European Commission

JRC SCIENTIFIC AND POLICY REPORTS

a4a short research project
-
Stock assessment and multi-fleet forecast methods forsome Mediterranean stocks

Ernesto Jardim, Dimitrios Damalas, Iago Mosqueira, Chato Osio, Tristan Rouyer,

Giuseppe Scarcella & Finlay Scott

Report EUR26716 EN

Joint Research Centre

European Commission

Joint Research Centre

Institute for the Protection and Security of the Citizen

Contact information

Ernesto Jardim

Address: Maritime Affairs Unit, Via Enrico Fermi 2749, 21027Ispra VA, Italy

E-mail: ernesto.jardim@jrc.ec.europa.eu

Tel.: 0039 0332 785311

Fax: 0039 0332 789658

http://ipsc.jrc.ec.europa.eu/

http://www.jrc.ec.europa.eu/

JRC 91120

EUR 26716 EN

ISBN 978-92-79-38997-9

ISSN 1831-9424

doi:10.2788/88416

Luxembourg: Publications Office of the European Union, 2014

Printed in Italy

a4a short research project
Stock assessment and multi-fleet forecast methods for some Mediterranean stocks

Ernesto Jardim (Editor)[1], Dimitrios Damalas[1], Iago Mosqueira[1], Chato Osio[1], Tristan Rouyer[2], Giuseppe Scarcella[3], and Finlay Scott[1]

[1]*JRC - Institute of the Protection and Security of the Citizen, European Commission, TP 051, Via Enrico Fermi 2749, I-21027 Ispra (VA), Italy*
[2]*IFREMER, UMR 212 EME, Avenue Jean Monnet, BP 171, 34203 Sete (France)*
[3]*National Research Council - Institute of Marine Sciences, L.go fiera della pesca SNC, 60125 Ancona (Italy)*

September 24, 2014

Contents

1 Introduction

Ernesto Jardim

Under the scope of the a4a Initiative, the JRC is promoting cooperative activities between fisheries scientists with the aim to test, disseminate and promote a4a methods. These Small Research Projects (SRP) are focus on comparing the results of assessments from other models to assessments obtained from the a4a statistical catch-at-age model, and explore research questions using case studies.

The Workshop dedicated to the Mediterranean took place in Ispra, Italy, the 23th to the 27th of June. The main objectives were to compare assessment models and develop multi-fleet forecasts methodologies. These can be applied in the context of ex-ante/ex-post evaluations of multi-annual plans, performed by STECF in order to provide scientific advice to the European Commission.

1.1 ToR and Agenda

The terms of reference of the workshop were:

- Assess the stocks of hake in GSA 7 and sole in GSA 17 with a4a and compare results with other models.
- Develop fleet forecasting algorithms considering requirements of multi-annual management plans.
- Test the methods above in other stocks.
- Report to STECF and other relevant management bodies.

The first three days of the workshop were dedicated to fit the a4a statistical catch-at-age method to the stocks of sardine, anchovy and sole in the North Adriatic (GSA17), Hake in the Gulf of Lions (GSA08) and in the South Adriatic (GSA18). The fourth and fifth days were dedicated to compute partial fishing mortalities for the fleets targeting Sole in the North Adriatic and runing forecasts under distinct scenarios. The scenarios were designed to test possible management of the fleets by constraining their effort in different management objectives.

1.2 The a4a Initiative

(This section is based on Jardim, et.al, 2014)

The volume and availability of data useful for fisheries stock assessment is continually increasing. Time series of traditional sources of information, such as surveys and landings data are not only getting longer, but also cover an increasing number of species.

For example, in Europe the 2009 revision of the Data Collection Regulation (EU, 2008a) has changed the focus of fisheries sampling programmes away from providing data for individual assessments of key stocks (i.e. those that are economically important) to documenting fishing trips, thereby shifting the perspective to a large coastal monitoring programme. The result has been that data on growth and reproduction of fish stocks are being collected for more than 300 stocks in waters where the European fleets operate.

Recognizing that the context above required new methodological developments, the European Commission Joint Research Centre (JRC) started its Assessment for All Initiative (`a4a`), with the aim to develop, test, and distribute methods to assess a large numbers of stocks in an operational time frame, and to build the necessary capacity/expertise on stock assessment and advice provision.

The long-term strategy of `a4a` is to increase the number of stock assessments while simultaneously promoting the inclusion of the major sources of uncertainty in scientific advice. Our aim is to reduce the required workload by developing a software framework with the methods required to run the analysis a stock assessment needs, including methods to deal with recognized bottlenecks, *e.g.* model averaging to deal with model selection (Millar, et.al, 2014). Moreover, we aim to make the analysis more intuitive, thereby attracting more experts to join stock assessment teams. Having more scientists/analysts working in fisheries management advice will increase the human resource basis, which is currently recognized to be limited. Regarding the former, `a4a` promotes a risk analysis approach to scientific advice through a wider usage of Operating Model/MSE approaches. We're focused on developing methods that can deal with the most common settings these type of analysis require, and creating the conditions for scientists to develop their own methods. Our expectation is that having a common framework, with clear data structures and workflows, will promote research in this area and make it simpler to implement and share methods.

To achieve these objectives, the Initiative identified a series of tasks, which were or are being carried out, namely:

- define a moderate data stock;
- develop a stock assessment framework;
- develop a forecasting algorithm based on MSE;
- organize training courses for marine scientists.

1.3 The a4a approach to stock assessment and management advice

As stated before, one of the main objectives of `a4a` is to promote a risk type of analysis, so that scientific advice provides policy and decision makers a perspective of the uncertainty existing on stock assessments and its propagation into the scenarios being analyzed.

The sources of uncertainty implemented so far are related with the processes of growth, natural mortality and reproduction (stock-recruitment); and with the estimation of population abundance and fishing mortality by the stock assessment model. In all cases the framework can include sampling error.

The approach is split into 4 steps: (i) converting length data to age data using a growth model, (ii) modeling natural mortality, (iii) assessing the stock, and (iv) MSE.

These steps may be followed in sequence or independently, depending on the user's preferences. All that is needed is to use the objects provided by the previous step and provide the objects required by the next, so that data flows between steps smoothly. One can make the analogy with building with Lego, where for each layer the builder may use the pieces provided by a particular boxset, or make use of pieces from other boxsets. Figure 1.1 shows the process, including the class of the objects that carry the data (in black).

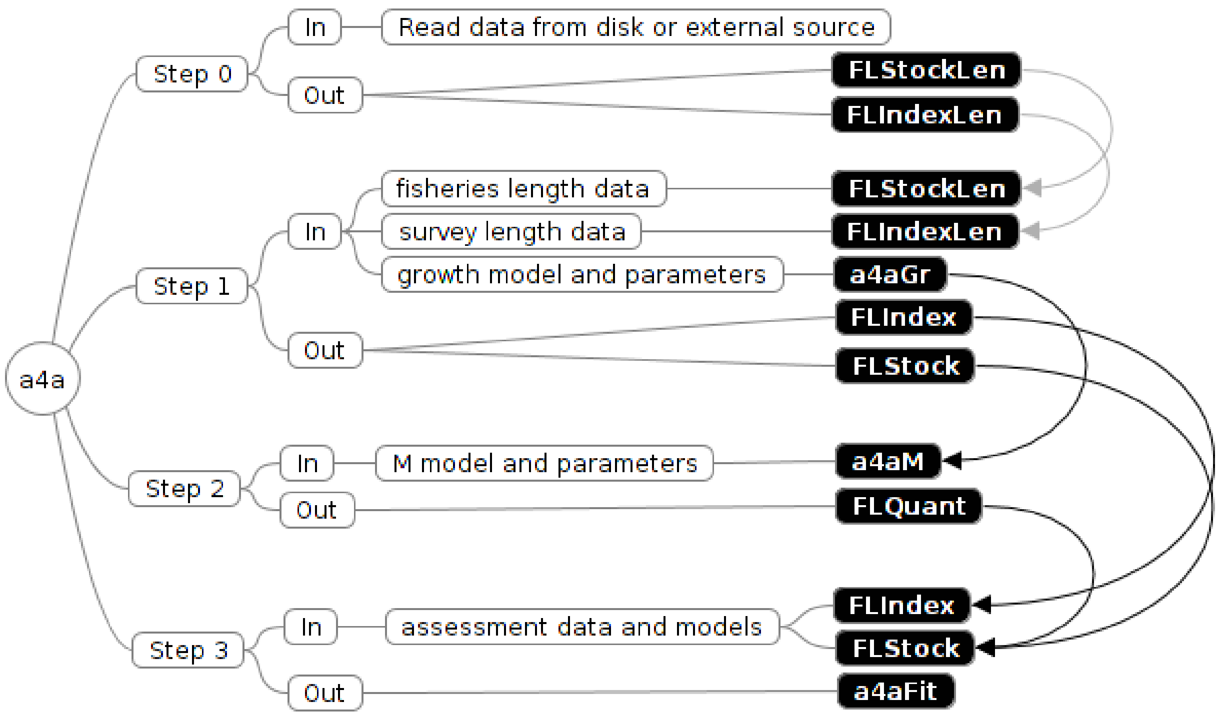


Figure 1.1: In/out process of the `a4a` approach. The boxes in black represent the classes of the objects that carry the information in and out of each step.

Analysis related to projections and biological reference points are dealt with by the `FLR` packages `FLash` and `FLBRP`.

In Steps 1 and 2 there is no fitting of growth models or natural mortality models. The rationale is to provide tools that allow the uncertainty associated with these processes to be carried on into the stock assessment, e.g. through parameter uncertainty. This approach allows the users to pick up the required information from other sources of information such as papers, PhDs, Fishbase, other stocks, etc. If the stock under analysis does not have specific information on the growth or natural mortality processes, generic information about life history invariants may be used such as the generic priors suggested by Bentley, (2014).

Note that an environment like the one distributed by `a4a` promotes the exploration of different models for each process, giving the analyst a lot of flexibility. It also opens the possibility to efficiently include distinct models in the analysis. For example, a stock assessment using two growth, or several models for natural mortality could be performed. Our suggestion to streamline the assessment process is to combine the final outcomes using model averaging (Miller, et. al, 2014). Other solutions may be implemented, like scenario analysis, etc. What is important is to keep the data flowing smoothly and the models clear. R (R Core Team, 2014) and `FLR` (Kell, et.al, 200) provide powerful platforms for this approach.

1.4 How to read this document

The target audience for this document are stock assessment experts. It presents a mixture of text and code that shows how the analysis can be run with R/FLR/FLa4a. Moreover, having the code allows the reader to copy/paste and replicate the analysis presented here.

The chapters are as independent as possible, so they can be extracted and runed individually.

1.5 Software packages - FLR & FLa4a

To run the `FLa4a` methods the reader will need to install the package and its dependencies and load them, together with a couple of other packages. The data sets can be made available upon request.

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

To replicate the analysis carried out in this document the user will need the following additional packages:

```
# from CRAN
install.packages(c("plyr", "xtable", "plot3D", "gridExtra", "ggplot2"))
# from FLR
pkgs <- c("FLXSA", "FLAssess", "FLSAM", "FLash", "FLBRP")
install.packages(pkgs, repos = "http://flr-project.org/R")
```

Loading !

```r
library(FLa4a)
library(FLBRP)
library(FLXSA)
library(xtable)
library(plyr)
library(plot3D)
library(FLSAM)
library(gridExtra)
```

2 Anchovy in the North Adriatic Sea

Chato Osio

2.1 Explore Anchovy stock in GSA 17

Bring in the ANCHOVY data from GSA 17, from the assessment performed during STECF EWG 13-19 and stored on github

```
load("data/Anchovy GSA 17.RData")
```

Explore the raw data, catch matrix and index, plus index internal consistency

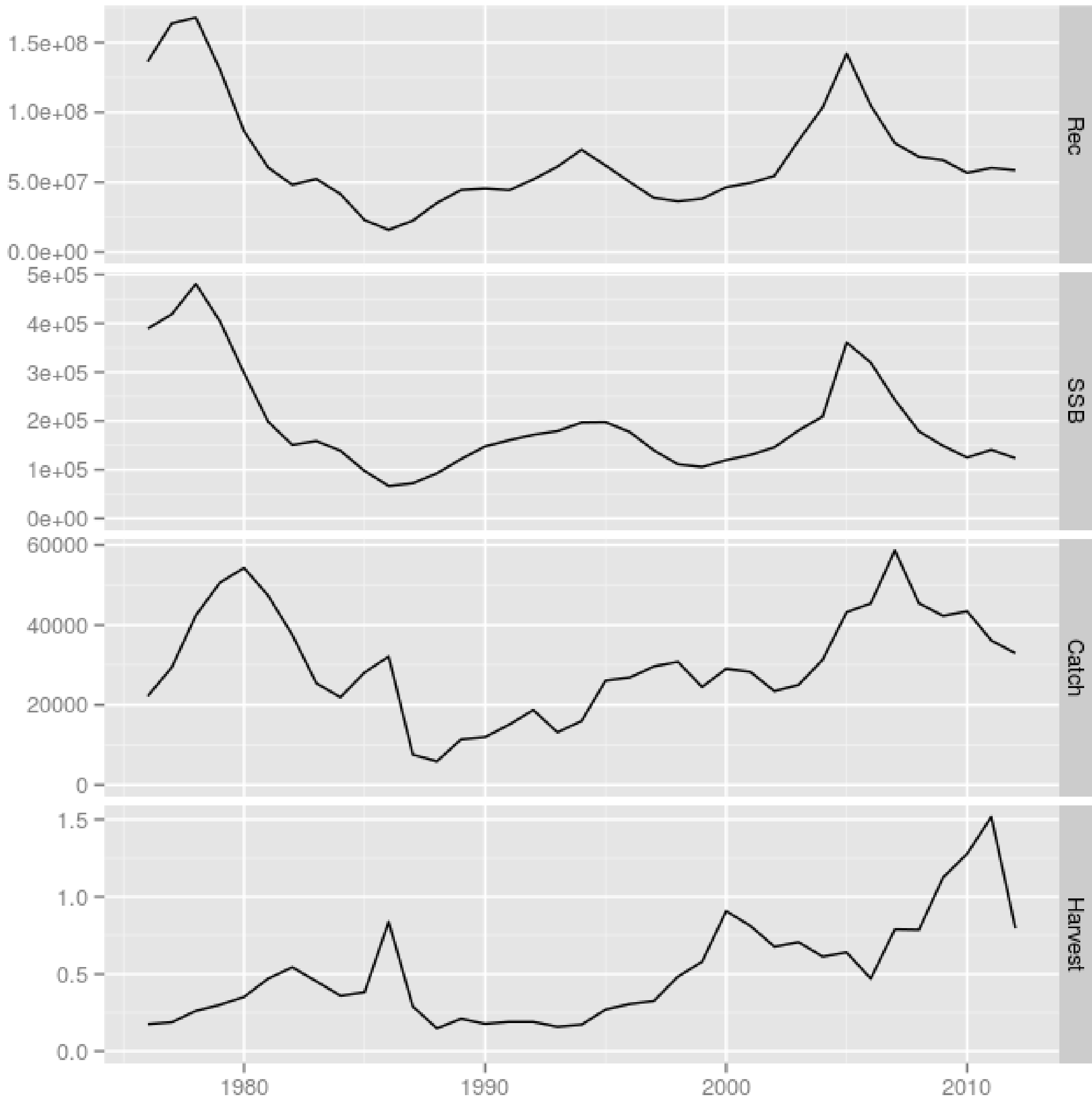


Figure 2.1: Anchovy stock

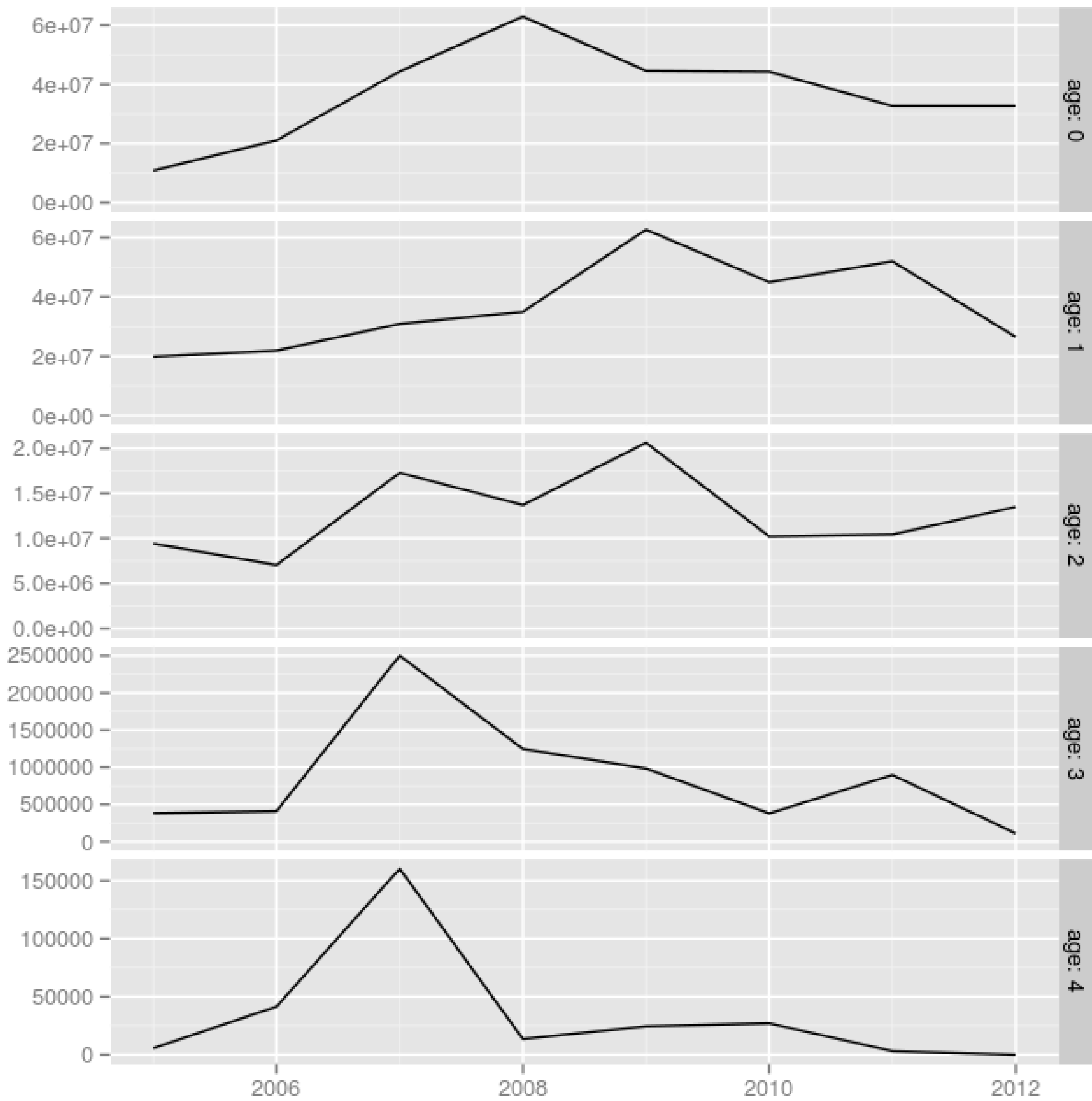


Figure 2.2: Anchovy tuning index

The internal consistency in the West+East survey is not good for almost all age classes indicating problems with this merged index.

