a4a stock assessment framework
DRAFT

Ernesto Jardim[1], Colin Millar[1], and Finlay Scott[1]

[1]European Commission, Joint Research Centre, IPSC / Maritime Affairs Unit, 21027 Ispra (VA), Italy

[*]Corresponding author ernesto.jardim@jrc.ec.europa.eu

April 9, 2014

Contents

1 Running assessments

There are two basic types of assessments available from using `a4a`: the management procedure (MP) fit and the full assessment fit. The MP fit does not compute estimates of covariances and is therefore quicker to execute, while the full assessment fit returns parameter estimates and their covariances and hence retains the ability to simulate from the model at the expense of longer fitting time.

In the `a4a` assessment model, the model structure is defined by submodels. These are models for the different parts of a statistical catch at age model that requires structural assumptions, such as the selectivity of the fishing fleet, or how F-at-age changes over time. It is advantageous to write the model for F-at-age and survey catchability as linear models (by working with log F and log Q) becuase it allows us to use the linear modelling tools available in R: see for example gam formulas, or factorial design formulas using lm. In R's linear modelling lanquage, a constant model is coded as ~ 1, while a slope over age would simply be $\sim$ age. Extending this we can write a traditional year / age seperable F model like $\sim$ factor(age) + factor(year).

There are effectively 5 submodels in operation: the model for F-at-age, a model for initial age structure, a model for recruitment, a (list) of model(s) for survey catchability-at-age, and a list of models for the observation variance of catch.n and the survey indices. In practice, we fix the variance models and the initial age structure models, but in theory these can be changed. A basic set of submodels would be

```
fmodel <- ~factor(age) + factor(year)
qmodel <- list(~factor(age))
```

```
library(FLa4a)

## Loading required package:  Matrix
## Loading required package:  copula
## Loading required package:  triangle
## Loading required package:  mgcv
## Loading required package:  nlme
## This is mgcv 1.7-28.  For overview type 'help("mgcv-package")'.
## Loading required package:  lattice
## Loading required package:  latticeExtra
## Loading required package:  RColorBrewer
## Loading required package:  splines
## Loading required package:  np
## Loading required package:  boot
##
## Attaching package:  'boot'
##
## The following object is masked from 'package:lattice':
##
##    melanoma
##
## Loading required package:  cubature
## Nonparametric Kernel Methods for Mixed Datatypes (version 0.50-1)
## [vignette("np_faq",package="np") provides answers to frequently asked questions]
## Loading required package:  FLCore
## Loading required package:  grid
## Loading required package:  MASS
## FLCore 2.5.0 development version
##
##
## Attaching package:  'FLCore'
##
## The following object is masked from 'package:Matrix':
##
```

```
##     expand
##
## The following objects are masked from 'package:base':
##
##     cbind, rbind
##
## This is FLa4a 0.82.0.  For overview type 'help("FLa4a-package")'
##
##
## Attaching package:  'FLa4a'
##
## The following object is masked from 'package:lattice':
##
##     qqmath

library(diagram)

## Loading required package:  shape

data(ple4)
data(ple4.indices)
source("funs.R")
```

1.1 Quick and dirty

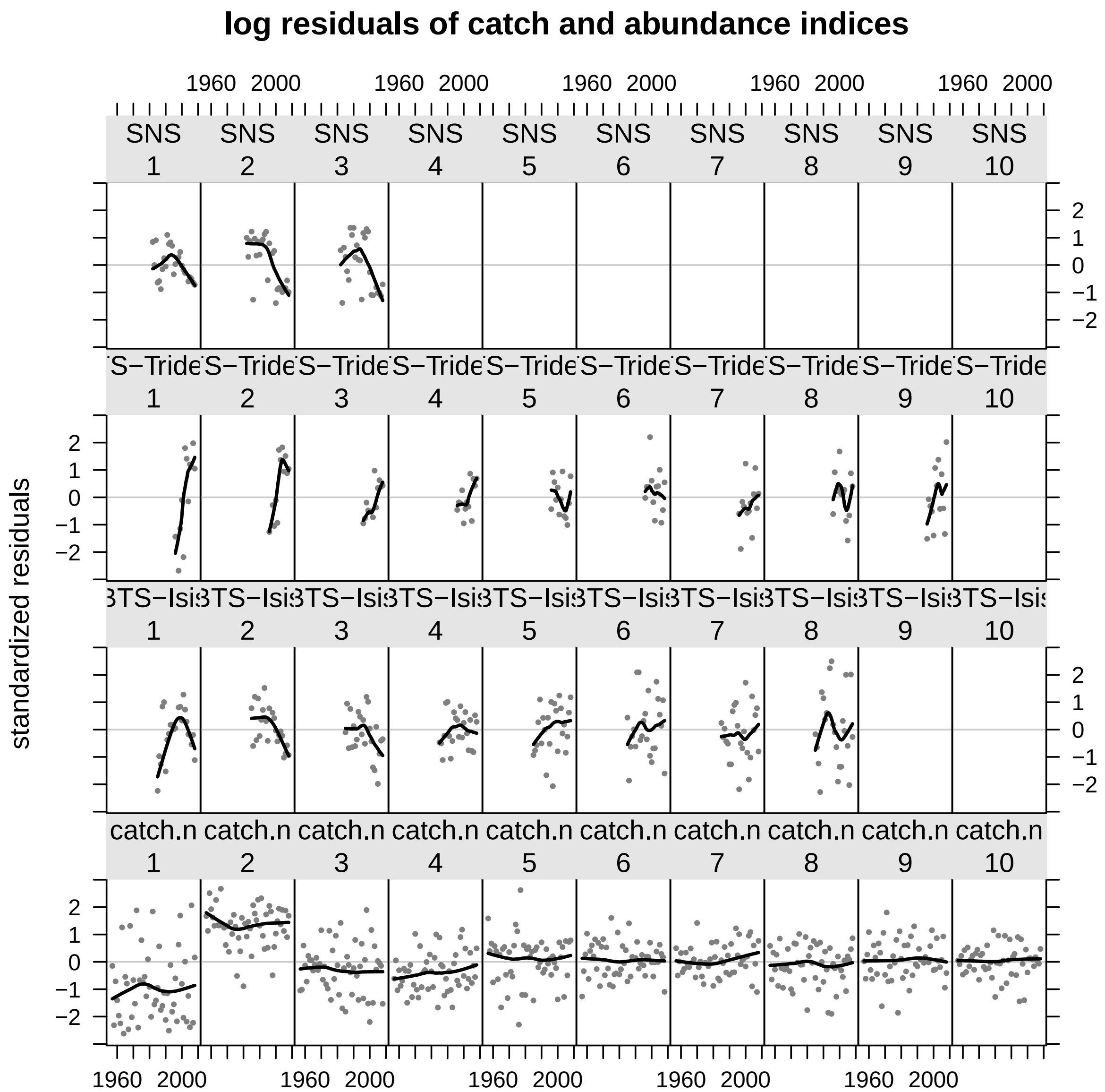
The default settings of the stock assessment model work reasoably well. It's an area of research that will improve with time.

```
data(ple4)
data(ple4.indices)
fit <- sca(ple4, ple4.indices)

## Note:  The following observations are treated as being missing at random:
##        fleet year age
##      BTS-Isis 1997  1
##      BTS-Isis 1997  2
## BTS-Tridens 1997  1
## BTS-Tridens 1997  2
##          SNS 1997  1
##          SNS 1997  2
##          SNS 2003  1
##          SNS 2003  2
##          SNS 2003  3
##       Predictions will be made for missing observations.
```

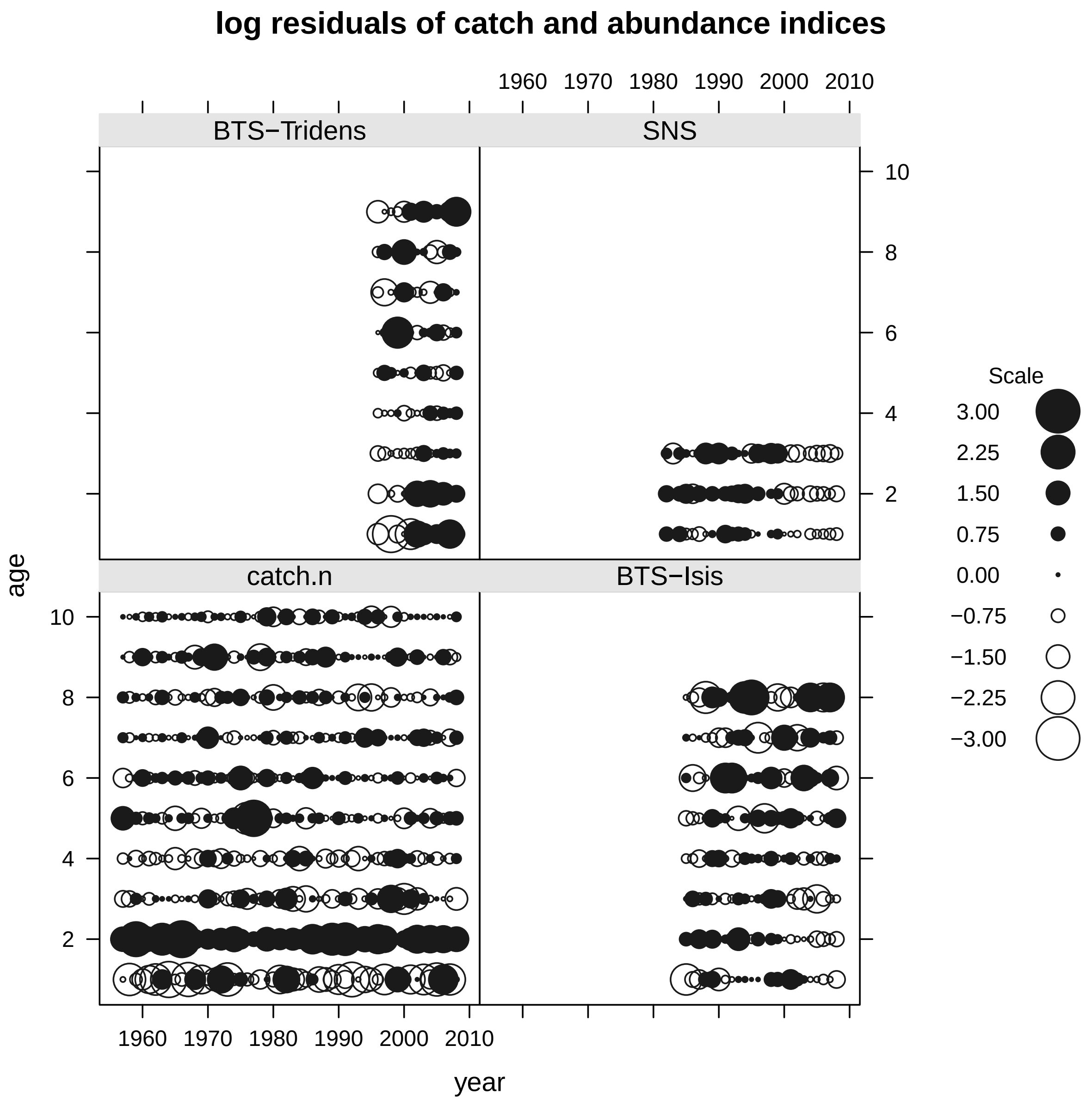
Note that because the survey index for plaice has missing values we get a warning saying that we assume these values are missing at random, and not because the observations were zero.

```
res <- residuals(fit, ple4, ple4.indices)
```

```
plot(res, main = "Residuals")
```


log residuals of catch and abundance indices
standardized residuals
SNS 1 SNS 2 SNS 3 SNS 4 SNS 5 SNS 6 SNS 7 SNS 8 SNS 9 SNS 10
catch.n 1 catch.n 2 catch.n 3 catch.n 4 catch.n 5 catch.n 6 catch.n 7 catch.n 8 catch.n 9 catch.n 10
1960 2000

