tsb"
```

# FLStock plot



# FLBiol

## Represents a biological population

```
> showClass("FLBiol")

Class "FLBiol" [package "FLCore"]

Slots:

Name:          n         m        wt       fec      spwn      name      desc
Class:   FLQuant   FLQuant   FLQuant   FLQuant   FLQuant character character

Name:      range
Class:   numeric

Extends: "FLComp"
```

# FLBiol example



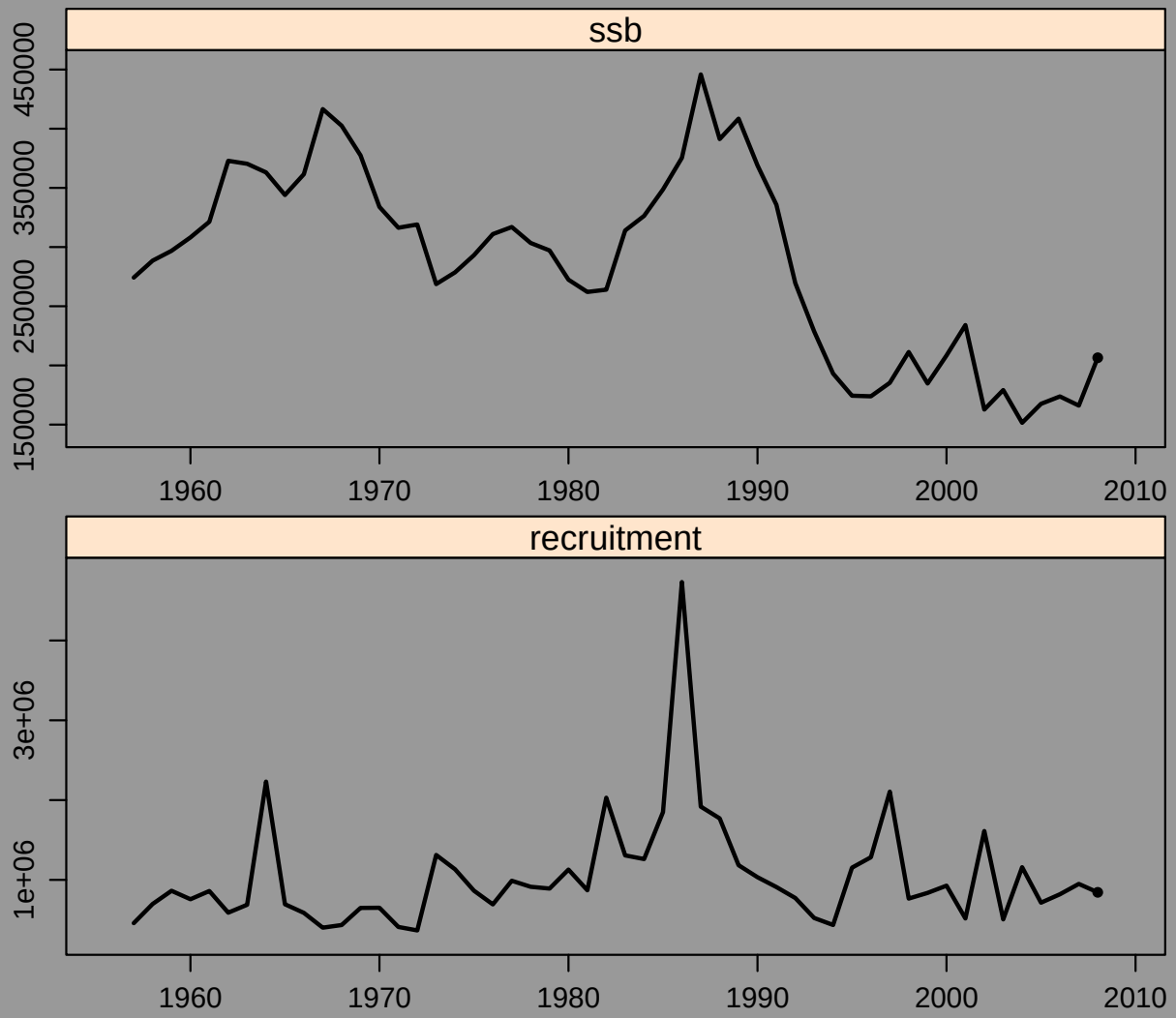
```
> summary(flbiol)

An object of class "FLBiol"