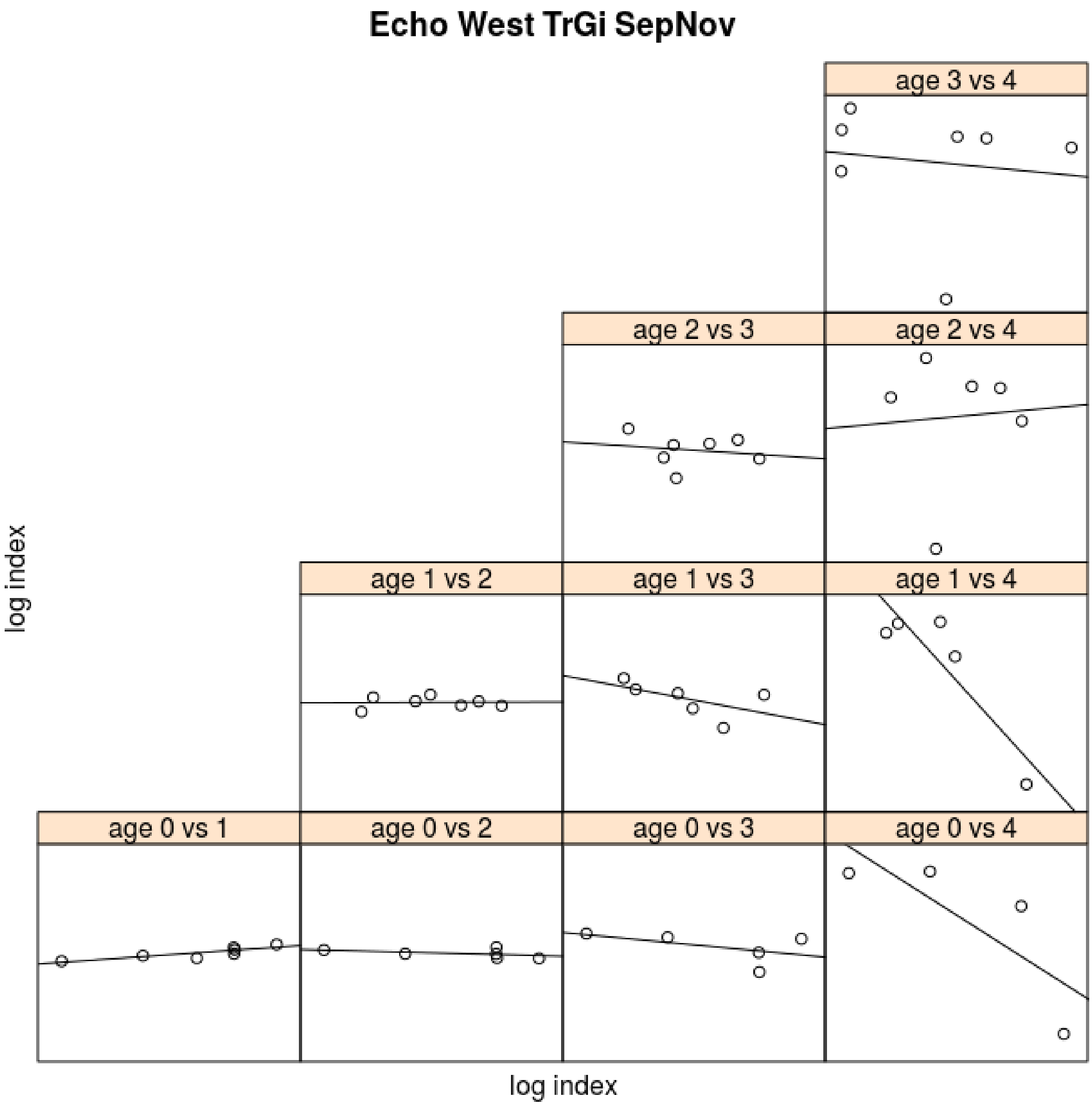


Figure 2.3: Anchovy tuning index internal consistency

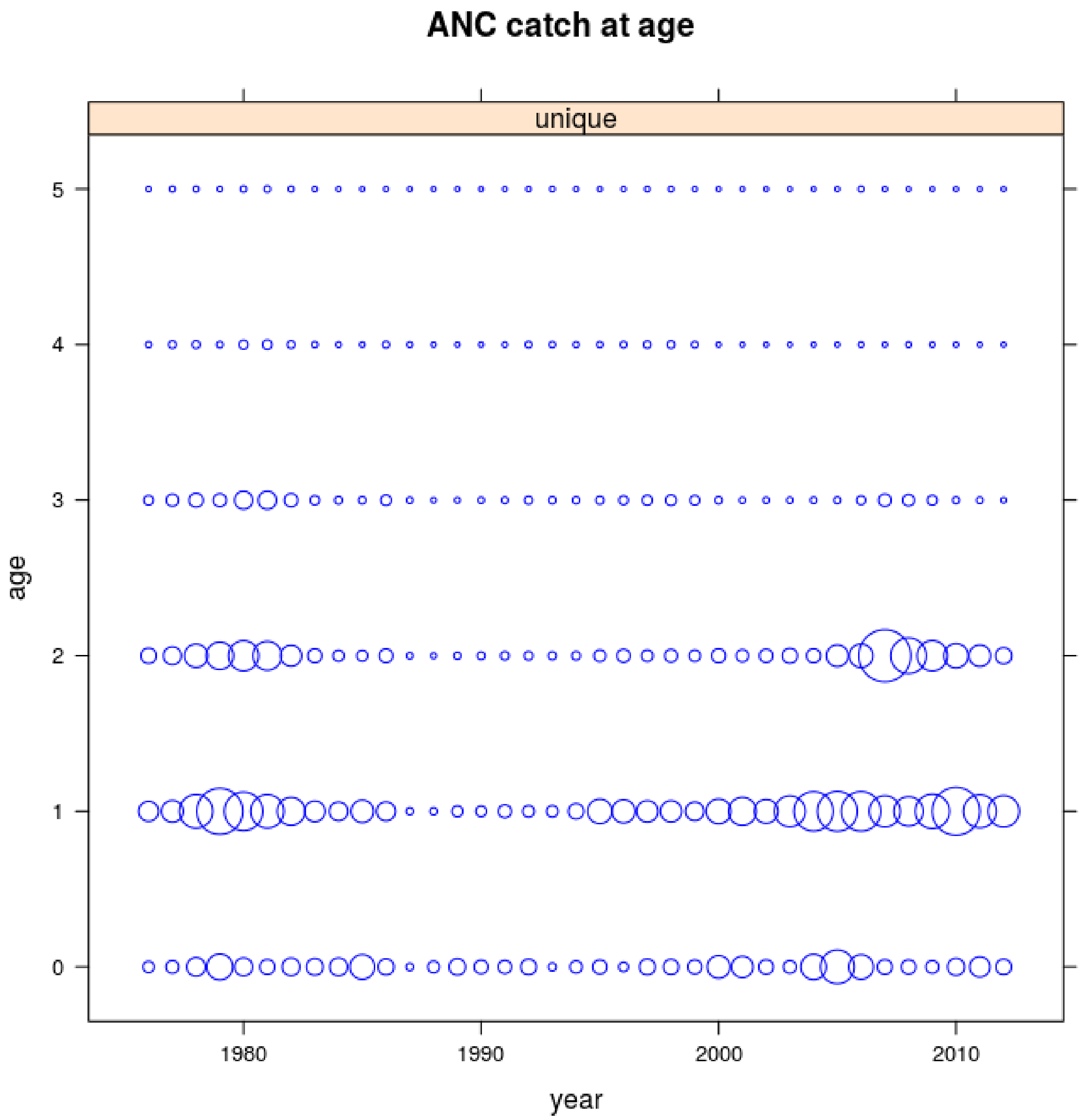


Figure 2.4: anchovy bubble plot catch numbers

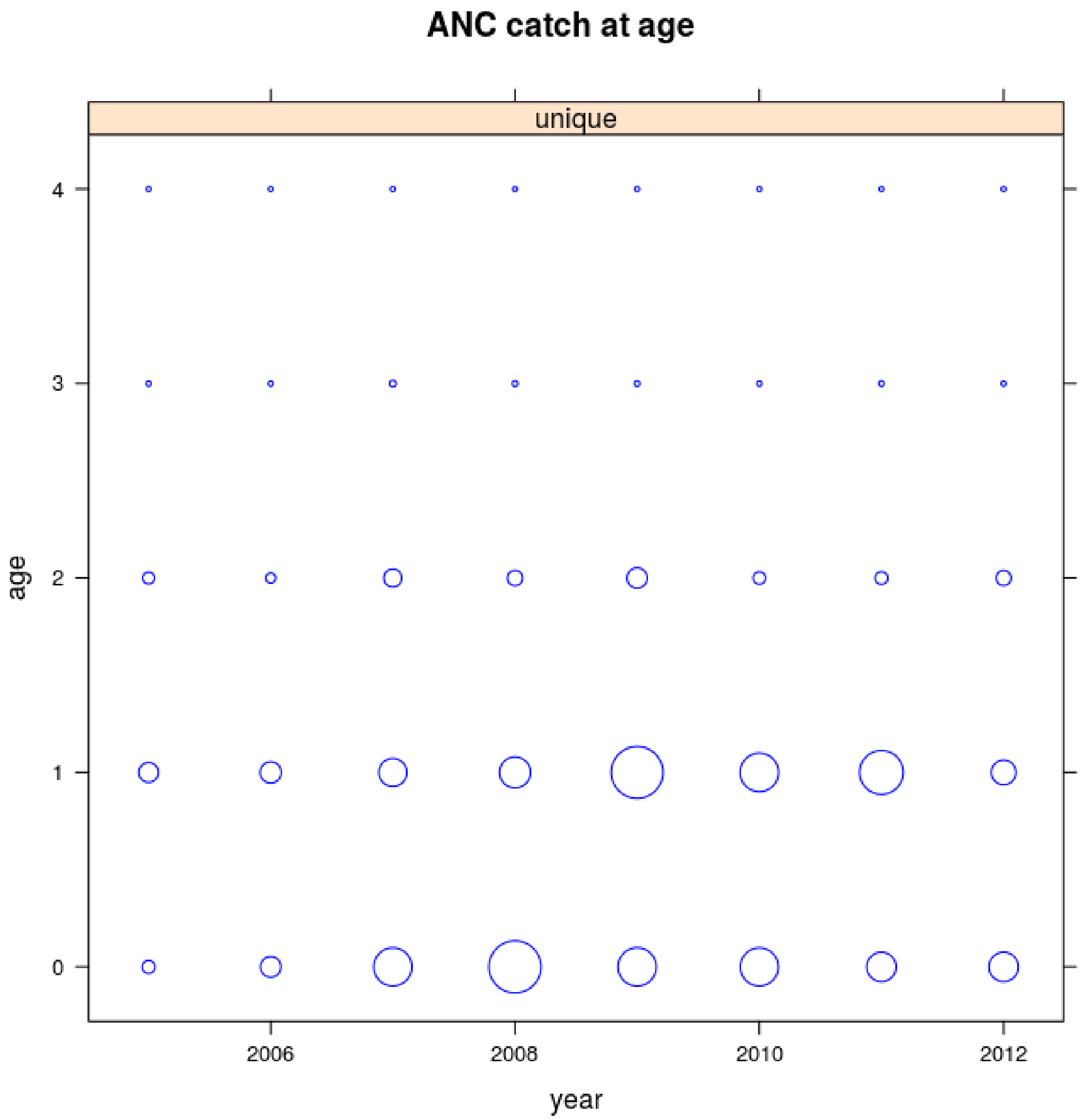


Figure 2.5: bubble plots of anchovy tuning index

2.2 Replicate SAM assessment

The latest accepted assessment performed in STECF working group was performed with FLSAM. Here the objective is to compare the SAM with a4a fit (e.g. achieve the same results) and see if the fit can be better. The assessment results from SAM model are in figure below for the Fishing mortality at age. It clearly appears that there are some very high Fs in the last ages and years. From the EWG 13-12 report it is clear that there were some problems with residual fits of the tuning index, for full details see the original report.

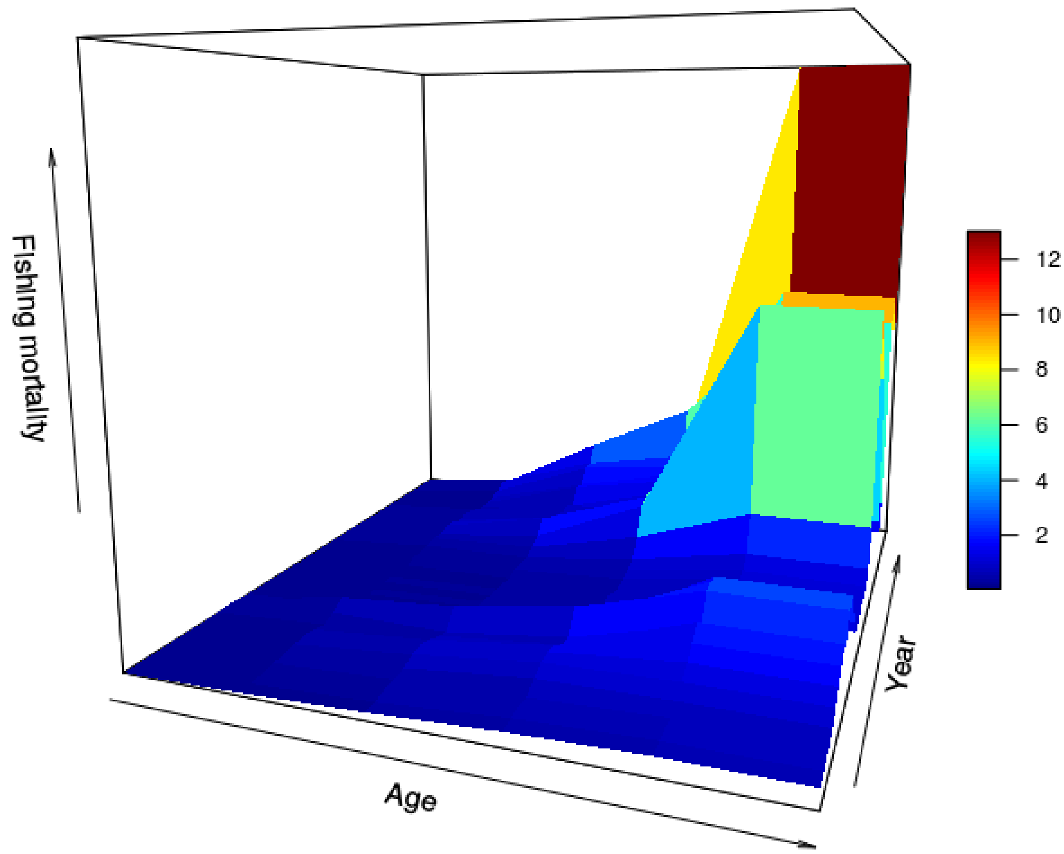


Figure 2.6: 3D Fishing mortality surface from the SAM assessment performed in STECF EWG 13-19

To rerun the assessment in a4a, once the data is loaded we reset the plus group at age class 4+ as in the SAM assessment.

```
ANC17 <- setPlusGroup(ANC17, 4)
```

Start simple and fit an a4a base model with the default settings for fmod, qmod and srmod.

```
fmod <- ~factor(year) + factor(age)
qmod <- list(~factor(age))
srmod <- ~factor(year)
fit0 <- a4aSCA(stock = ANC17, indices = ANC17.tun, fmodel = fmod,
    qmodel = qmod, fit = "MP")
```

The assessment with the default settings does not converge, and returns the following warning: " Hessian was not positive definite". To get results the fit must be done without computing the hessian, with the argument 'fit' set to 'MP'. The main diagnostics for the assessment fit are the residual patterns by age for survey and catches. The survey displays bad yearly trends while the catches present a level problem in Age 0.

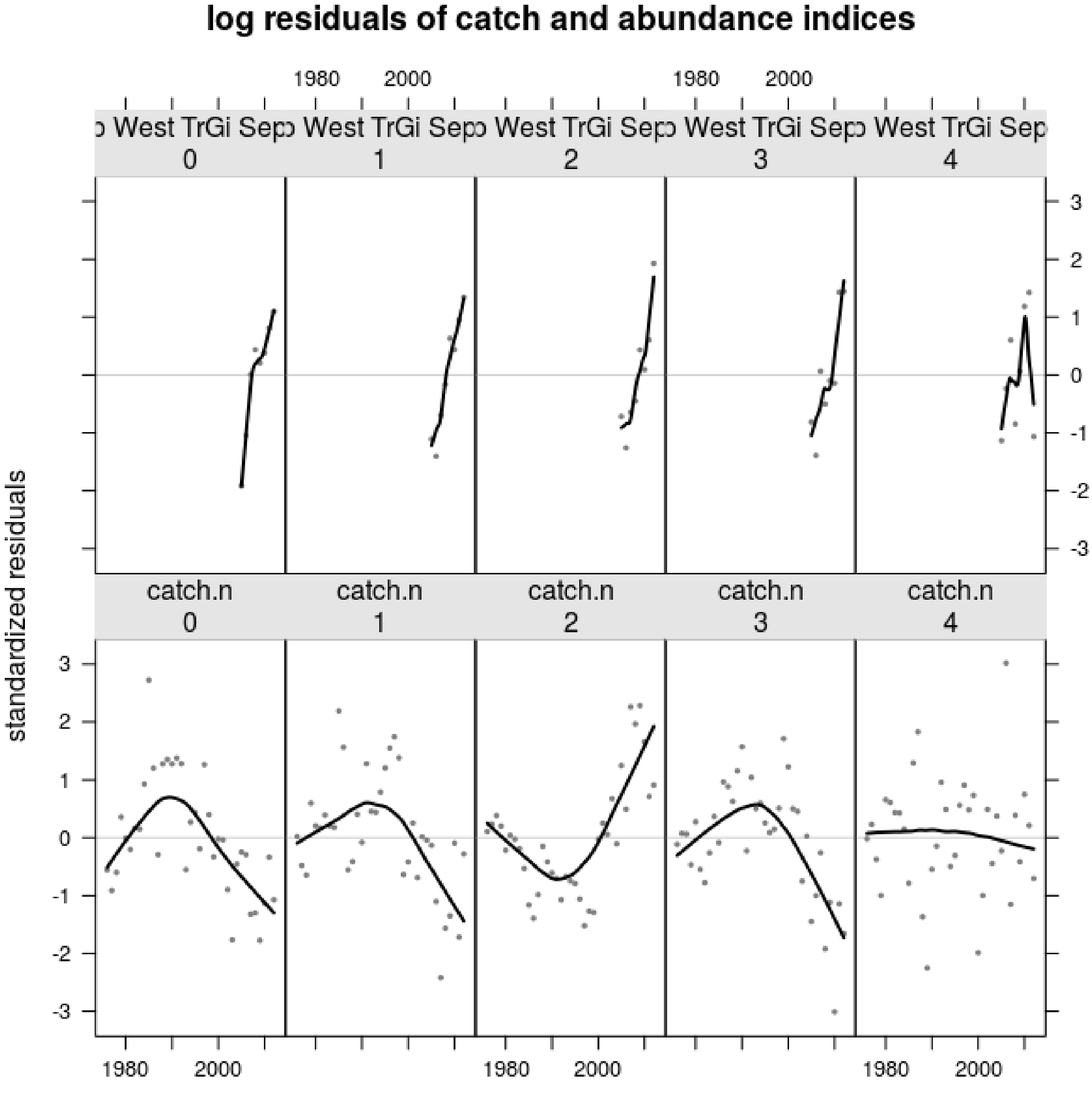


Figure 2.7: Residual patterns for tuning survey and year classes in catch numbers for fit0

We can plot fitted against observed numbers at age for survey, where model fitted numbers are lines in blue and observed is pink line. The prediction is quite far off the observed and will need more flexibility in either the fmod or qmod.

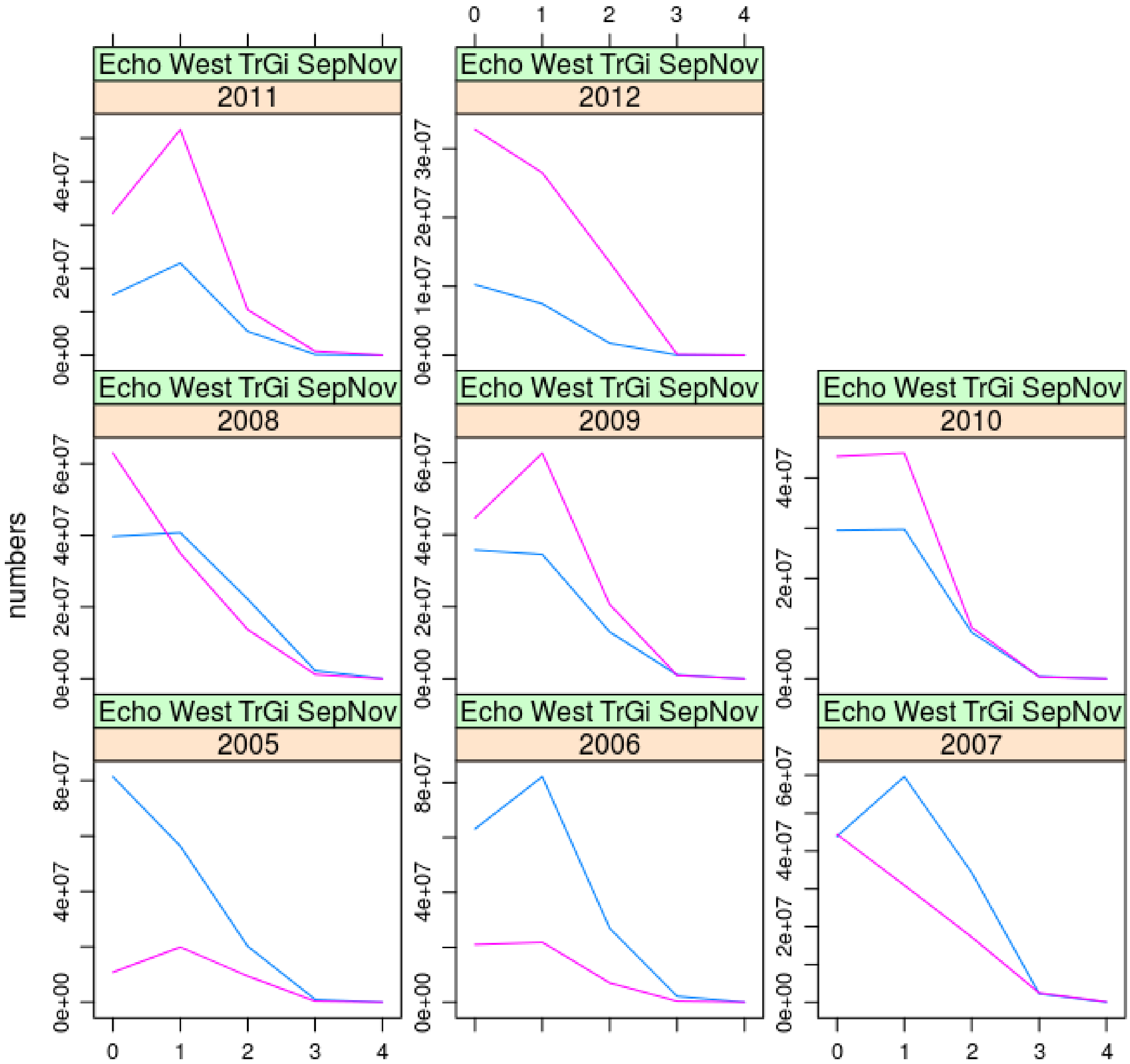


Figure 2.8: Predicted vs observed numbers at age in tuning index

Similarly we can plot fitted catch at age against observed where fitted is blue line, observed is pink line. In this case also there are large discrepancies in the early part of the series and in the 1990's.

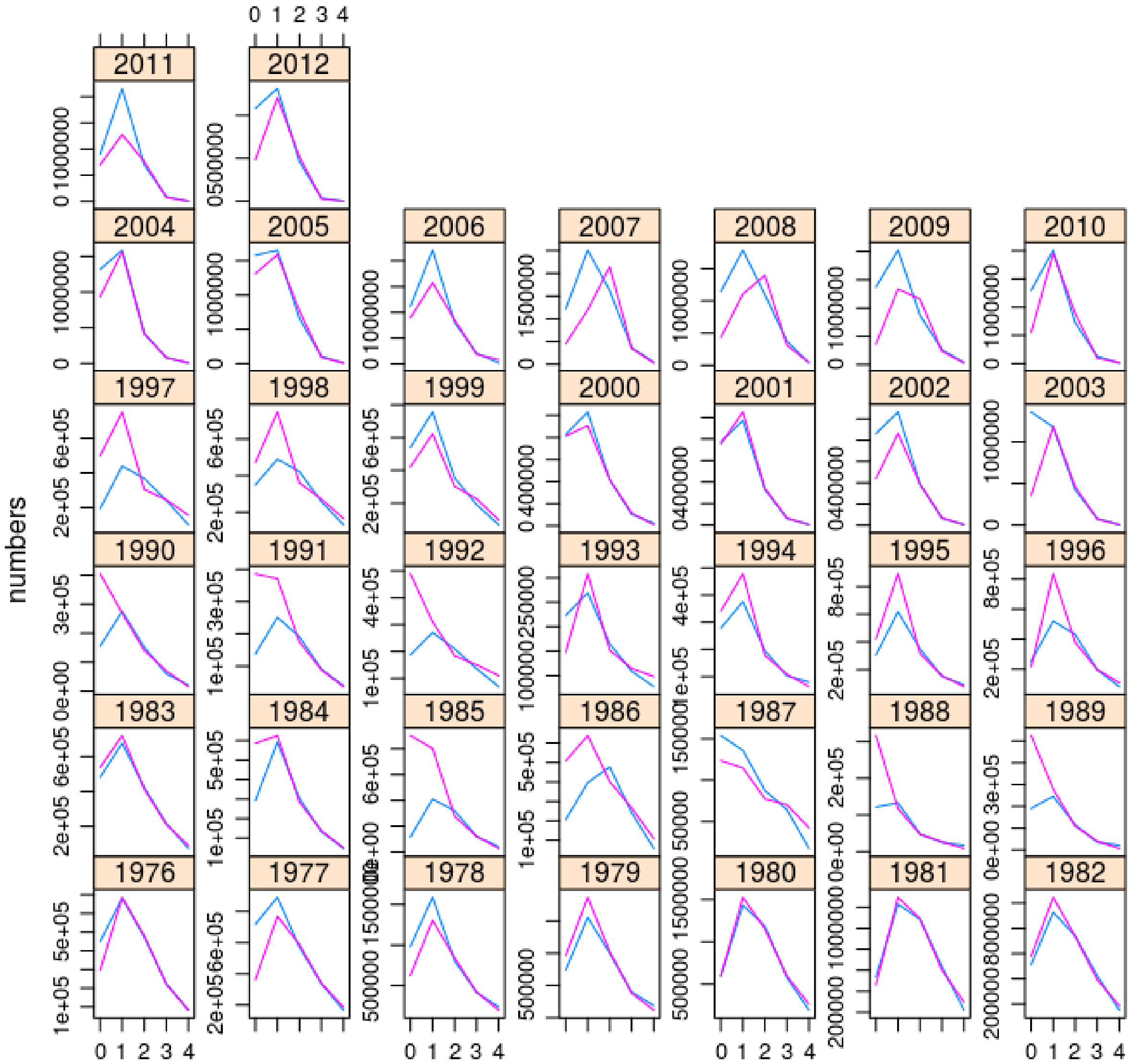


Figure 2.9: Predicted vs observed catch numbers at age for fit0

Try now to model with greater flexibility and add a smoother in the fmod on age and year with K set differently to account for the available numbers of ages and years.

```
fmod <- ~s(age, k = 4) + s(year, k = 20)
fit1 <- a4aSCA(stock = ANC17, indices = ANC17.tun, fmodel = fmod,
    qmodel = qmod)
```

By running the same set of diagnostic plots, the fit of age 0 in the catches improves but there are still strong residual year trends in the survey for all ages.