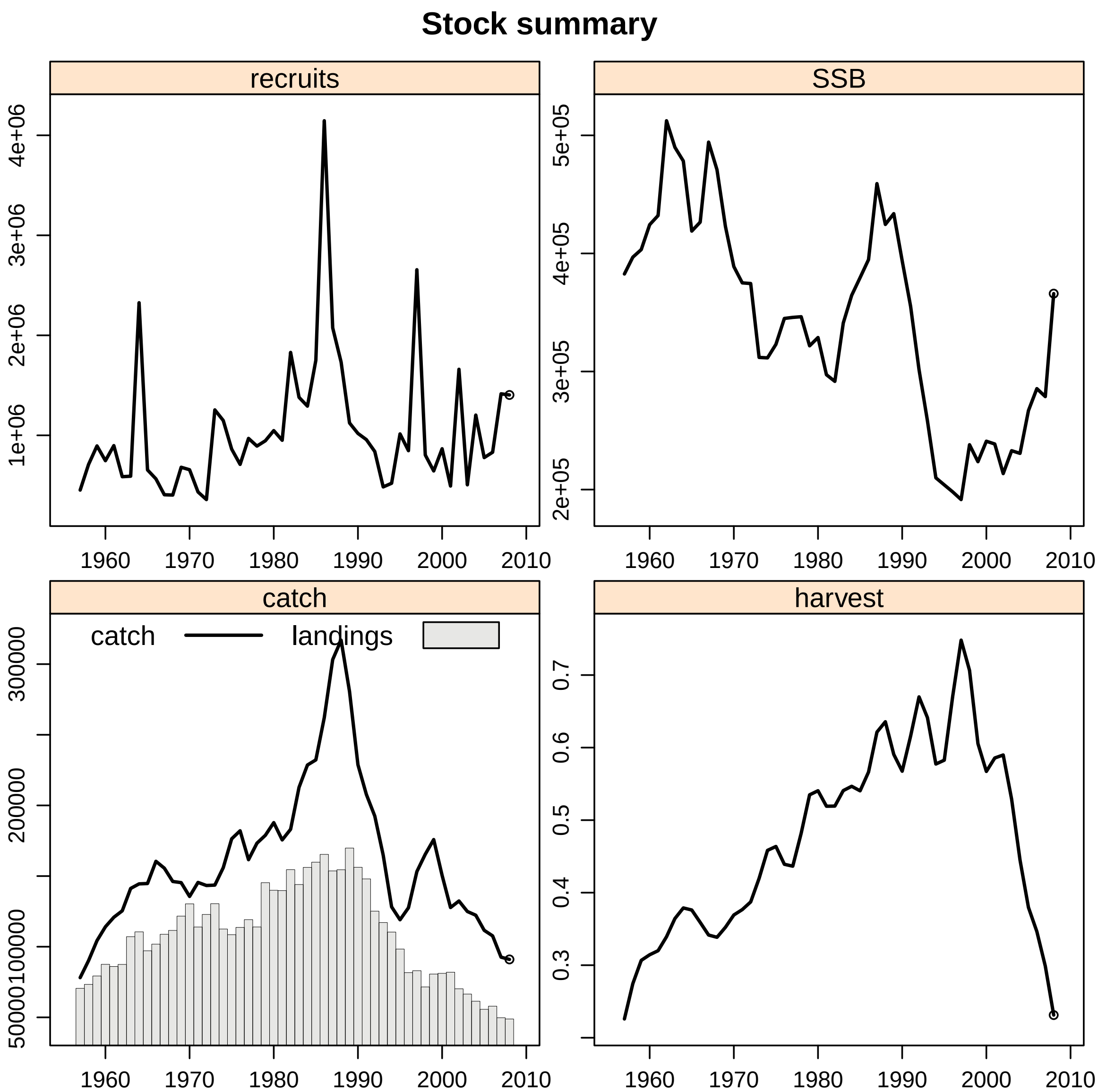
```
bubbles(res)
```



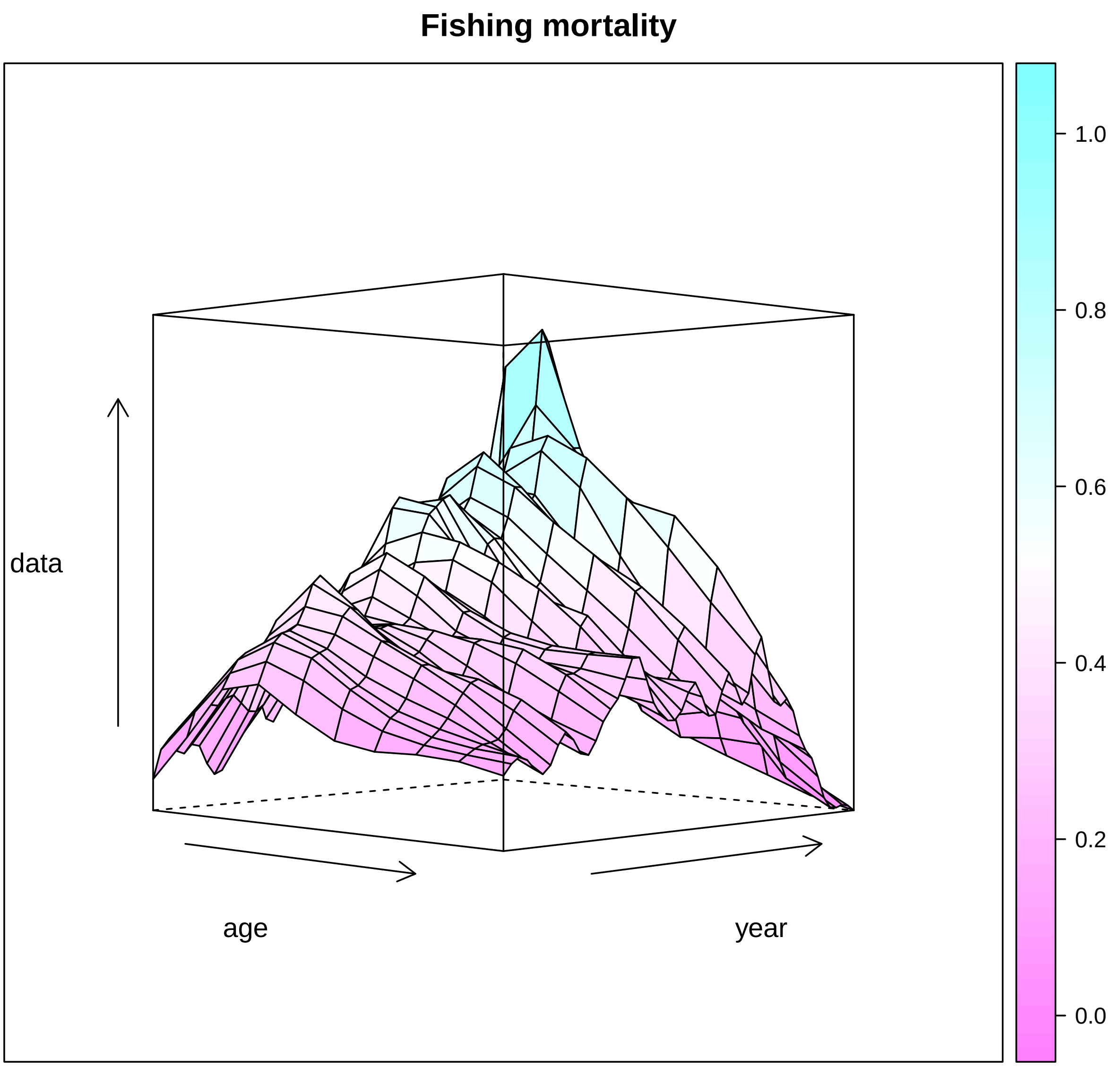

```
qqmath(res)

## Error:  unable to find an inherited method for function 'qqmath' for signature '"formula",
"data.frame"'
```

We can inspect the summaries from this fit my adding it to the origional stock object, for example to see the fitted fbar we can do

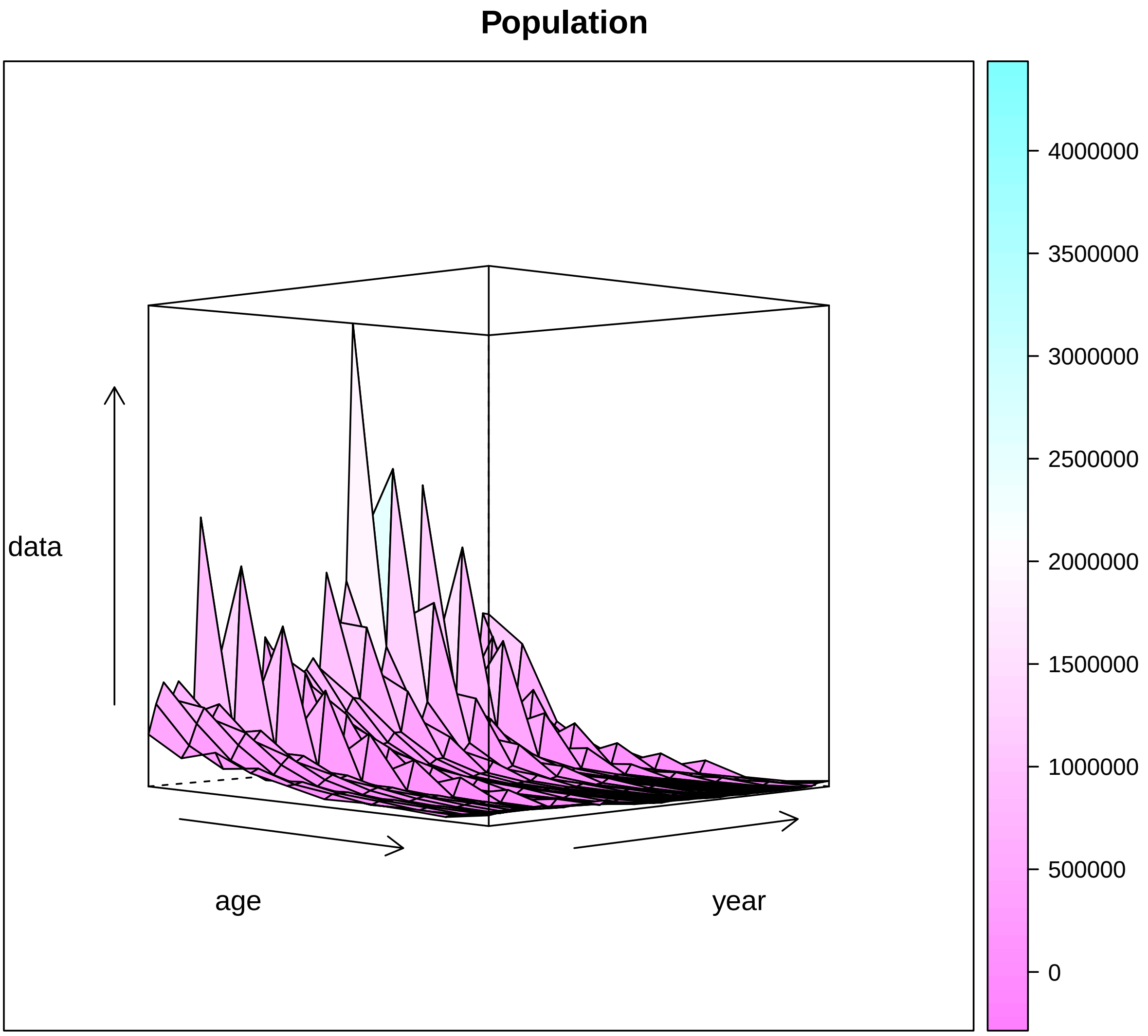
```
stk <- ple4 + fit
plot(stk, main = "Stock summary")

## Warning:  invalid factor level, NA generated
```



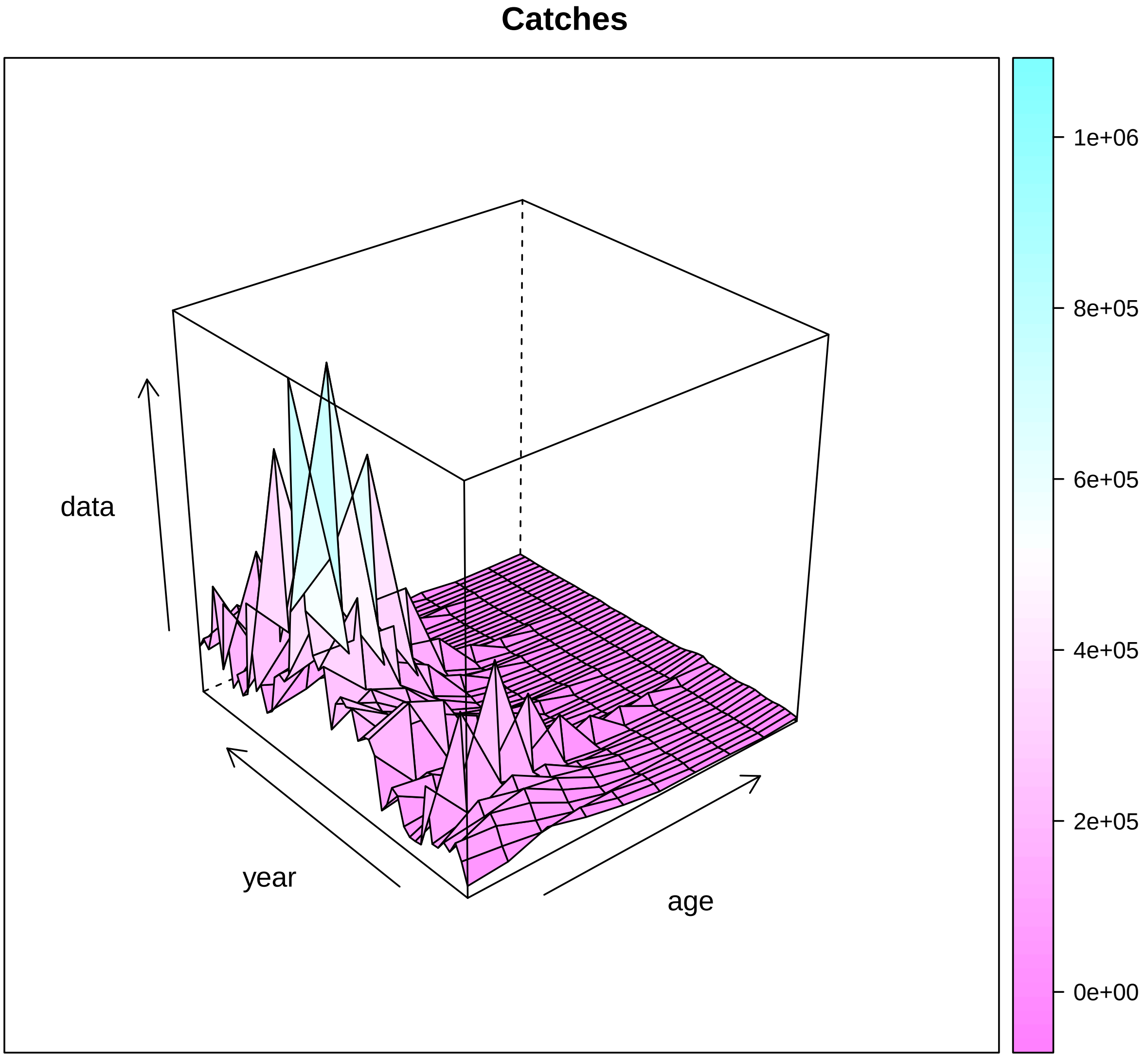

```
wireframe(data ~ age + year, data = as.data.frame(harvest(stk)), drape = TRUE,
    main = "Fishing mortality", screen = list(x = -90, y = -45))
```




```
wireframe(data ~ age + year, data = as.data.frame(stock.n(stk)), drape = TRUE,
    main = "Population", screen = list(x = -90, y = -45))
```




```
wireframe(data ~ age + year, data = as.data.frame(catch.n(stk)), drape = TRUE,
    main = "Catches")
```

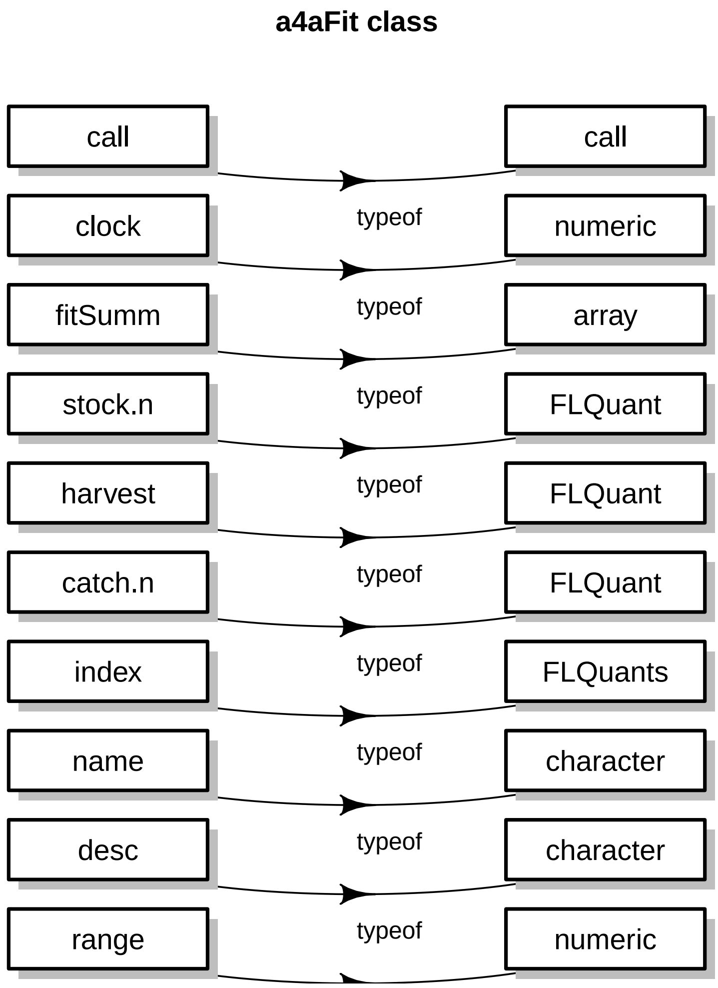


1.2 Data structures

The basic model output is contained in the `a4aFit` class. This object contains only the fitted values.

```
showClass("a4aFit")

## Class "a4aFit" [package "FLa4a"]
##
## Slots:
##
## Name:        call     clock   fitSumm   stock.n   harvest   catch.n
## Class:       call   numeric     array   FLQuant   FLQuant   FLQuant
##
## Name:       index      name      desc     range
## Class:  FLQuants character character   numeric
##
## Extends: "FLComp"
##
## Known Subclasses: "a4aFitSA"
```

```r
plotS4("a4aFit", main = "a4aFit class", lwd = 1, box.lwd = 2, cex.txt = 0.8,
    box.size = 0.1, box.type = "square", box.prop = 0.3)
```