Name: Plaice in IV
Description: Imported from a VPA file. ( N:\Projecten\ICES WG\Demersale werkgroep WGNSSK\2009\st
Range:         min        max        pgroup       minyear        maxyear        minfbar
        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
```

# FLBiol methods

```
> getClassMethods("FLBiol", "package:FLCore")

 [1] "as.FLBiol"     "as.FLSR"       "catch.n"       "coerce"
 [5] "computeStock"  "fbar"          "fec<-"         "fec"
 [9] "harvest"       "leslie"        "mean.lifespan" "m<-"
[13] "m"             "n<-"           "n"             "plot"
[17] "rec"           "r"             "setPlusGroup"  "spwn<-"
[21] "spwn"          "ssb"           "ssn"           "summary"
[25] "survprob"      "tsb"           "wt<-"          "wt"
```

# FLBiol plot





# FLIndex

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

```
> showClass("FLIndex")

Class "FLIndex" [package "FLCore"]

Slots:

Name:          type distribution         index     index.var       catch.n
Class:    character    character      FLQuant       FLQuant       FLQuant

Name:      catch.wt       effort  sel.pattern       index.q          name
Class:      FLQuant      FLQuant      FLQuant       FLQuant     character

Name:          desc        range
Class:    character      numeric

Extends: "FLComp"
```

# FLIndex example

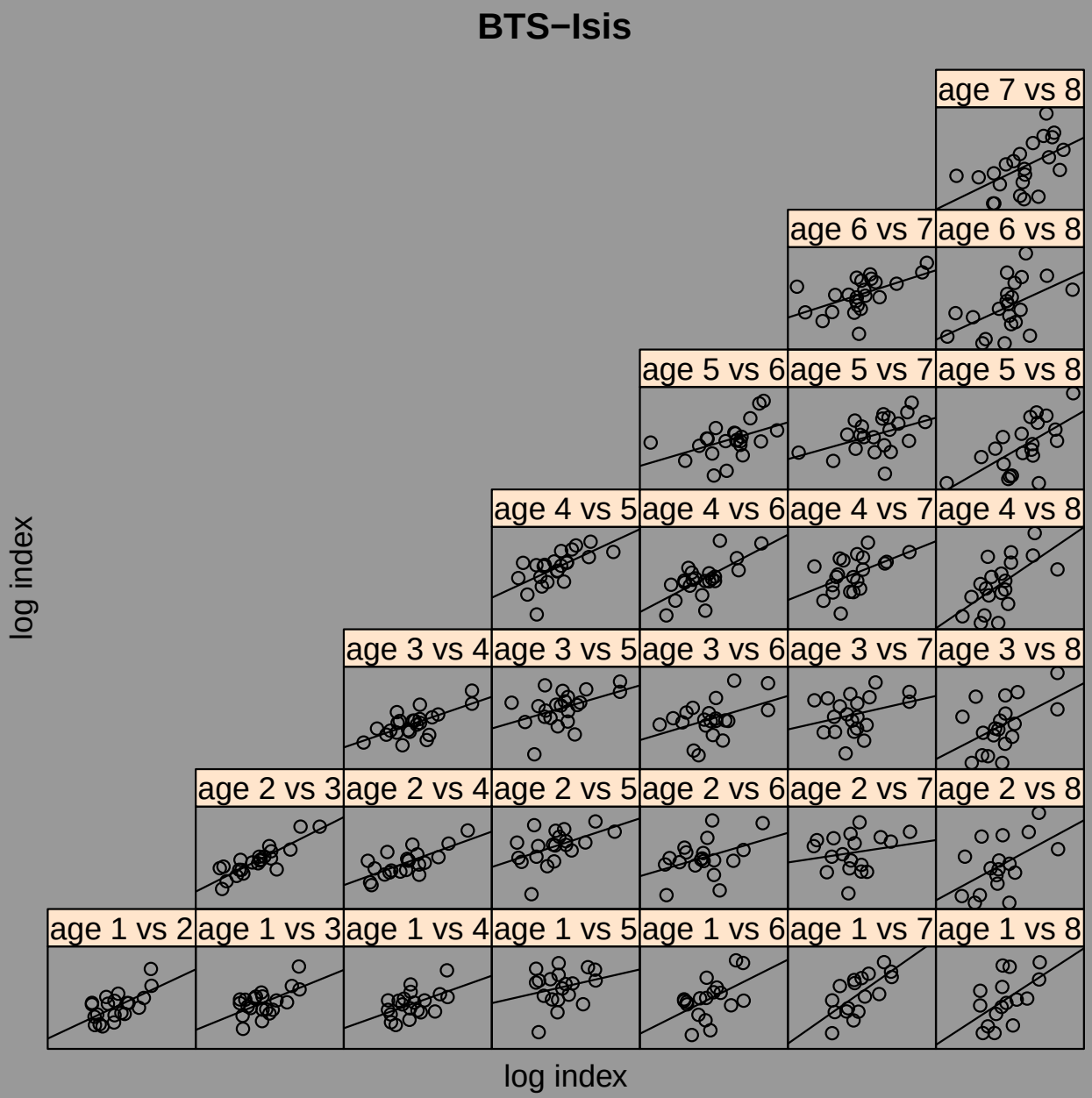