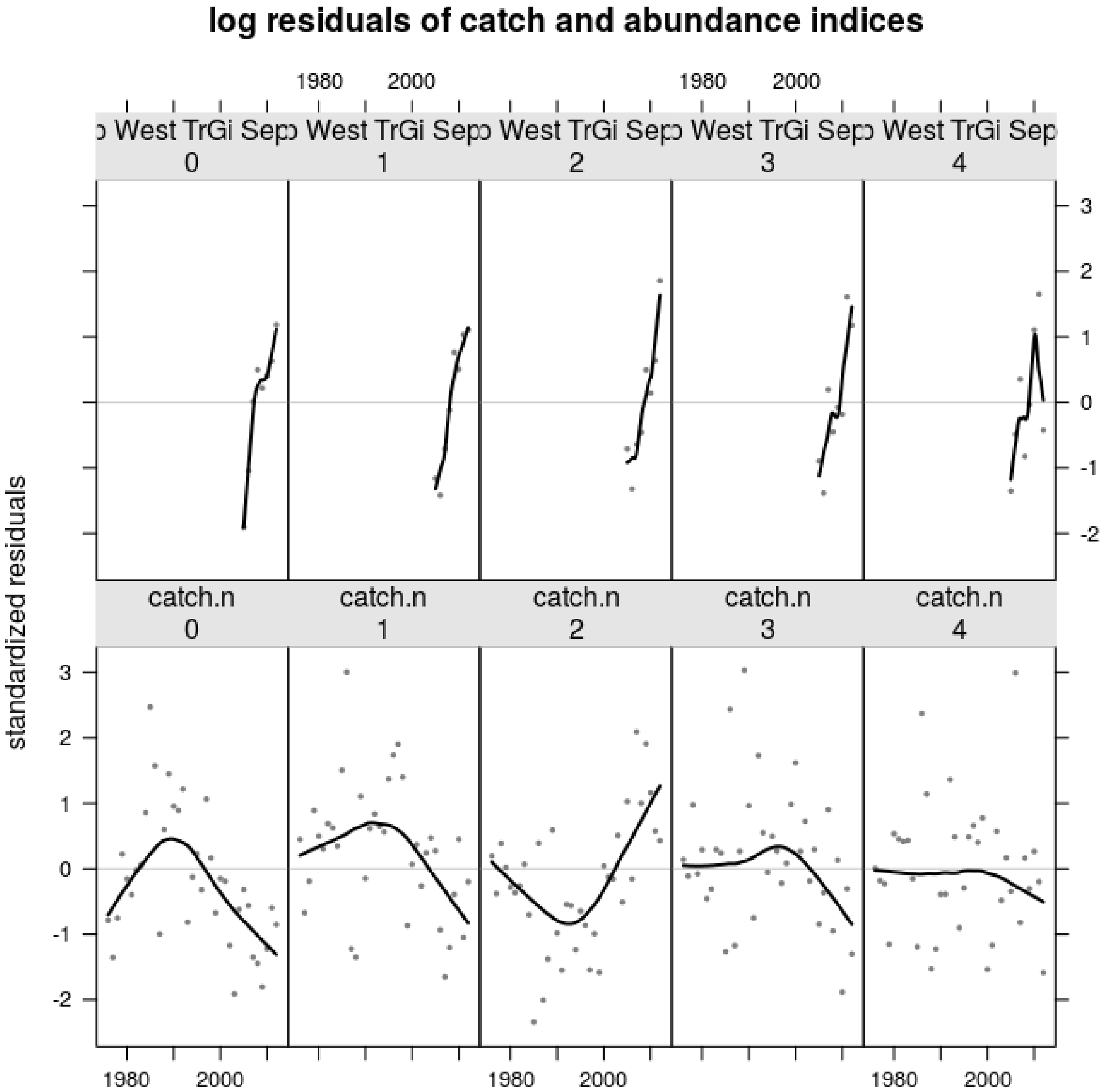


Figure 2.10: Diagnostics for fit1

Survey diagnostics are again not very good, while the predicted vs observed catch numbers are better than in fit1.

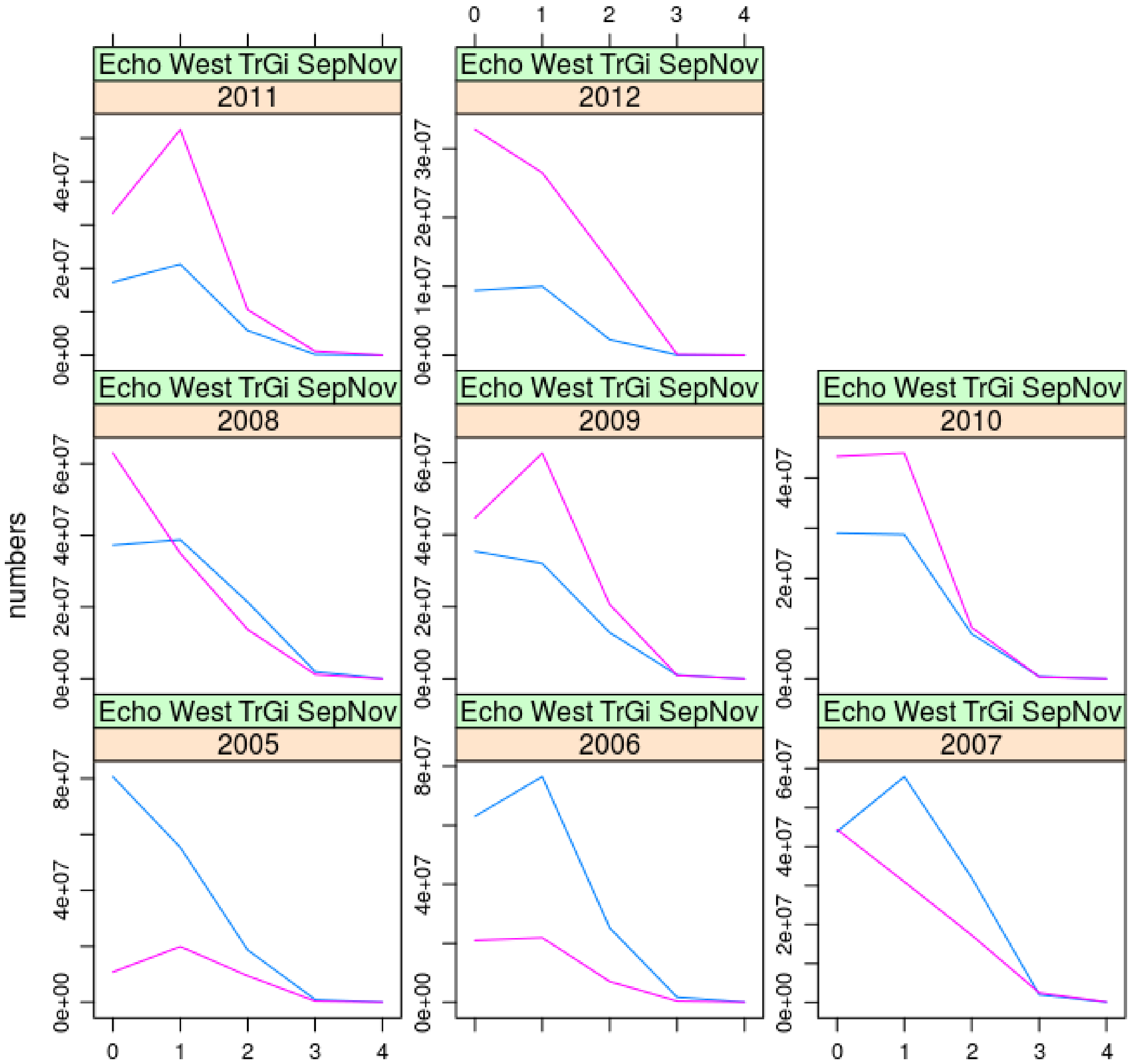


Figure 2.11: Diagnostics for fit1

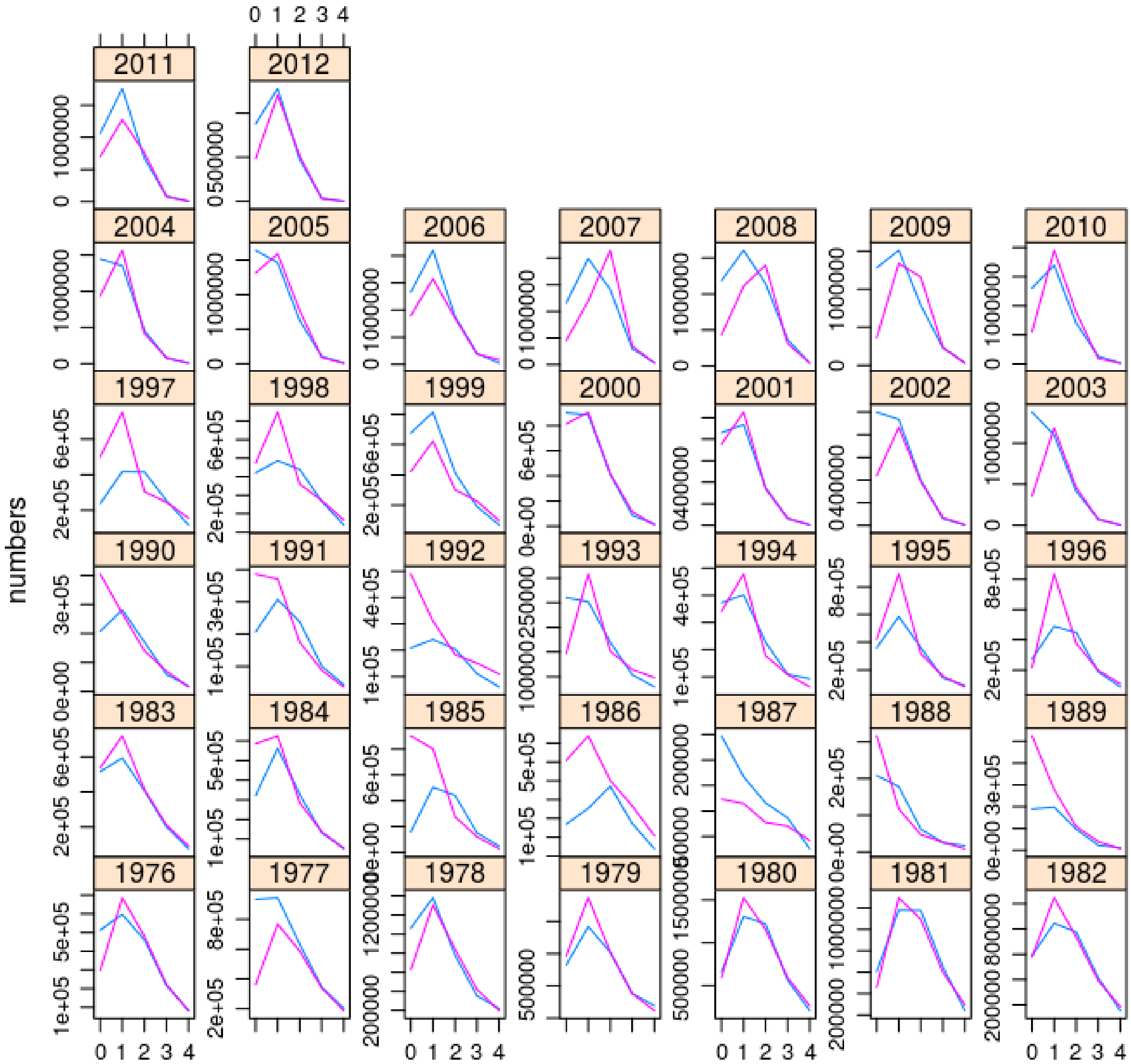


Figure 2.12: Diagnostics for fit1

By keeping the same F model, we model catchability (qmod) as a smooth of age plus year.

```
fmod <- ~s(age, k = 4) + s(year, k = 20)
qmod <- list(~s(age, k = 4) + year)

fit2 <- a4aSCA(stock = ANC17, indices = ANC17.tun, fmodel = fmod,
    qmodel = qmod)
```

By rerunning diagnostics plots on fit2 there is an improvement in the survey yearly trends in residuals. Also the predicted vs observed in survey numbers improves. The same is true in the early part of the catch numbers.

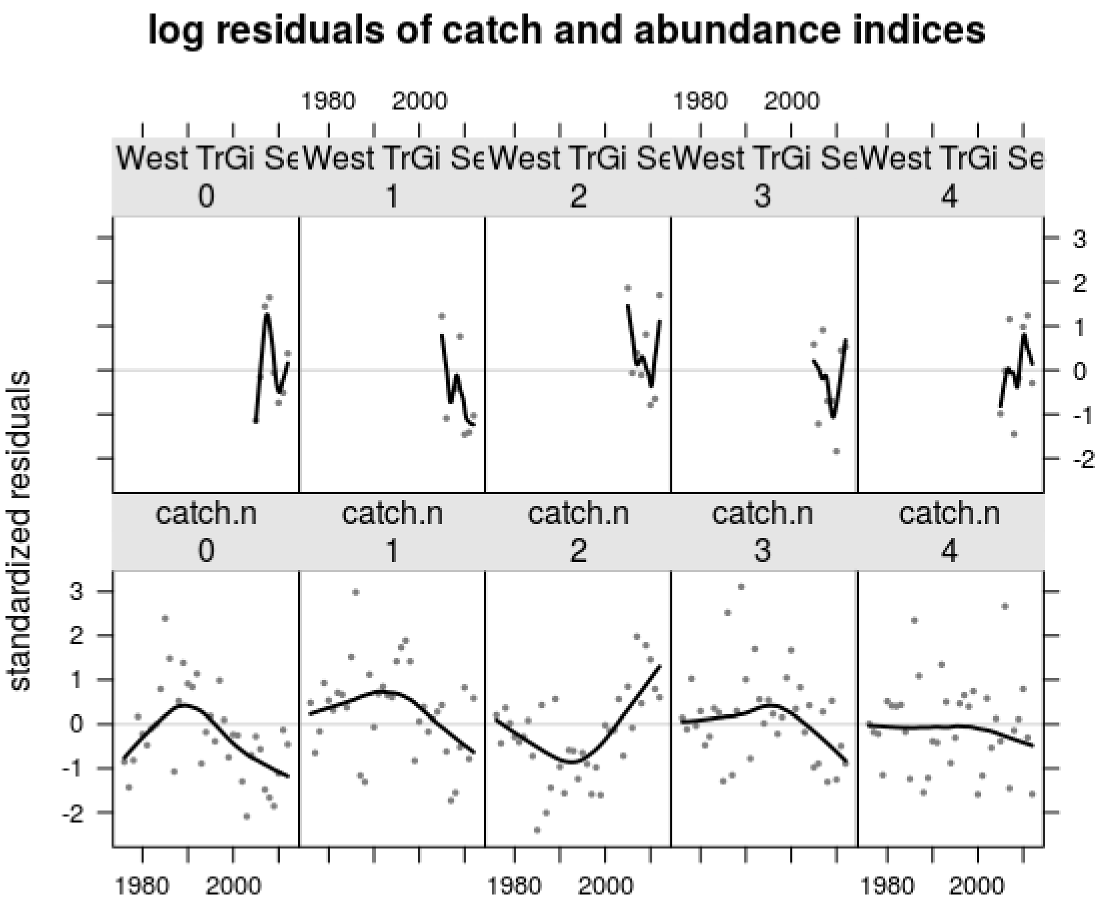


Figure 2.13: Diagnostics for fit2

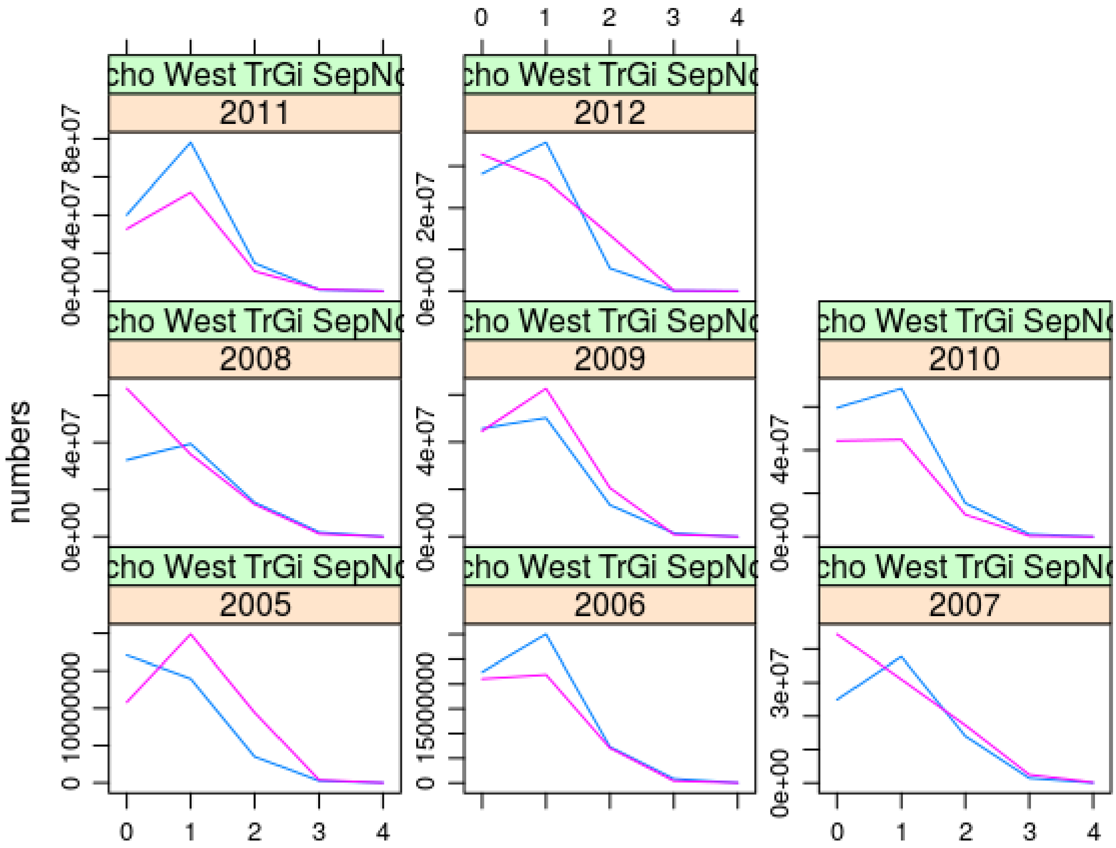


Figure 2.14: Diagnostics for fit2

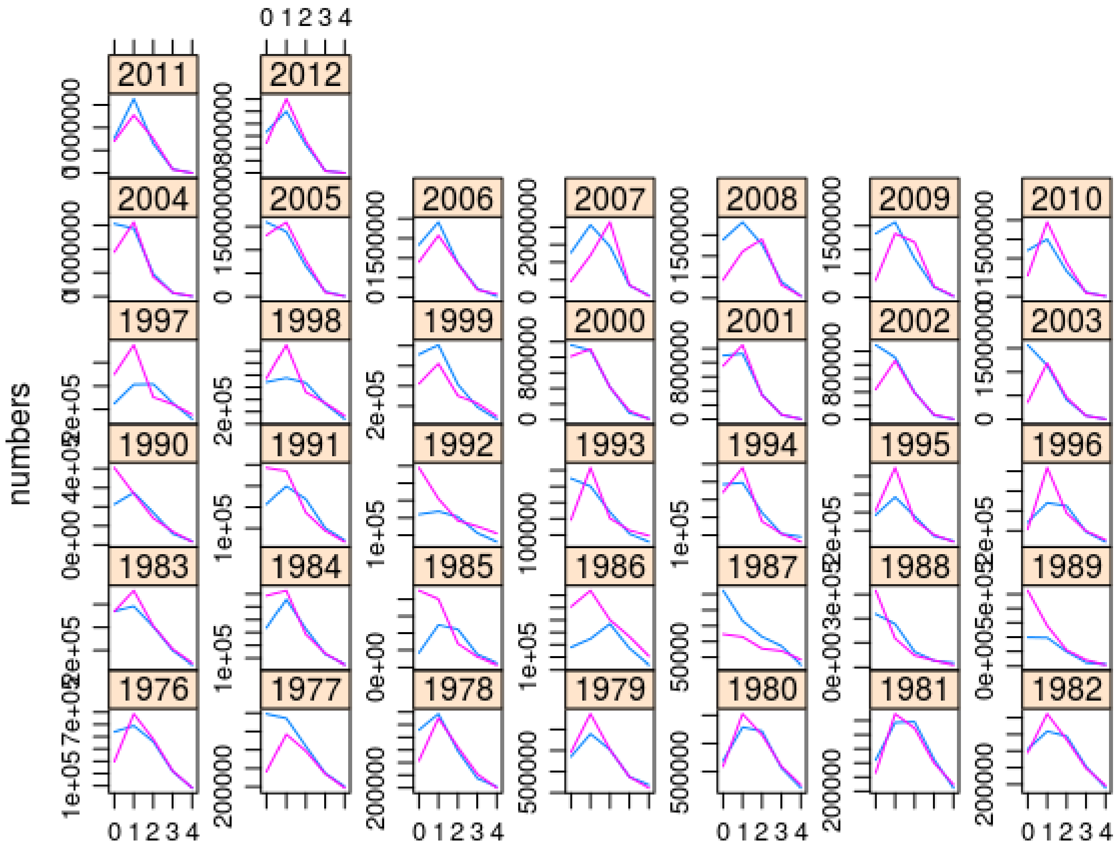


Figure 2.15: Diagnostics for fit2

Looks like there are problems with the survey catchability, while the catch residuals are more or less ok. The 3D F surface shows very high mortality in the terminal year and age and this is closest result to the SAM fit.

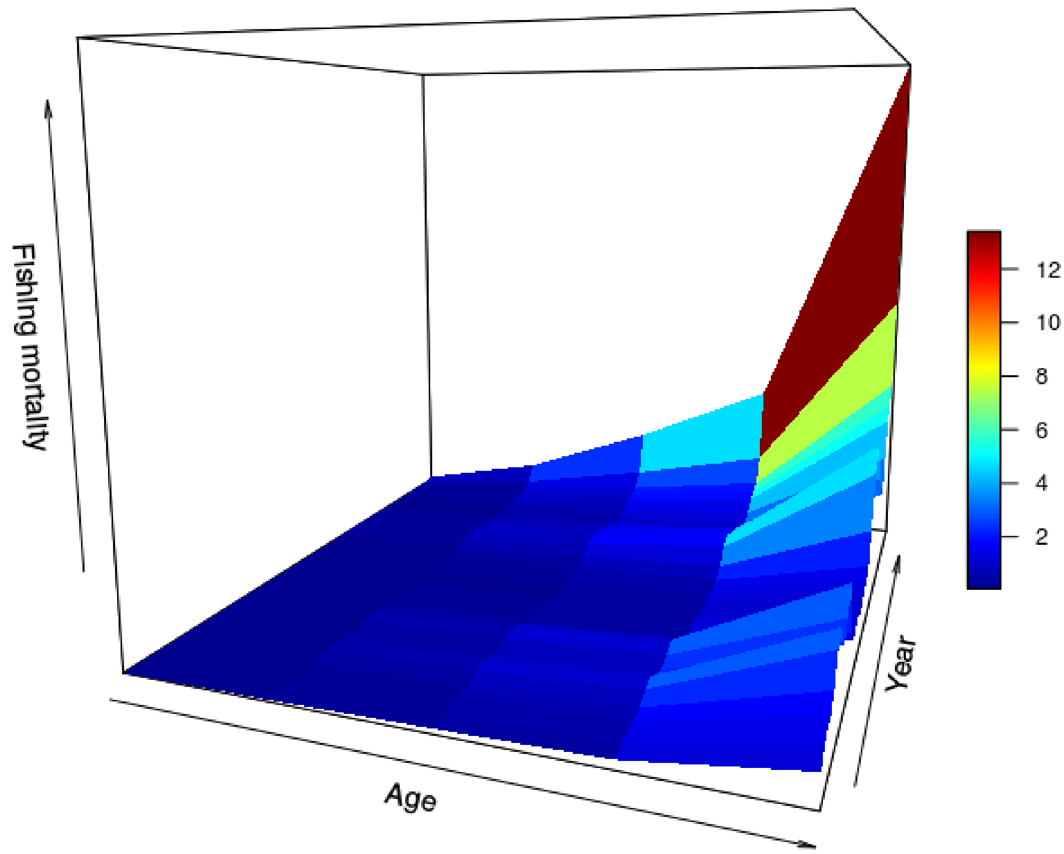


Figure 2.16: 3D surface of F at age for fit2

2.3 Improving the stock assessment fit

To try to improve the residual patterns, the overall model fit and get more realistic F's at age in terminal year we can work on survey catchability. We thus try to model catchability with a smoother on age and introduce breakpoints in 2006 and 2011. The choice of these years aims at allowing more flexibility in these years to account for changes in catchability.

```
fmod <- ~s(age, k = 4) + s(year, k = 20)
qmod <- list(~s(age, k = 4, by = breakpts(year, c(2006, 2011))))
fit3 <- sca(stock = ANC17, indices = ANC17.tun, fmodel = fmod,
    qmodel = qmod, srmodel = srmod, fit = "assessment")
```

Upon inspection of residual patterns, survey residuals still present some trend. Survey predicted vs observed numbers at age are better but with some discrepancies in 200-2006 and 2009-2011.