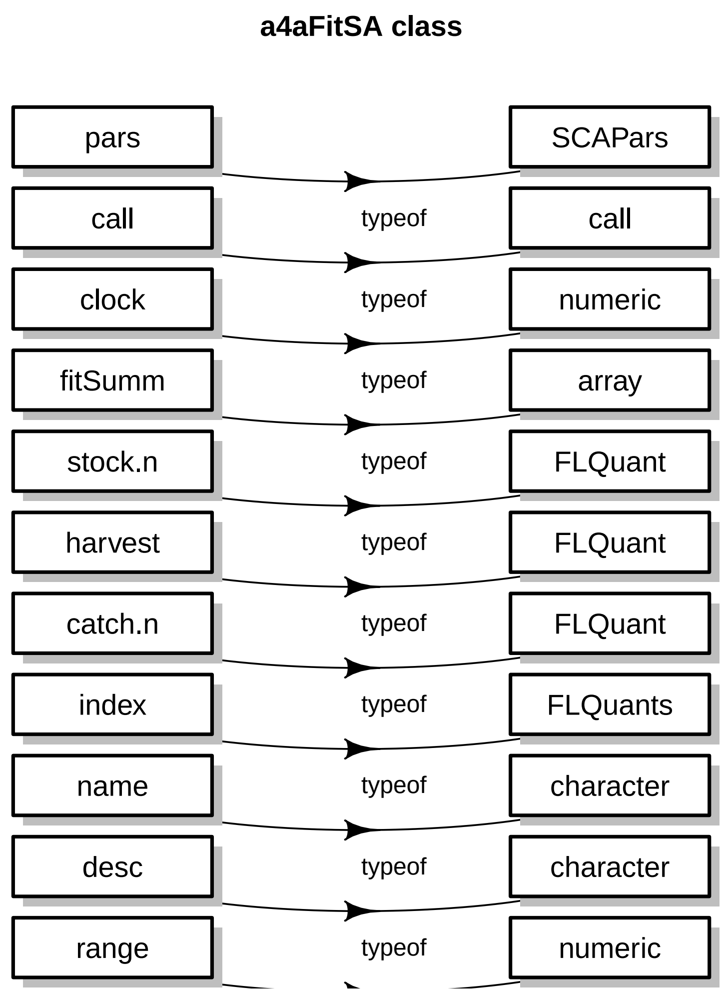
Fitted values are stored in the `stock.n`, `harvest`, `catch.n` and `index` slots. It also contains information carried over from the stock object used to fit the model: the name of the stock in `name`, any description provided in `desc` and the age and year range and mean F range in `range`. There is also a wall clock that has a breakdown of the time taken o run the model.

The full assessment fit returns an object of `a4aFitSA` class:

```r
showClass("a4aFitSA")

## Class "a4aFitSA" [package "FLa4a"]
##
## Slots:
##
## Name:       pars      call     clock    fitSumm   stock.n   harvest
## Class:   SCAPars      call   numeric      array   FLQuant   FLQuant
##
```

```
## Name:     catch.n      index       name       desc      range
## Class:    FLQuant  FLQuants character character   numeric
##
## Extends:
## Class "a4aFit", directly
## Class "FLComp", by class "a4aFit", distance 2
```

```
plotS4("a4aFitSA", main = "a4aFitSA class", lwd = 1, box.lwd = 2, cex.txt = 0.8,
    box.size = 0.1, box.type = "square", box.prop = 0.3)
```



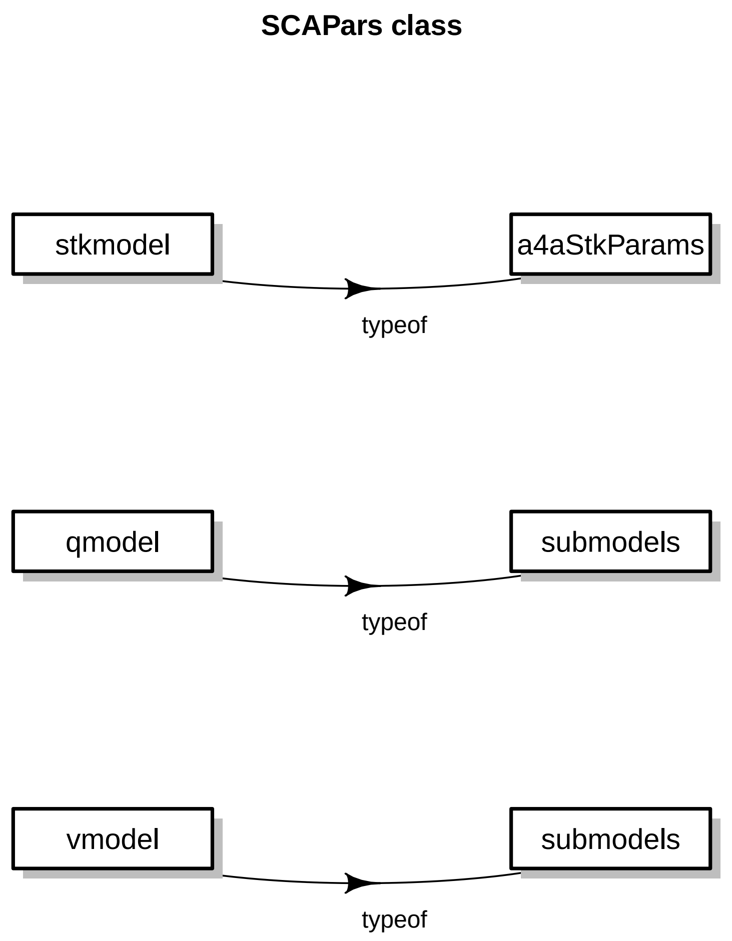
The additional slots in the assessment output is the `fitSumm` and `pars` slots which are containers for model summaries and the model parameters. The `pars` slot is a class of type `SCAPars` which is itself composed of sub-classes, designed to contain the information necessary to simulate from the model.

```
showClass("SCAPars")
```

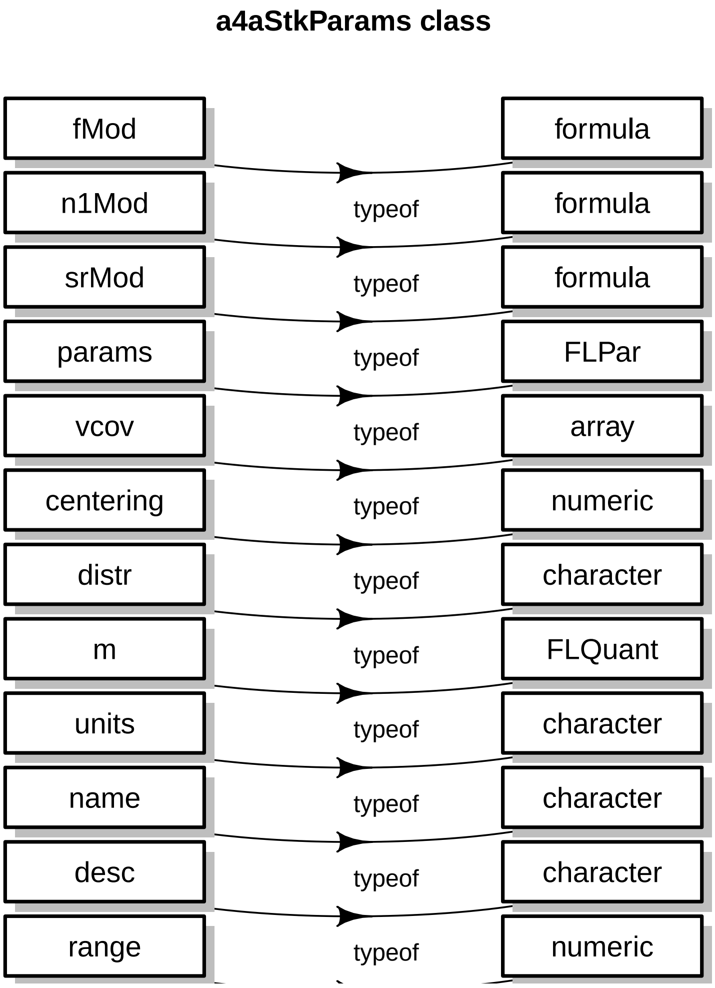
```
## Class "SCAPars" [package "FLa4a"]
##
```

```
## Slots:
##
## Name:      stkmodel       qmodel       vmodel
## Class: a4aStkParams    submodels    submodels

showClass("a4aStkParams")

## Class "a4aStkParams" [package "FLa4a"]
##
## Slots:
##
## Name:       fMod     n1Mod     srMod    params      vcov centering
## Class:   formula   formula   formula     FLPar     array   numeric
##
## Name:      distr         m     units      name      desc     range
## Class: character   FLQuant character character character   numeric
##
## Extends: "FLComp"

showClass("submodel")

## Class "submodel" [package "FLa4a"]
##
## Slots:
##
## Name:        Mod    params      vcov centering     distr      name
## Class:   formula     FLPar     array   numeric character character
##
## Name:       desc     range
## Class: character   numeric
##
## Extends: "FLComp"
```

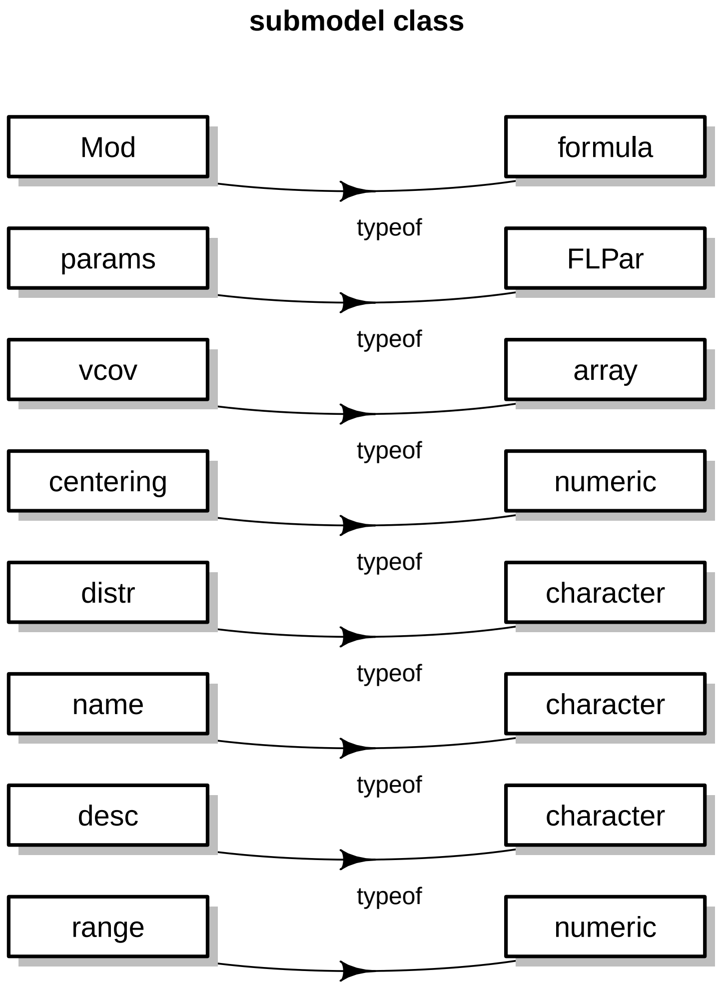
```
plotS4("SCAPars", main = "SCAPars class", lwd = 1, box.lwd = 2, cex.txt = 0.8,
    box.size = 0.1, box.type = "square", box.prop = 0.3)
```




```
plotS4("a4aStkParams", main = "a4aStkParams class", lwd = 1, box.lwd = 2, cex.txt = 0.8,
    box.size = 0.1, box.type = "square", box.prop = 0.3)
```




```
plotS4("submodel", main = "submodel class", lwd = 1, box.lwd = 2, cex.txt = 0.8,
    box.size = 0.1, box.type = "square", box.prop = 0.3)
```



for example, all the parameters required so simulate a time-series of mean F trends is contained in the `stkmodel` slot, which is a class of type `a4aStkParams`. This class contains the relevant submodels (see later), their parameters `params` and the joint covariance matrix `vcov` for all stock related parameters.

1.3 The sca method - statistical catch-at-age

We will now take a look at some examples for F models and the forms that we can get. Lets start with a separable model in which we model selectivity at age as an (unpenalised) thin plate spline. We will use the North Sea Plaice data again, and since this has 10 ages we will use a simple rule of thumb that the spline should have fewer than $\frac{10}{2} = 5$ degrees of freedom, and so we opt for 4 degrees of freedom. We will also do the same for year and model the change in F through time as a smoother with 20 degrees of freedom.

