Step by step demo for the `a4a` statistical catch-at-age framework

Ernesto Jardim[1], Danai Mantopoulou Palouka[2], and Colin Millar[3]

[1]Independent Expert, Brussels, Belgium
[2]European Commission, Joint Research Centre, Sustainable resources directorate, Water and Marine Resources unit, 21027 Ispra (VA), Italy
[3]International Council for the Exploration of the Sea (ICES), H. C. Andersens Boulevard 44-46, 1553 Copenhagen V, Denmark
[*]Corresponding author ernesto.jardim@gmail.com

October 9, 2024

Abstract

Contents

1 Before starting

1.1 License, documentation and development status

The software is released under the EUPL 1.1.

For more information on the `a4a`methodologies refer to Jardim, et.al, 2014, Millar, et.al, 2014 and Scott, et.al, 2016.

Documentation can be found at `http://flr-project.org/FLa4a`. You are welcome to:

- Submit suggestions and bug-reports at: `https://github.com/flr/FLa4a/issues`

- Send a pull request on: `https://github.com/flr/FLa4a/`

- Compose a friendly e-mail to the maintainer, see 'packageDescription('FLa4a')'

1.2 Installing and loading libraries

To run the `FLa4a` methods the reader will need to install the package and its dependencies and load them. Some datasets are distributed with the package and as such need to be loaded too.

```r
# from CRAN
install.packages(c("copula", "triangle", "coda", "grid", "gridExtra", "latticeExtra"))
# from FLR
install.packages(c("FLCore", "FLa4a"), repos = "http://flr-project.org/R")
```

```r
# libraries
library(devtools)
library(FLa4a)
library(XML)
library(reshape2)
library(ggplotFL)
# datasets
data(ple4)
data(ple4.index)
```

```r
packageVersion("FLCore")
```

```
## [1] '2.6.20.920'
```

```r
packageVersion("FLa4a")
```

```
## [1] '1.9.0'
```

1.3 How to read this document

The target audience for this document are readers with some experience in R and some background on stock assessment.

The document explains the approach being developed by `a4a`for fish stock assessment and scien-

tific advice. It presents a mixture of text and code, where the first explains the concepts behind the methods, while the last shows how these can be run with the software provided. Moreover, having the code allows the reader to copy/paste and replicate the analysis presented here.

The sections and subsections are as independent as possible, so it can be used as a reference document for the `FLa4a`.

2 Step by step

2.1 The "mean" model

To start the analysis we'll fit a "mean" model, where all submodels will be set to an overall average, by using the 1 formula. This will be our reference model to see how adding age and year effects will show up in the diagnostic tools, in particular in the residuals.

```
fit01 <- sca(ple4, ple4.index, fmod = ~1, qmod = list(~1), srmod = ~1, vmod = list(~1,
    ~1), n1mod = ~1)
res01 <- residuals(fit01, ple4, ple4.index)
```

The common residuals plot clearly shows a trend across ages (figure 2) for both datasets.
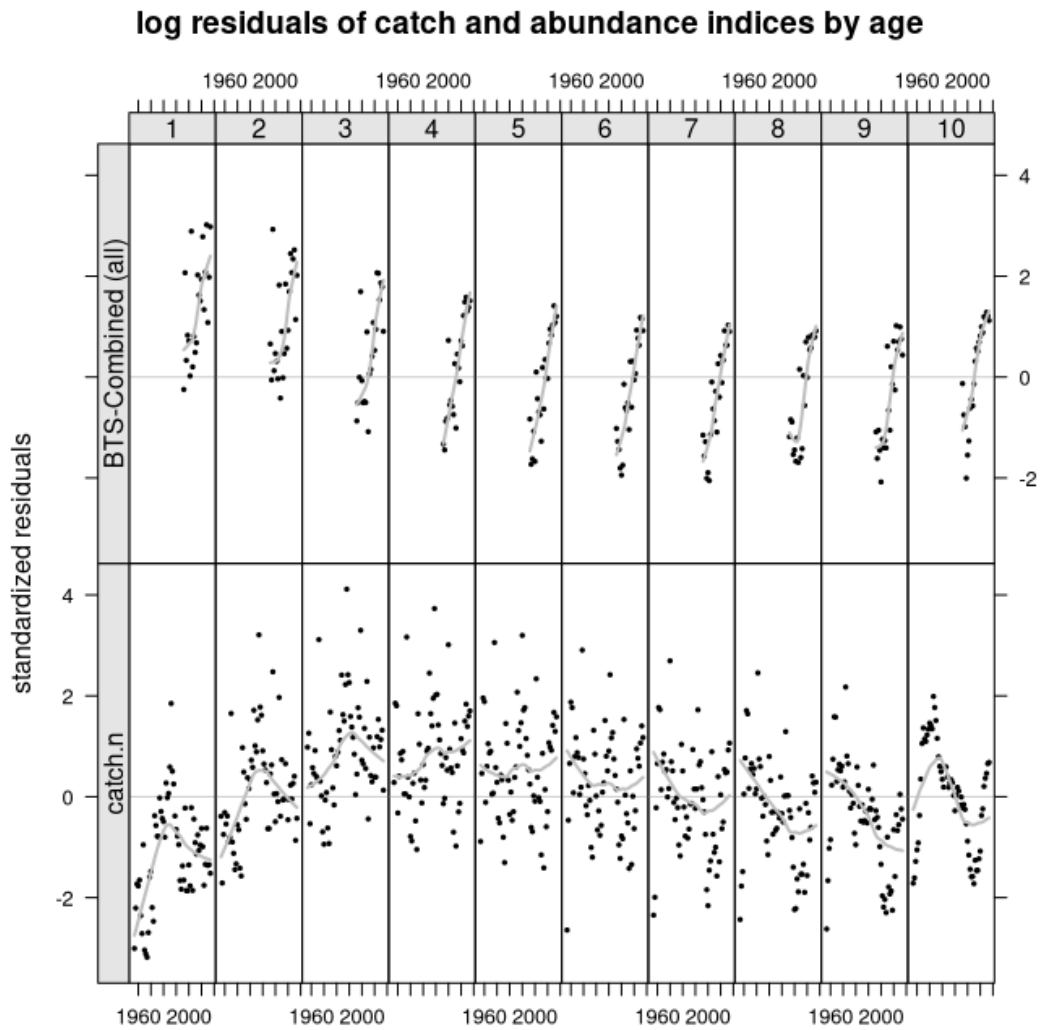
```
plot(res01)
```

Figure 1: Mean fit residuals by year)

Which is even clearer when plotting the residuals by age across years.
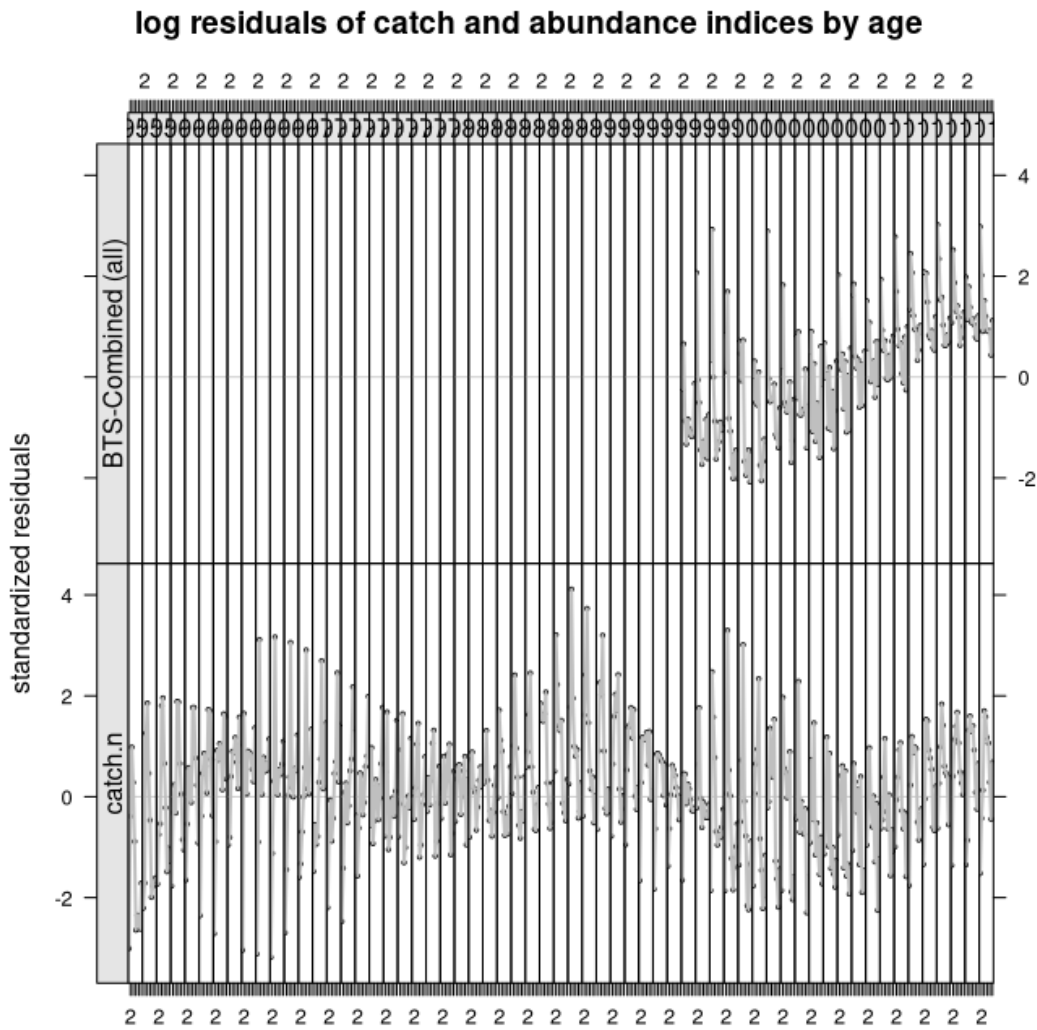
```
plot(res01, auxline = "l", by = "age")
```

Figure 2: Mean fit residuals by age)

2.2 The age effects

These models will introduce age effects in the fishing mortality submodel and catchability submodel. First in the fishinf mortality submodel.

```
fit02 <- sca(ple4, ple4.index, fmod = ~factor(age), qmod = list(~1), srmod = ~1,
    vmod = list(~1, ~1), n1mod = ~1)
res02 <- residuals(fit02, ple4, ple4.index)
```