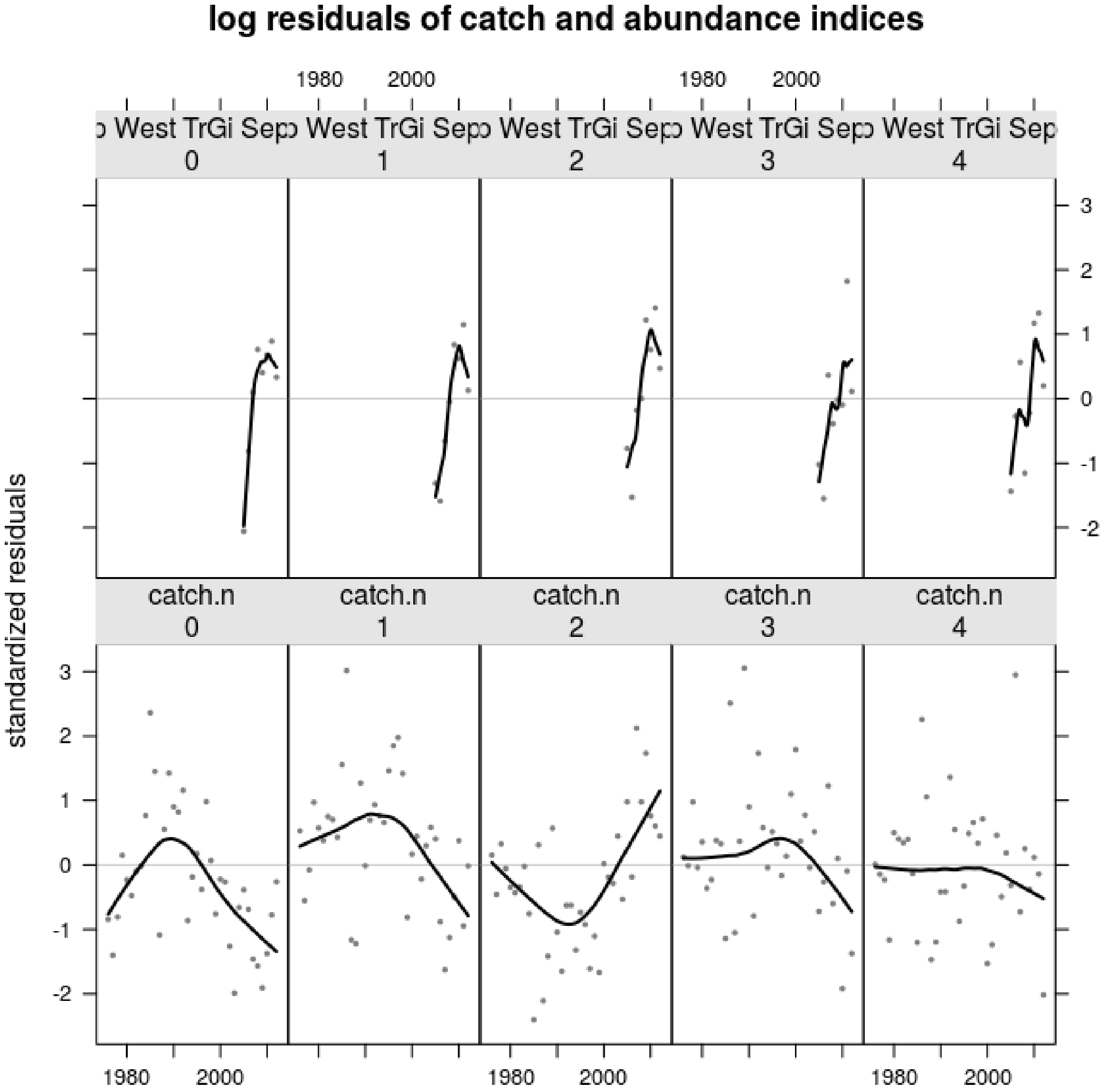


Figure 2.17: Diagnostics for fit3

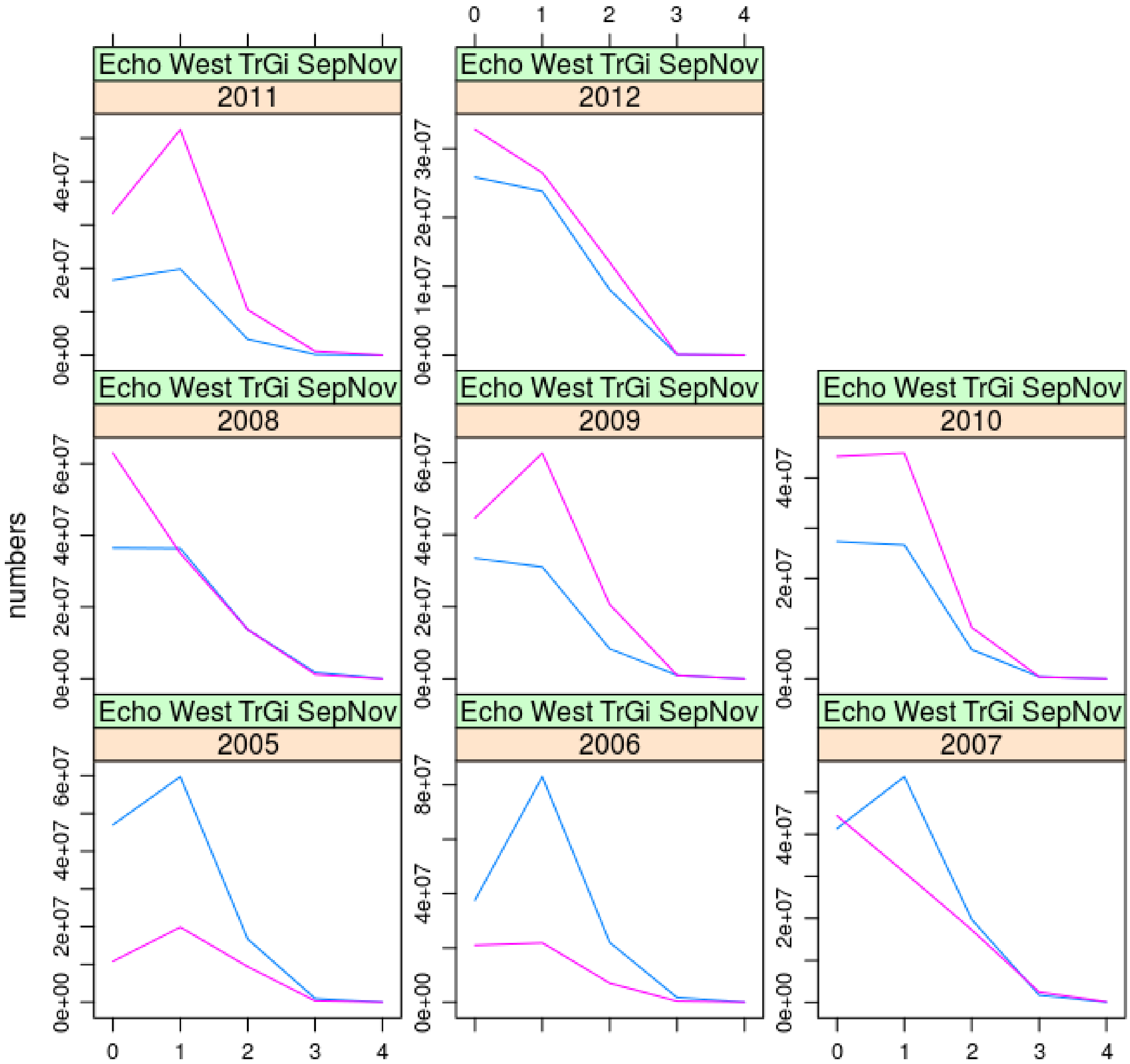


Figure 2.18: Diagnostics fit3

However in the predicted vs observed numbers in the catch there is a large discrepancy starting from 1985 to 1987, then 1990-1992, 1995-1998

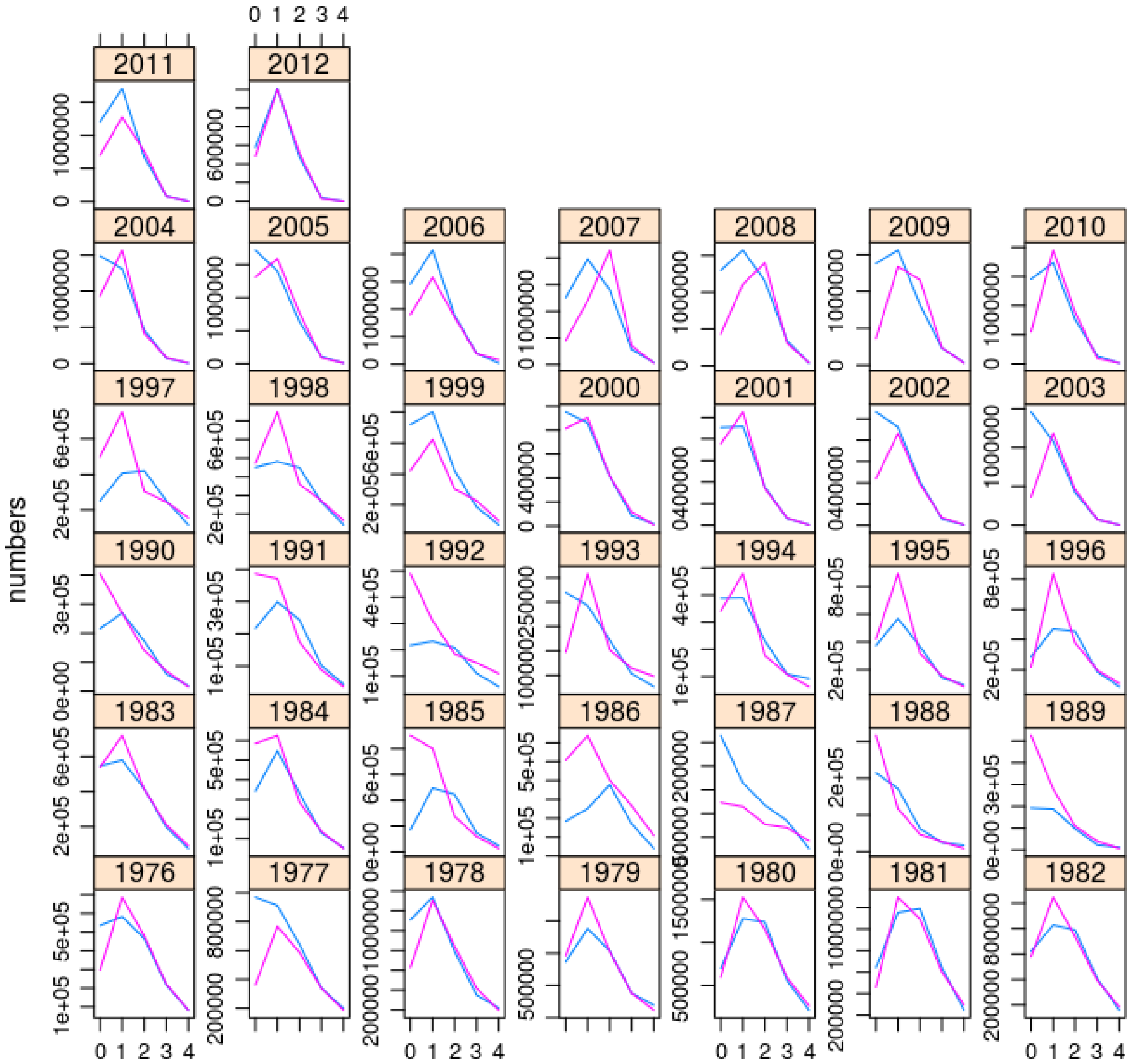


Figure 2.19: Diagnostics for fit3

The next attempt to improve the model fits is to model catchability as changing after 2006 and to insert breakpoints at ages in the fmod.

```
fmod <- ~s(age, k = 4) + s(year, k = 20, by = breakpts(age, c(1.5,
    2.5)))
qmod <- list(~s(age, k = 4, by = breakpts(year, 2006)))
# fmod <- ~ s(age, k=4) + s(year, k = 20)
fit4 <- a4aSCA(stock = ANC17, indices = ANC17.tun, fmodel = fmod,
    qmodel = qmod, srmodel = srmod, fit = "assessment")
```

The diagnostics for fit4 are better than fit3 for catch residuals but not for age 2 of the survey.

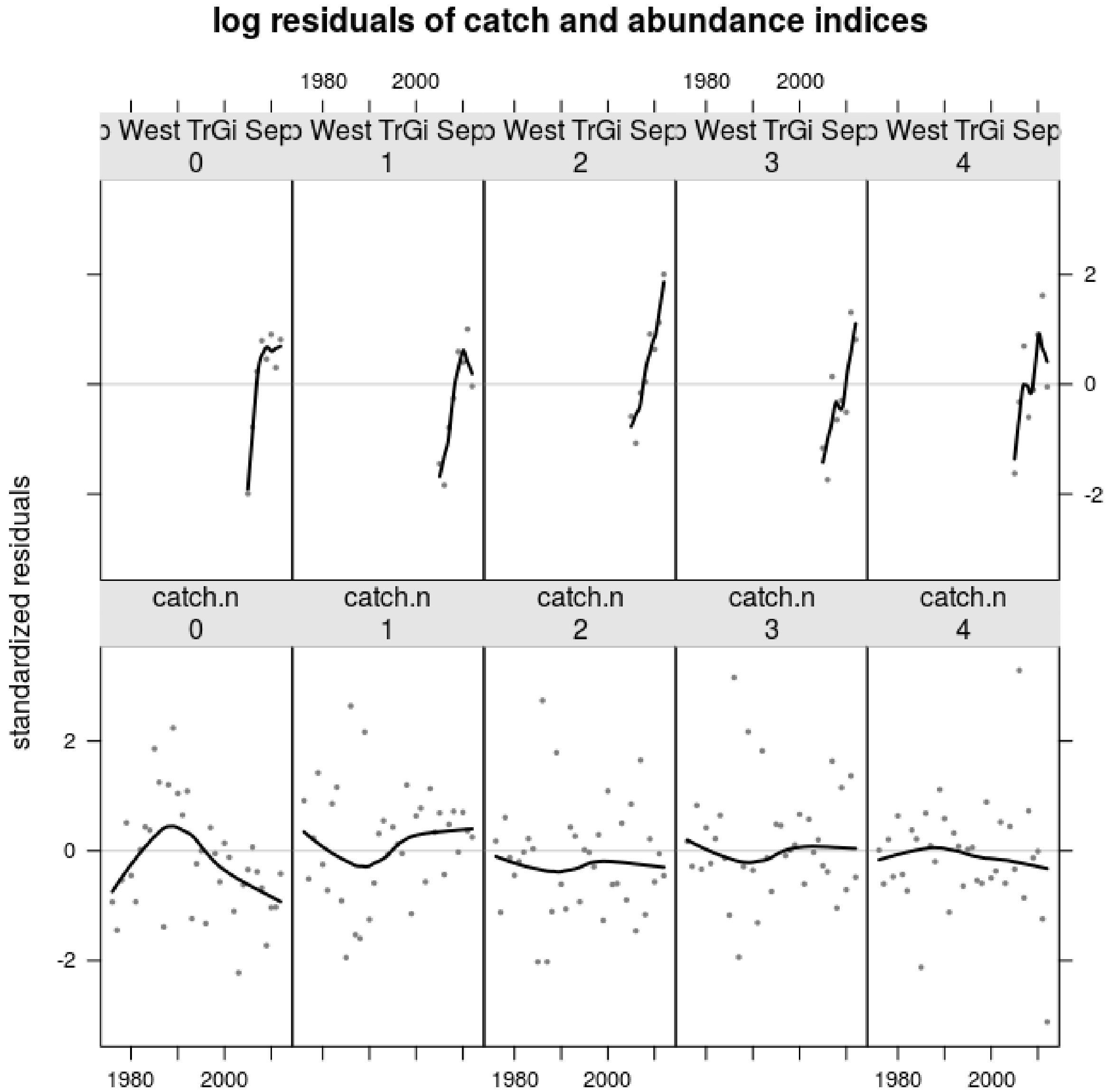


Figure 2.20: Anchovy diagnostics for fit4

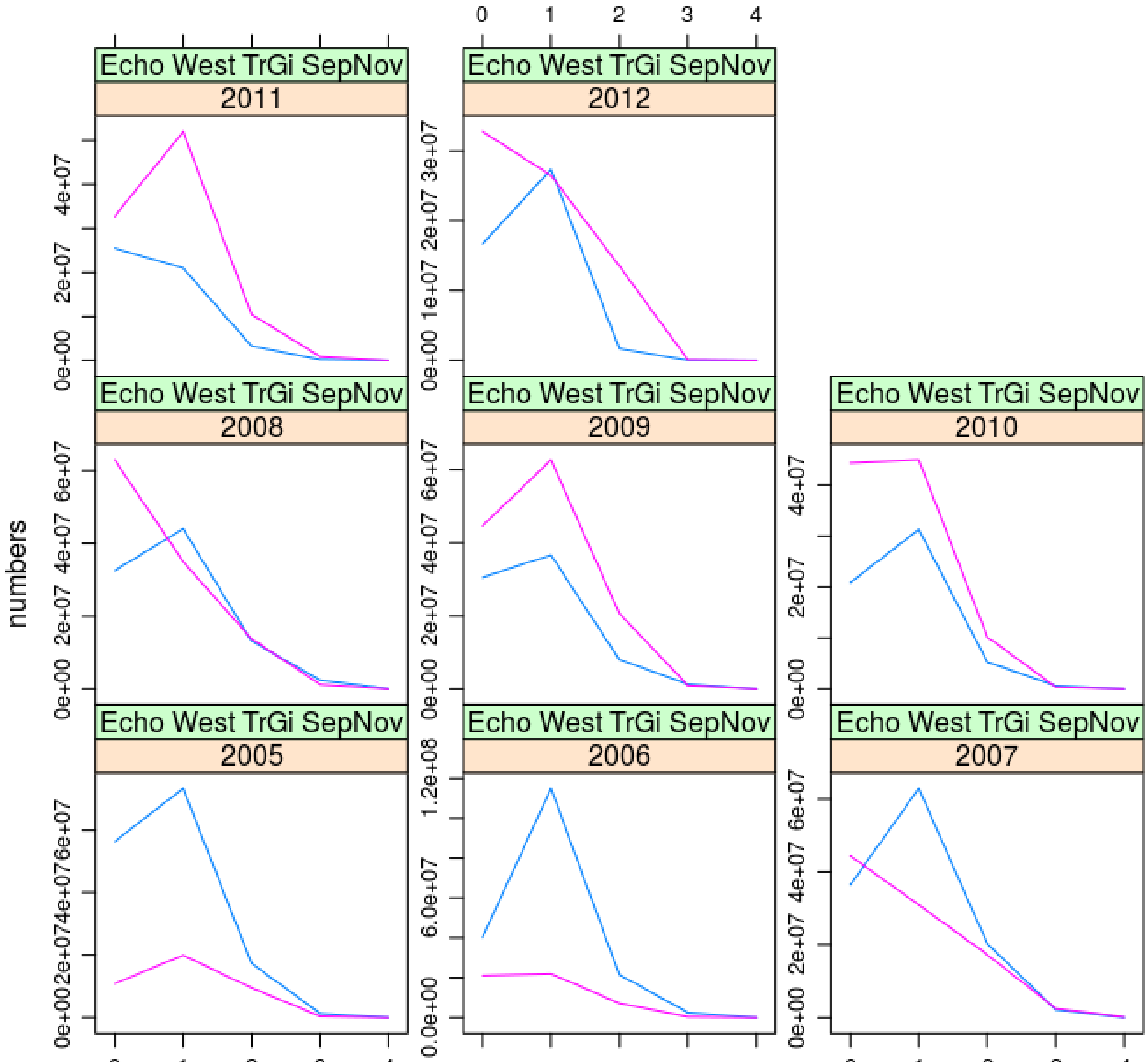


Figure 2.21: Diagnostics for fit4

This model with the exception of year 1986-1988 does a good job at predicting catch numbers.

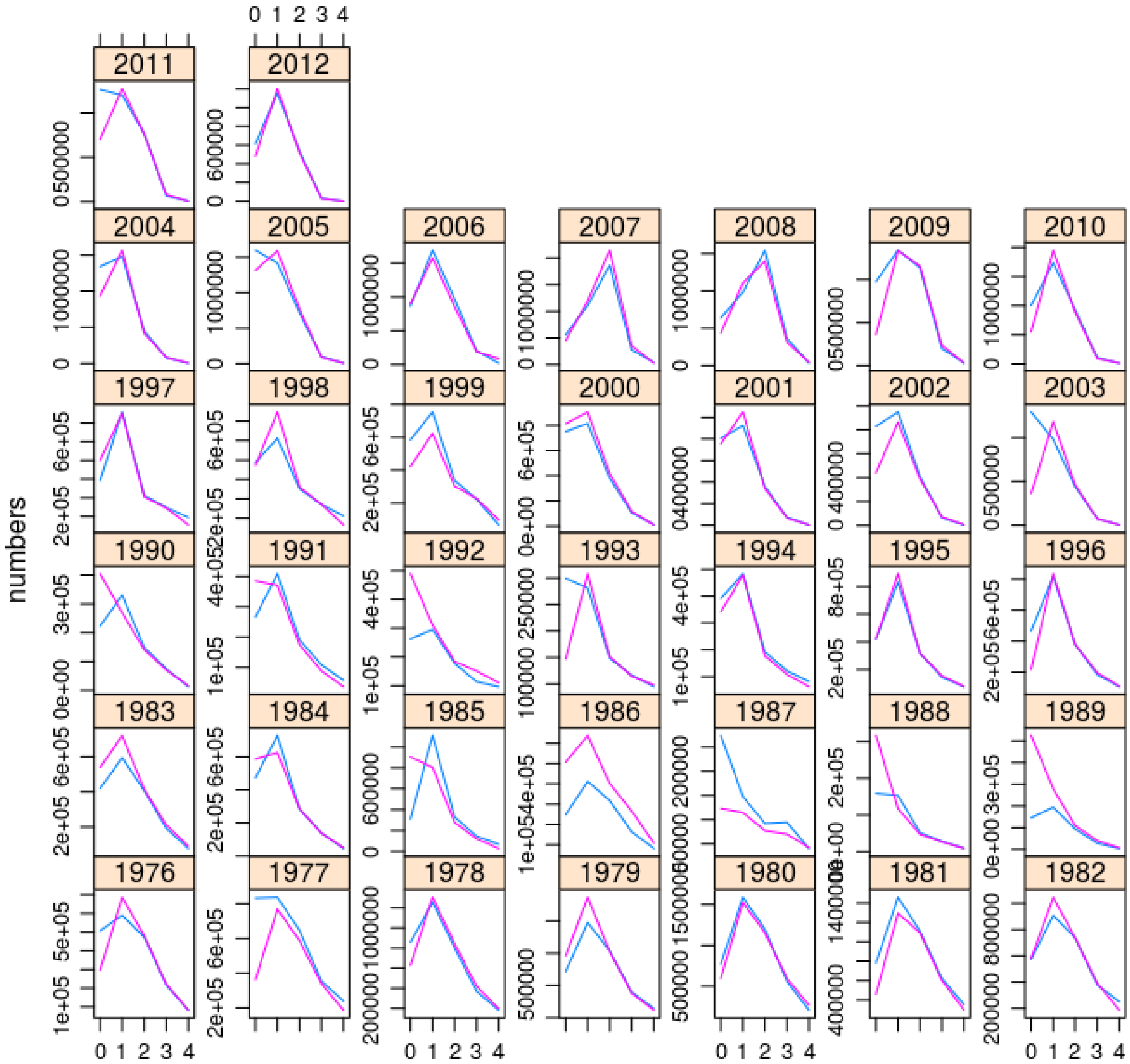


Figure 2.22: Anchovy diagnostics for fit4

The F surface remains however particularly high in the last ages and years.