Lets now investigate some variations in the selectivity shape with time, but only a little... we can do this by adding a smooth interaction term in the fmodel

A further move is to free up the Fs to vary more over time

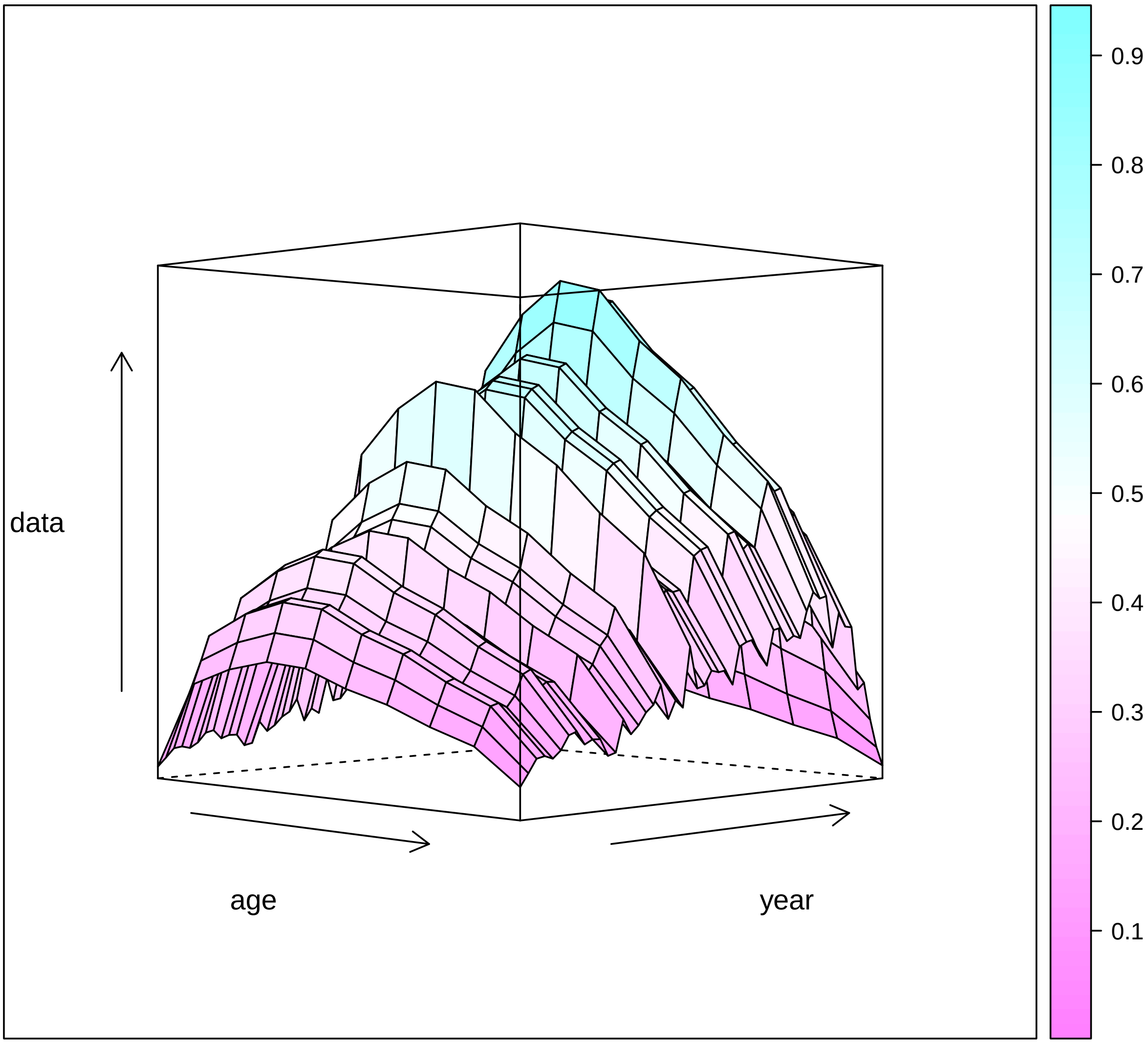
In the last examples the Fs are linked across age and time. What if we want to free up a specific age class because in the residuals we see a consistent pattern. This can happen, for example, if the spatial distribution of juvenilles is disconnected to the distribution of adults. The fishery focuses on the adult fish, and therefore the the F on young fish is a function of the distribution of the juveniles and could deserve a seperate model. This can be achieved by

Please note that each of these model *structures* lets say, have not been tuned to the data. The degrees of freedom of each model can be better tuned to the data by using model selection procedures such as AIC or BIC.

1.3.1 Fishing mortality submodel

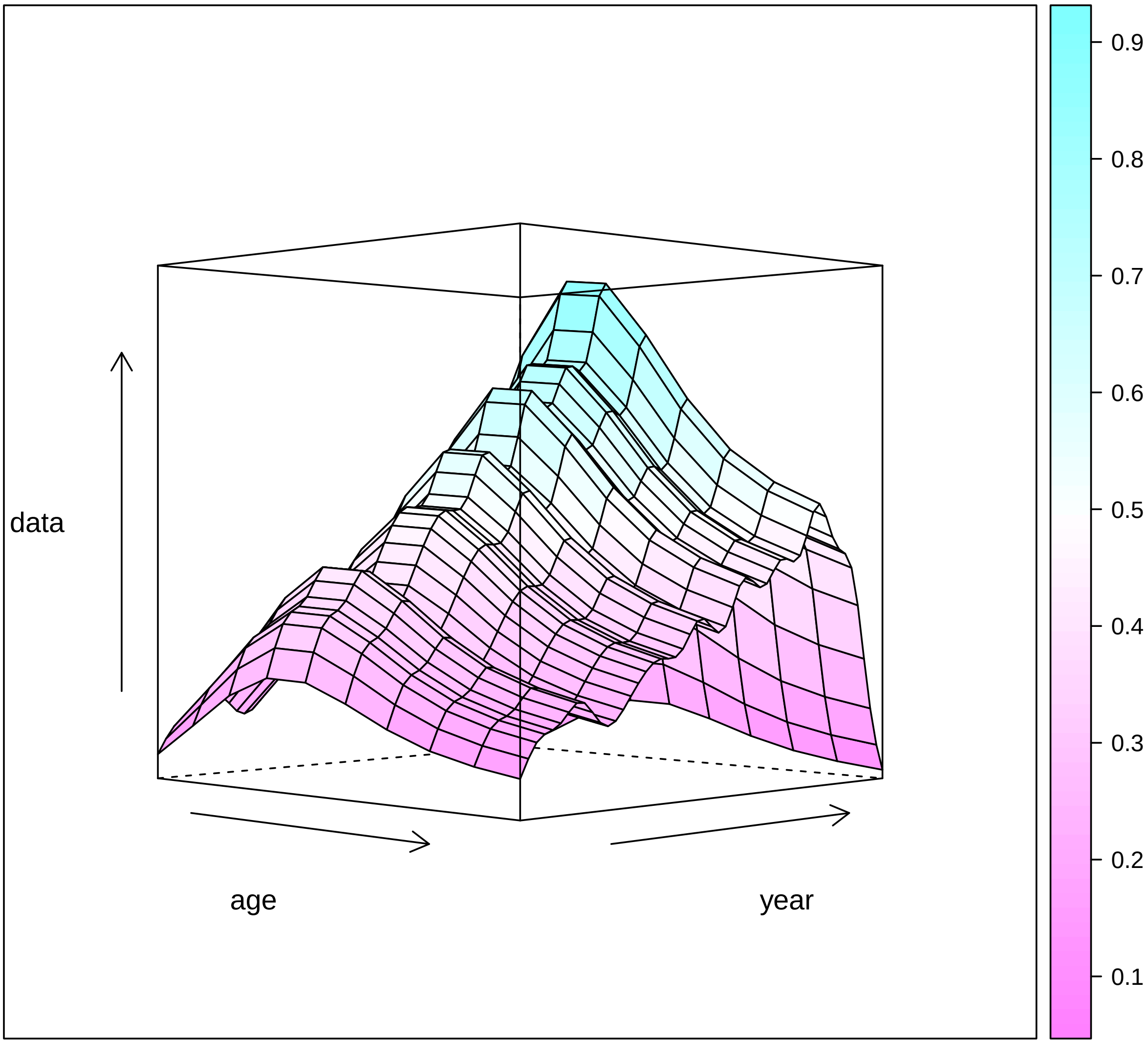
```
qmodel <- list(~factor(age))
fmodel <- ~factor(age) + factor(year)
fit <- sca(stock = ple4, indices = ple4.indices[1], fmodel = fmodel, qmodel = qmodel)

## Note:  The following observations are treated as being missing at random:
##     fleet year age
## BTS-Isis 1997  1
## BTS-Isis 1997  2
##      Predictions will be made for missing observations.
```

```
wireframe(data ~ age + year, data = as.data.frame(harvest(fit)), drape = TRUE,
    screen = list(x = -90, y = -45))
```



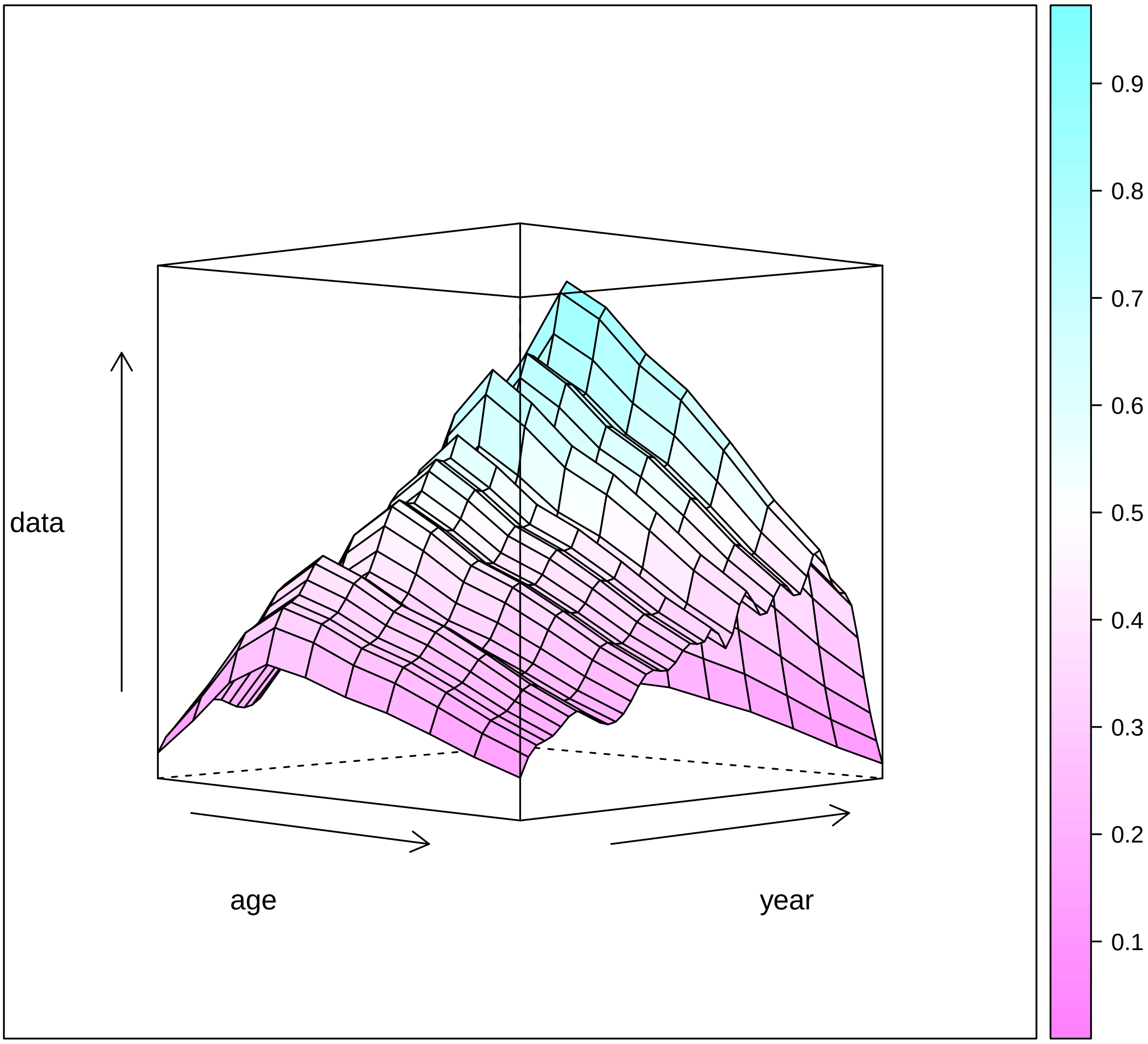

```r
fmodel <- ~s(age, k = 4) + s(year, k = 20)
fit1 <- sca(ple4, ple4.indices[1], fmodel, qmodel)

## Note:  The following observations are treated as being missing at random:
##       fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##        Predictions will be made for missing observations.
```

```r
wireframe(data ~ age + year, data = as.data.frame(harvest(fit1)), drape = TRUE,
    screen = list(x = -90, y = -45))
```



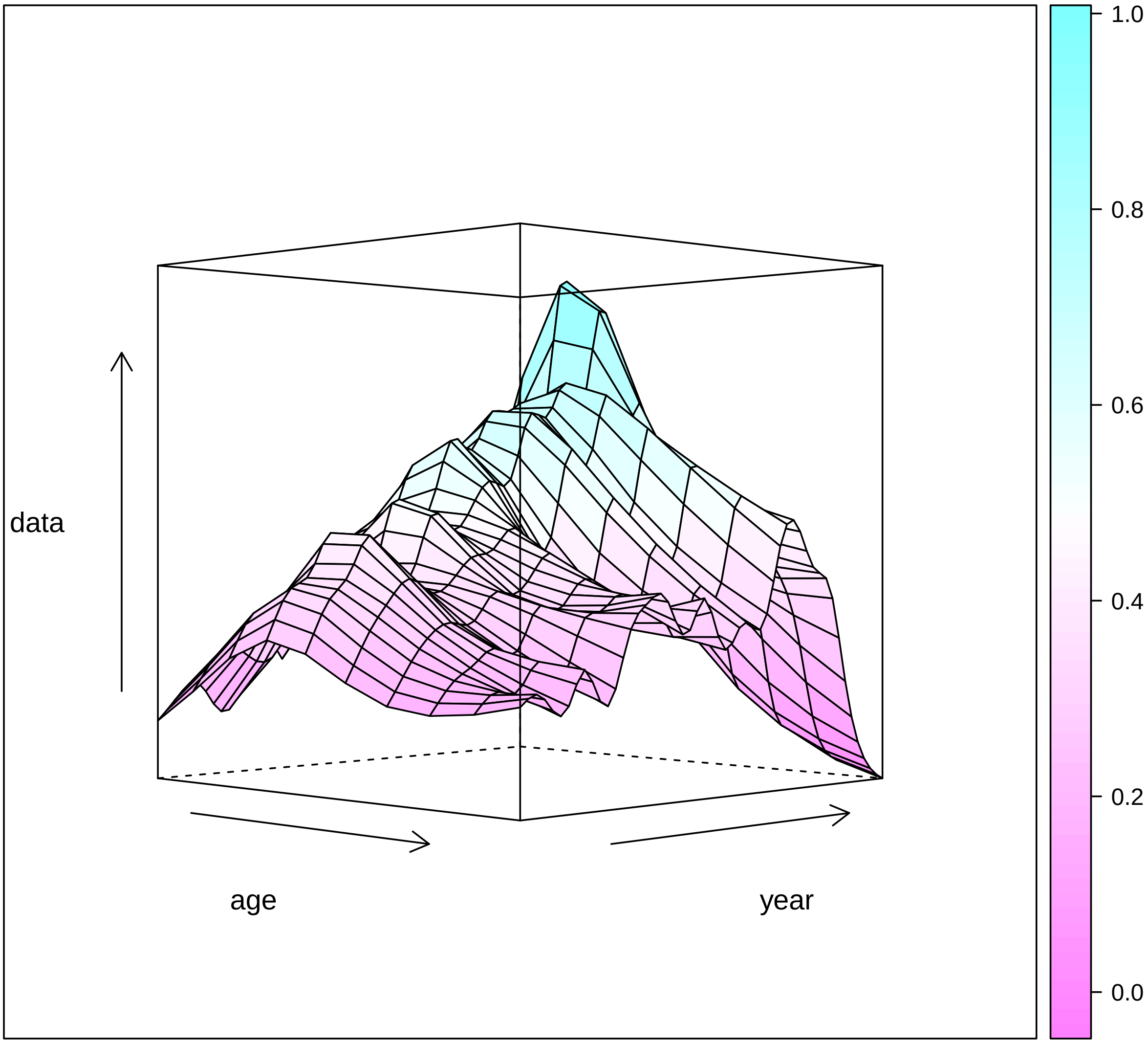

```
fmodel <- ~s(age, k = 4) + s(year, k = 20) + te(age, year, k = c(3, 3))
fit2 <- sca(ple4, ple4.indices[1], fmodel, qmodel)

## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##       Predictions will be made for missing observations.
```

```
wireframe(data ~ age + year, data = as.data.frame(harvest(fit2)), drape = TRUE,
    screen = list(x = -90, y = -45))
```