The residuals plot now shows catch at age residuals less stagered, reflecting the modelling of the age effect.
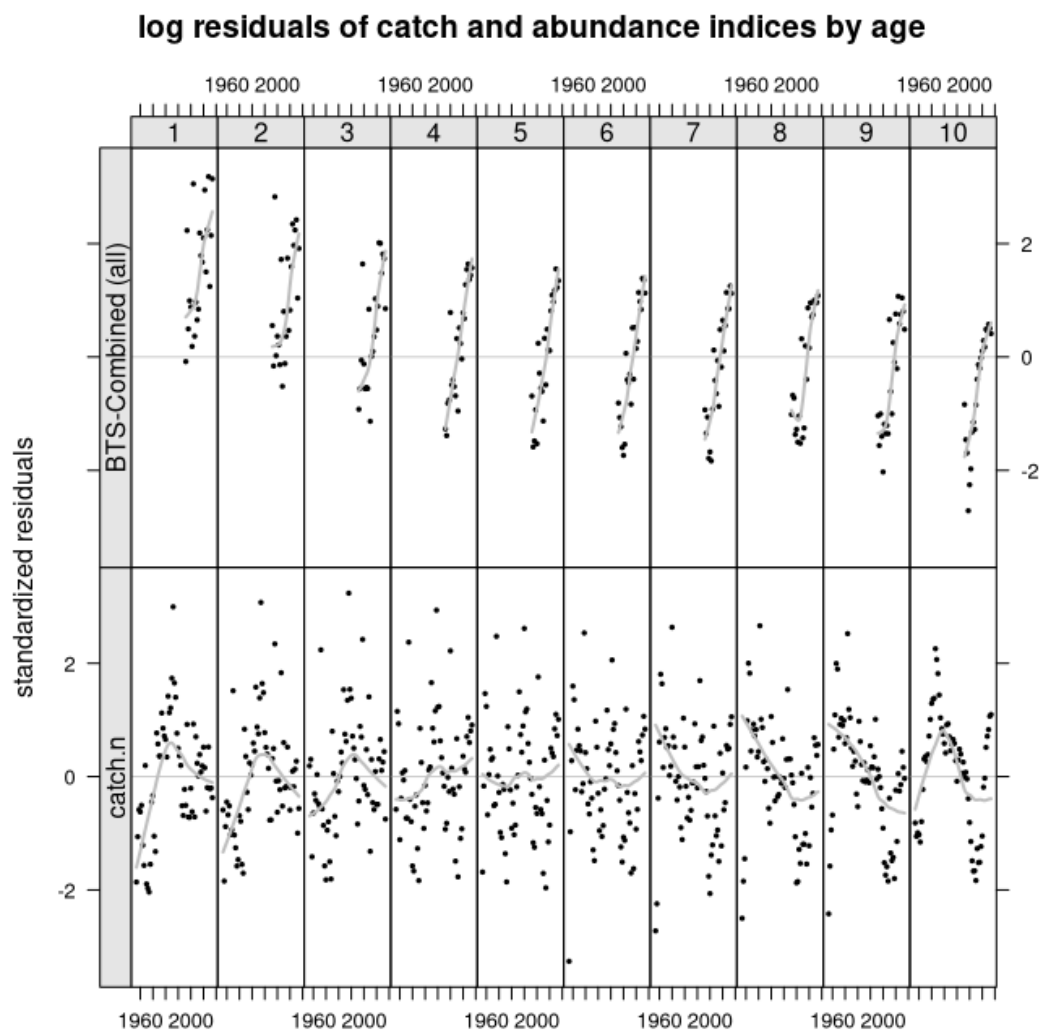
```
plot(res02)
```


Figure 3: f age effect fit residuals by year)

The residuals plot by age shows the same outcome.
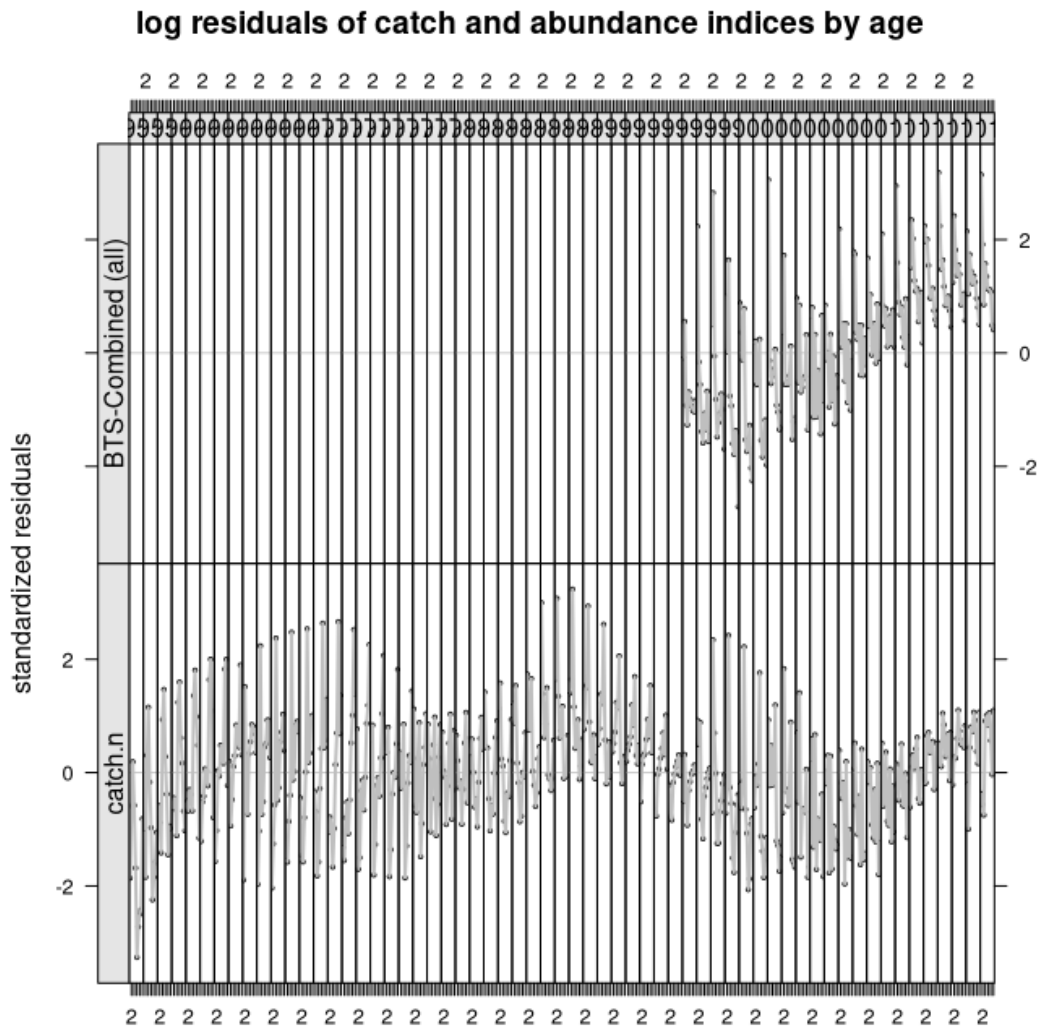
```
plot(res02, auxline = "l", by = "age")
```

Figure 4: f age effect fit residuals by age)

Follwed by the same addition to the catchability model.

```
fit03 <- sca(ple4, ple4.index, fmod = ~1, qmod = list(~factor(age)), srmod = ~1,
    vmod = list(~1, ~1), n1mod = ~1)
res03 <- residuals(fit03, ple4, ple4.index)

plot(res03)
```
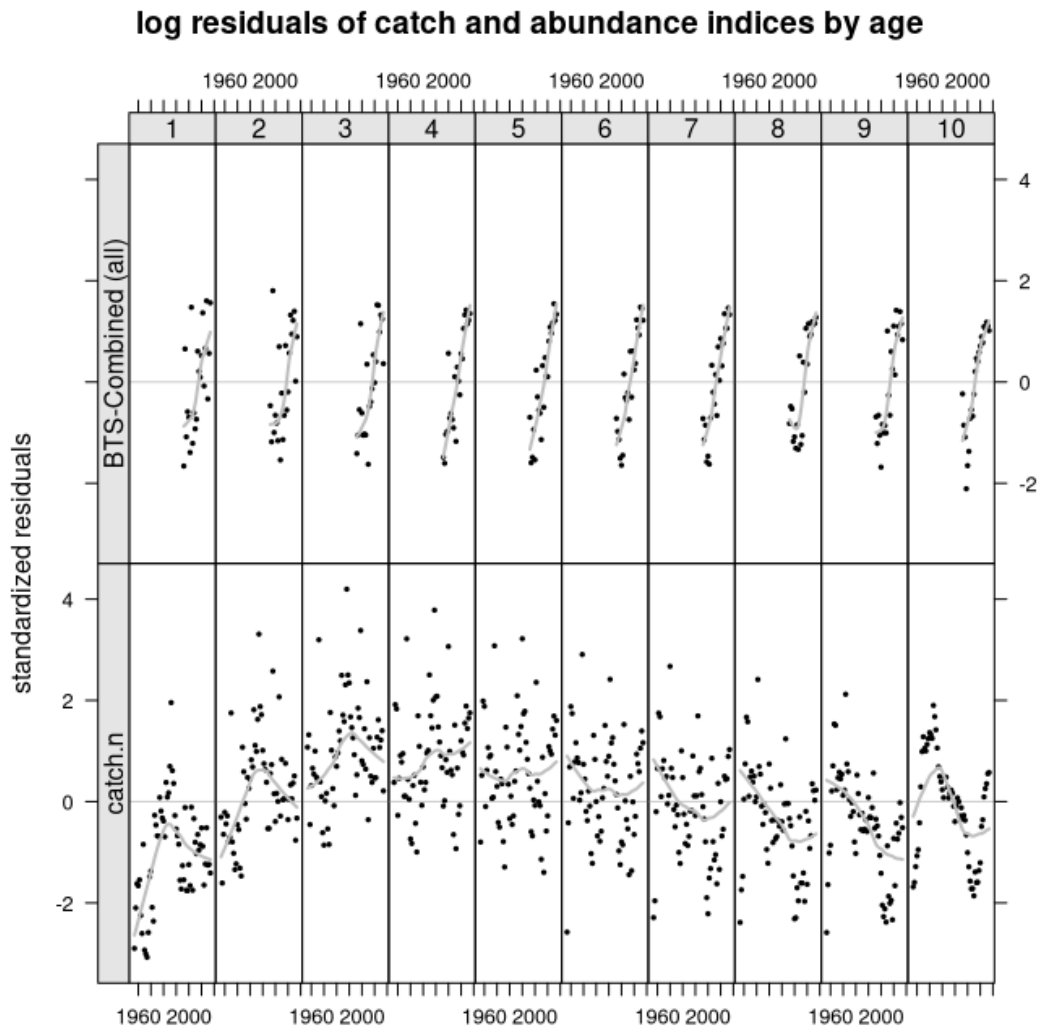
Figure 5: q age effect fit residuals by year)