```
> summary(ple4.index)

An object of class "FLIndex"

Name: BTS-Isis
Description: Plaice in IV . Imported from VPA file.
Range:         min        max        pgroup        minyear        maxyear        startf        e
        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
```

# FLIndex methods

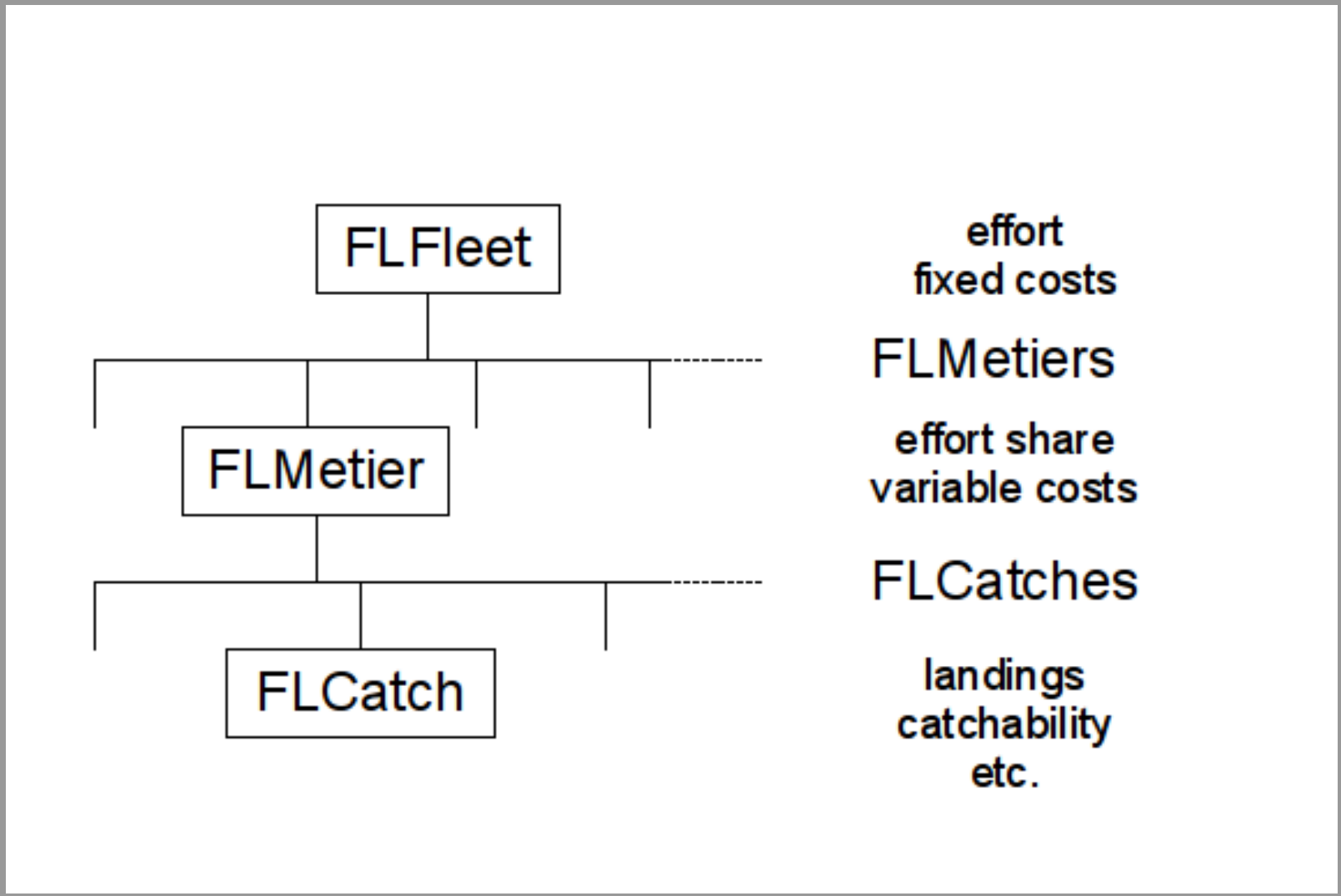
```
> getClassMethods("FLIndex", "package:FLCore")