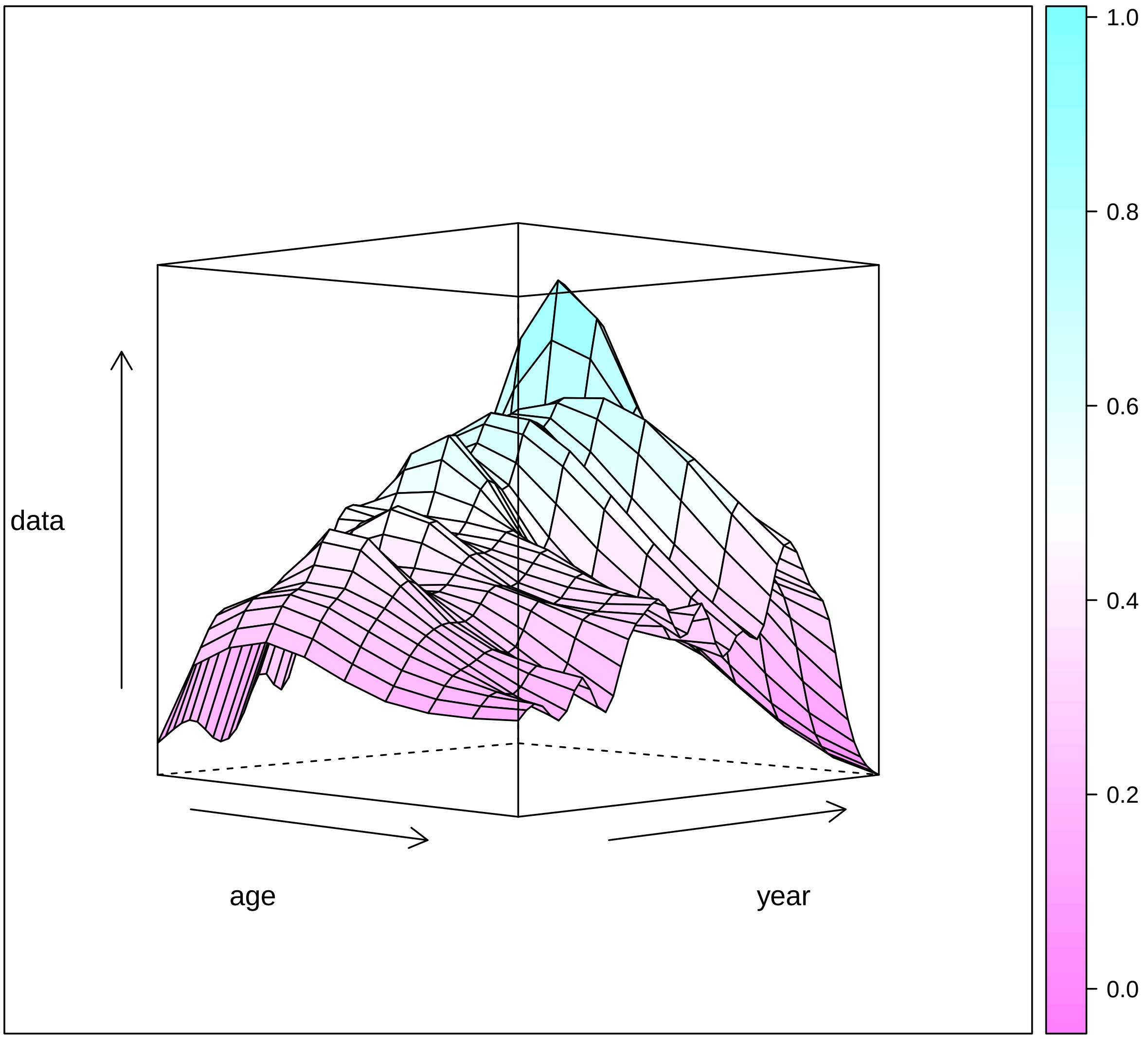

```
fmodel <- ~te(age, year, k = c(4, 20))
fit3 <- sca(ple4, ple4.indices[1], fmodel, qmodel)

## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997  1
## BTS-Isis 1997  2
##       Predictions will be made for missing observations.
```

```
wireframe(data ~ age + year, data = as.data.frame(harvest(fit3)), drape = TRUE,
    screen = list(x = -90, y = -45))
```




```
fmodel <- ~te(age, year, k = c(4, 20)) + s(year, k = 5, by = as.numeric(age ==
    1))
fit4 <- sca(ple4, ple4.indices[1], fmodel, qmodel)

## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##       Predictions will be made for missing observations.
```

```
wireframe(data ~ age + year, data = as.data.frame(harvest(fit4)), drape = TRUE,
    screen = list(x = -90, y = -45))
```

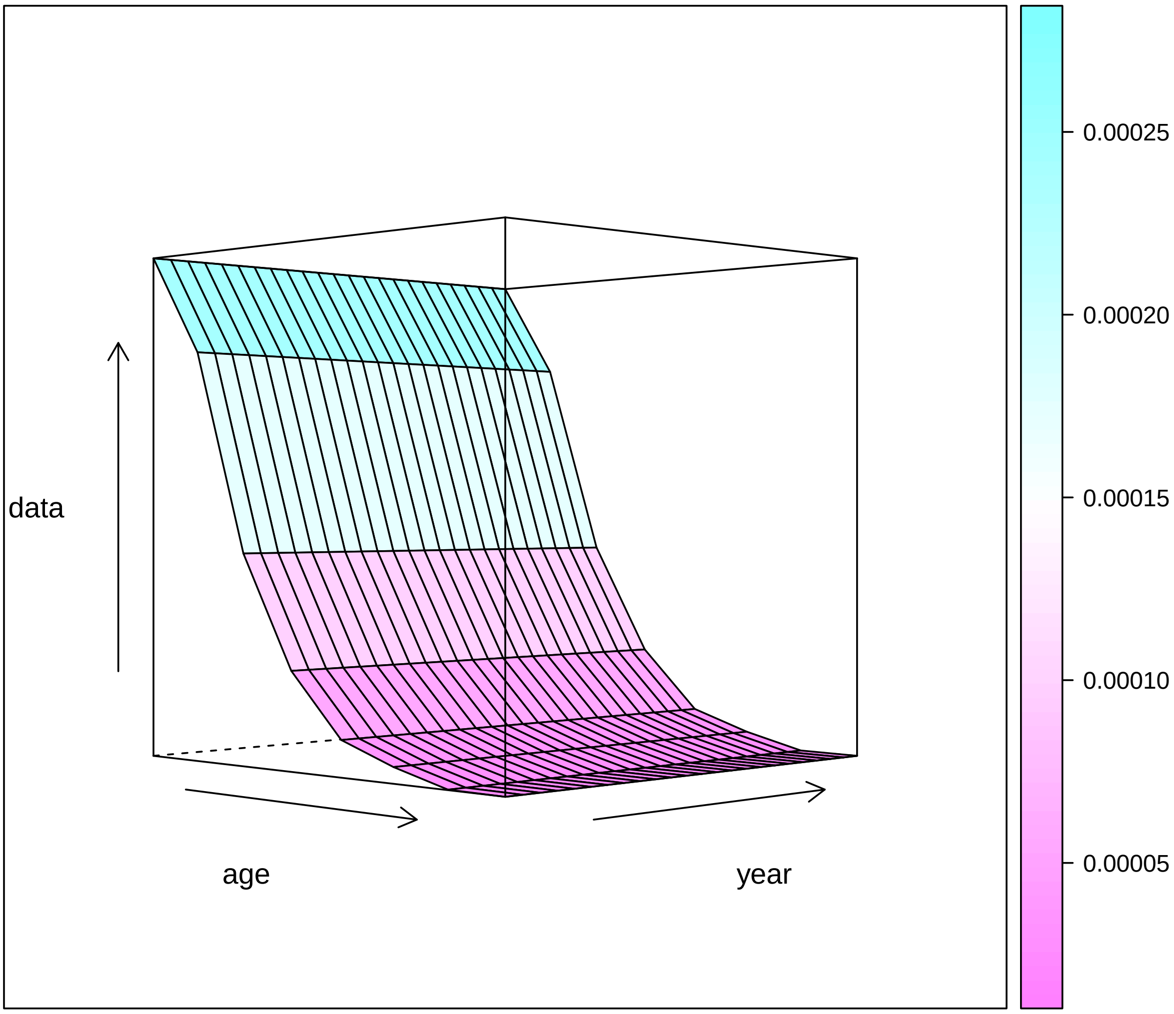


1.3.2 Catchability submodel

```
sfrac <- mean(range(ple4.indices[[1]])[c("startf", "endf")])
fmodel <- ~factor(age) + factor(year)
```

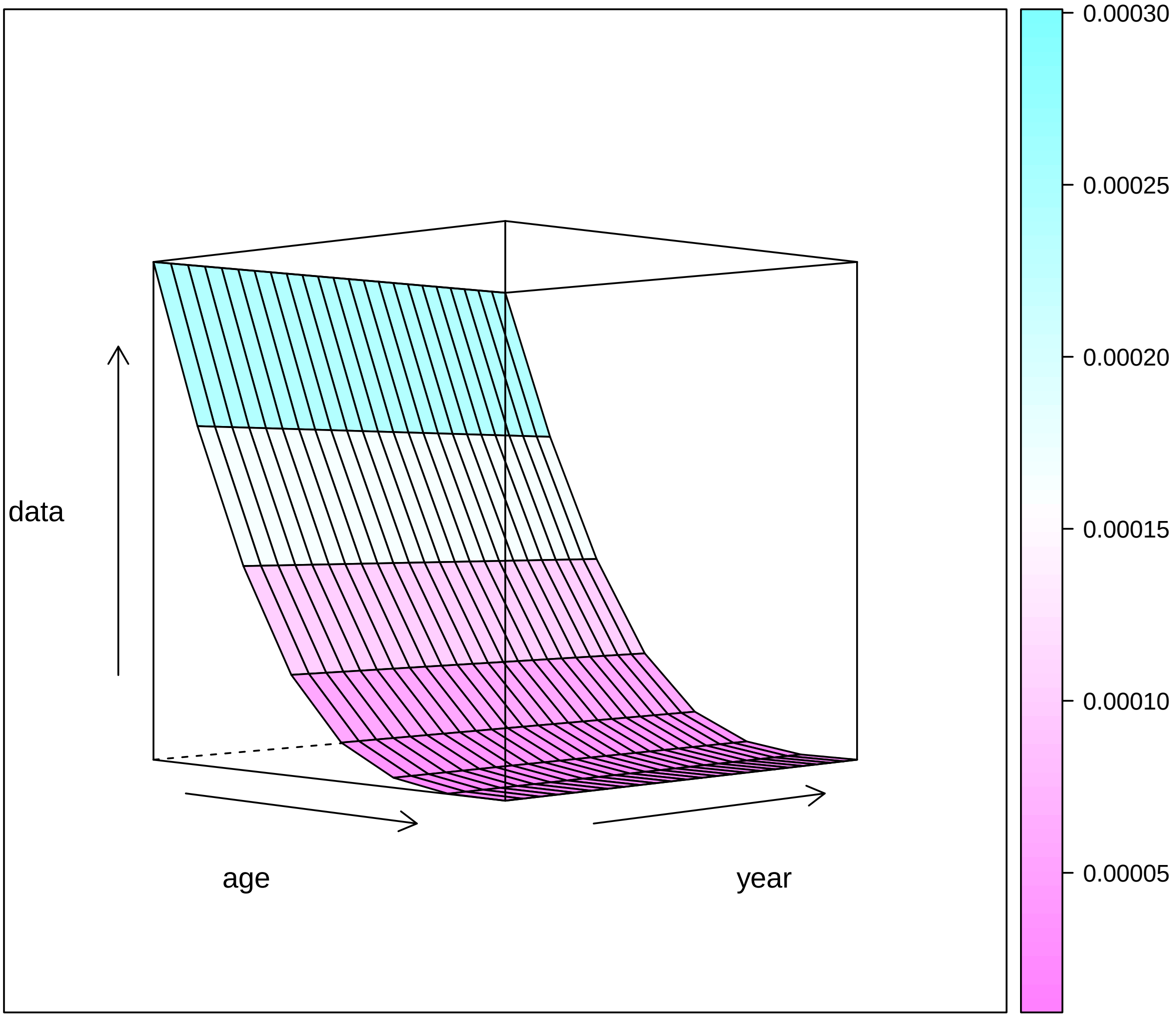
```
qmodel <- list(~factor(age))
fit <- sca(ple4, ple4.indices[1], fmodel, qmodel)

## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##       Predictions will be made for missing observations.

Z <- m(ple4) + harvest(fit) * sfrac
lst <- dimnames(fit@index[[1]])
lst$x <- stock.n(fit) * exp(-Z)
stkn <- do.call("trim", lst)
```

```r
wireframe(data ~ age + year, data = as.data.frame(index(fit)[[1]]/stkn), drape = TRUE,
    screen = list(x = -90, y = -45))
```