The residuals plot by age shows the same outcome.

```
plot(res03, auxline = "l", by = "age")
```
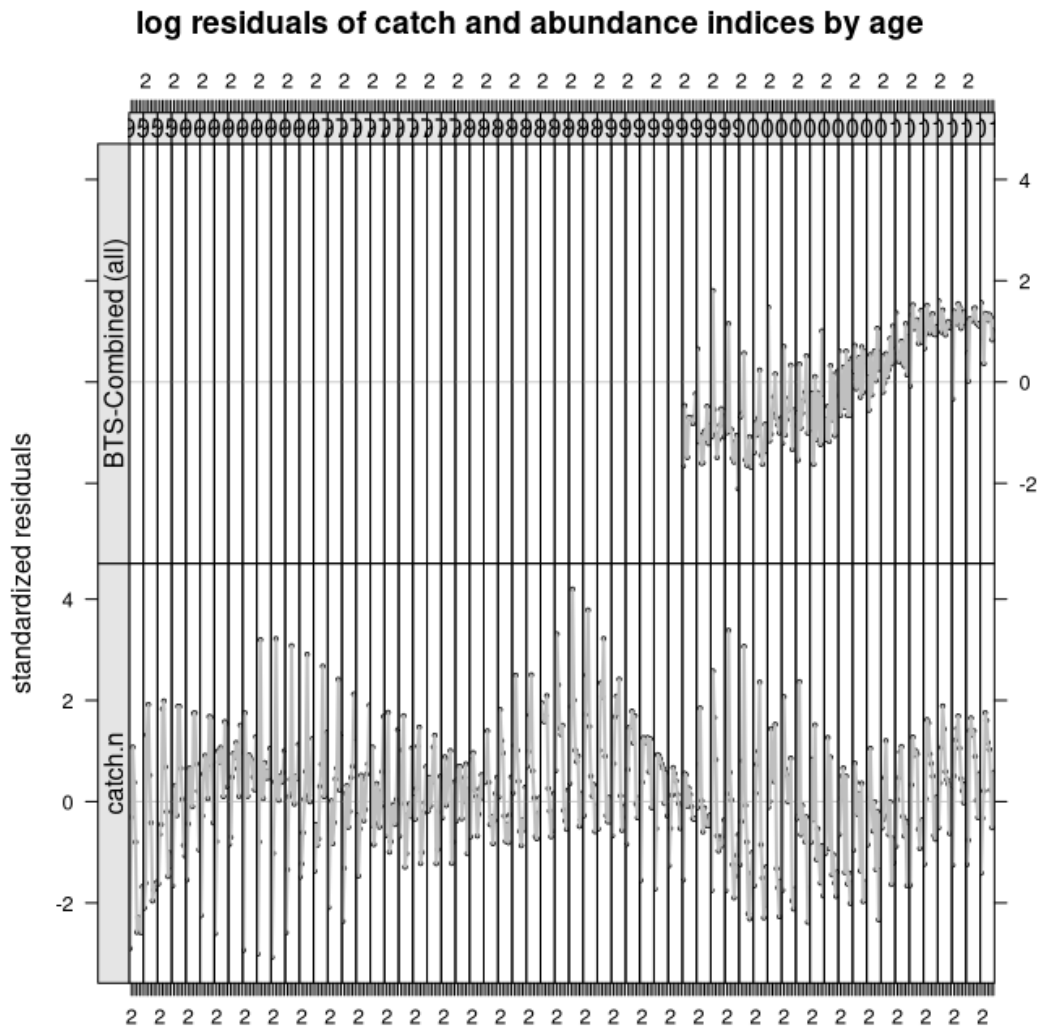
Figure 6: q age effect fit residuals by age)

Finally both effects are brought together.

```
fit04 <- sca(ple4, ple4.index, fmod = ~factor(age), qmod = list(~factor(age)), srmod = ~1,
    vmod = list(~1, ~1), n1mod = ~1)
res04 <- residuals(fit04, ple4, ple4.index)

plot(res04)
```
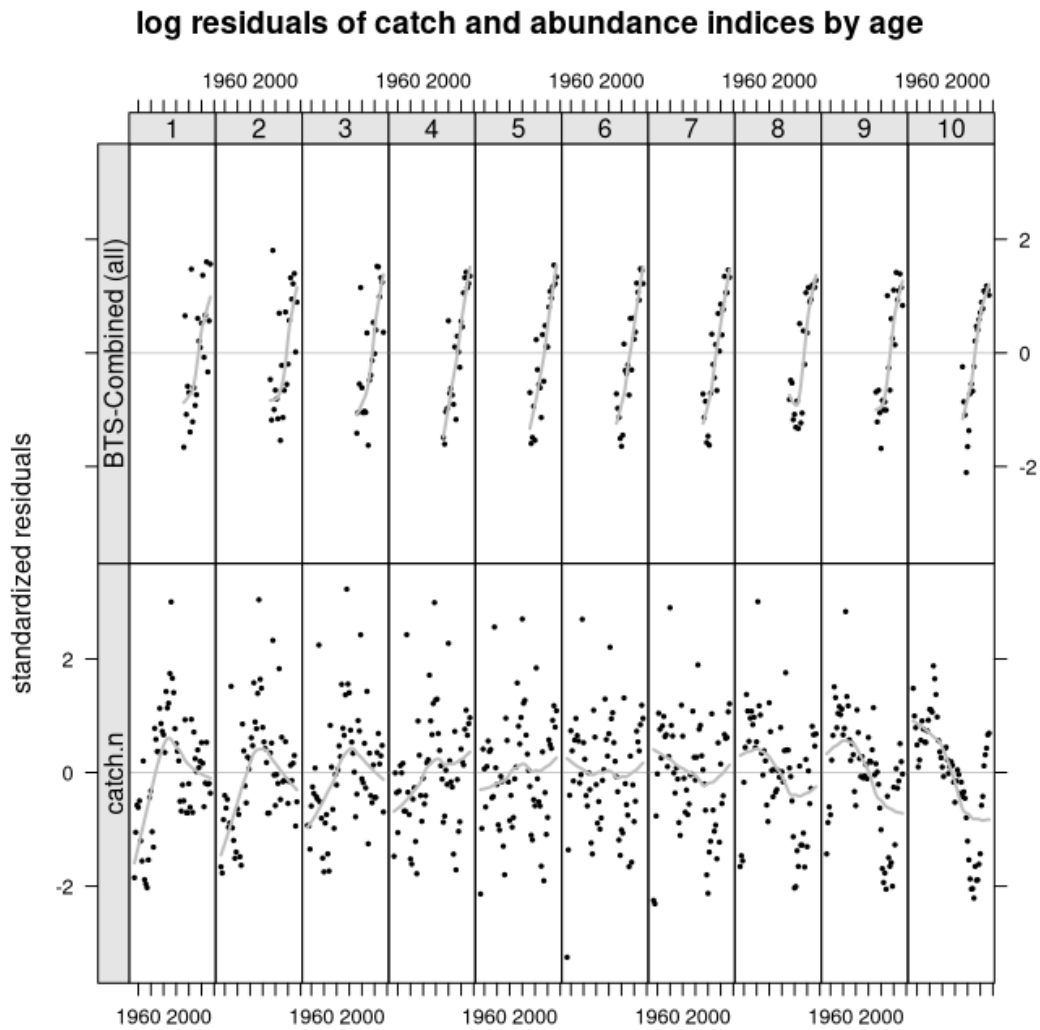
Figure 7: q age effect fit residuals by year)

The residuals plot by age shows the same outcome.

```
plot(res04, auxline = "l", by = "age")
```
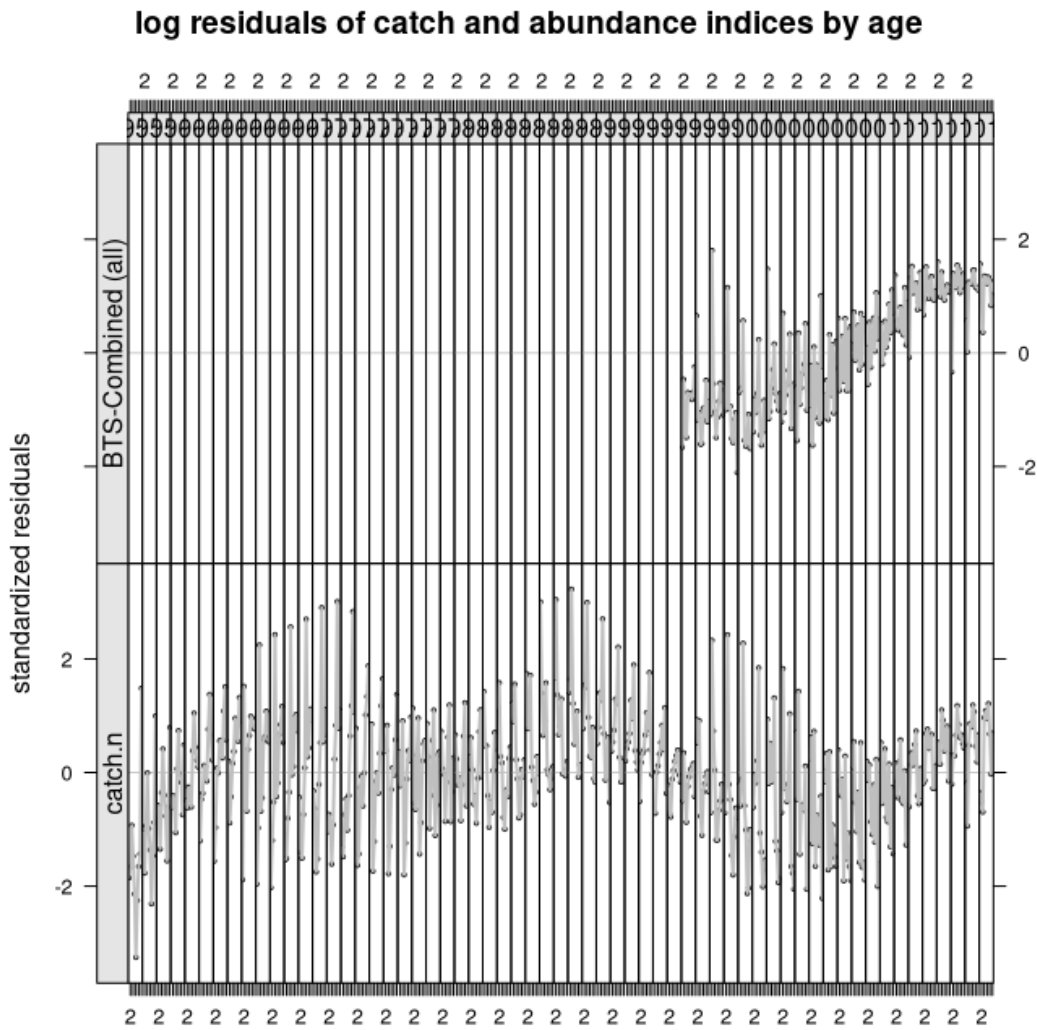
Figure 8: q age effect fit residuals by age)

2.3 The fishing mortality year model

This model will introduce an year effect in the fishing mortality submodel on top of the age effects added before.

```
fit05 <- sca(ple4, ple4.index, fmod = ~factor(age) + factor(year), qmod = list(~factor(age)),
    srmod = ~1, vmod = list(~1, ~1), n1mod = ~1)
res05 <- residuals(fit05, ple4, ple4.index)
```