 [1] "["             "catch.n<-"     "catch.n"       "catch.wt<-"
 [5] "catch.wt"      "coerce"        "computeCatch"  "dims"
 [9] "effort<-"      "effort"        "index<-"       "index"
[13] "index.q<-"     "index.q"       "index.var<-"   "index.var"
[17] "plot"          "sel.pattern<-" "sel.pattern"   "summary"
[21] "trim"          "type<-"        "type"
```

# FLBiol plot


BTS−Isis
age 7 vs 8
age 6 vs 7
age 6 vs 8
age 5 vs 6
age 5 vs 7
age 5 vs 8
age 4 vs 5
age 4 vs 6
age 4 vs 7
age 4 vs 8
age 3 vs 4
age 3 vs 5
age 3 vs 6
age 3 vs 7
age 3 vs 8
age 2 vs 3
age 2 vs 4
age 2 vs 5
age 2 vs 6
age 2 vs 7
age 2 vs 8
age 1 vs 2
age 1 vs 3
age 1 vs 4
age 1 vs 5
age 1 vs 6
age 1 vs 7
age 1 vs 8
log index
log index

# FLFleet

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



# FLFleet

```
> showClass("FLFleet")

Class "FLFleet" [package "FLCore"]

Slots:

Name:     effort     fcost  capacity crewshare   metiers      name      desc
Class:   FLQuant   FLQuant   FLQuant   FLQuant FLMetiers character character

Name:      range
Class:   numeric

Extends: "FLComp"
```

# FLFleet example



```
> summary(bt4)

An object of class "FLFleet"

Name: beam trawl fleet
Description: Example of an FLFleet
Range:         min        max        pgroup        minyear        maxyear
        0        0         NA        1957         2001
Quant: age

effort        : [ 1 45 1 1 1 1 ], units =  NA
fcost         : [ 1 45 1 1 1 1 ], units =  NA
capacity      : [ 1 45 1 1 1 1 ], units =  NA
crewshare     : [ 1 45 1 1 1 1 ], units =  NA

Metiers:
         TBB :
                ple : [ 15 45 1 1 1 1 ]
                sol : [ 10 45 1 1 1 1 ]
```

# FLFleet methods

```
> getClassMethods("FLFleet", "package:FLCore")