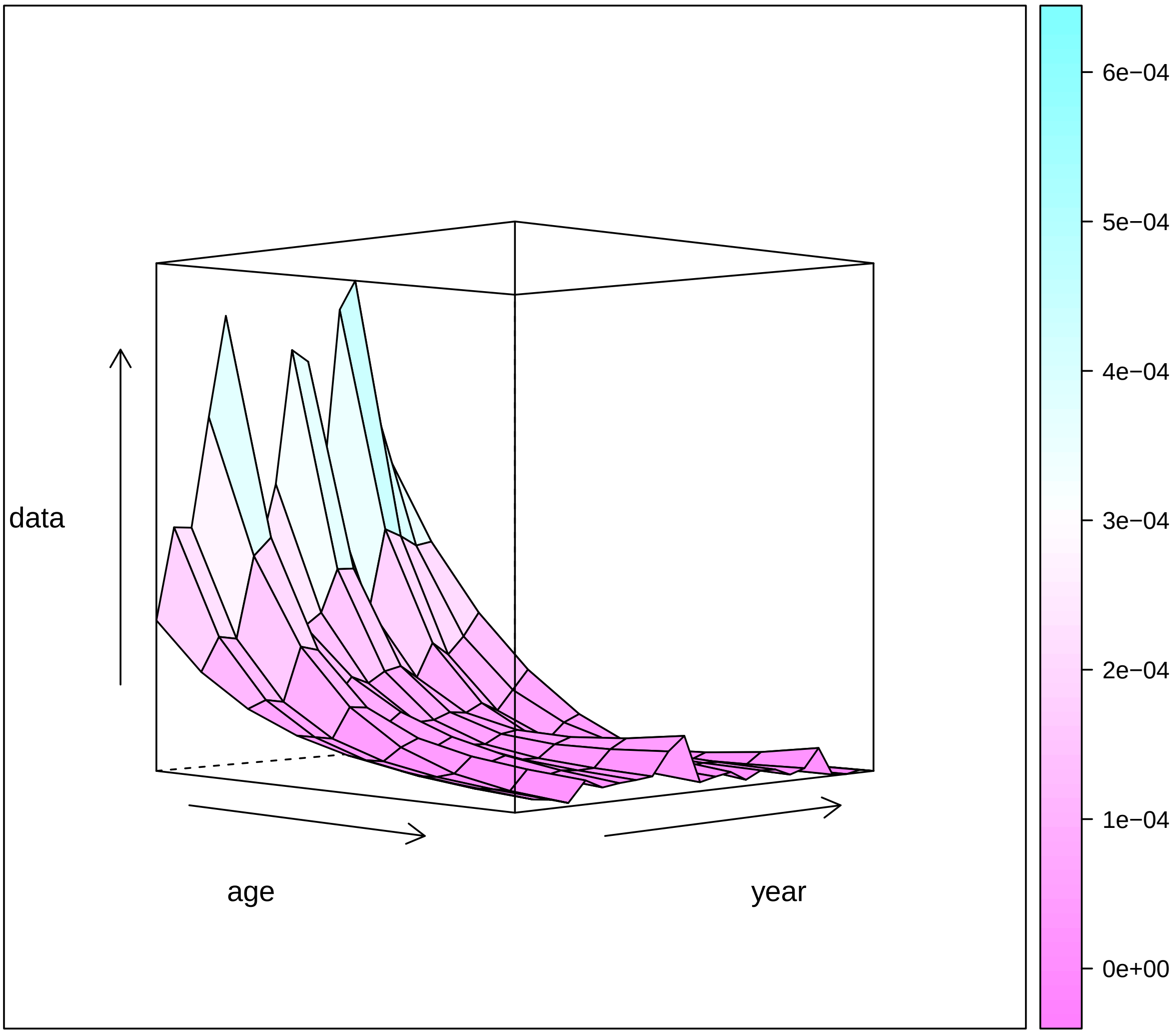

```r
qmodel <- list(~s(age, k = 4))
fit1 <- sca(ple4, ple4.indices[1], fmodel, qmodel)

## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##       Predictions will be made for missing observations.

Z <- m(ple4) + harvest(fit1) * sfrac
lst <- dimnames(fit1@index[[1]])
lst$x <- stock.n(fit1) * exp(-Z)
stkn <- do.call("trim", lst)
```

```r
wireframe(data ~ age + year, data = as.data.frame(index(fit1)[[1]]/stkn), drape = TRUE,
    screen = list(x = -90, y = -45))
```



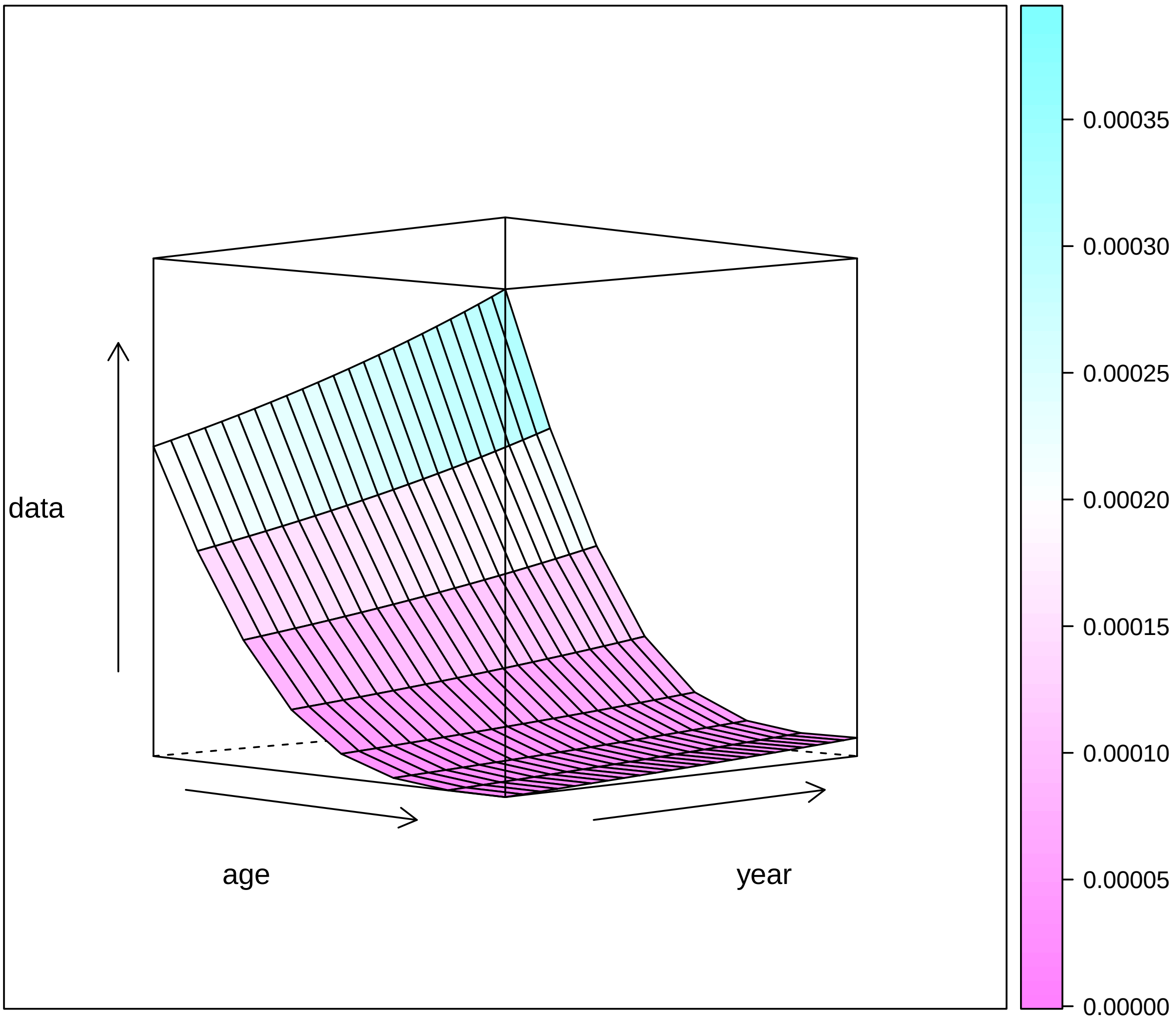

```r
qmodel <- list(~te(age, year, k = c(3, 40)))
fit2 <- sca(ple4, ple4.indices[1], fmodel, qmodel)

## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##       Predictions will be made for missing observations.

Z <- m(ple4) + harvest(fit2) * sfrac
lst <- dimnames(fit2@index[[1]])
lst$x <- stock.n(fit2) * exp(-Z)
stkn <- do.call("trim", lst)
```

```r
wireframe(data ~ age + year, data = as.data.frame(index(fit2)[[1]]/stkn), drape = TRUE,
    screen = list(x = -90, y = -45))
```




```r
qmodel <- list(~s(age, k = 4) + year)
fit3 <- sca(ple4, ple4.indices[1], fmodel, qmodel)

## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##       Predictions will be made for missing observations.

Z <- m(ple4) + harvest(fit3) * sfrac
lst <- dimnames(fit3@index[[1]])
lst$x <- stock.n(fit3) * exp(-Z)
stkn <- do.call("trim", lst)
```

```r
wireframe(data ~ age + year, data = as.data.frame(index(fit3)[[1]]/stkn), drape = TRUE,
    screen = list(x = -90, y = -45))
```



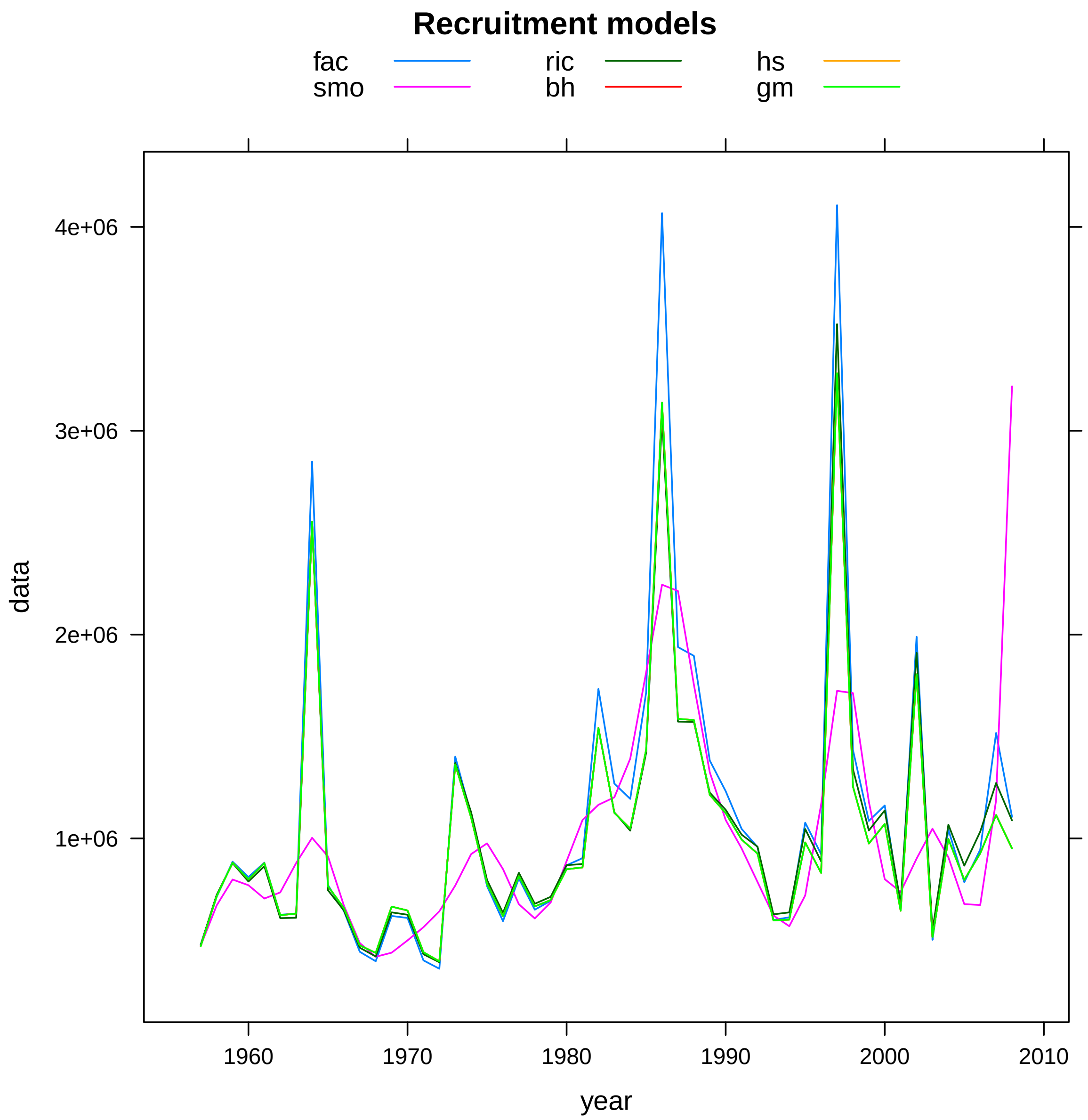
1.3.3 Stock-recruitment submodel

```r
fmodel <- ~s(age, k = 4) + s(year, k = 20)
qmodel <- list(~s(age, k = 4))
```

```r
srmodel <- ~factor(year)
fit <- sca(ple4, ple4.indices[1], fmodel = fmodel, qmodel = qmodel, srmodel = srmodel)

## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##       Predictions will be made for missing observations.
```

```r
srmodel <- ~s(year, k = 20)
fit1 <- sca(ple4, ple4.indices[1], fmodel, qmodel, srmodel)

## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##       Predictions will be made for missing observations.

srmodel <- ~ricker(CV = 0.05)
fit2 <- sca(ple4, ple4.indices[1], fmodel, qmodel, srmodel)

## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##       Predictions will be made for missing observations.

srmodel <- ~bevholt(CV = 0.05)
fit3 <- sca(ple4, ple4.indices[1], fmodel, qmodel, srmodel)

## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##       Predictions will be made for missing observations.

srmodel <- ~hockey(CV = 0.05)
fit4 <- sca(ple4, ple4.indices[1], fmodel, qmodel, srmodel)

## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##       Predictions will be made for missing observations.

srmodel <- ~geomean(CV = 0.05)
fit5 <- sca(ple4, ple4.indices[1], fmodel, qmodel, srmodel)

## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##       Predictions will be made for missing observations.

flqs <- FLQuants(fac = stock.n(fit)[1], smo = stock.n(fit1)[1], ric = stock.n(fit2)[1],
    bh = stock.n(fit3)[1], hs = stock.n(fit4)[1], gm = stock.n(fit5)[1])
```

```r
xyplot(data ~ year, groups = qname, data = flqs, type = "l", main = "Recruitment models",
    auto.key = list(points = FALSE, lines = TRUE, columns = 3))
```