The residuals plot now shows catch at age residuals stagered as before. The year trends are less pronounced although, because the data doesn't have a very strong year effect, it's less clear than when modelling the age effect.
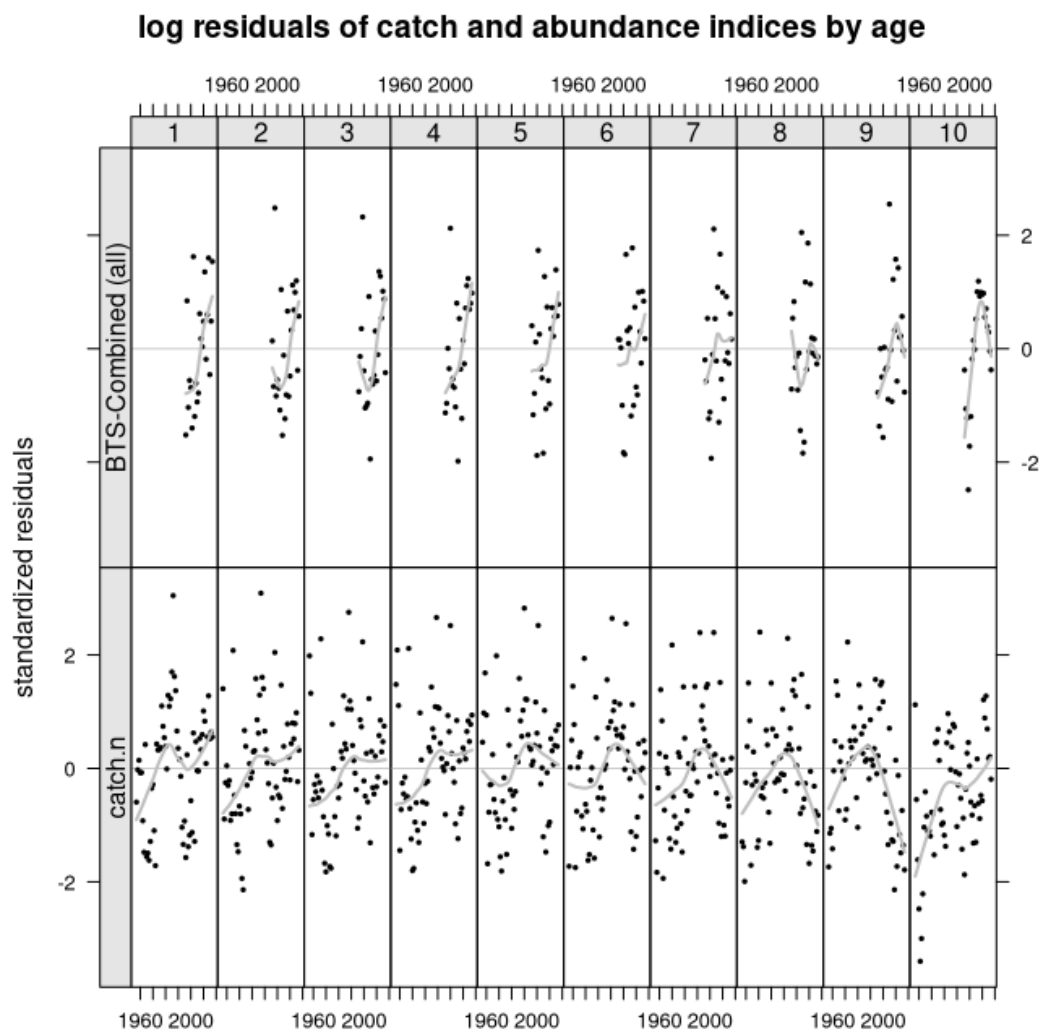
```
plot(res05)
```


Figure 9: f year effect fit residuals by year)

The residuals plot by age shows the same outcome.

```
plot(res05, auxline = "l", by = "age")
```
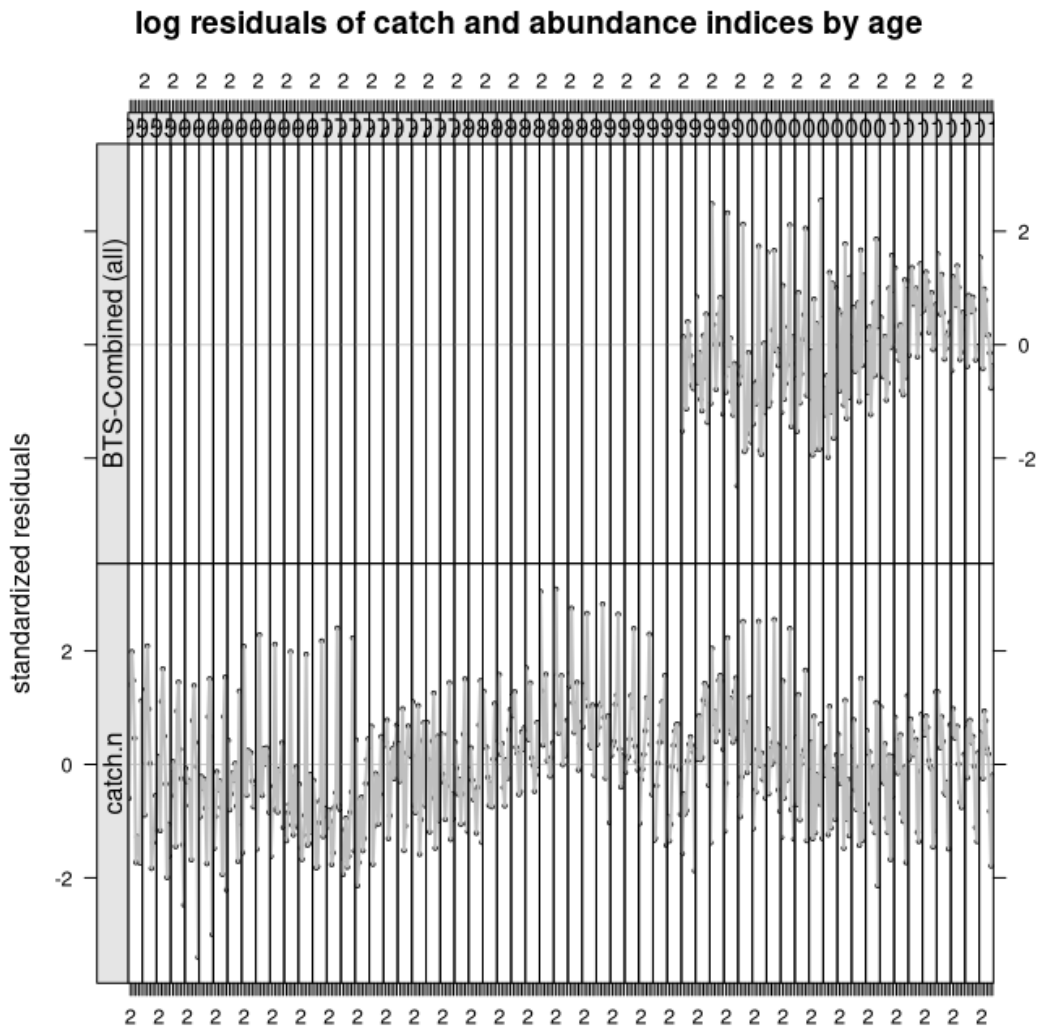
Figure 10: f year effect fit residuals by age)

We can see now that the residuals show a lot less patterns than before. There's still some issues, the survey catchability seems to have an year trend. However the model is not fully specified yet, stock recruitment is modelled as constant over time, the initial population abundance is also modelled as a constant as well as the variance models.

2.4 The initial year population abundance model, aka N1

This model will introduce an age effect in the population abundance in the first year of the time series. This model sets the n-at-age in the first year of the time series, which is needed due to the lack of previous data to reconstruct those cohorts.

```
fit06 <- sca(ple4, ple4.index, fmod = ~factor(age) + factor(year), qmod = list(~factor(age)),
    srmod = ~1, vmod = list(~1, ~1), n1mod = ~factor(age))
res06 <- residuals(fit06, ple4, ple4.index)
```

The residuals plot now shows catch at age residuals stagered as before. The year trends are less

pronounced although, because the data doesn't have a very strong year effect, it's less clear than when modelling the age effect.
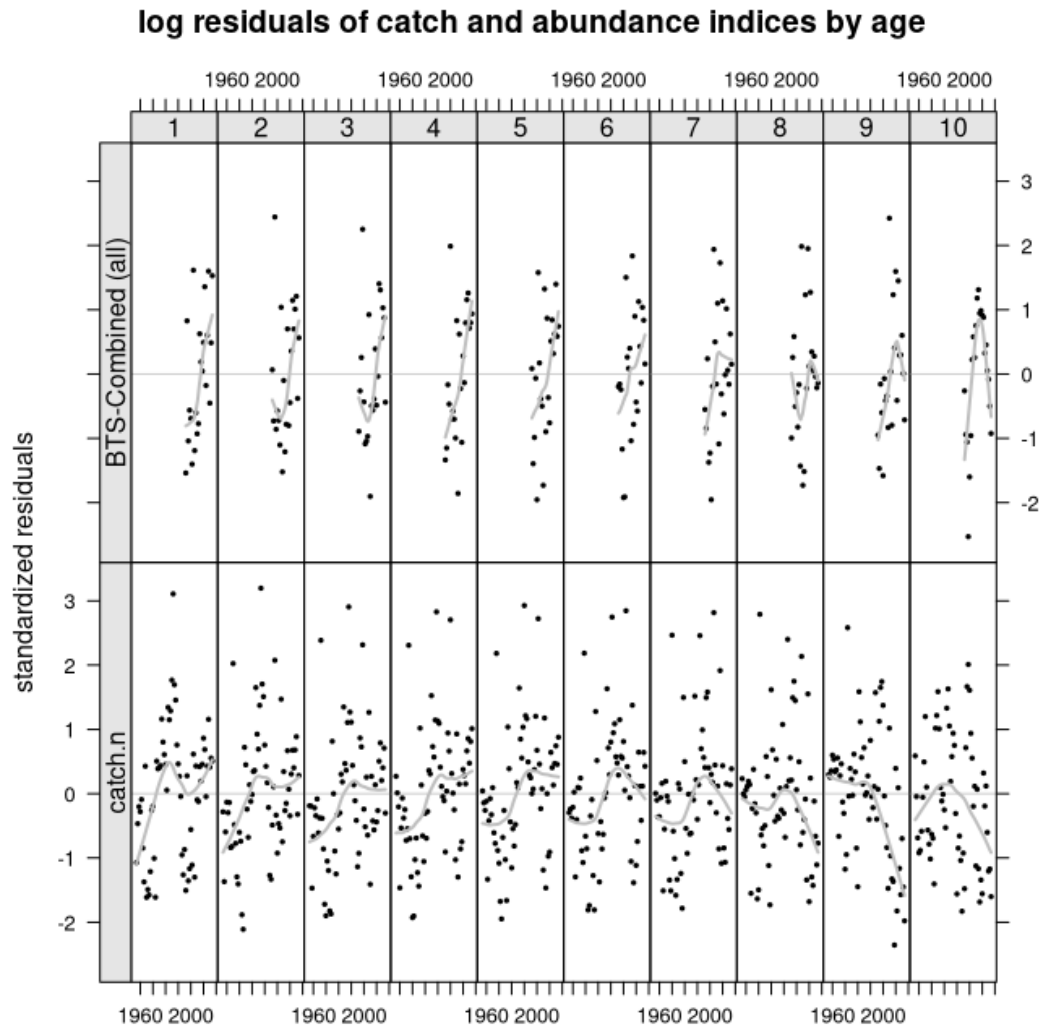
```
plot(res06)
```


Figure 11: f year effect fit residuals by year)