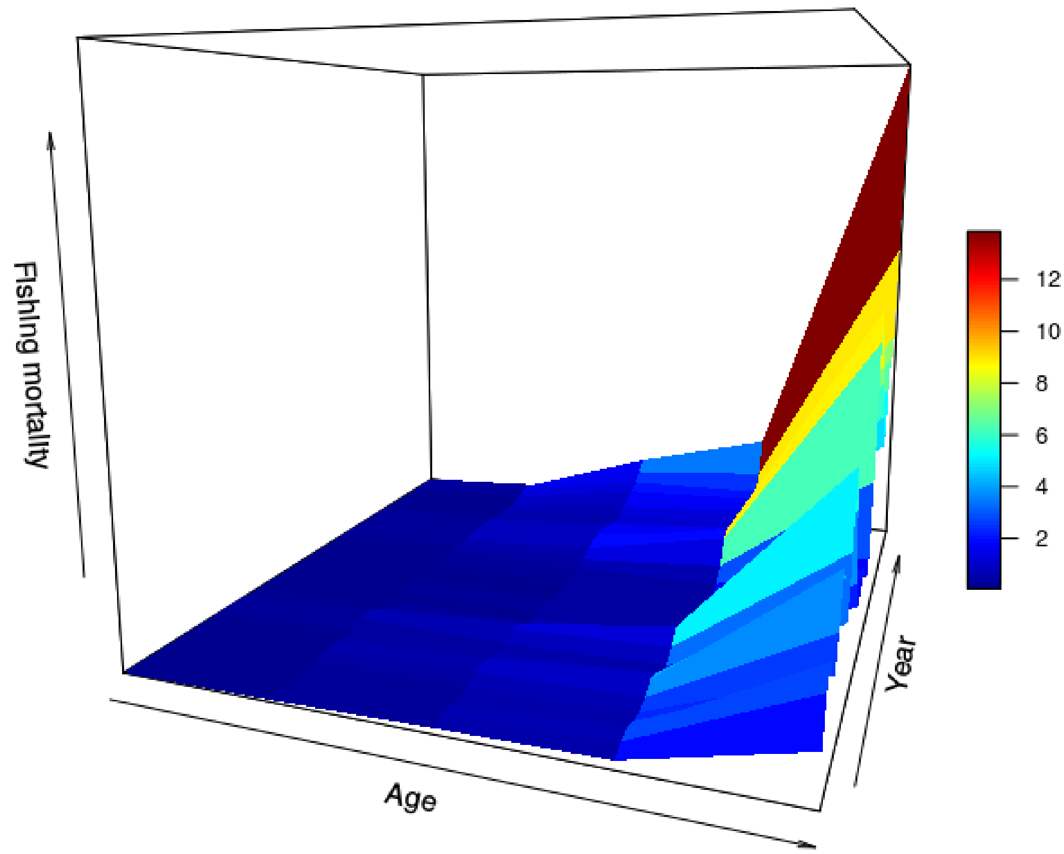


Figure 2.23: 3D surface of F at age for model fit4

Another possibility to improve the model fit is have a fmodel where F is a function of a smoother of year with breakpoints on year 1985 and 1998. The qmodel is kept as in fit4.

```
fmod <- ~s(age, k = 5, by = breakpts(year, c(1985, 1998)))
qmod <- list(~s(age, k = 4, by = breakpts(year, 2006)))

fit5 <- a4aSCA(stock = ANC17, indices = ANC17.tun, fmodel = fmod,
    qmodel = qmod, fit = "assessment")
```

Overall fit5 has acceptable standardized residuals for both survey and catches.

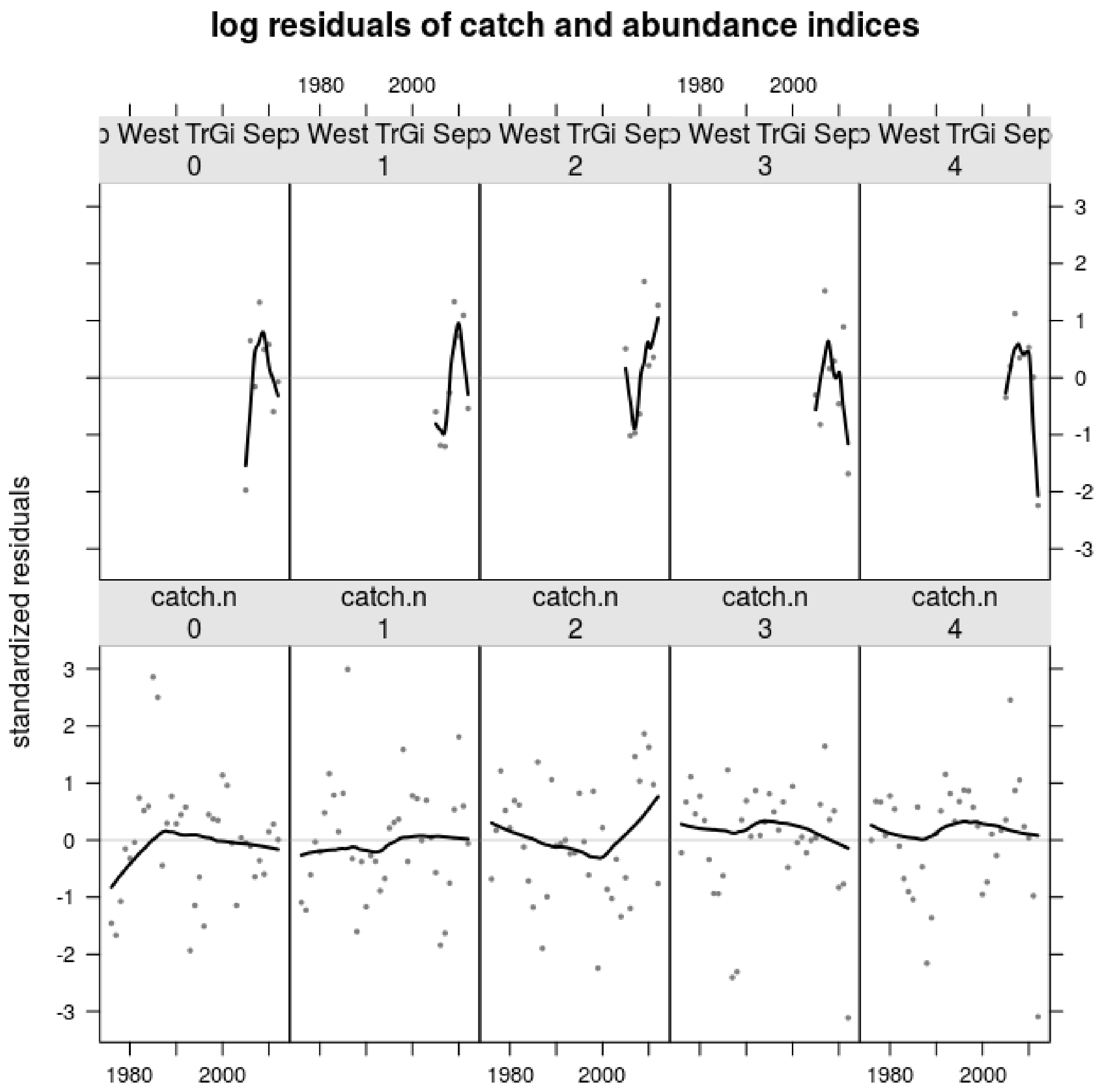


Figure 2.24: Anchovy diagnostics for fit5

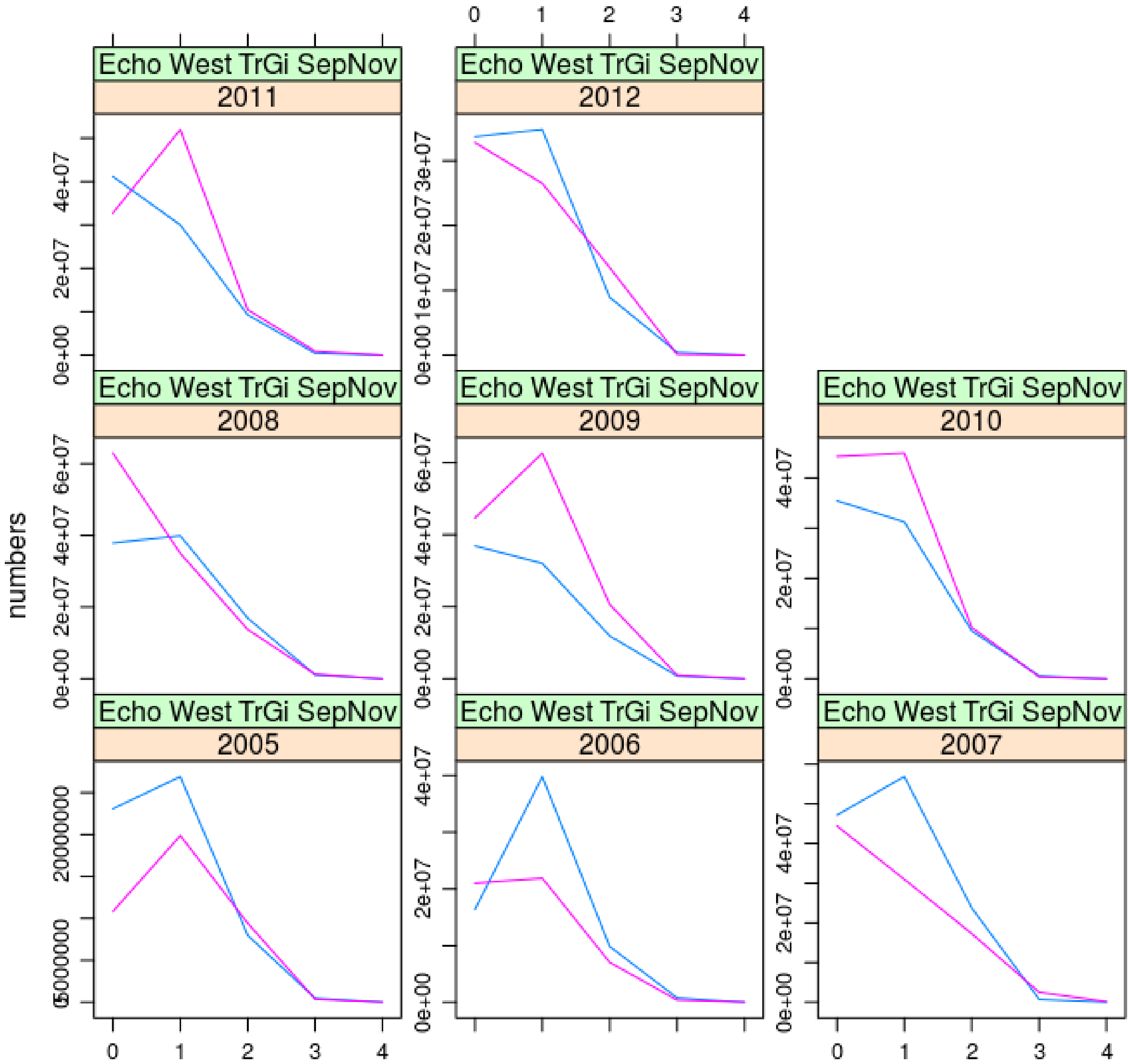


Figure 2.25: Anchovy diagnostics for fit5

Predicted vs observed catches still present problems in 1986:1988.

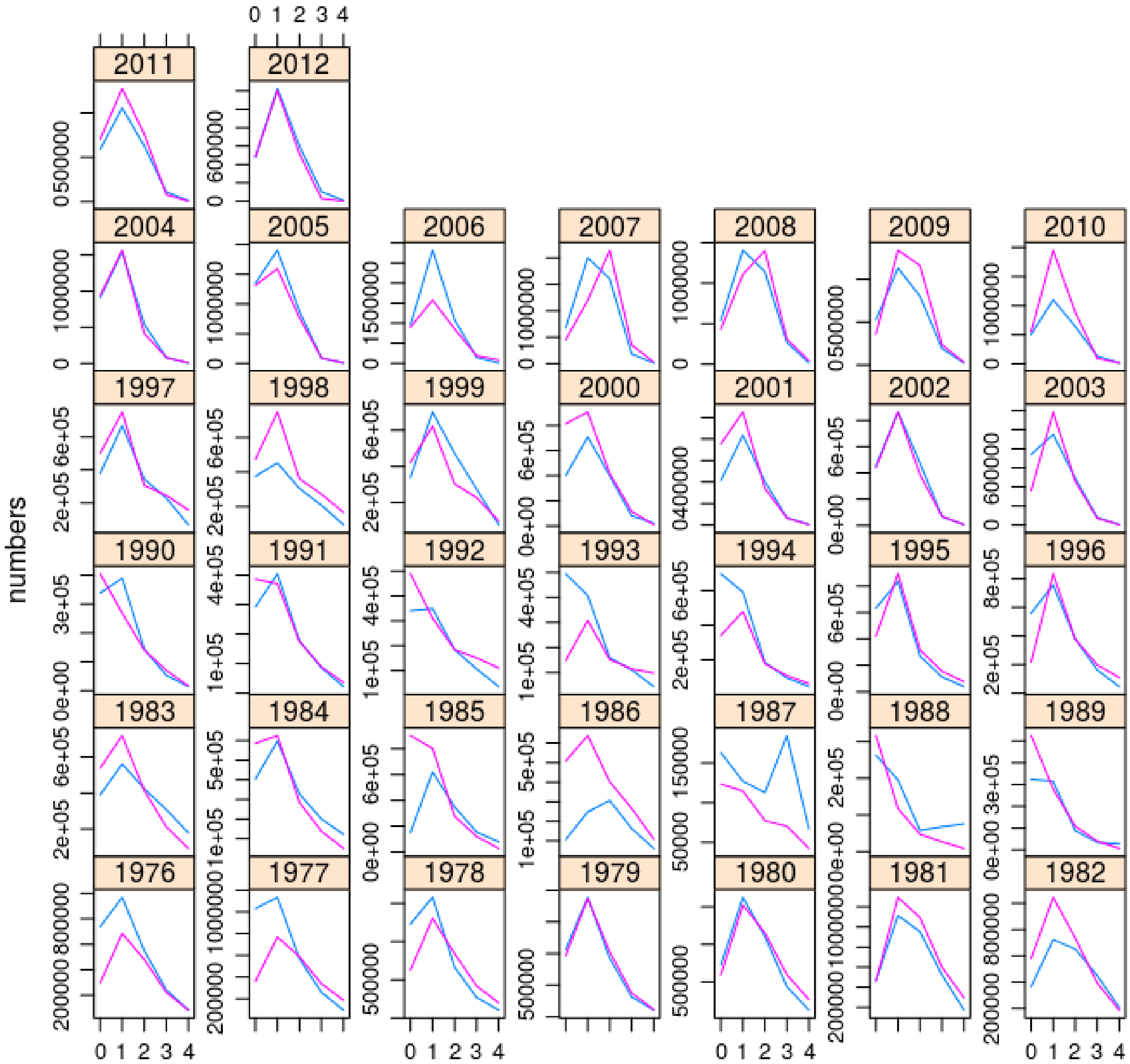


Figure 2.26: Anchovy diagnostics for fit5

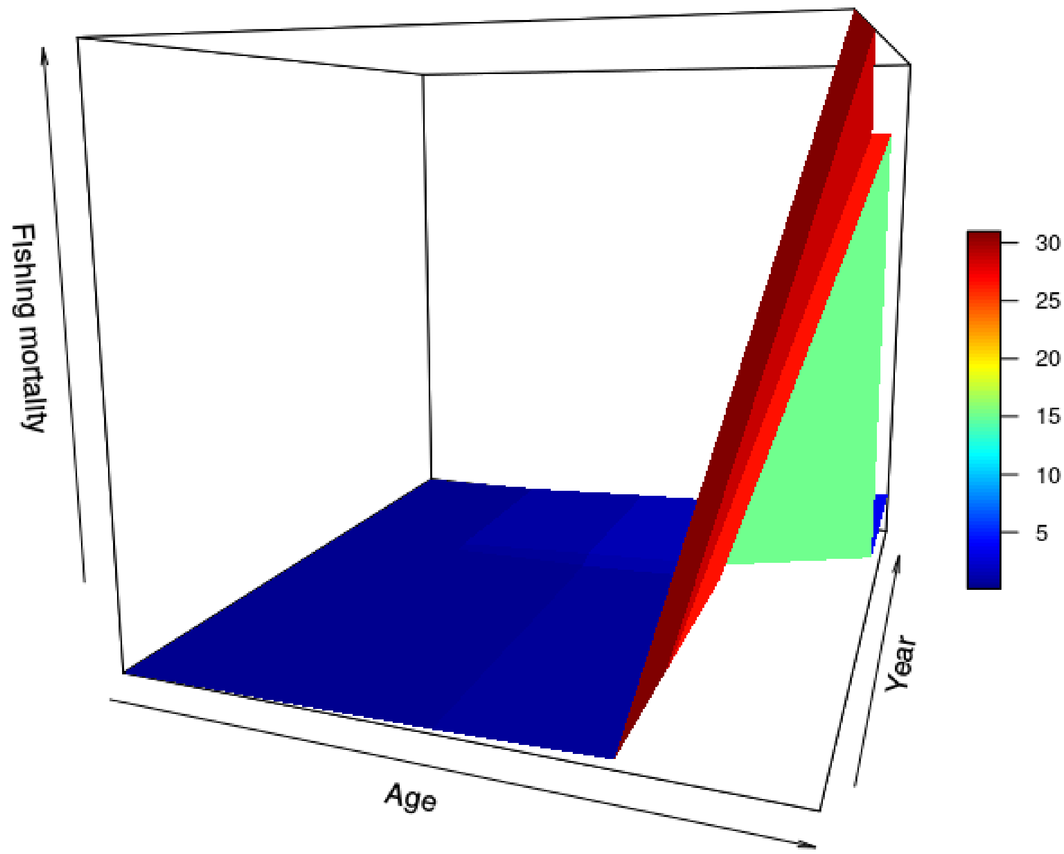


Figure 2.27: 3D surface of F at age for model fit5

Try to change default of the variance model on the survey since there have been a number of assumptions when combining the east and west surveys. Specify that variance can change in both survey and catch by a smooth of age with k=3

```
fmod <- ~s(age, k = 5, by = breakpts(year, c(1987, 1995))) +
    s(year, k = 20, by = breakpts(age, 4))
qmod <- list(~s(age, k = 4, by = breakpts(year, 2006)))
vmod <- list(~s(age, k = 3), ~s(age, k = 3))
fit6 <- a4aSCA(stock = ANC17, indices = ANC17.tun, fmodel = fmod,
    qmodel = qmod, vmod = vmod, fit = "assessment")
```

Fit6 is the best model obtained, the residuals, predicted vs observed in catch and survey are quite acceptable.