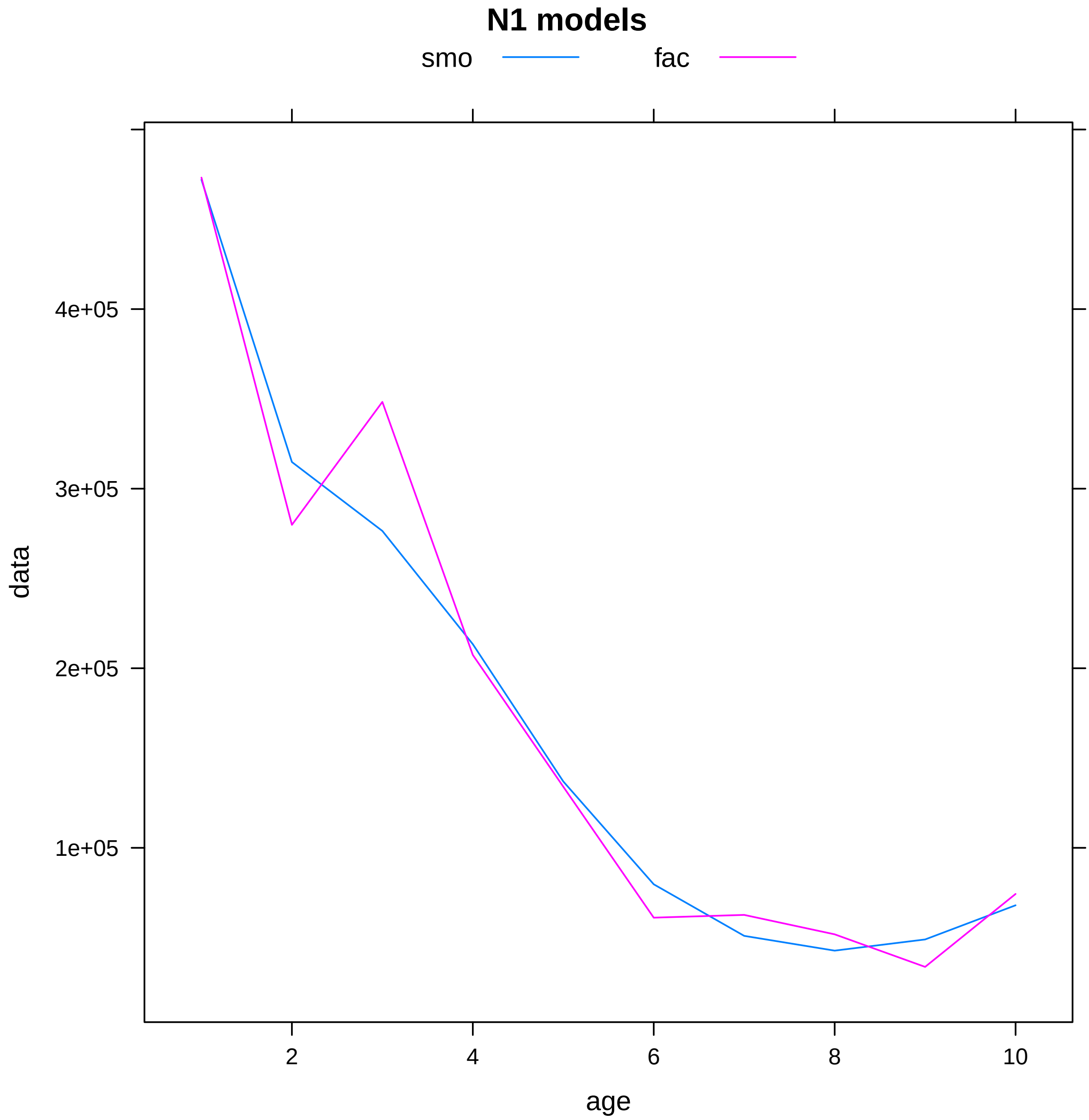
1.4 The a4aSCA method - advanced features

```
fmodel <- ~s(age, k = 4) + s(year, k = 20)
qmodel <- list(~s(age, k = 4) + year)
srmodel <- ~s(year, k = 20)
fit <- a4aSCA(ple4, ple4.indices[1], fmodel, qmodel, srmodel)

## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997  1
## BTS-Isis 1997  2
##       Predictions will be made for missing observations.
```

1.4.1 N1 model

```
n1model <- ~s(age, k = 4)
fit1 <- a4aSCA(ple4, ple4.indices[1], fmodel, qmodel, srmodel, n1model)

## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##       Predictions will be made for missing observations.

flqs <- FLQuants(smo = stock.n(fit1)[, 1], fac = stock.n(fit)[, 1])
```

```
xyplot(data ~ age, groups = qname, data = flqs, type = "l", main = "N1 models",
    auto.key = list(points = FALSE, lines = TRUE, columns = 2))
```



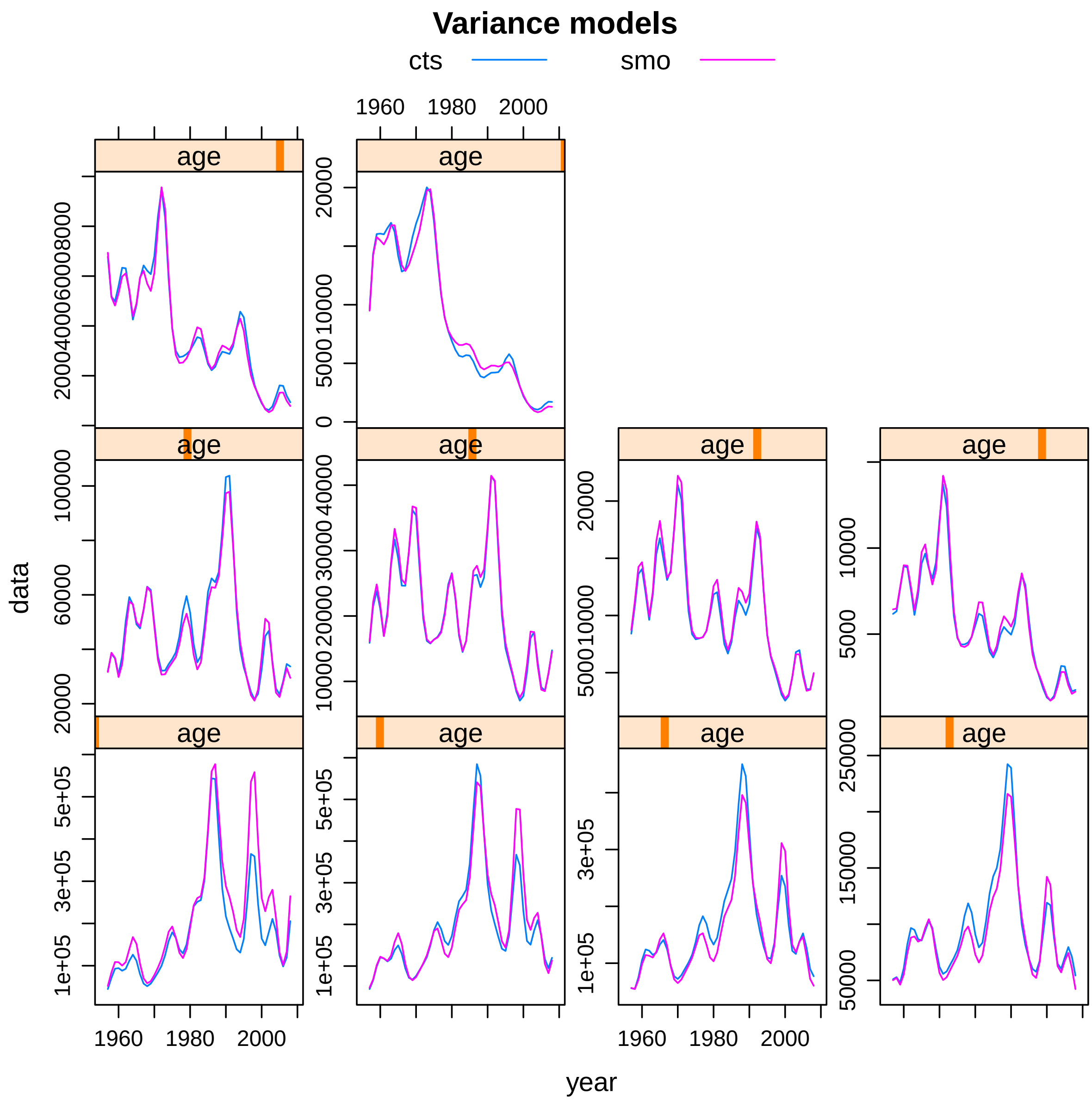
1.4.2 Variance model

One important subject related with fisheries data used for input to stock assessment models is the shape of the variance of the data. It's quite common to have more precision on the most represented ages and less precision on the less frequent ages. Due to the fact that the last do not show so often on the auction markets, on the fishing operations or on survey samples.

By default the model assumes constant variance over time and ages (1 model) but it can use other models specified by the user. This feature requires a call to the a4aInternal method, which gives more options than the a4a method, which in fact is a wrapper.

```
vmodel <- list(~1, ~1)
fit <- a4aSCA(ple4, ple4.indices[1], fmodel, qmodel, srmodel, n1model, vmodel)

## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##       Predictions will be made for missing observations.

vmodel <- list(~s(age, k = 4), ~1)
fit1 <- a4aSCA(ple4, ple4.indices[1], fmodel, qmodel, srmodel, n1model, vmodel)

## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##       Predictions will be made for missing observations.

flqs <- FLQuants(cts = catch.n(fit), smo = catch.n(fit1))
```

```
xyplot(data ~ year | age, groups = qname, data = flqs, type = "l", main = "Variance models",
    scales = list(y = list(relation = "free")), auto.key = list(points = FALSE,
        lines = TRUE, columns = 2))
```



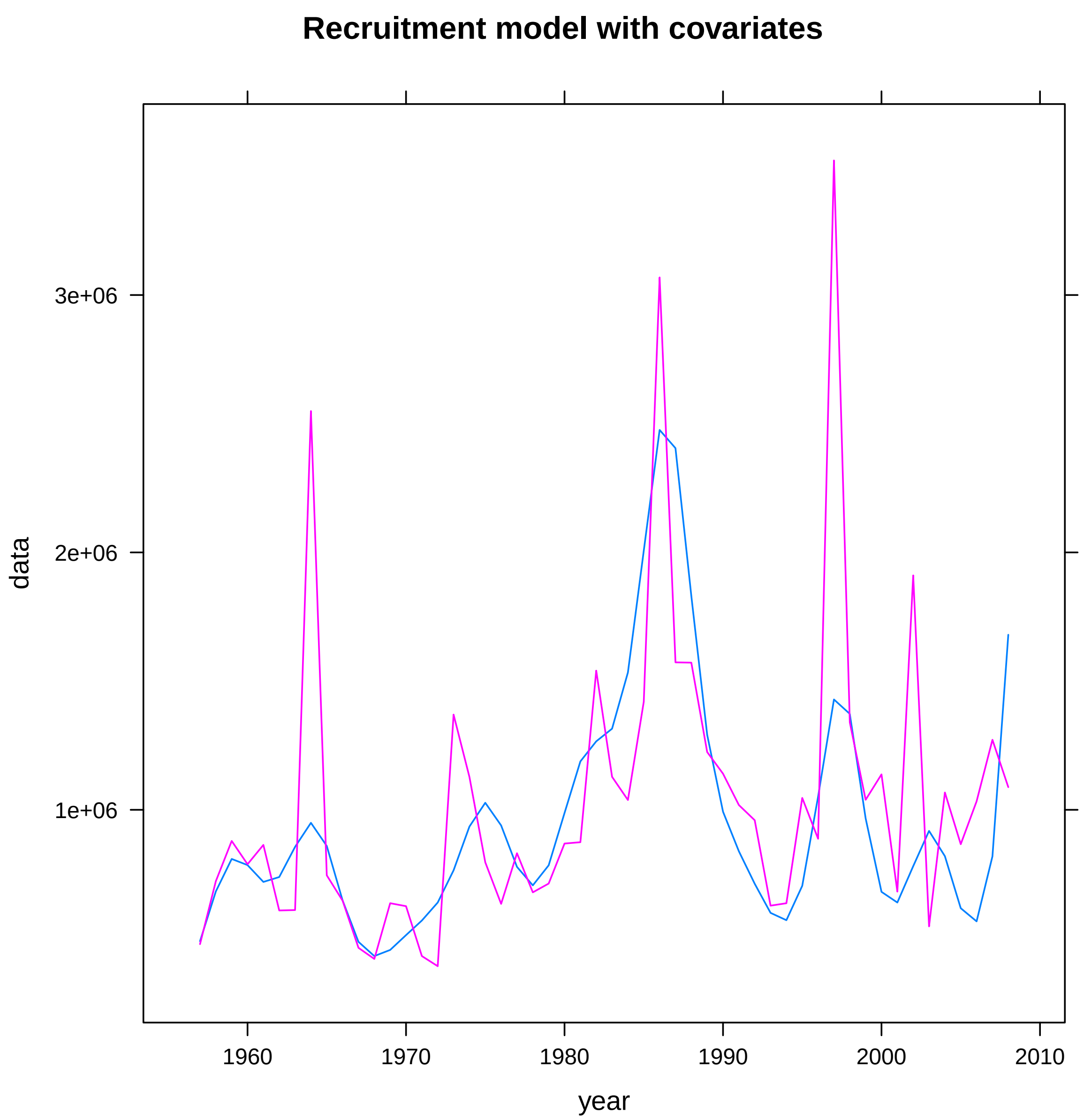
1.4.3 Working with covariates

```
nao <- read.table("http://www.cdc.noaa.gov/data/correlation/nao.data", skip = 1,
    nrow = 62, na.strings = "-99.90")

## Warning:  cannot open:  HTTP status was '503 Service Temporarily Unavailable'

## Error:  cannot open the connection

dnms <- list(quant = "nao", year = 1948:2009, unit = "unique", season = 1:12,
    area = "unique")
nao <- FLQuant(unlist(nao[, -1]), dimnames = dnms, units = "nao")

## Error:  error in evaluating the argument 'object' in selecting a method for function
'FLQuant':  Error in unlist(nao[, -1]) :  object 'nao' not found

nao <- seasonMeans(trim(nao, year = dimnames(stock.n(ple4))$year))
```

```
## Error:  error in evaluating the argument 'x' in selecting a method for function 'seasonMeans':
Error in trim(nao, year = dimnames(stock.n(ple4))$year) :
##  error in evaluating the argument 'x' in selecting a method for function 'trim':  Error:
object 'nao' not found

nao <- as.numeric(nao)

## Error:  object 'nao' not found
```

```
srmodel <- ~nao
fit2 <- a4aSCA(ple4, ple4.indices[1], fmodel, qmodel, srmodel)

## Note:  The following observations are treated as being missing at random:
##     fleet year age
## BTS-Isis 1997  1
## BTS-Isis 1997  2
##      Predictions will be made for missing observations.

## Error:  object 'nao' not found

flqs <- FLQuants(fac = stock.n(fit)[1], cvar = stock.n(fit2)[1])
```

```
xyplot(data ~ year, groups = qname, data = flqs, type = "l", main = "Recruitment model with covariates"
```




```
srmodel <- ~ricker(a = ~nao, CV = 0.1)
fit3 <- a4aSCA(ple4, ple4.indices[1], fmodel, qmodel, srmodel)

## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997  1
## BTS-Isis 1997  2
##       Predictions will be made for missing observations.

## Error:  object 'nao' not found

flqs <- FLQuants(fac = stock.n(fit)[1], cvar = stock.n(fit3)[1])
```

```
# fmodel <- ~ s(age, k=4) + s(year, k = 20) qmodel <- list( ~ s(age, k=4) +
# year) srmodel <- ~s(year, k=20) fit <- a4aSCA(ple4, ple4.indices[1],
# fmodel, qmodel, srmodel) fits <- simulate(fit, 25) stks <- ple4 + fits
# idxs <- ple4.indices[1] index(idxs[[1]]) <- index(fits)[[1]]
# library(parallel) options(mc.cores=1) lst <- mclapply(split(1:25, 1:25),
```