The residuals by age (figure 12) the residuals' improvement in the first year of the catch at age time series (bottom left plots).

```
plot(res06, auxline = "l", by = "age")
```
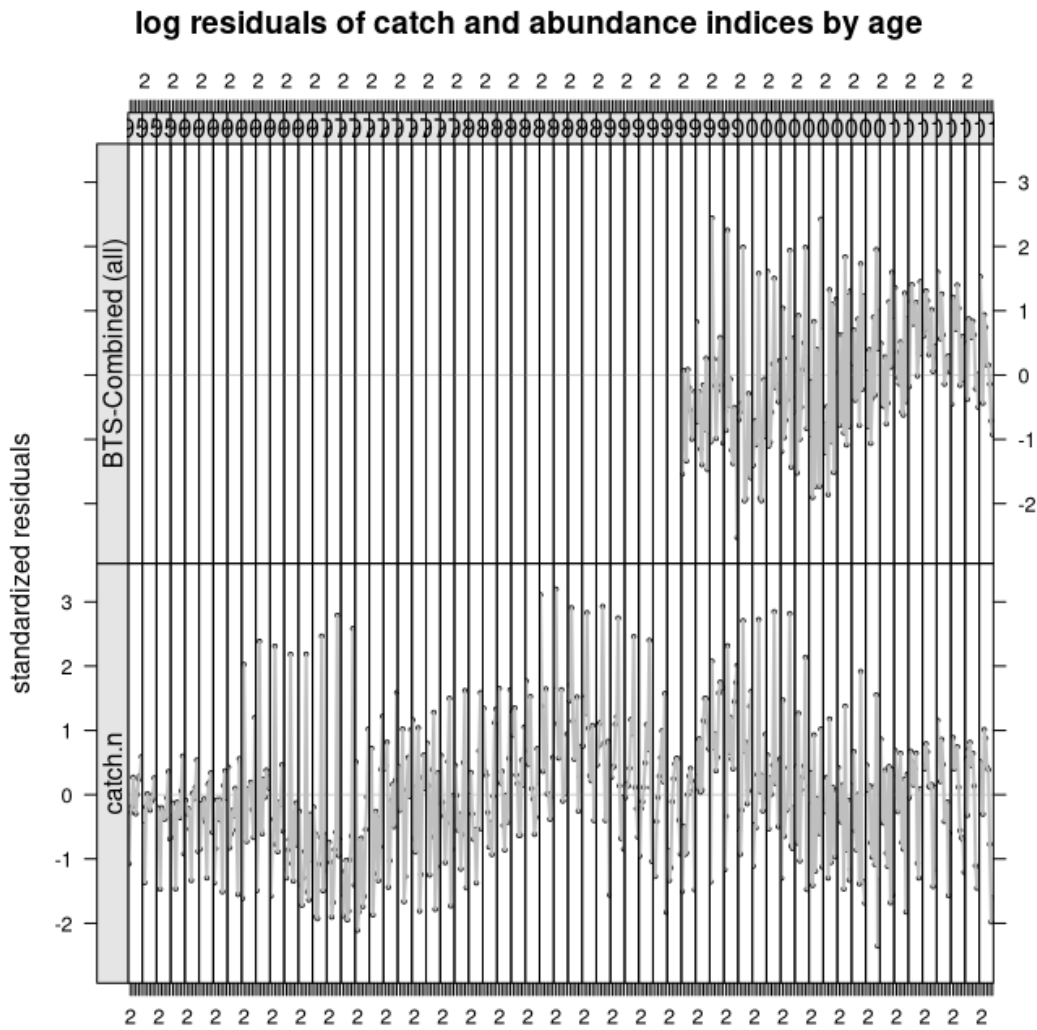
Figure 12: f year effect fit residuals by age)

2.5 The stock recruitment submodel

There's a lot to say about this submodel (check the `a4a`manual for more). In this example we'll simply add a model to allow recruitment to vary over time and we'll see how to track potential improvements in the residuals.

```
fit07 <- sca(ple4, ple4.index, fmod = ~factor(age) + factor(year), qmod = list(~factor(age)),
    srmod = ~factor(year), vmod = list(~1, ~1), n1mod = ~factor(age))
res07 <- residuals(fit07, ple4, ple4.index)
```

The residuals plot by year are very useful to see the effect of adding a varying stock recruitment model. The year trends present in previous models are not absent. Recruitment variability when left unmodelled was being picked up by trends in the survey catchability and catch at age. And due to the cohort dynamics underlying the catch at age model, where propagating into other ages' estimates.
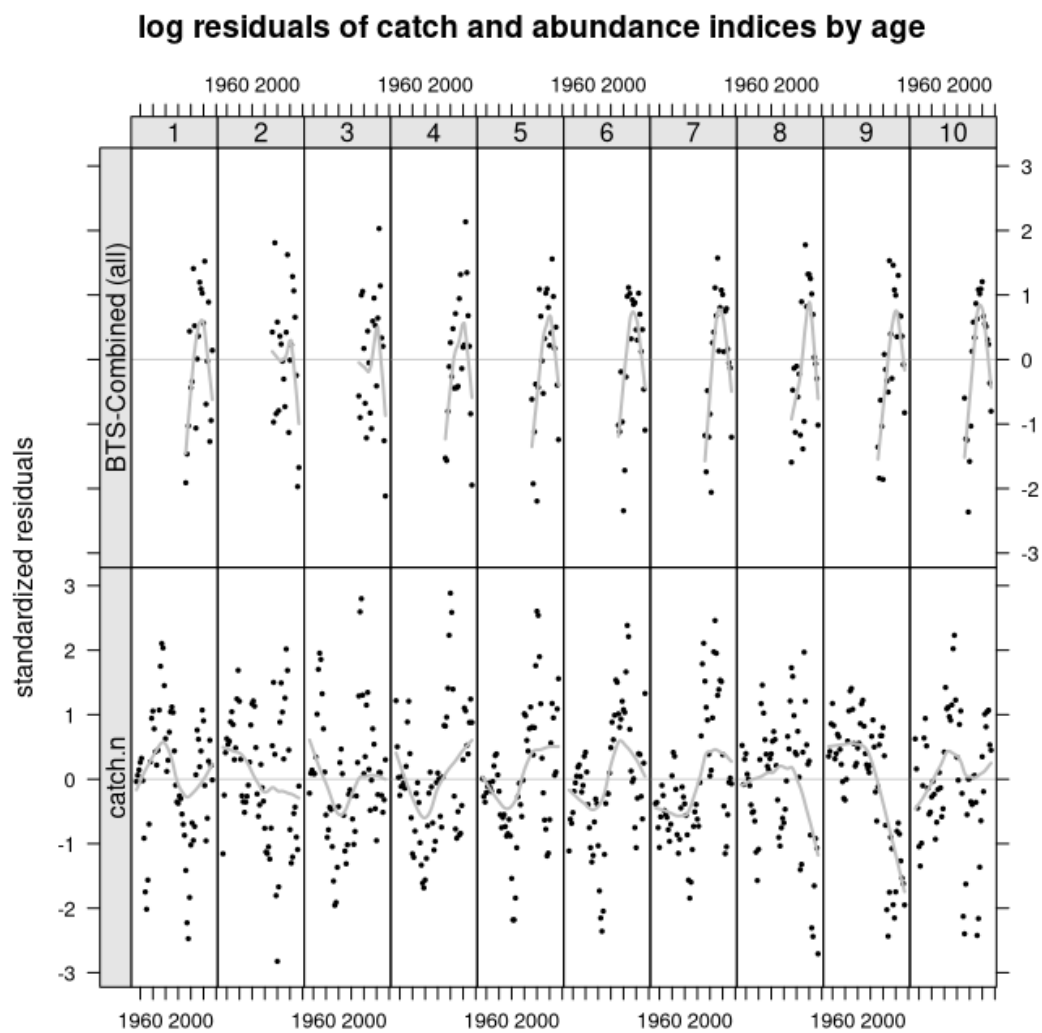
```
plot(res07)
```

log residuals of catch and abundance indices by age

Figure 13: f year effect fit residuals by year)

```
plot(res07, auxline = "l", by = "age")
```
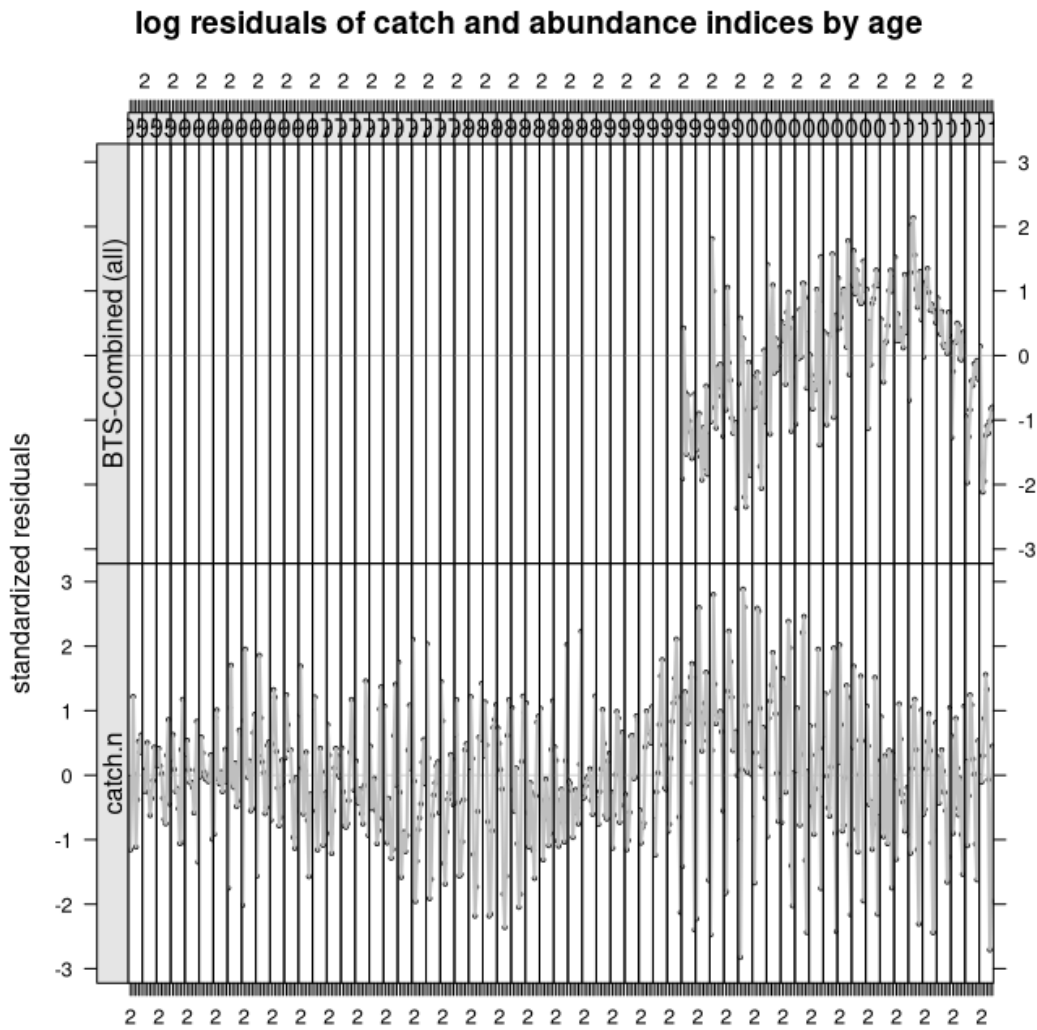
Figure 14: f year effect fit residuals by age)

2.6 Reducing the number of parameters

In this section we'll look into how to use smoothers to reduce the number of parameters of the model. The process is laid out line by line but it could easily be wrapped in a function call to compute a BIC profile.