 [1] "as.data.frame"   "as.FLIndex"      "["               "[["
 [5] "capacity<-"      "capacity"        "catches"         "catch"
 [9] "catchNames"      "catch.n"         "catch.q<-"       "catch.q"
[13] "catch.sel"       "catch.wt"        "coerce"          "computeCatch"
[17] "computeDiscards" "computeLandings" "crewshare<-"     "crewshare"
[21] "dims"            "discards<-"      "discards"        "discards.n<-"
[25] "discards.n"      "discards.sel<-"  "discards.sel"    "discards.wt<-"
[29] "discards.wt"     "effort<-"        "effort"          "effshare"
[33] "fcost<-"         "fcost"           "FLFleet"         "iter"
[37] "landings<-"      "landings"        "landings.n<-"    "landings.n"
[41] "landings.sel<-"  "landings.sel"    "landings.wt<-"   "landings.wt"
[45] "metier<-"        "metier"          "metiers<-"       "metiers"
[49] "price<-"         "price"           "propagate"       "revenue"
[53] "summary"         "trim"            "vcost"           "window"
```

# List classes

| | parent | nSlots | virtual | child | distance |
|---|---|---|---|---|---|
| FLBiols | FLlst | 4 | FALSE | | |
| FLCatches | FLlst | 4 | FALSE | | |
| FLCohorts | FLlst | 4 | FALSE | | |
| FLFleets | FLlst | 4 | FALSE | | |
| FLIndices | FLlst | 4 | FALSE | | |
| FLMetiers | FLlst | 4 | FALSE | | |
| FLQuants | FLlst | 4 | FALSE | | |
| FLStocks | FLlst | 4 | FALSE | | |

# Model classes

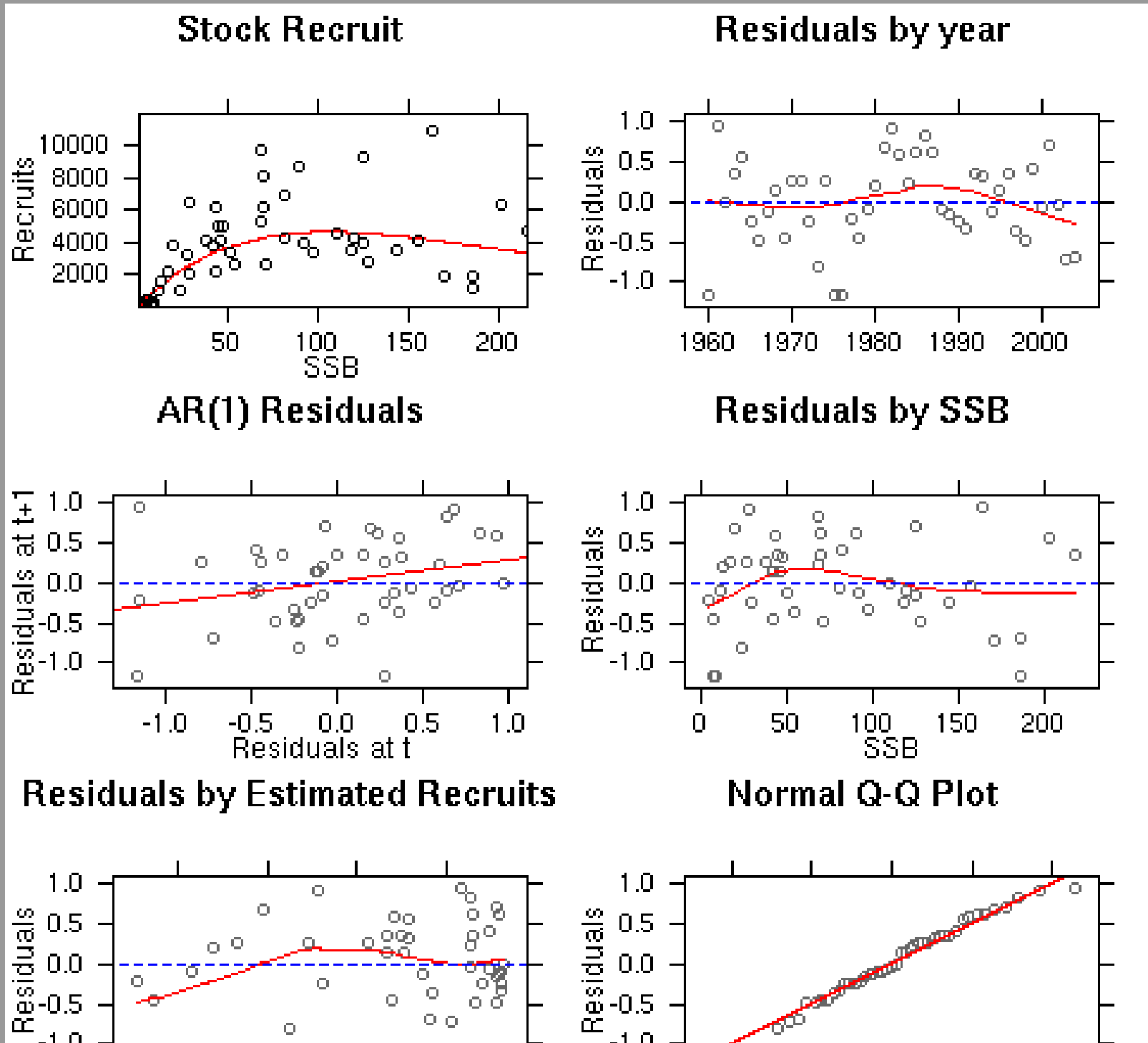


| | parent | nSlots | virtual | child | distance |
|---|---|---|---|---|---|
| FLGrowth | FLModel | 15 | FALSE | | |
| FLSR | FLModel | 18 | FALSE | | |

# 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: 0x560f3e0>
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
```


Stock Recruit
Recruits
SSB
Residuals by year
Residuals
AR(1) Residuals
Residuals at t+1
Residuals at t
Residuals by SSB
Residuals by Estimated Recruits
Normal Q-Q 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