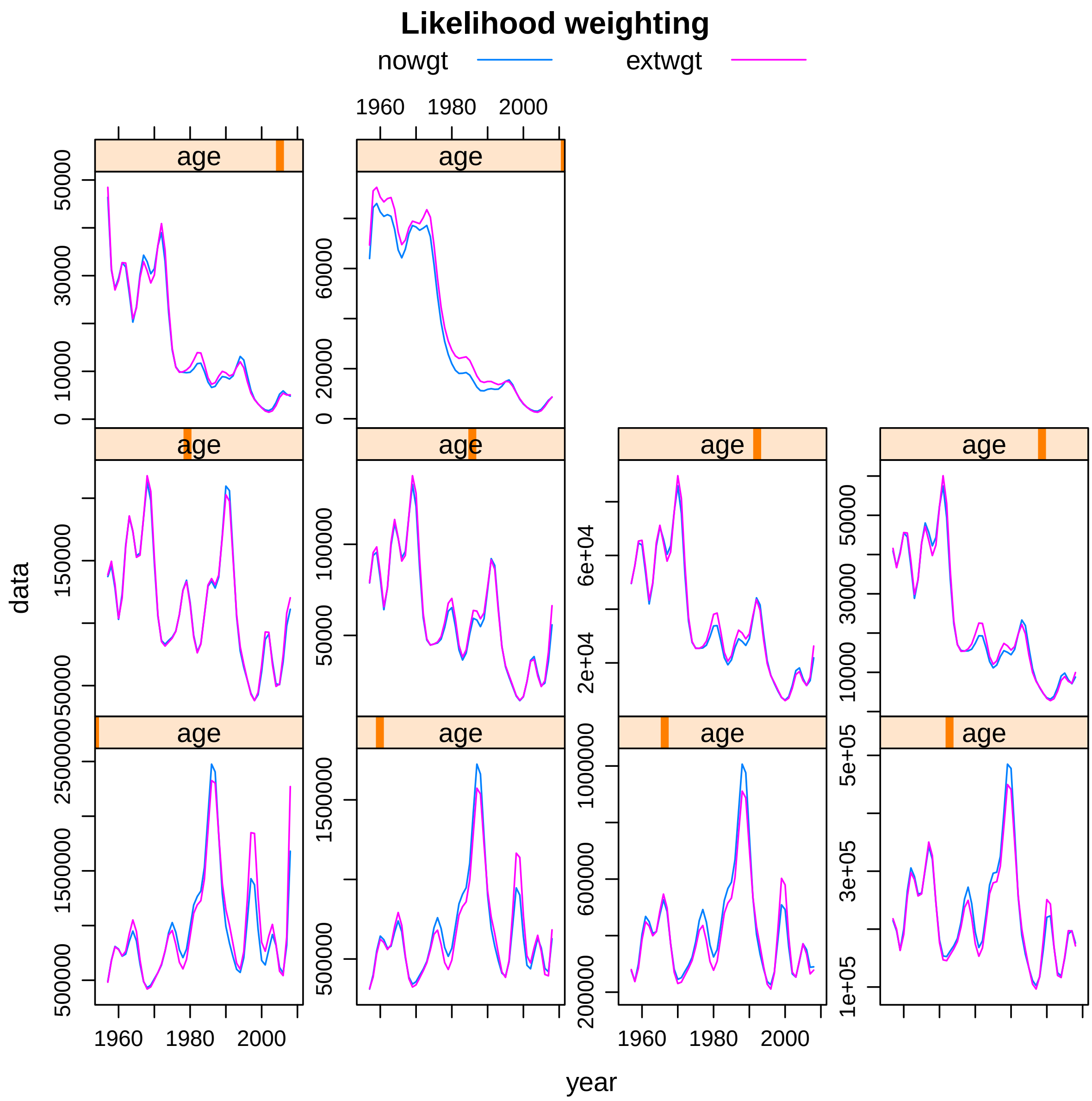
```
# function(x){ fit <- a4aSCA(stks[,,,,,x], FLIndices(idxs[[1]][,,,,,x]),
# fmodel, qmodel, srmodel, fit='MP') })
```

1.4.4 External weigthing of likelihood components

By default the likelihood components are weighted using inverse variance of the parameters estimates. However the user may change this weights by setting the variance of the input parameters, which is done by adding a variance matrix to the catch.n and index.n slots of the stock and index objects.

```
stk <- ple4
idx <- ple4.indices[1]
# variance of observed catches
varslt <- catch.n(stk)
varslt[] <- 1
catch.n(stk) <- FLQuantDistr(catch.n(stk), varslt)
# variance of observed indices
varslt <- index(idx[[1]])
varslt[] <- 0.05
index.var(idx[[1]]) <- varslt
# run
fit1 <- a4aSCA(stk, idx, fmodel, qmodel, srmodel, n1model, vmodel = list(~1,
    ~1))

## Note:  Provided variances will be used to weight observations.
## Weighting assumes variances are on the log scale or equivalently log(CV^2 + 1).
## Note:  The following observations are treated as being missing at random:
##     fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##      Predictions will be made for missing observations.

## Error:  object 'nao' not found

flqs <- FLQuants(nowgt = stock.n(fit), extwgt = stock.n(fit1))
```

```
xyplot(data ~ year | age, groups = qname, data = flqs, type = "l", main = "Likelihood weighting",
    scales = list(y = list(relation = "free")), auto.key = list(points = FALSE,
        lines = TRUE, columns = 2))
```



1.4.5 Assessing ADMB files

To inspect the ADMB files the user must specify the working dir and all files will be left there.

```
fit1 <- a4aSCA(stk, idx, fmodel, qmodel, srmodel, n1model, vmodel = list(~1,
    ~1), wkdir = "mytest")

## Note:  Provided variances will be used to weight observations.
## Weighting assumes variances are on the log scale or equivalently log(CV^2 + 1).
## Note:  The following observations are treated as being missing at random:
##      fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##       Predictions will be made for missing observations.
## Error:  object 'nao' not found
```

1.5 Predict and simulate

```
fmodel <- ~s(age, k = 4) + s(year, k = 20)
qmodel <- list(~s(age, k = 4) + year)
srmodel <- ~s(year, k = 20)
fit <- a4aSCA(ple4, ple4.indices[1], fmodel, qmodel, srmodel)

## Note:  The following observations are treated as being missing at random:
##     fleet year age
## BTS-Isis 1997   1
## BTS-Isis 1997   2
##      Predictions will be made for missing observations.
```

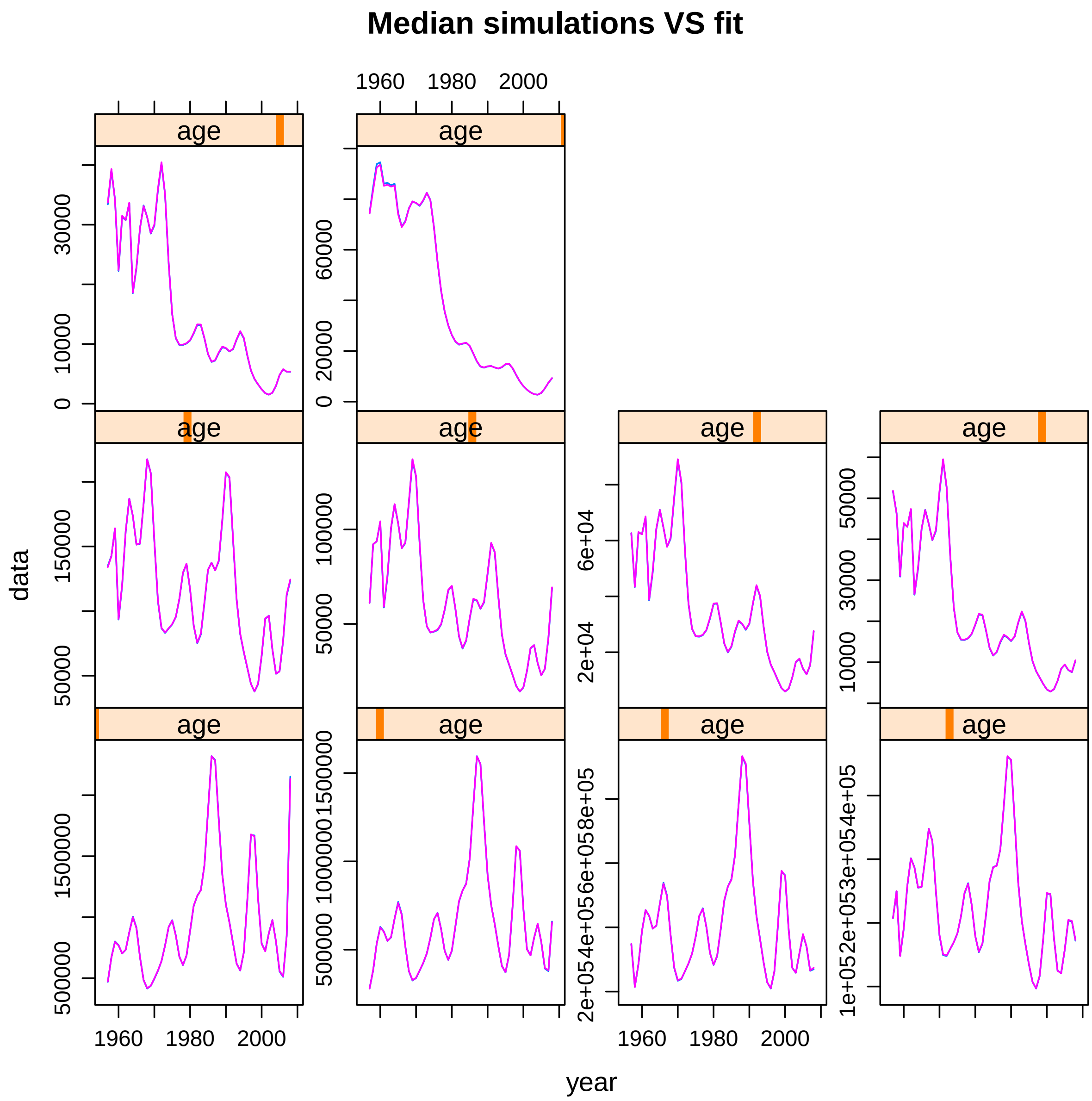
1.5.1 Predict

```
fit.pred <- predict(fit)
lapply(fit.pred, names)

## $stkmodel
## [1] "harvest" "rec"     "ny1"
##
## $qmodel
## [1] "BTS-Isis"
##
## $vmodel
## [1] "catch"    "BTS-Isis"
```

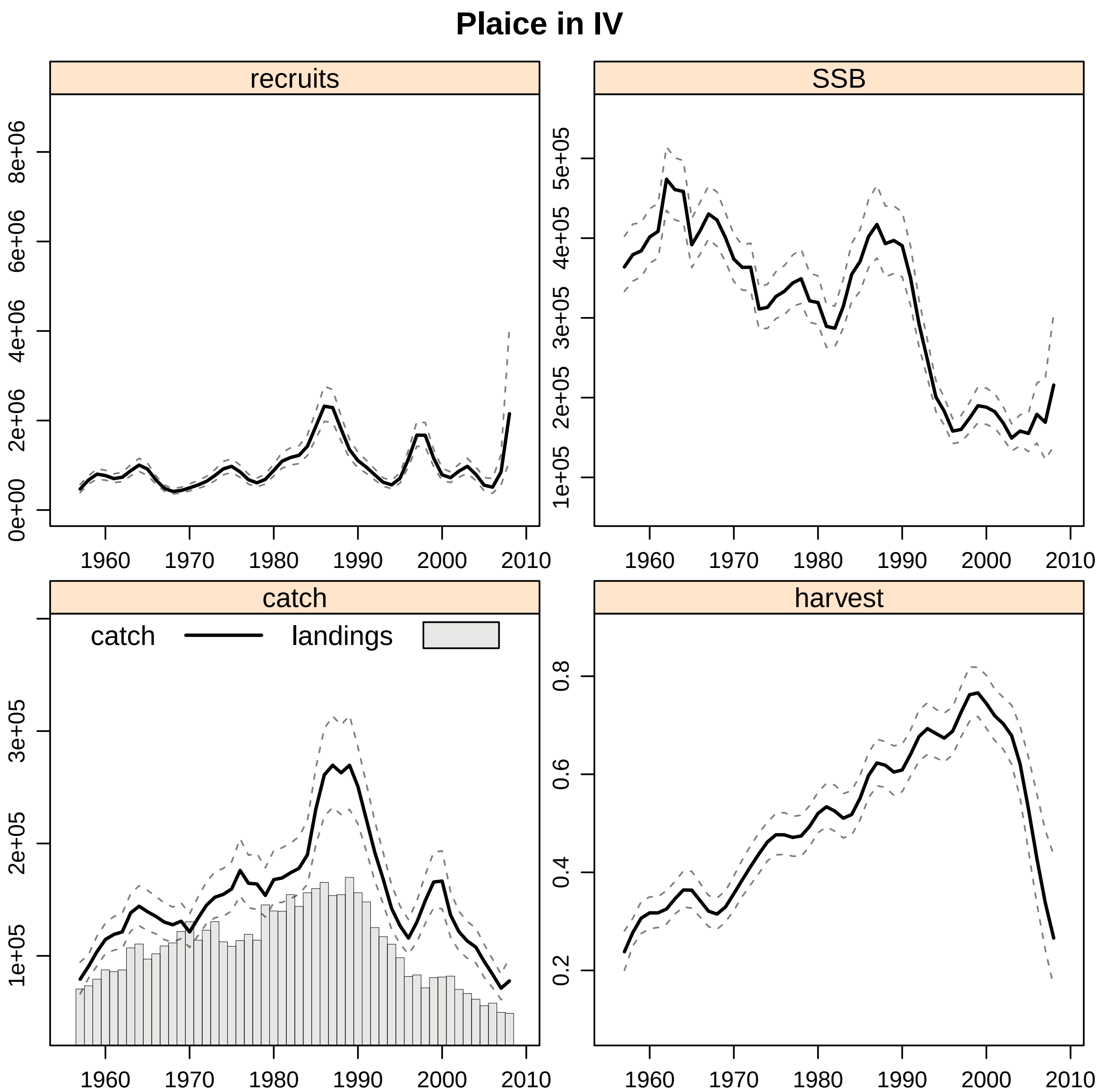
1.5.2 simulate

```
fits <- simulate(fit, 1000)
flqs <- FLQuants(sim = iterMedians(stock.n(fits)), det = stock.n(fit))
```

```
xyplot(data ~ year | age, groups = qname, data = flqs, type = "l", main = "Median simulations VS fit",
    scales = list(y = list(relation = "free")))
```




```
stks <- ple4 + fits
plot(stks)

## Warning:  invalid factor level, NA generated
```


Plaice in IV
recruits
SSB
catch
harvest
catch
landings
0e+00
2e+06
4e+06
6e+06
8e+06
1e+05
2e+05
3e+05
4e+05
5e+05
0.2
0.4
0.6
0.8
1960
1970
1980
1990
2000
2010