Due to the number of ages in the data, 4, and the minimum degrees of freedom one can use in a `MGCV` smoother, 3, there's not a lot to gain by playing with age effects. Our first attempt will be with both year effects, in fishing mortality and stock recruitment. Starting with a sparser vector of ks and later focusing in the region of minimal BIC.

```
fit08a <- sca(ple4, ple4.index, fmod = ~factor(age) + s(year, k = 15), qmod = list(~factor(age
    srmod = ~factor(year), vmod = list(~1, ~1), n1mod = ~factor(age))
fit08b <- sca(ple4, ple4.index, fmod = ~factor(age) + s(year, k = 10), qmod = list(~factor(age
    srmod = ~factor(year), vmod = list(~1, ~1), n1mod = ~factor(age))
fit08c <- sca(ple4, ple4.index, fmod = ~factor(age) + s(year, k = 5), qmod = list(~factor(age)
```

```
      srmod = ~factor(year), vmod = list(~1, ~1), n1mod = ~factor(age))

BIC(fit08a, fit08b, fit08c)

##           df       BIC
## fit08a 106 637.7772
## fit08b 101 648.0972
## fit08c  96 673.7392

fit08d <- sca(ple4, ple4.index, fmod = ~factor(age) + s(year, k = 7), qmod = list(~factor(age)
      srmod = ~factor(year), vmod = list(~1, ~1), n1mod = ~factor(age))
fit08e <- sca(ple4, ple4.index, fmod = ~factor(age) + s(year, k = 6), qmod = list(~factor(age)
      srmod = ~factor(year), vmod = list(~1, ~1), n1mod = ~factor(age))
fit08f <- sca(ple4, ple4.index, fmod = ~factor(age) + s(year, k = 4), qmod = list(~factor(age)
      srmod = ~factor(year), vmod = list(~1, ~1), n1mod = ~factor(age))
fit08g <- sca(ple4, ple4.index, fmod = ~factor(age) + s(year, k = 3), qmod = list(~factor(age)
      srmod = ~factor(year), vmod = list(~1, ~1), n1mod = ~factor(age))


BIC(fit08a, fit08b, fit08c, fit08d, fit08e, fit08f, fit08g)

##           df       BIC
## fit08a 106 637.7772
## fit08b 101 648.0972
## fit08c  96 673.7392
## fit08d  98 642.2048
## fit08e  97 636.7929
## fit08f  95 717.6289
## fit08g  94 912.6683
```

Using `k = 5` is the best option if using BIC for comparison.

```
fit08 <- sca(ple4, ple4.index, fmod = ~factor(age) + s(year, k = 5), qmod = list(~factor(age))
      srmod = ~factor(year), vmod = list(~1, ~1), n1mod = ~factor(age))
res08 <- residuals(fit08, ple4, ple4.index)
```
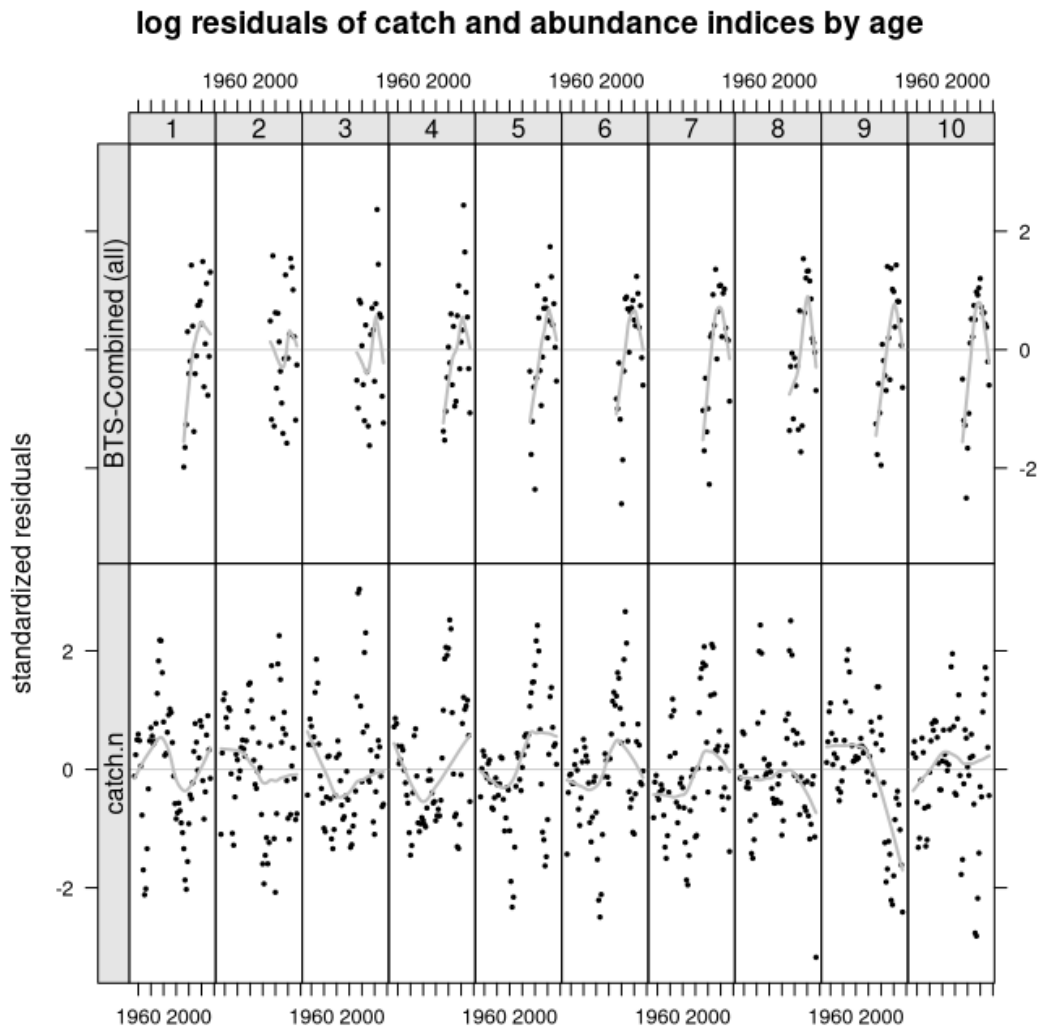
```
plot(res08)
```

Figure 15: f year effect fit residuals by year)

```r
plot(res08, auxline = "l", by = "age")
```
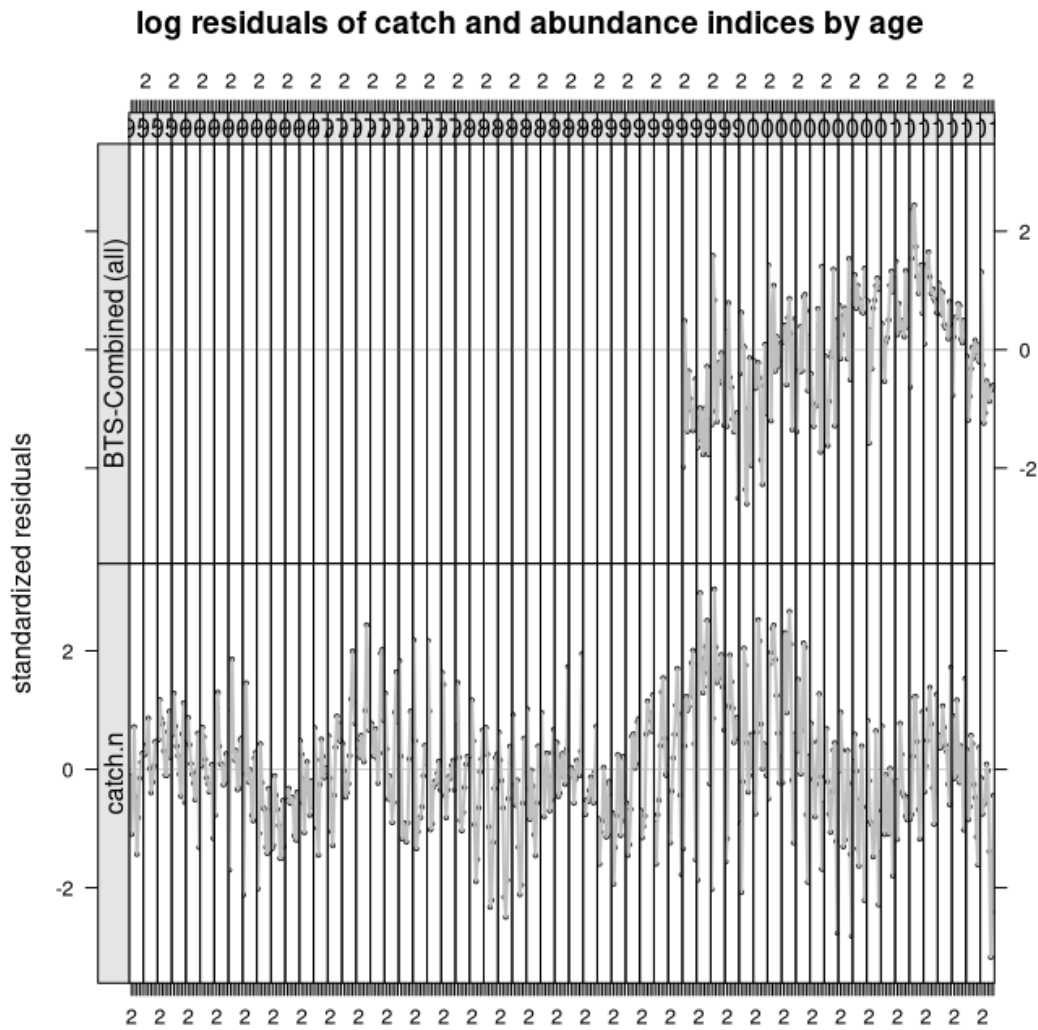
Figure 16: f year effect fit residuals by age)

Following the same approach let's check what can be done with regards to the recruitment model degrees of freedom.

```
fit09a <- sca(ple4, ple4.index, fmod = ~factor(age) + s(year, k = 5), qmod = list(~factor(age)
    srmod = ~s(year, k = 15), vmod = list(~1, ~1), n1mod = ~factor(age))
fit09b <- sca(ple4, ple4.index, fmod = ~factor(age) + s(year, k = 5), qmod = list(~factor(age)
    srmod = ~s(year, k = 10), vmod = list(~1, ~1), n1mod = ~factor(age))
fit09c <- sca(ple4, ple4.index, fmod = ~factor(age) + s(year, k = 5), qmod = list(~factor(age)
    srmod = ~s(year, k = 5), vmod = list(~1, ~1), n1mod = ~factor(age))

BIC(fit09a, fit09b, fit09c)

##          df       BIC
## fit09a 50 1204.243
## fit09b 45 1285.004
## fit09c 40 1326.823
```

```
fit09d <- sca(ple4, ple4.index, fmod = ~factor(age) + s(year, k = 5), qmod = list(~factor(age)
    srmod = ~s(year, k = 7), vmod = list(~1, ~1), n1mod = ~factor(age))
fit09e <- sca(ple4, ple4.index, fmod = ~factor(age) + s(year, k = 5), qmod = list(~factor(age)
    srmod = ~s(year, k = 6), vmod = list(~1, ~1), n1mod = ~factor(age))
fit09f <- sca(ple4, ple4.index, fmod = ~factor(age) + s(year, k = 5), qmod = list(~factor(age)
    srmod = ~s(year, k = 4), vmod = list(~1, ~1), n1mod = ~factor(age))
fit09g <- sca(ple4, ple4.index, fmod = ~factor(age) + s(year, k = 5), qmod = list(~factor(age)
    srmod = ~s(year, k = 3), vmod = list(~1, ~1), n1mod = ~factor(age))


BIC(fit09a, fit09b, fit09c, fit09d, fit09e, fit09f, fit09g)

##          df      BIC
## fit09a 50 1204.243
## fit09b 45 1285.004
## fit09c 40 1326.823
## fit09d 42 1302.922
## fit09e 41 1333.478
## fit09f 39 1420.317
## fit09g 38 1418.858
```