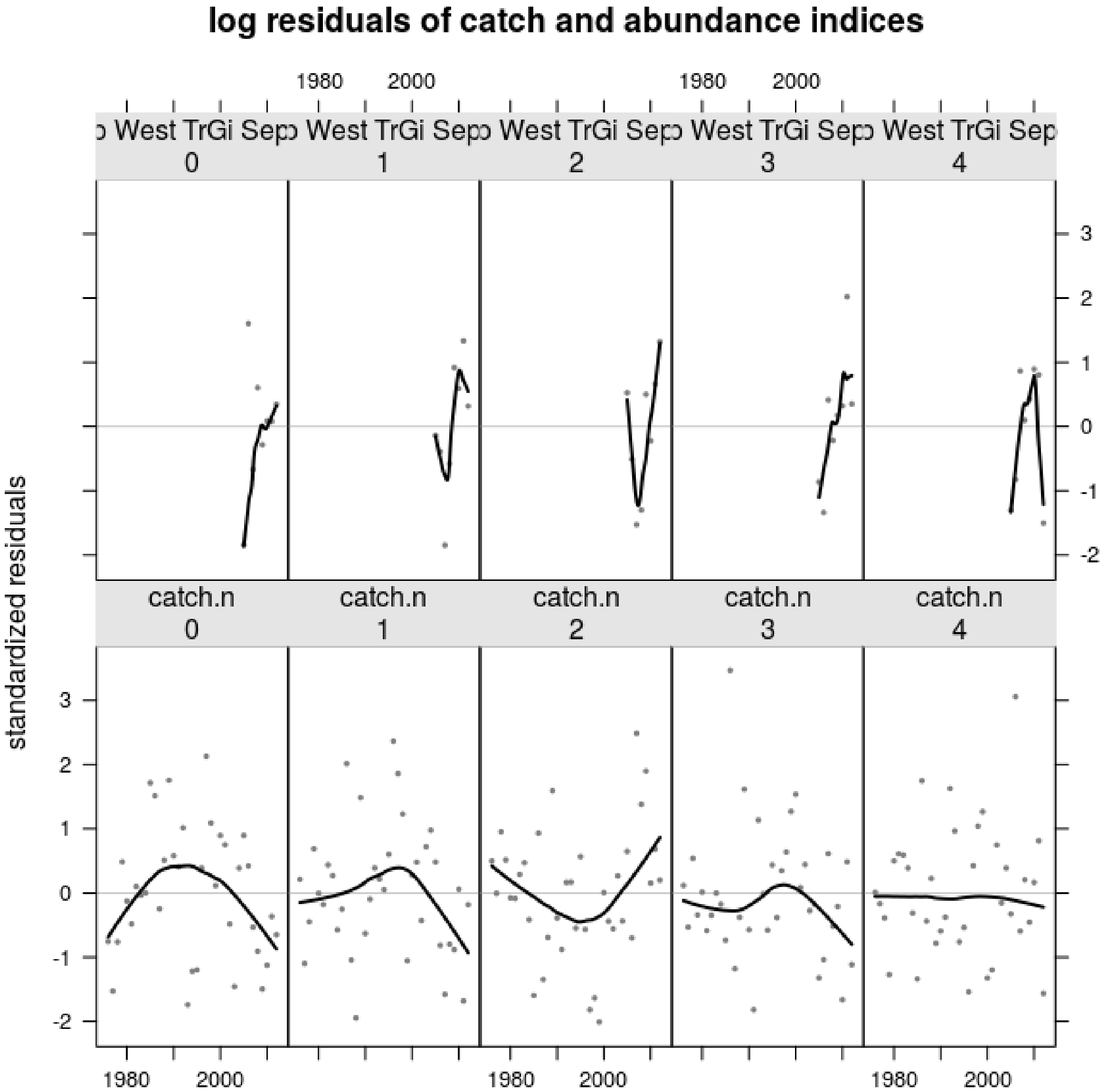


Figure 2.28: Anchovy diagnostics for fit6

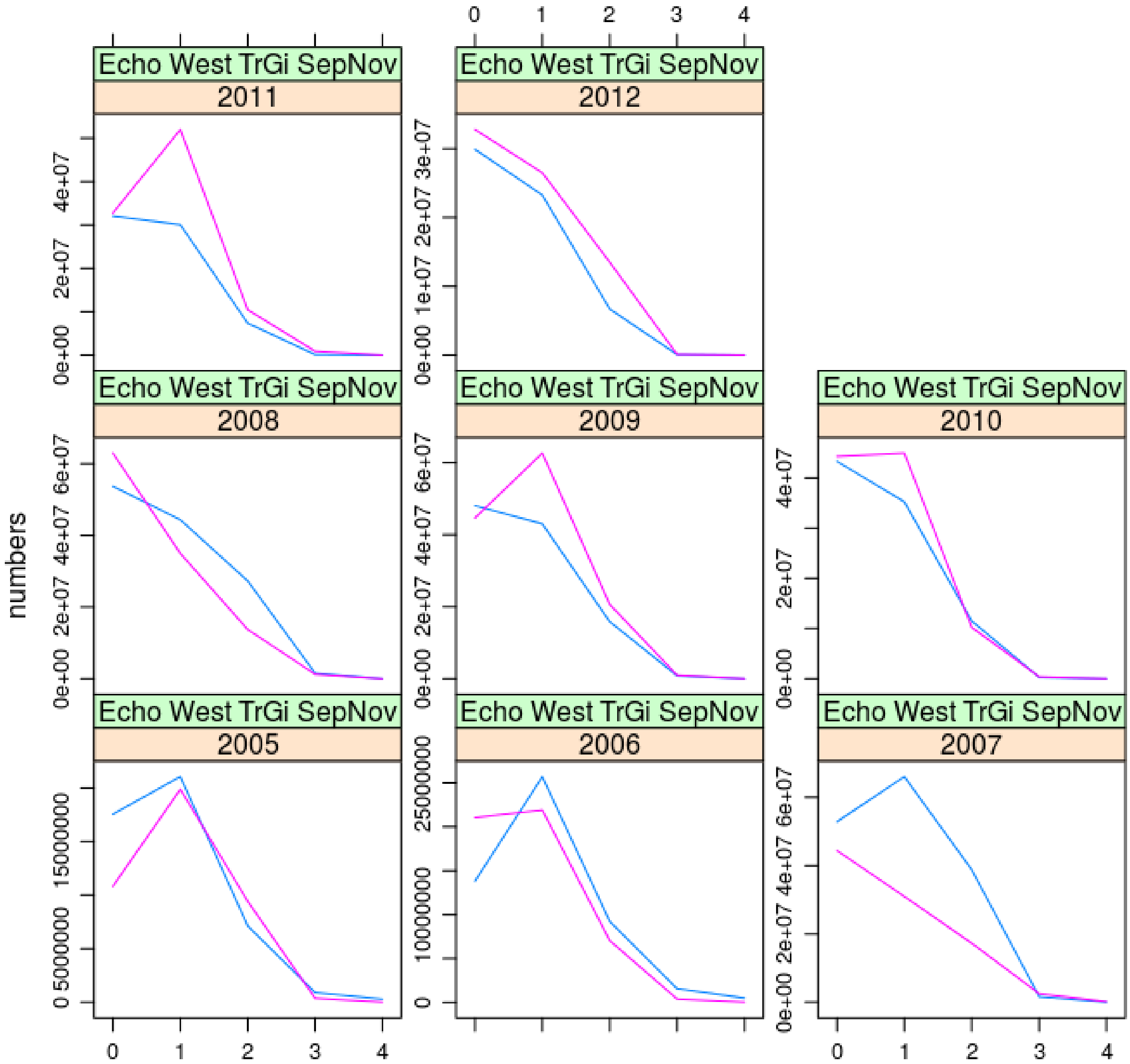


Figure 2.29: Diagnostics for fit6

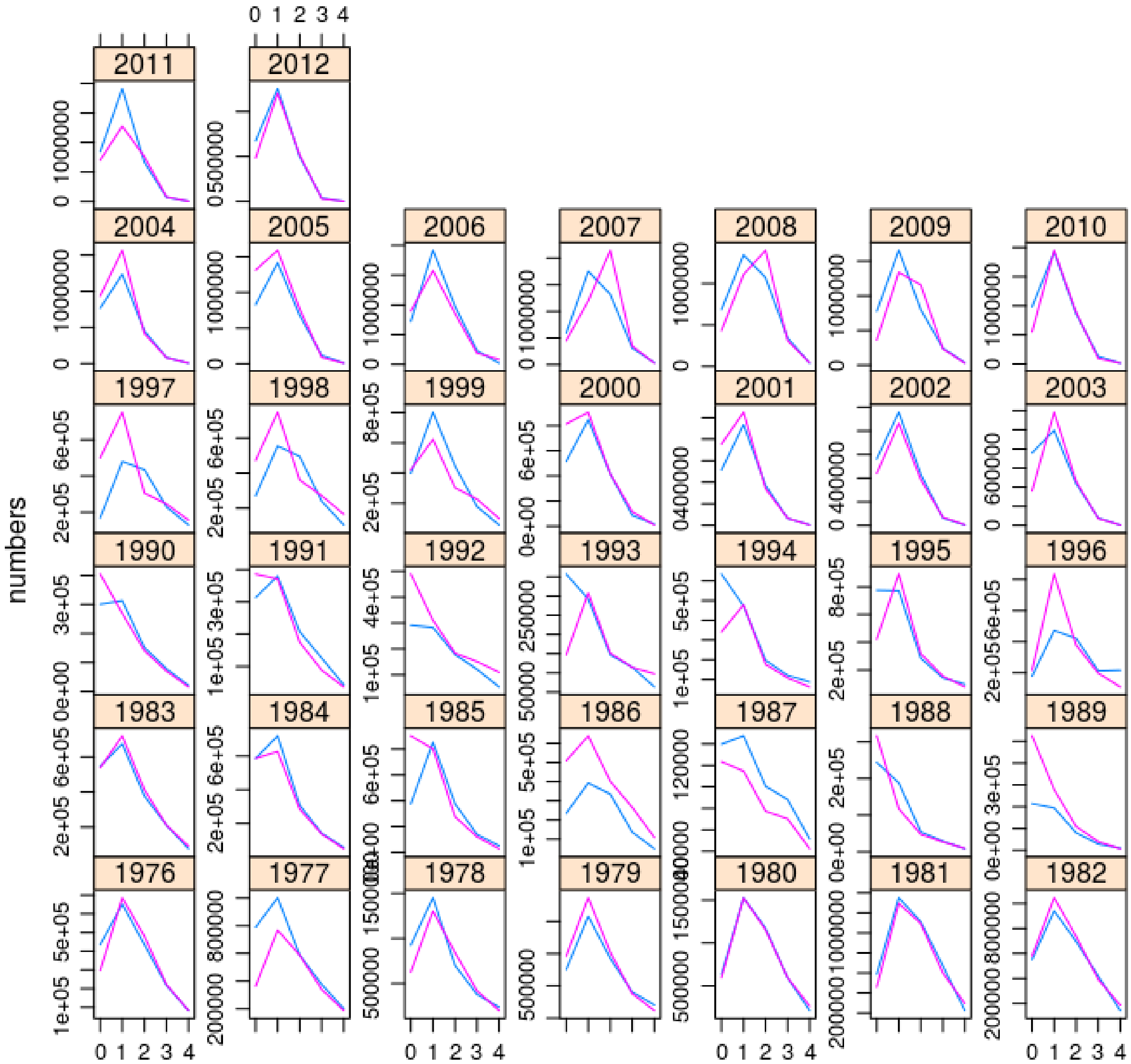


Figure 2.30: Diagnostics for fit6

The F surface is very reasonable. This model returns a max of F=3, while the SAM model in EWG 13-19 was around Fmax=12.

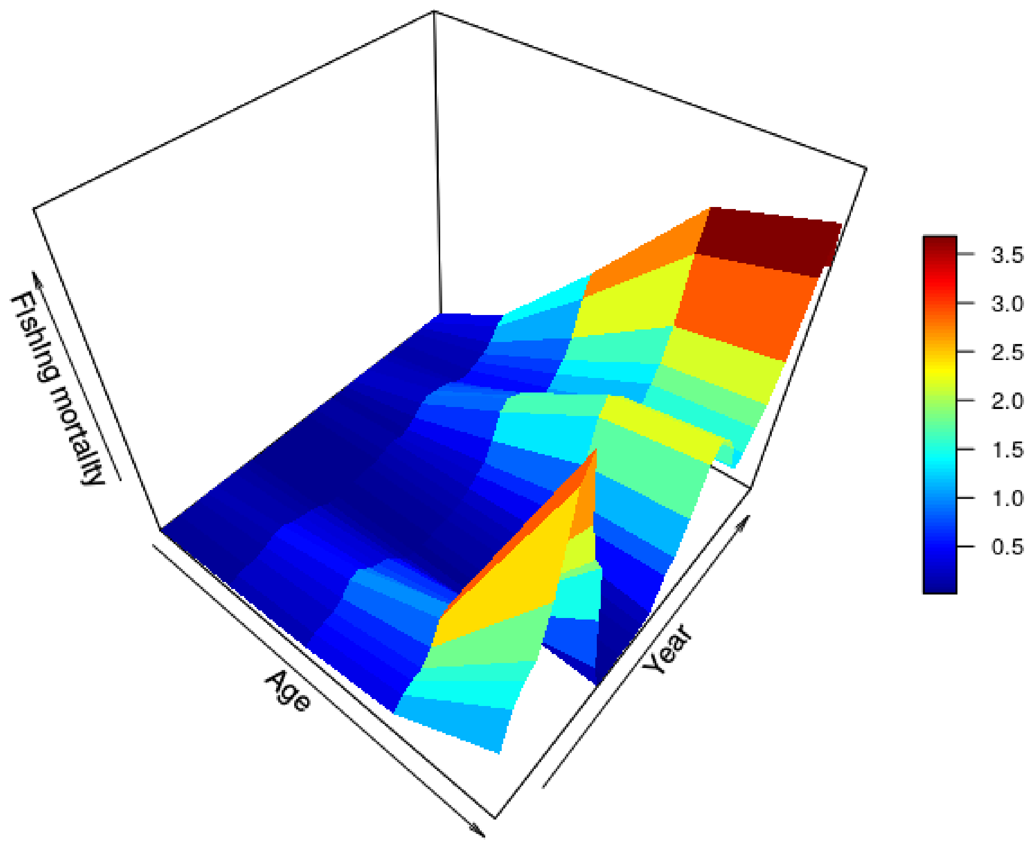


Figure 2.31: 3D surface of F at age for model fit6

2.4 Inclusion of Index Error in model fit

A different approach to improve model fit is to down weight the variance model of the survey since it's problematic on the F's in the last couple years. One attempt was made by down weighting the whole variance of the survey (code shown but no model results presented here) and one by down weighting only the last two years (plus a range of different settings)

```
# make temporary stocks not to override the original stock
stk <- ANC17
idx <- ANC17.tun[1]
# variance of observed catches, keep fixed variance for all
```

```
# years
varslt <- catch.n(stk)
varslt[] <- 0.3
catch.n(stk) <- FLQuantDistr(catch.n(stk), varslt)
# variance of observed indices
varslt <- index(idx[[1]])
# downweight the var for all years of the survey equally
# varslt[] <- 0.6 index.var(idx[[1]]) <- varslt

# downweight the var for all years of the survey equally a
# lot of weight in the last part of the tuning and high in
# the last two years w <- c(0.6, 0.6, 0.6, 0.6, 0.6, 0.6,
# 0.3, 0.3) index.var(idx[[1]]) <-rep(v, each=5) a lot of
# weight in the initial part of the tuning and low in the
# last two years.
w <- c(0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.9, 0.9)
index.var(idx[[1]]) <- rep(w, each = 5)

# run
fmod <- ~s(age, k = 5, by = breakpts(year, c(1987, 1995))) +
    s(year, k = 20, by = breakpts(age, 4))
qmod <- list(~s(age, k = 4, by = breakpts(year, 2006)))
fitvar <- a4aSCA(stk, idx, fmod = fmod, qmod = qmod)
```

Upon residual inspection, the patterns of the survey present problems.

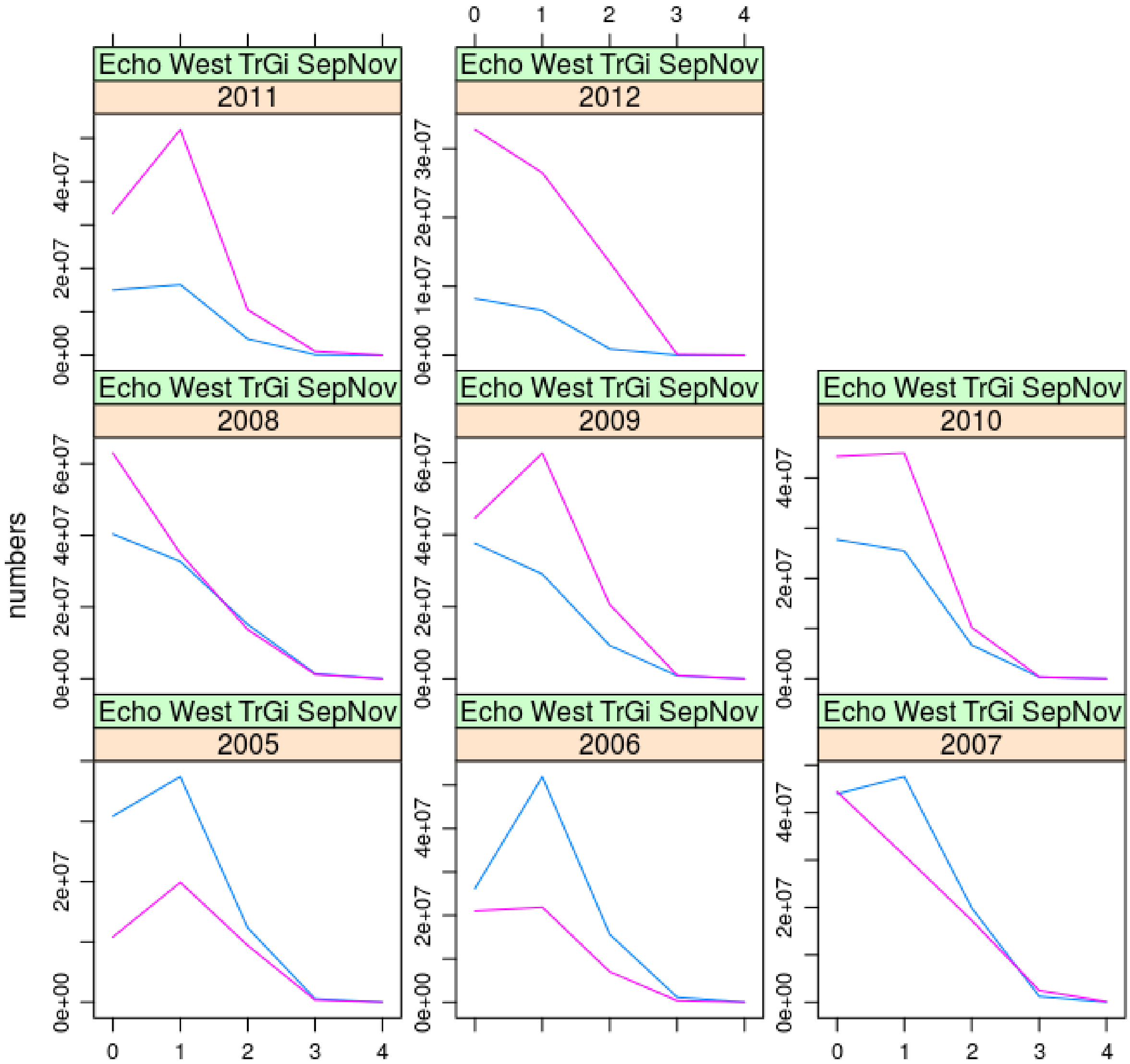


Figure 2.32: Anchovy diagnostics for fitvar

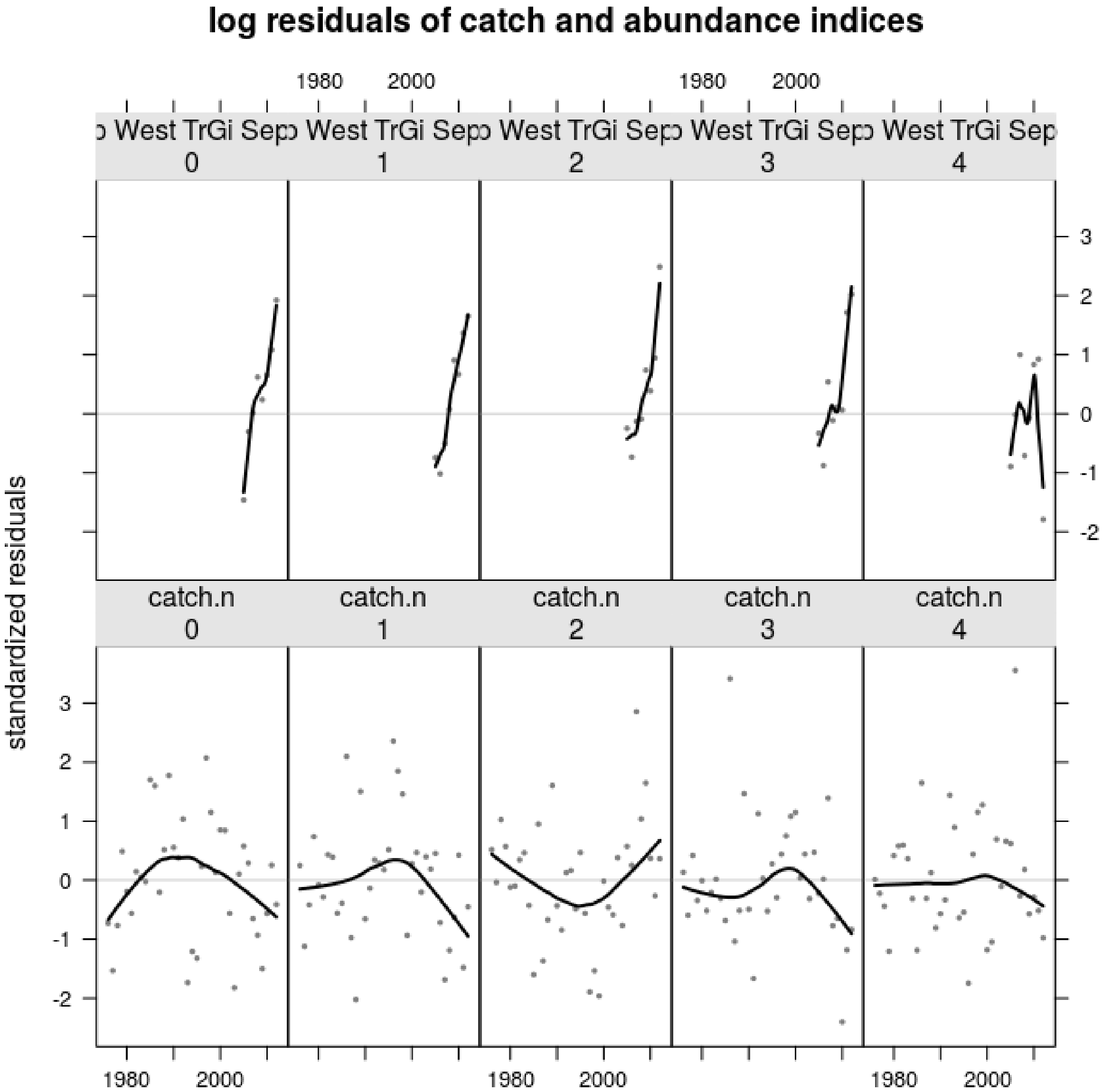


Figure 2.33: Anchovy diagnostics for fitvar

The fishing mortality surface while better than other models still is very high on age 4+ in the last years.

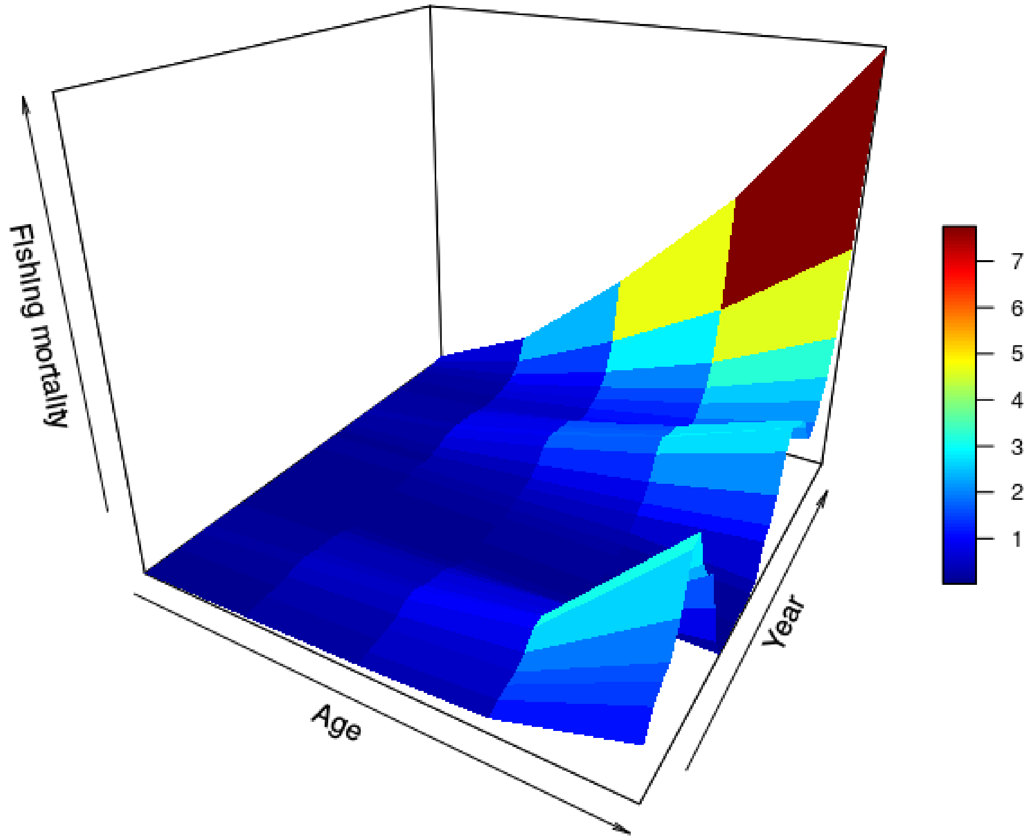


Figure 2.34: 3D surface of F at age for model fitvar

trim the survey index by removal of age = 4

The bubble plots of the cohort strengths show that the last year classes are very poorly represented in the catch matrix and while in the models before we set the plus group at 4+, here we attempt to remove catches of age = 4, while keeping the some of the fmod and qmod tested before.

```
fmod <- ~s(age, k = 5, by = breakpts(year, c(1987, 1995))) +
    s(year, k = 20, by = breakpts(age, 4))
qmod <- list(~s(age, k = 4, by = breakpts(year, 2006)))
fit7 <- a4aSCA(stock = ANC17, indices = FLIndices(trim(ANC17.tun[[1]],
    age = 0:3)), fmodel = fmod, qmodel = qmod, fit = "assessment")
```

Model fit can be compared with the AIC and BIC. In this case however the best model in terms of AIC is not fit6 which is indicated by the residuals but fit4 for AIC or fit2 according to BIC.

```
# same as above and bring in variance model. Model not
# converging
fit8 <- a4aSCA(stock = ANC17, indices = FLIndices(trim(ANC17.tun[[1]],
    age = 0:3)), fmodel = fmod, qmodel = qmod, vmodel = vmod,
    fit = "assessment")

AIC(fit0, fit1, fit2, fit3, fit4, fit5, fit6, fitvar)

##         df    AIC
## fit0    91  355.6
## fit1    73  373.6
## fit2    73  318.2
## fit3    78  373.6
## fit4   113  302.9
## fit5    65  491.1
## fit6    86  318.0
## fitvar  84 1092.6

BIC(fit0, fit1, fit2, fit3, fit4, fit5, fit6, fitvar)

##         df    BIC
## fit0    91  666.5
## fit1    73  622.9
## fit2    73  567.6
## fit3    78  640.0
## fit4   113  688.9
## fit5    65  713.1
## fit6    86  611.8
## fitvar  84 1379.5
```