The best model according to BIC is again `k = 5` for the recruitment submodel and as such the best model we get is:

```
fit09 <- sca(ple4, ple4.index, fmod = ~factor(age) + s(year, k = 5), qmod = list(~factor(age))
    srmod = ~s(year, k = 5), vmod = list(~1, ~1), n1mod = ~factor(age))
res09 <- residuals(fit09, ple4, ple4.index)
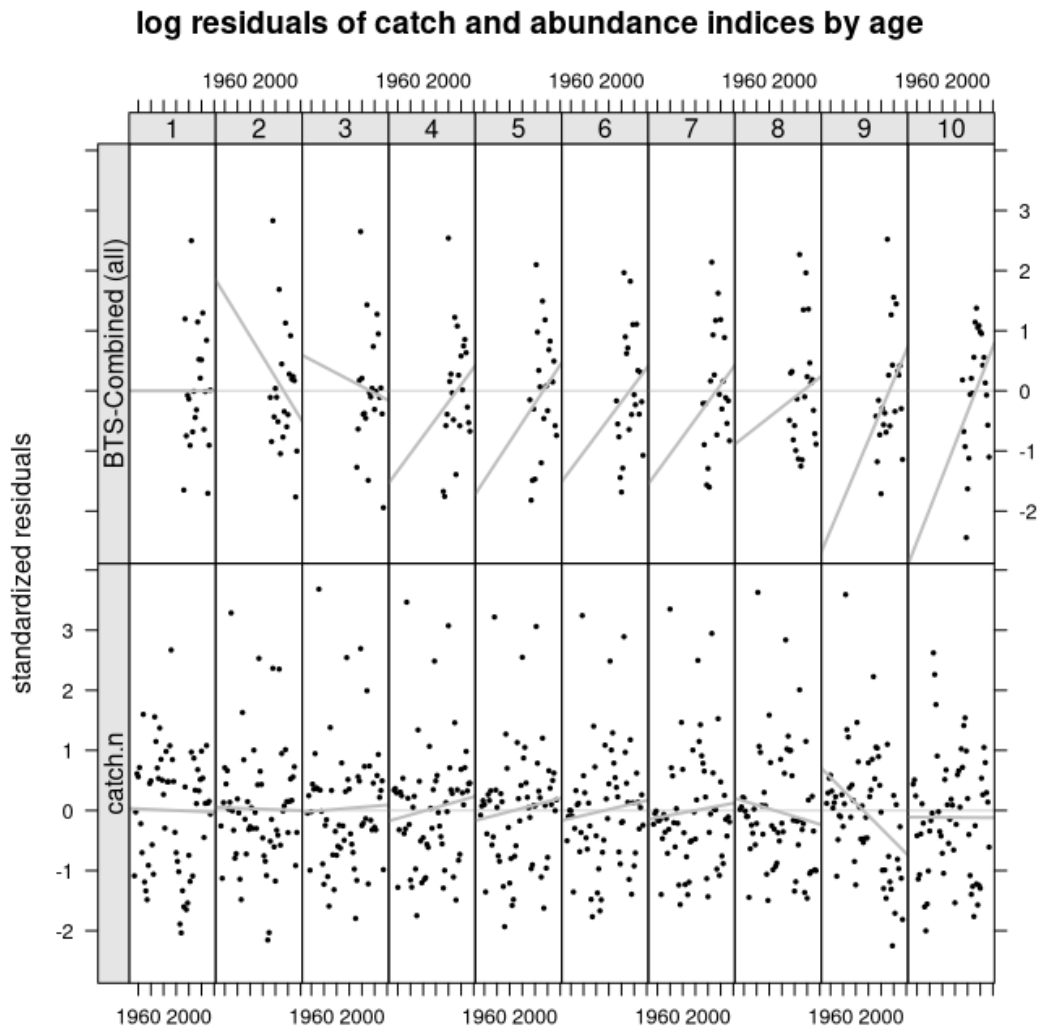```

```
plot(res09, auxline = "r")
```

Figure 17: f year effect fit residuals by year)

```
plot(res09, auxline = "l", by = "age")
```
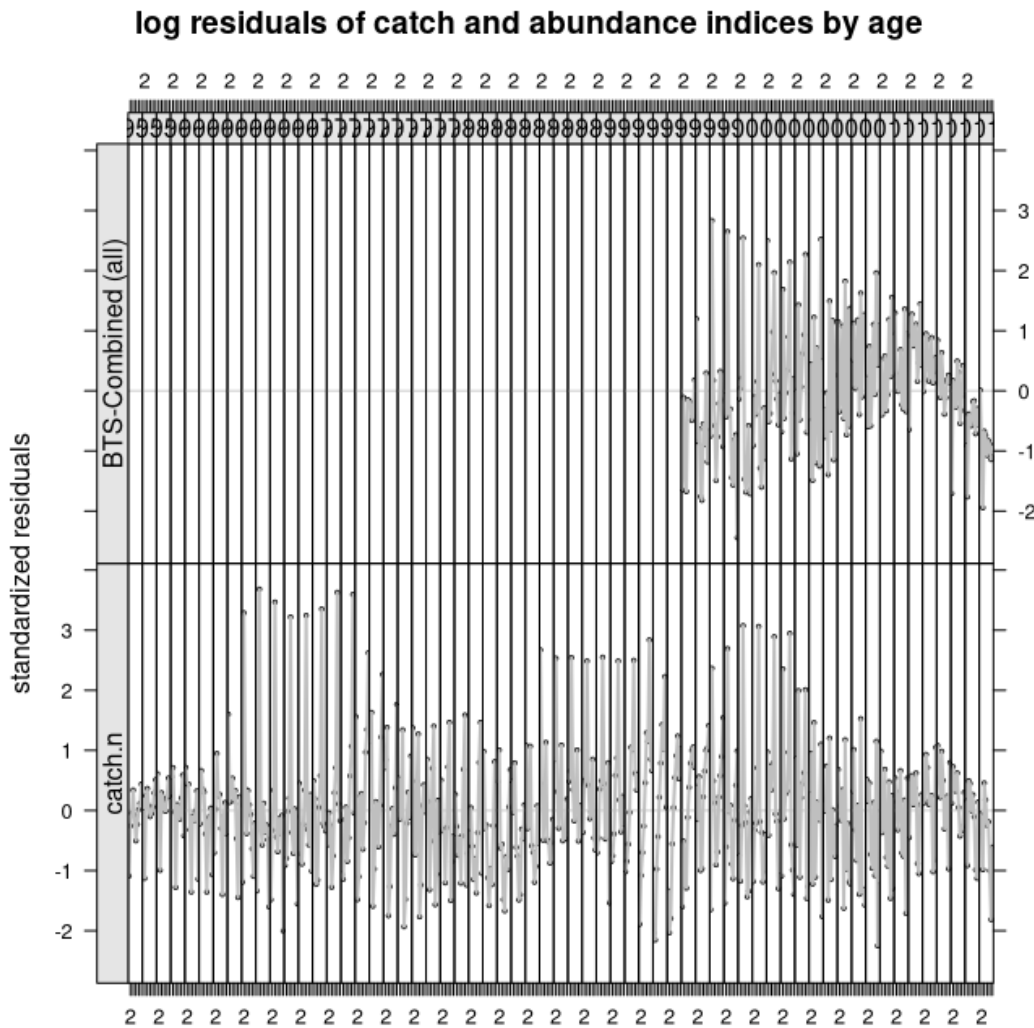
Figure 18: f year effect fit residuals by age)

2.7 The variance submodel

Finally, we're testing the variance submodel, specifically the catch at age variance model. We won't dig into the catchability variance model though. It's common to accept that a scientific survey following a well designed sampling protocol will have equal variance across ages since no preferential areas are sampled.

```
fit10 <- sca(ple4, ple4.index, fmod = ~factor(age) + s(year, k = 5), qmod = list(~factor(age))
    srmod = ~s(year, k = 5), vmod = list(~s(age, k = 4), ~1), n1mod = ~factor(age))
res10 <- residuals(fit10, ple4, ple4.index)
```

To see what's happening with the variance model one can use predict to plot the different models fitted.

```
flqs <- FLQuants(mod10 = predict(fit10)$vmodel$catch[, "2022"], mod09 = predict(fit09)$vmodel$
    "2022"])
```

```
## Error in x@.Data[i, j, k, l, m, n, drop = FALSE]: subscript out of bounds

xyplot(data ~ age, data = flqs, group = qname, type = "l", auto.key = T)
```
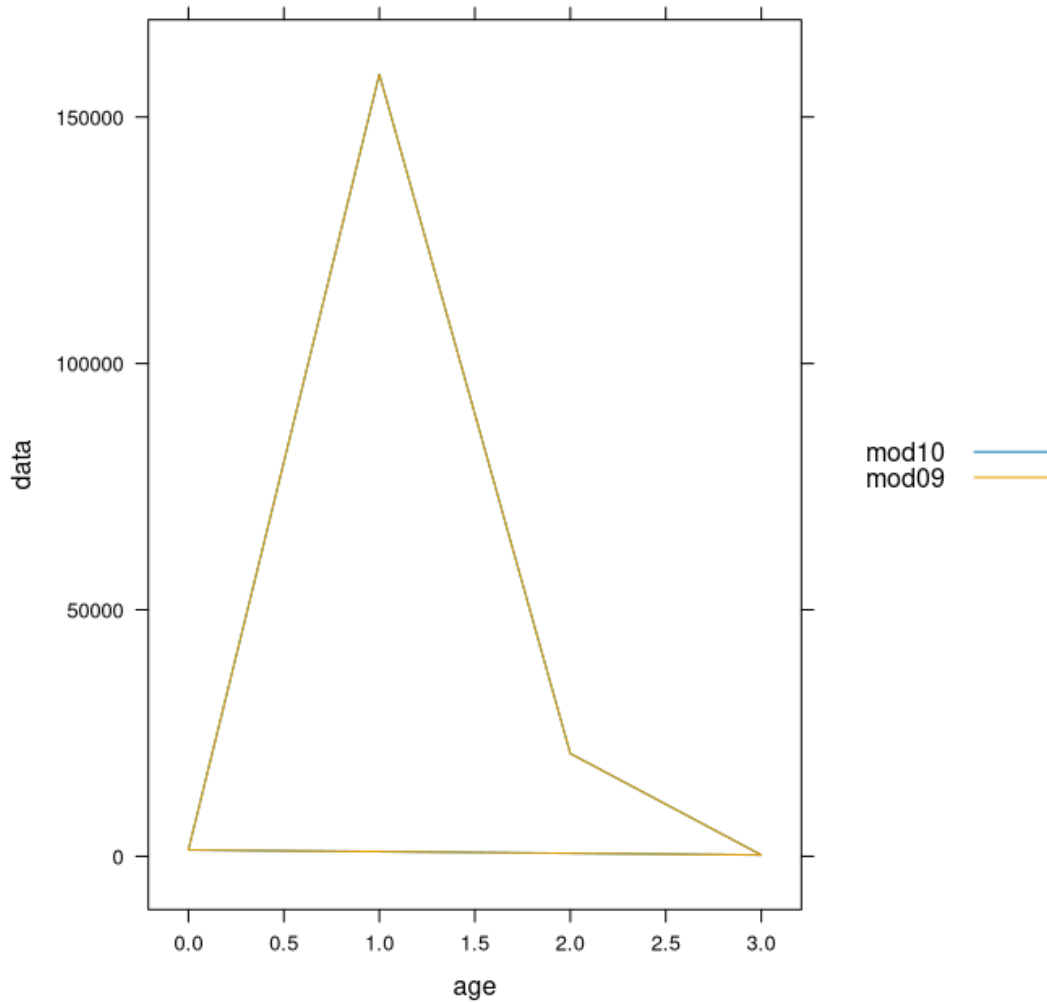

Figure 19: Variance models for catch at age

To see the effect these models have on the estimated quantities one can look at the variance of the estimates:

```
flqs <- FLQuants(mod10 = catch.n(ple4 + simulate(fit10, nsim = 500))[, "2022"], mod09 = catch.
    simulate(fit09, nsim = 500))[, "2022"])

## Error in x@.Data[i, j, k, l, m, n, drop = FALSE]: subscript out of bounds

bwplot(data ~ qname | factor(age), data = as.data.frame(flqs), scales = "free", auto.key = T)
```
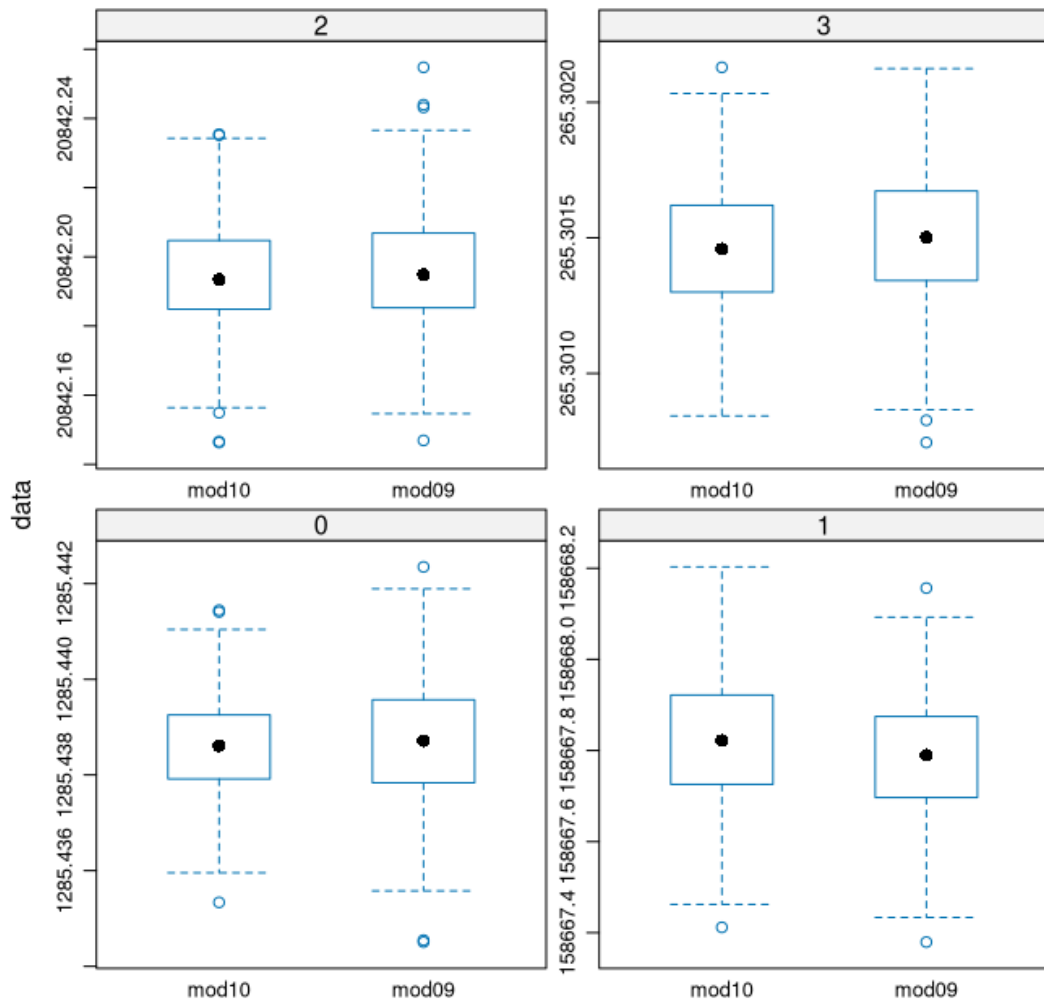
Figure 20: Estimates of population abundance with different variance models

and the usual residuals
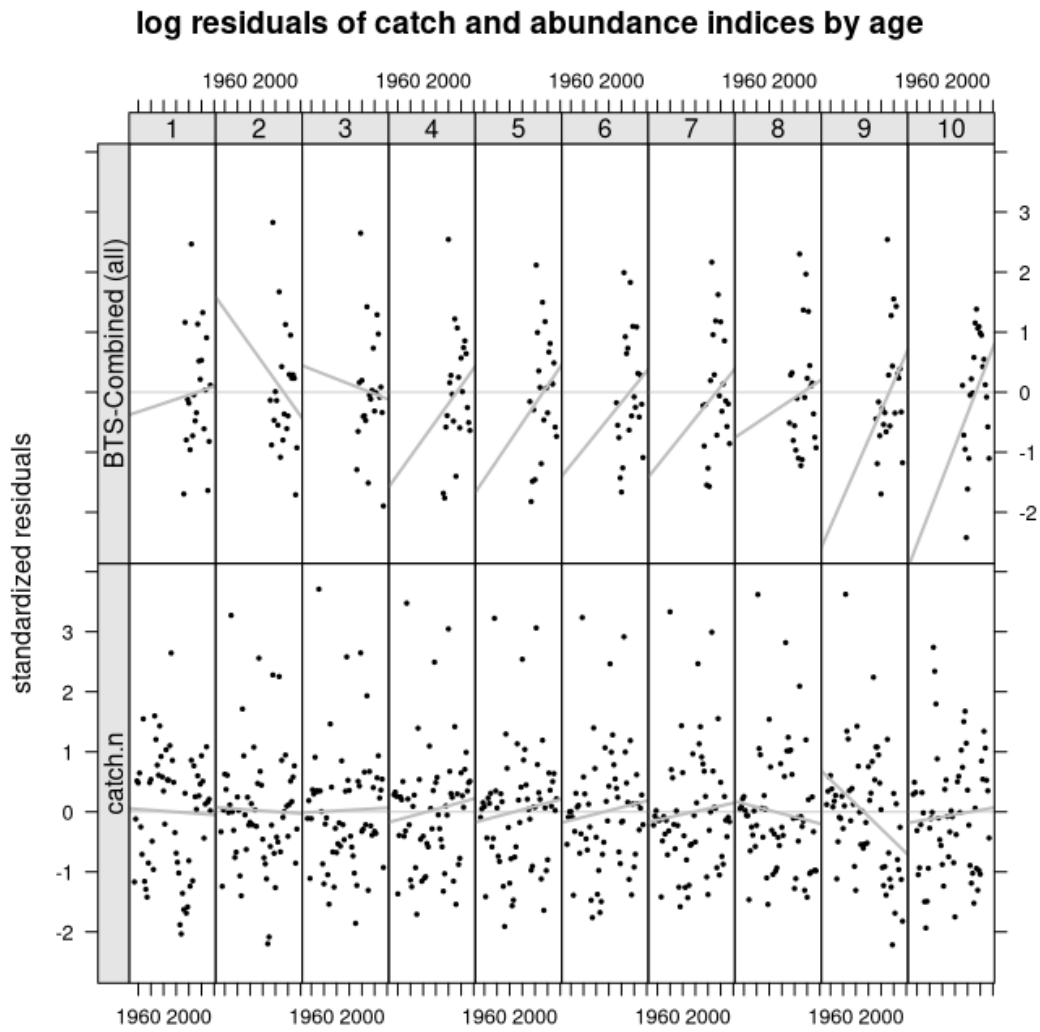
```
plot(res10, auxline = "r")
```

Figure 21: f year effect fit residuals by year)
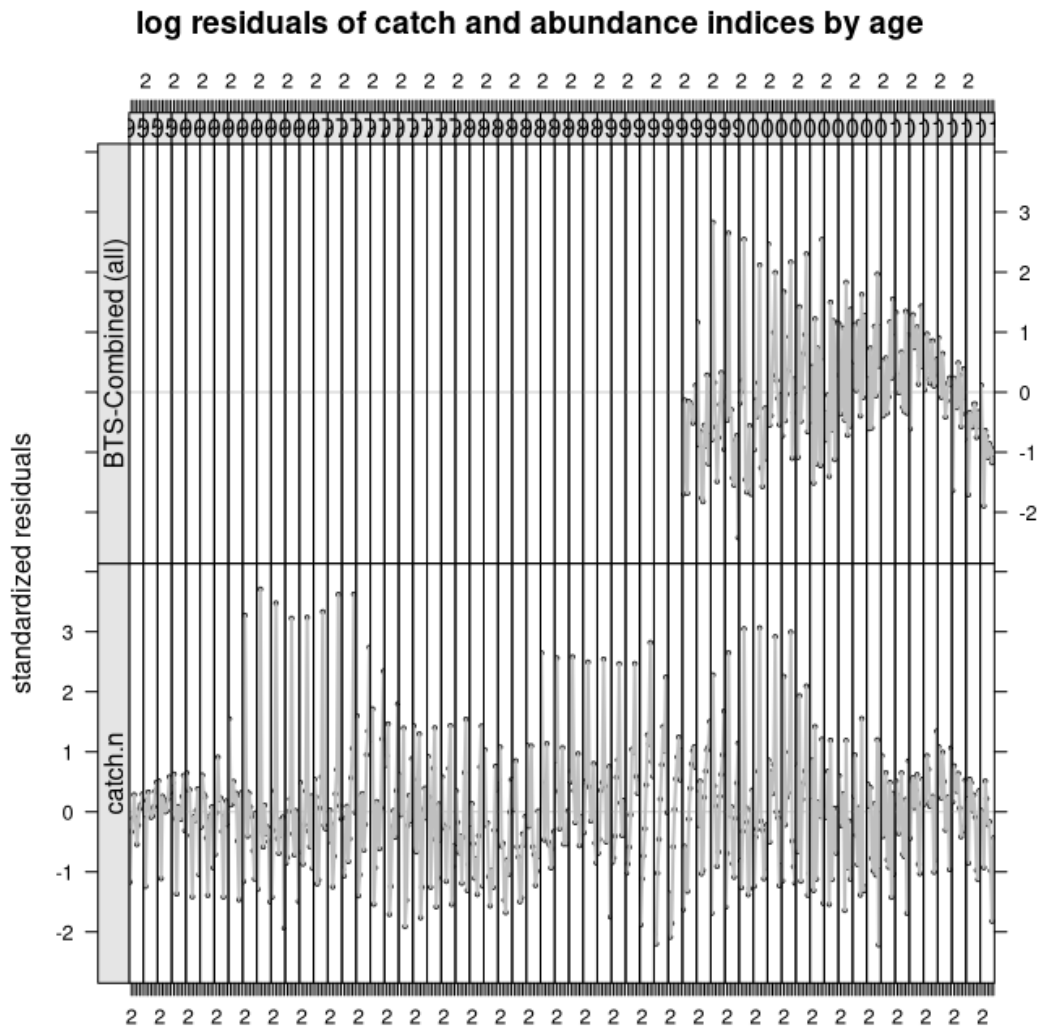
```
plot(res10, auxline = "l", by = "age")
```

Figure 22: f year effect fit residuals by age)