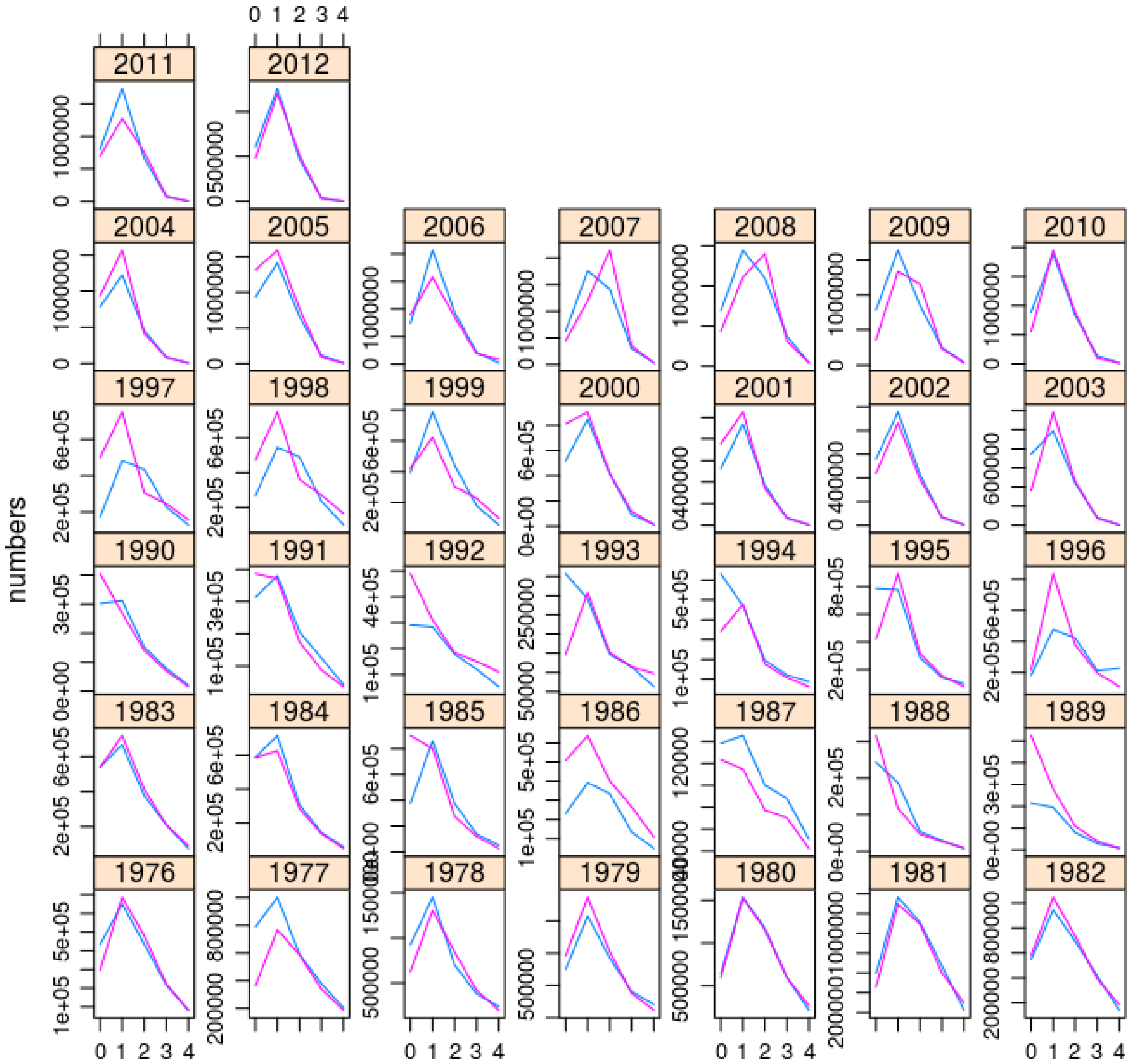


Figure 2.35: Diagnostics for model fit7

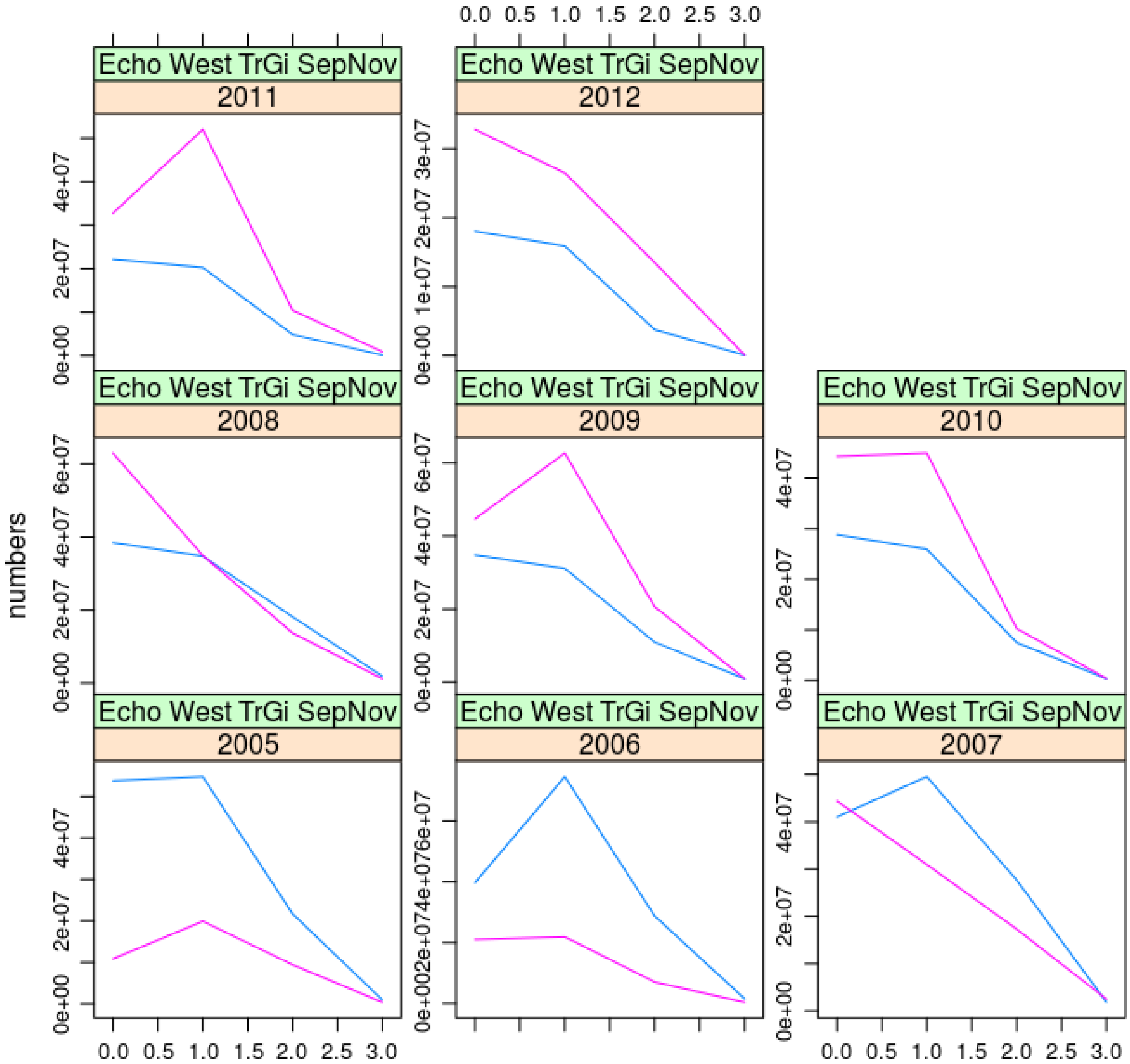


Figure 2.36: Anchovy diagnostics for fit7

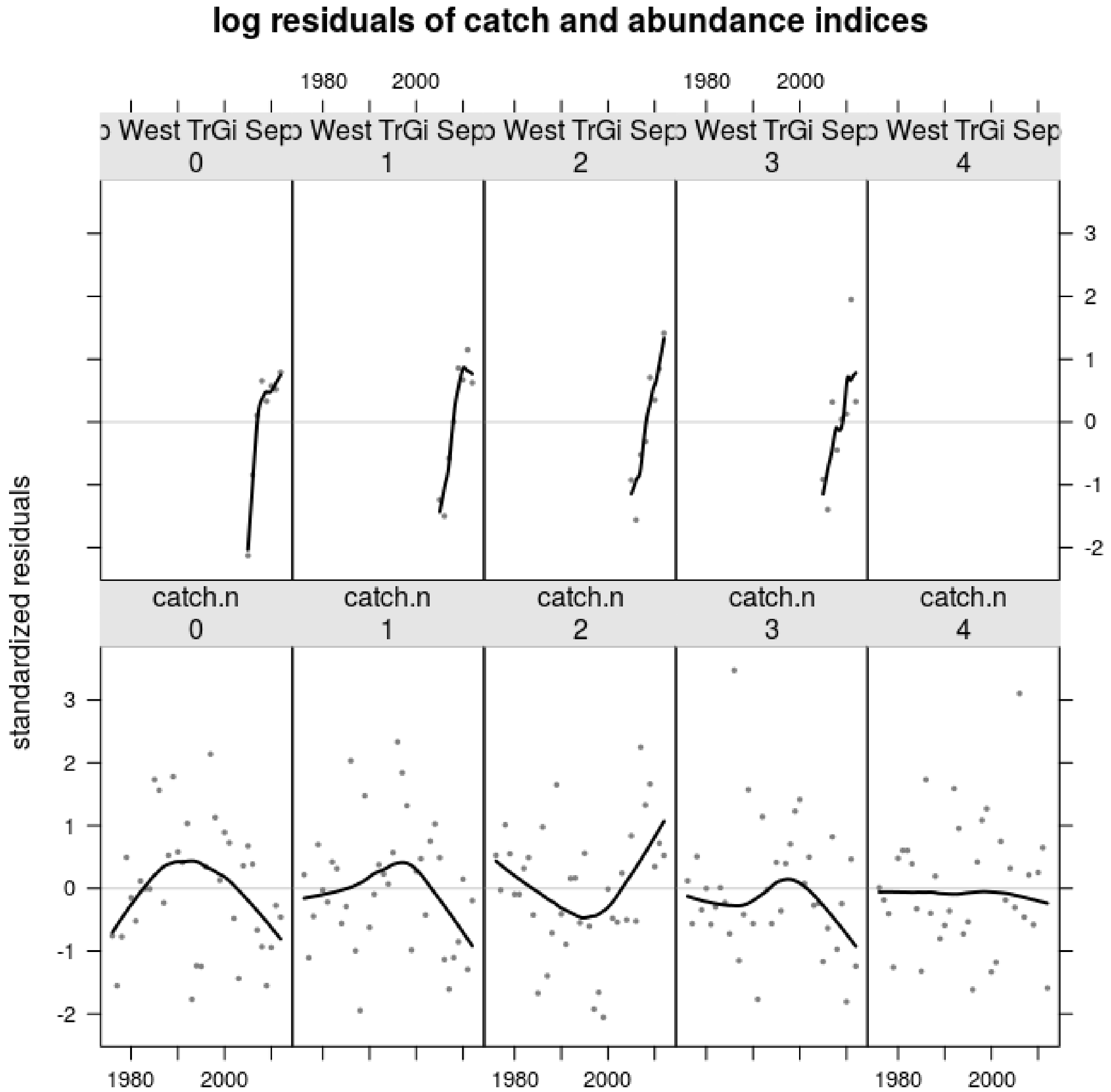


Figure 2.37: Anchovy diagnostics for fitvar

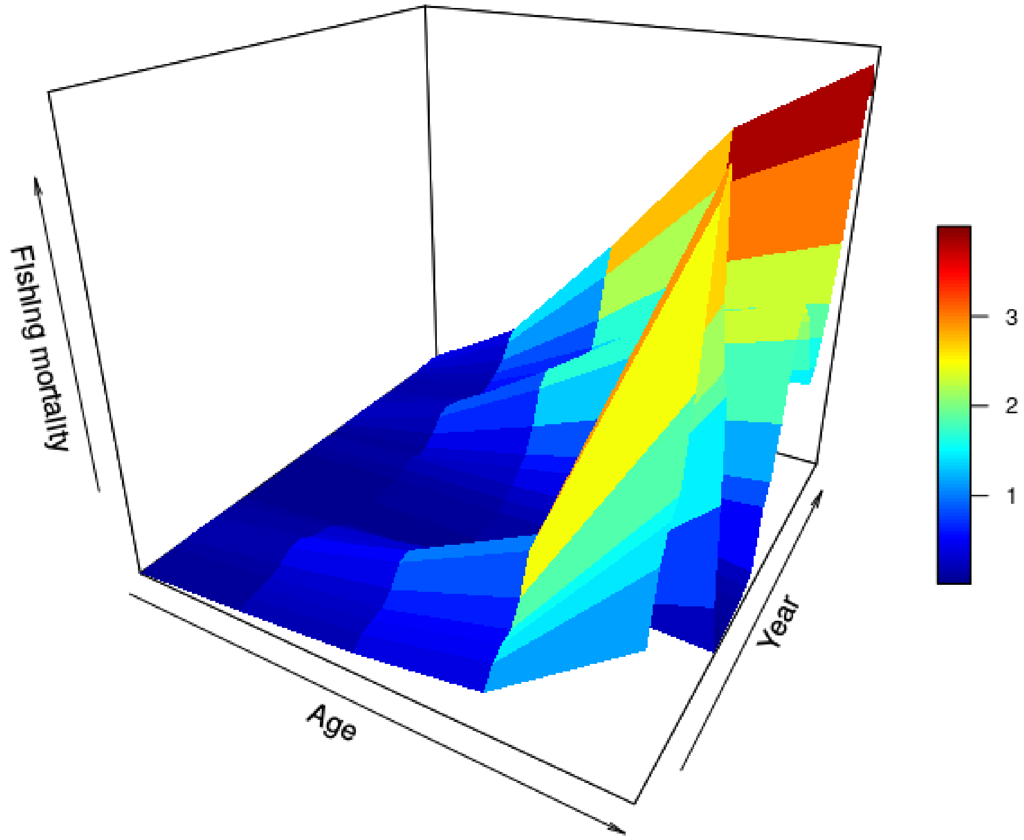


Figure 2.38: 3D surface of F at age for model fit7

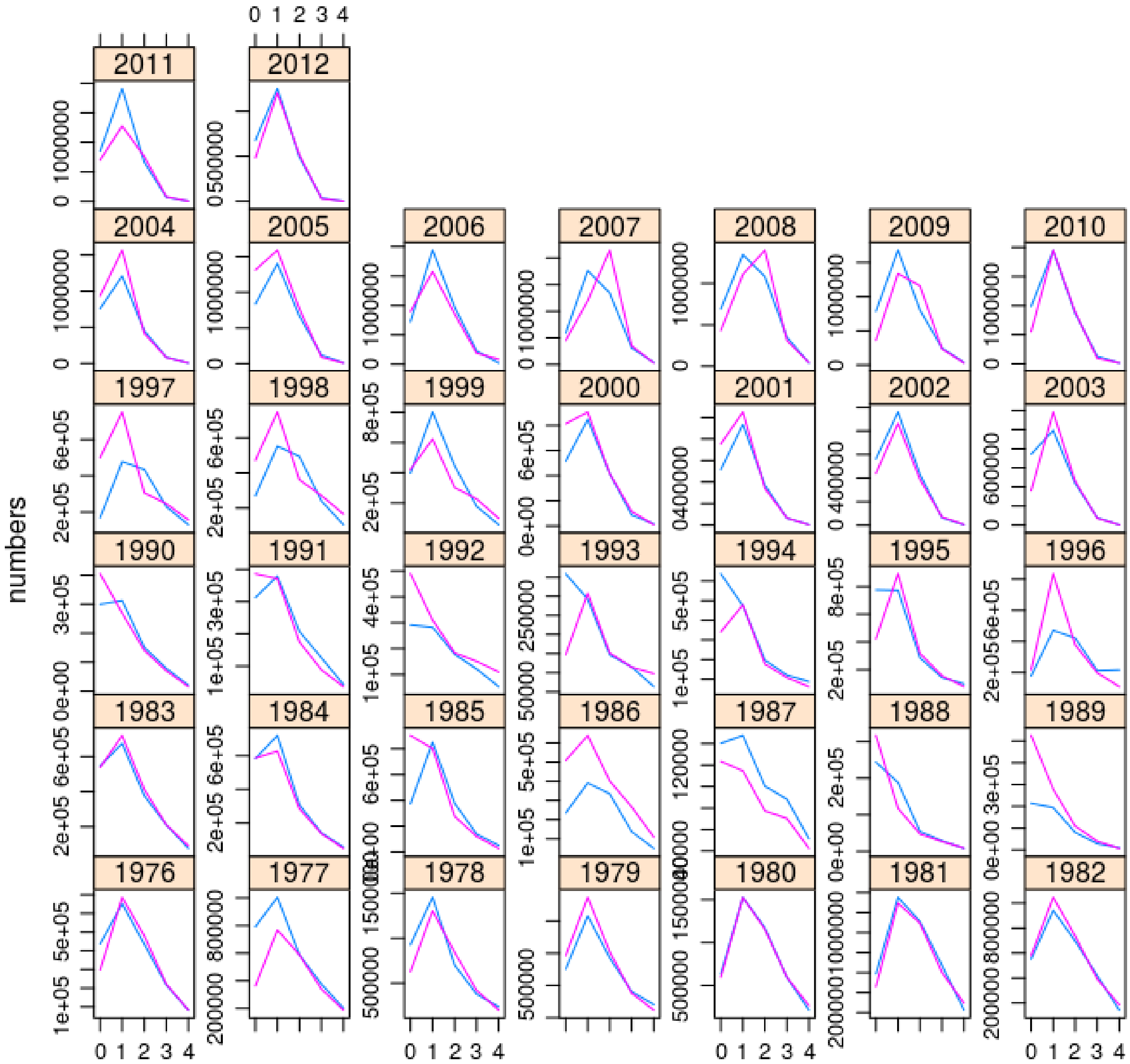


Figure 2.39: Diagnostics for model fit8

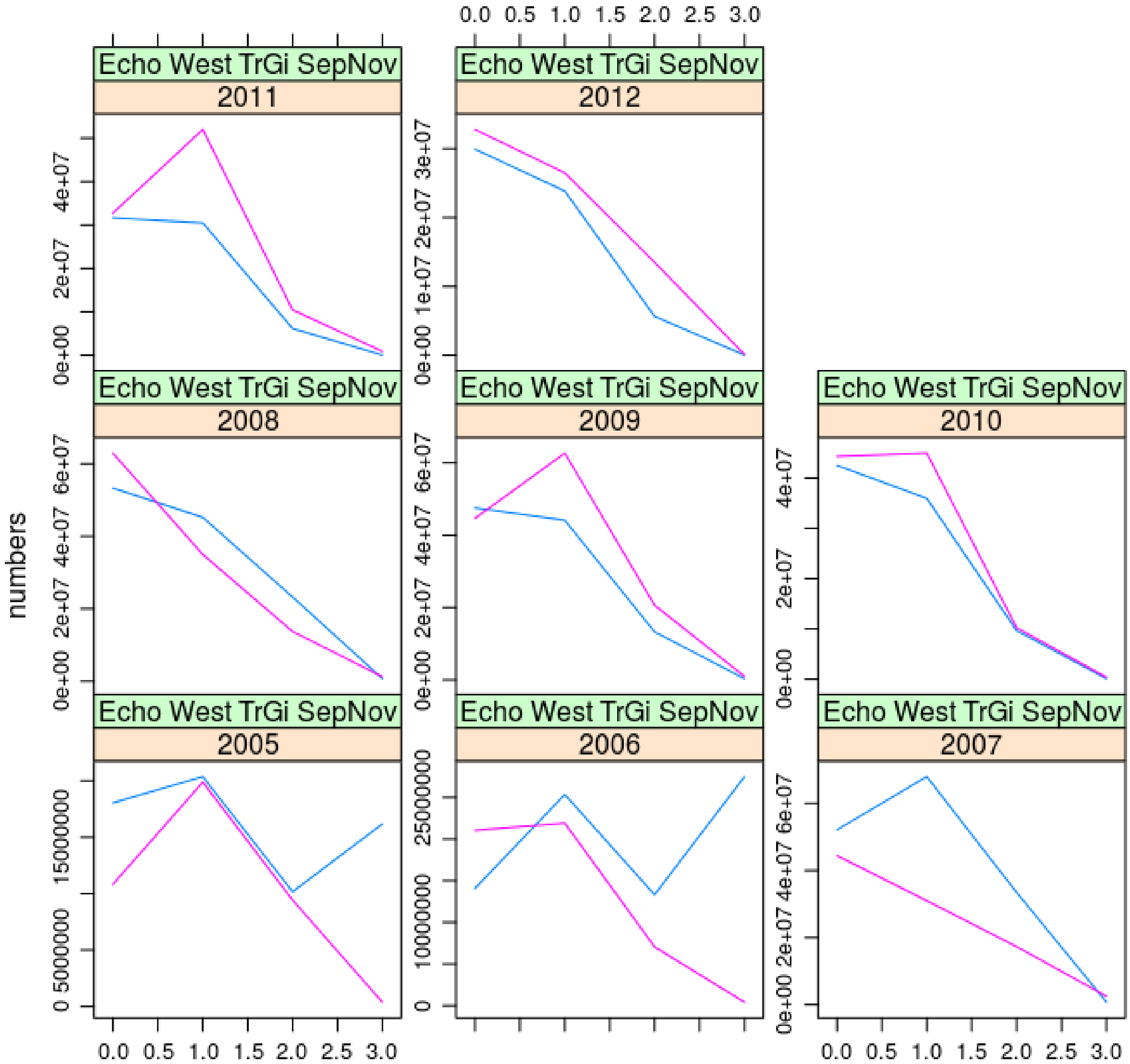


Figure 2.40: Diagnostics for model fit8

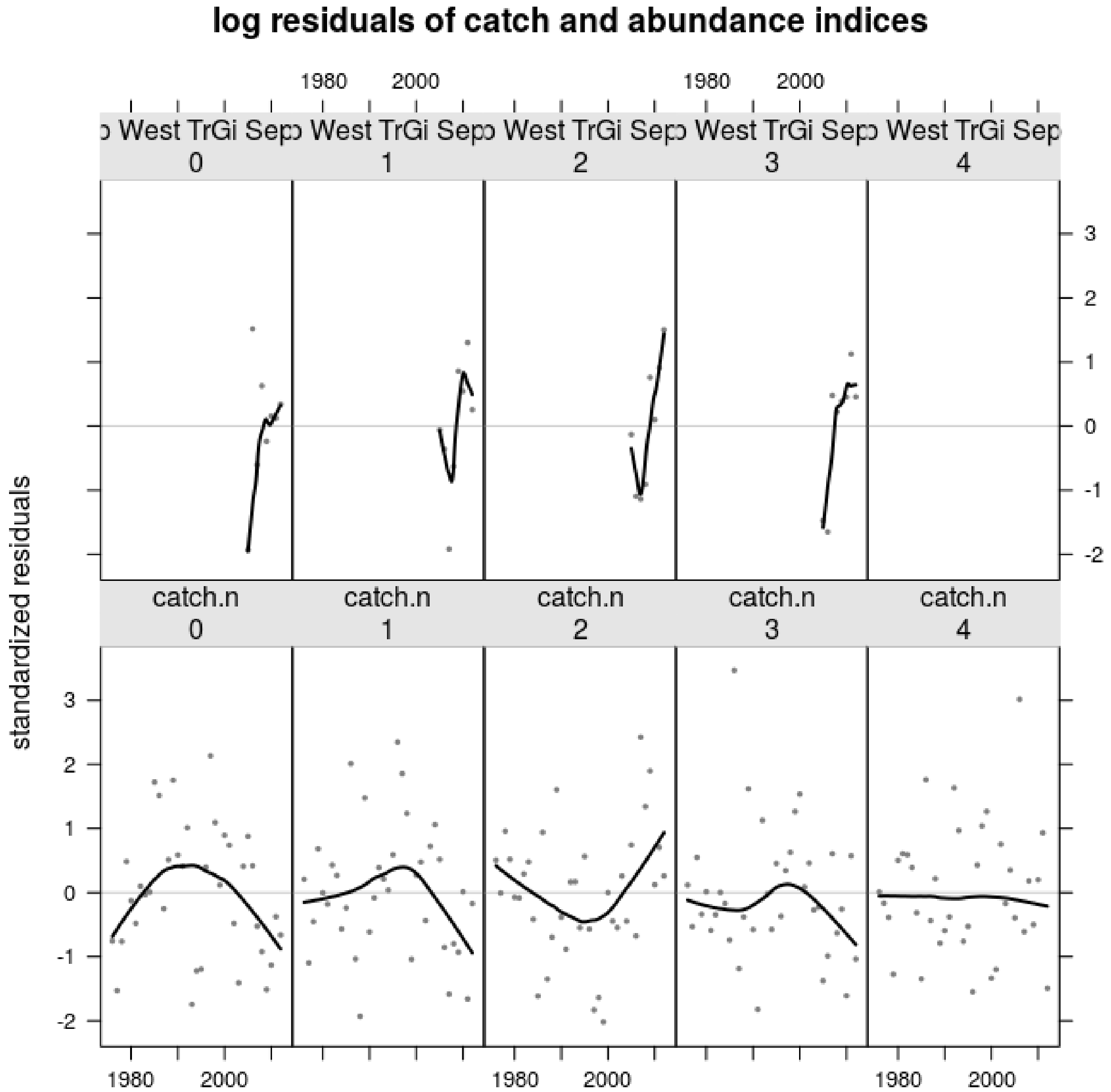


Figure 2.41: Diagnostics for model fit8

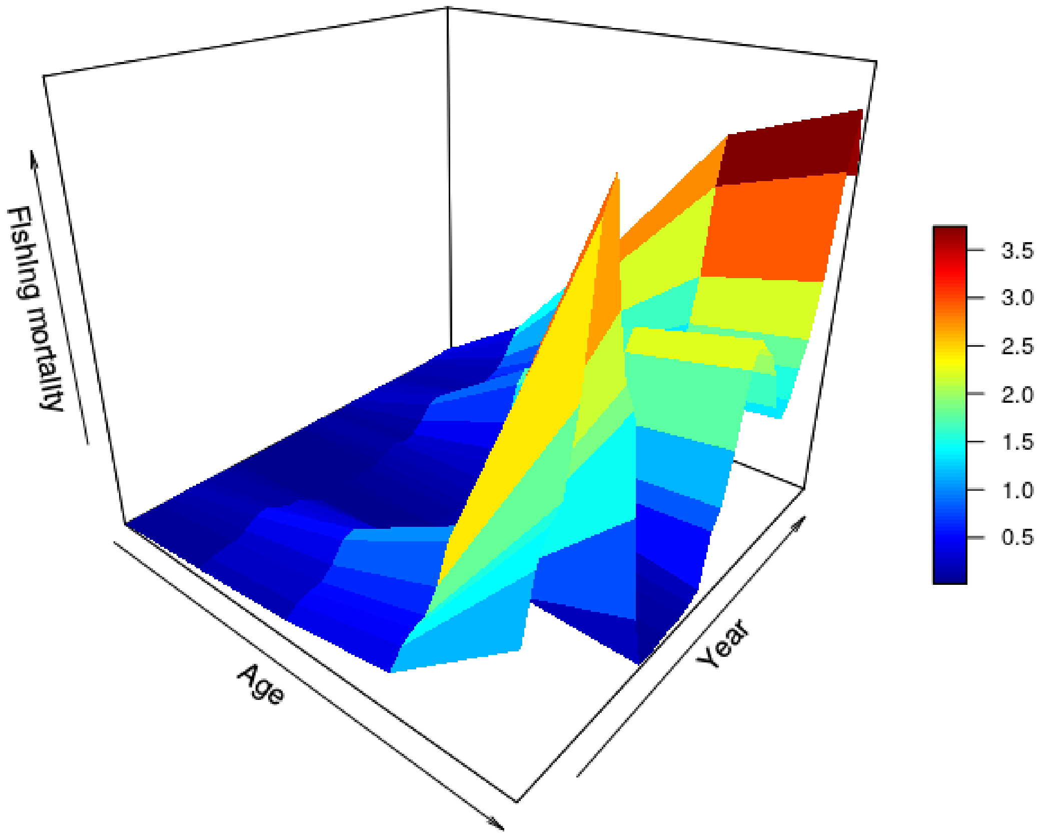


Figure 2.42: 3D shape of F at age for model fit8

An alternative to truncating age classes in the tuning index is by replacing age=4 with NAs, however the refit is not shown here.

```
# replace age = 4 with NAs
ANC17.tunNA <- ANC17.tun[[1]][5, ]
```

2.5 Use of Biomass Index in Anchovy assessment

Two historical acoustic surveys exist in the West Adriatic and were performed by CNR in ANCONA. The data from the surveys has been recovered from GFCM reports and is available as an aggregated biomass index expressed as $tons/nm^2$. We read in the data and create and FLIndex first.

```
ane_pil <- read.csv("data/ANE_PIL_acoustic.csv", sep = ";")
# Subset the stock and split the two surveys
ane <- ane_pil[ane_pil$specie == "ANE", ]
anenw <- ane[ane$survey == "nwacoustic_survey", ]
anecw <- ane[ane$survey == "midadr_acoustic", ]

anenw_ind <- anenw$tons_nm
anecw_ind <- anecw$tons_nm

# Define dimensions for the FLIndex
dnms <- list(age = "all", year = min(anenw$year):max(anenw$year))
nwidx <- FLIndexBiomass(index = FLQuant(anenw_ind, dimnames = dnms))

dnms2 <- list(age = "all", year = min(anecw$year):max(anecw$year))
cwidx <- FLIndexBiomass(index = FLQuant(anecw_ind, dimnames = dnms2))

# Assign the time of the year the survey is performed,
# uncertain here on how it is best to treat it given the
# split year

# range(nwidx)[c('startf', 'endf')] <- c(0.58, 0.66)
# range(cwidx)[c('startf', 'endf')] <- c(0.58, 0.66)
range(nwidx)[c("startf", "endf")] <- c(0, 0)
range(cwidx)[c("startf", "endf")] <- c(0, 0)
```

We then combine the 3 indexes, NW, CW and West+East in FLIndices that will be used in the assessment

```
# merge FLindexes
flis <- FLIndices(northwest = nwidx, centralwest = cwidx, west_east = ANC17.tun[[1]])
```

Now the main adjustment is in the variance model (vmod) where we can allow a different variance for each survey through a list following the order of the surveys in the "flis" FLindices. Two model fits are compared via AIC for different qmodels.

```
fmod <- ~s(age, k = 4, by = breakpts(year, c(1987, 1995))) +
    s(year, k = 20, by = breakpts(age, 4))
qmod <- list(~s(year, k = 10), ~s(year, k = 7), ~s(age, k = 4,
    by = breakpts(year, 2006)))
qmod1 <- list(~s(year, k = 20), ~s(year, k = 10), ~s(age, k = 4,
    by = breakpts(year, 2006)))
vmod <- list(~s(age, k = 3), ~1, ~1, ~s(age, k = 3))
fit6ts1 <- a4aSCA(stock = ANC17, indices = flis, fmodel = fmod,
    qmodel = qmod, vmodel = vmod)
fit6ts2 <- a4aSCA(stock = ANC17, indices = flis, fmodel = fmod,
    qmodel = qmod1)
AIC(fit6ts1, fit6ts2)


##          df   AIC
## fit6ts1 102 440.5
## fit6ts2 113 456.0
```

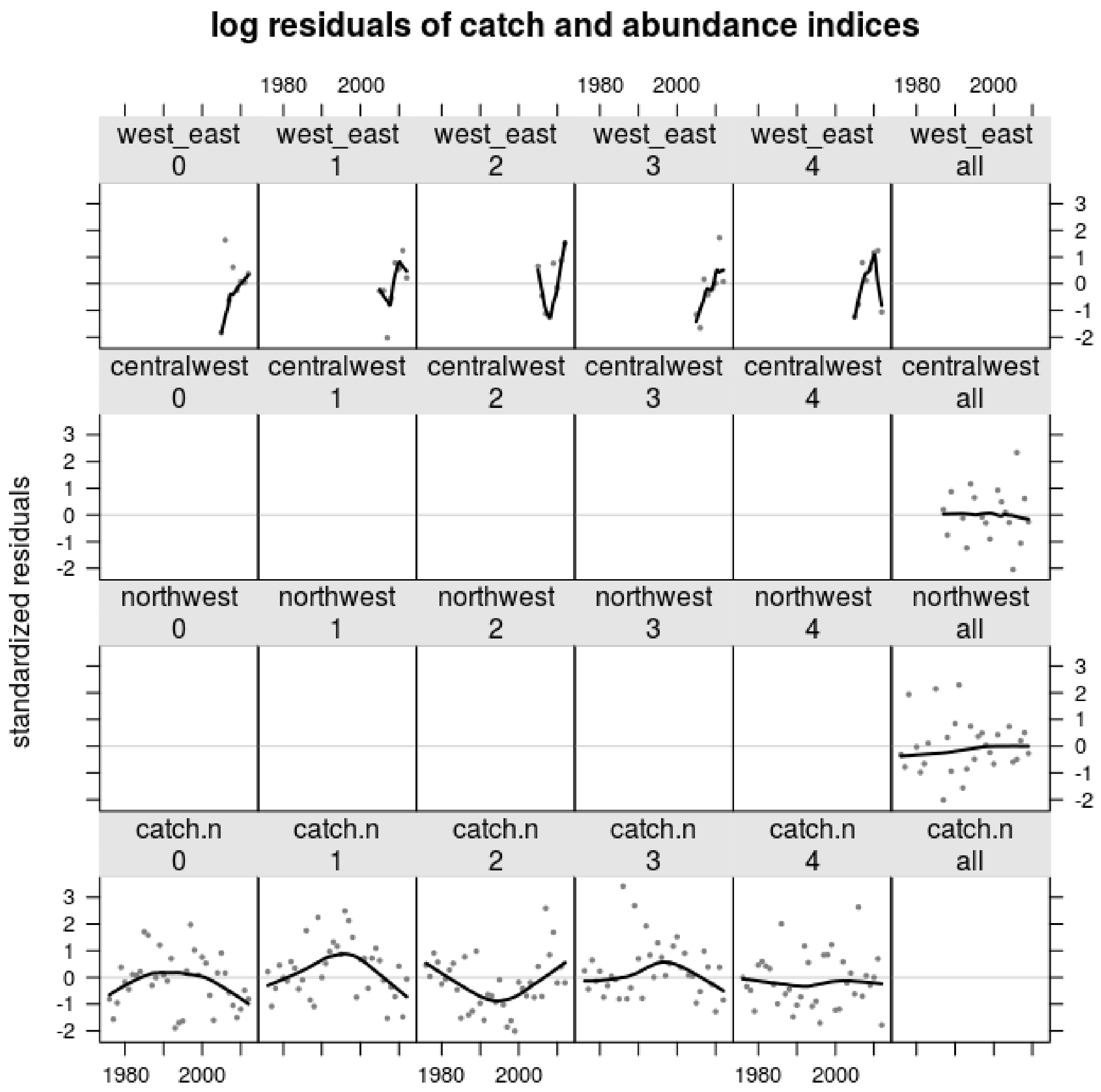


Figure 2.43: Residuals plots for fit6ts1

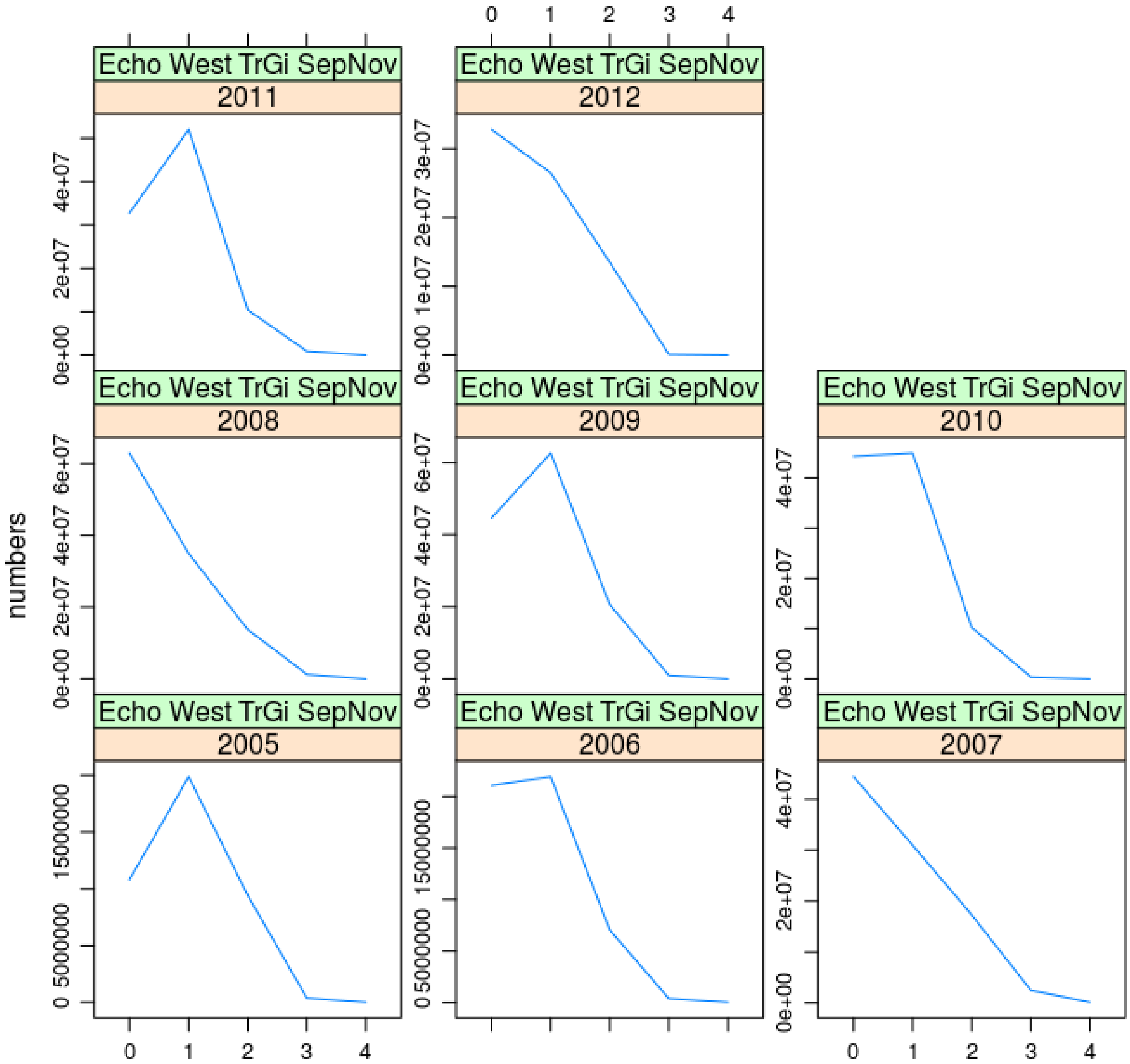


Figure 2.44: Residuals plots for fit6ts1

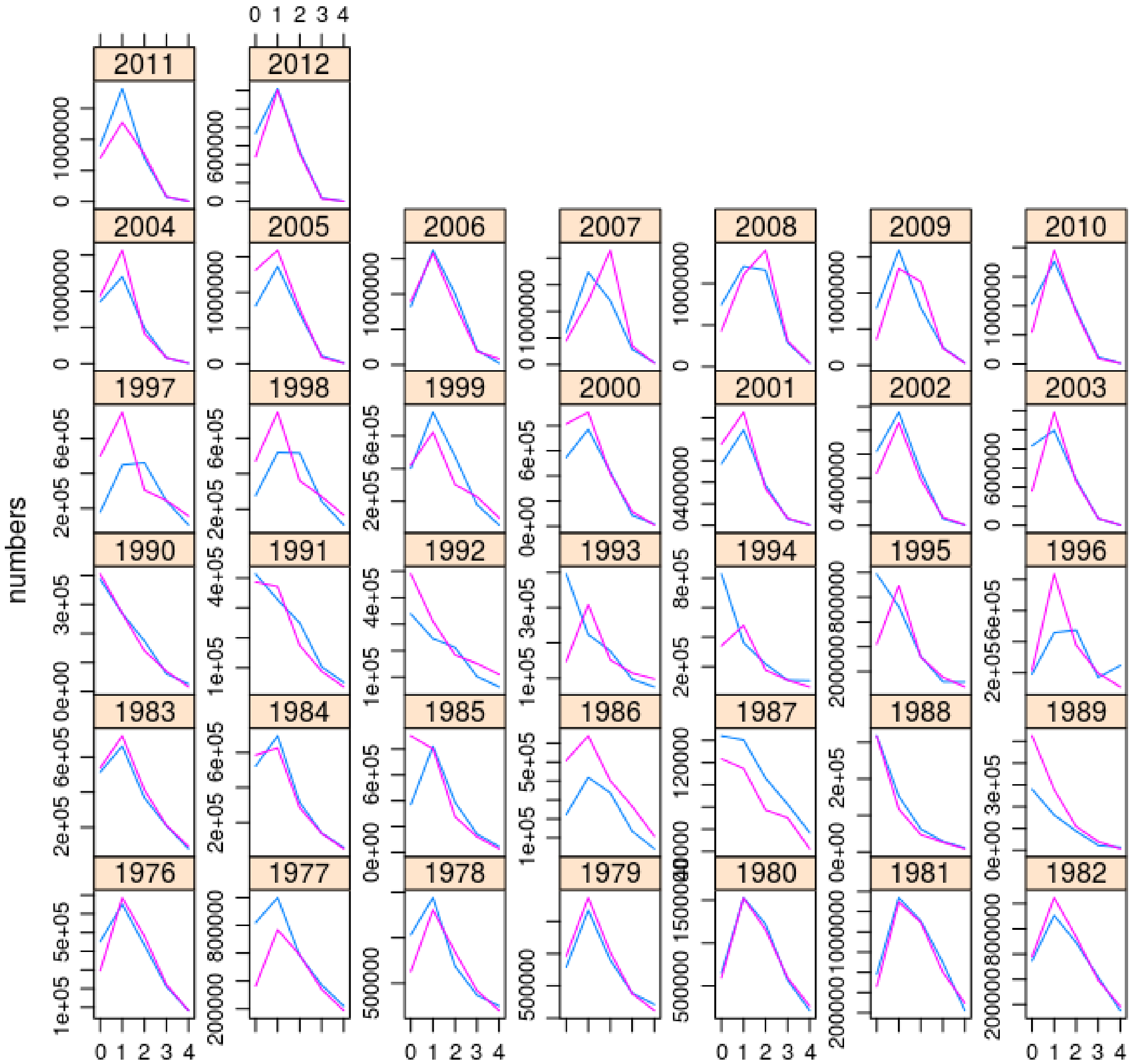


Figure 2.45: Residuals plots for fit6ts1

Fit6ts1 is the best model obtained, the residuals, predicted vs observed in catch and survey are quite acceptable and the F surface is very reasonable. This model returns a max of F=3, while the SAM model in EWG 13-19 was around Fmax=12.

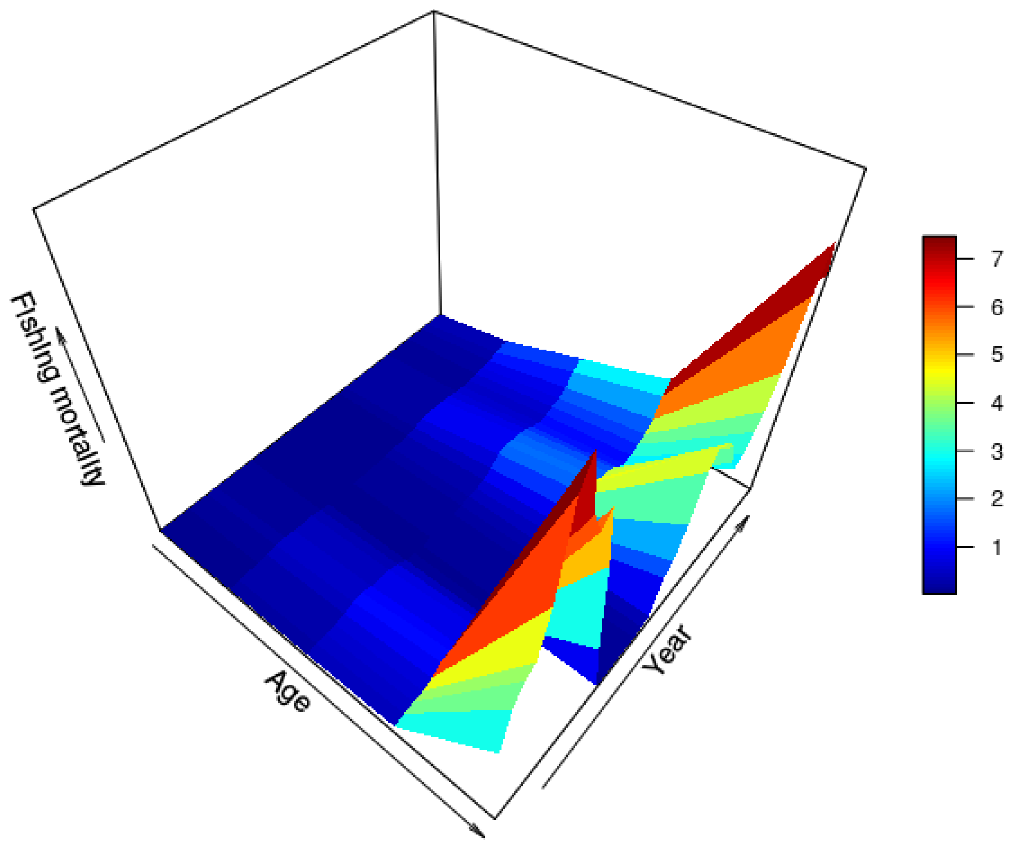


Figure 2.46: 3D shape of F at age for model fit6ts1

Another attempt with the biomass tuning indexes is with the following model

```
fmod <- ~s(age, k = 5, by = breakpts(year, c(1987, 1995))) +
    s(year, k = 20, by = breakpts(age, 4))
qmod <- list(~s(year, k = 10), ~s(year, k = 7), ~s(age, k = 4,
    by = breakpts(year, 2006)))
vmod <- list(~s(age, k = 3), ~1, ~1, ~s(age, k = 3))
fitk <- a4aSCA(stock = ANC17, indices = flis, fmodel = fmod,
    qmodel = qmod, vmodel = vmod, n1model = ~s(age, k = 3))
```

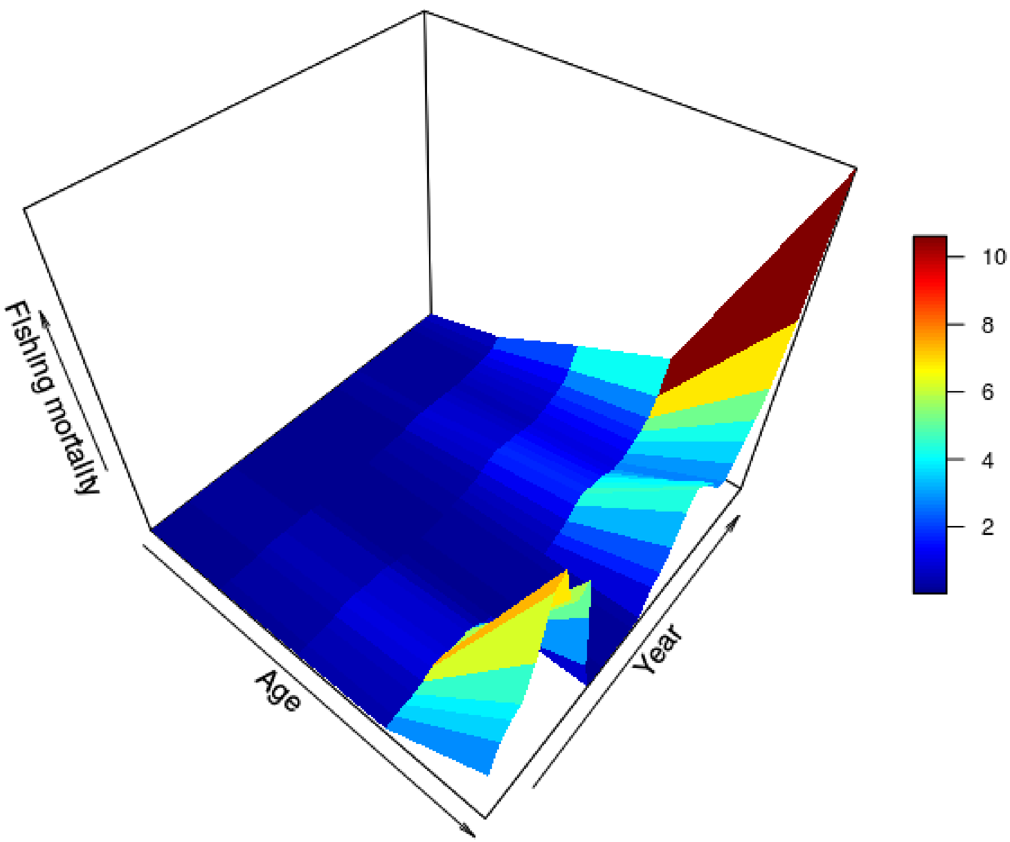


Figure 2.47: 3D surface of F at age for model fitk

```
##          df   BIC
## fit6ts1 102 808.6
## fit6ts2 113 863.9
```

To get confidence intervals around the estimates of the a4a model it is possible to simulate for a number of iteration, here 250.

```
# simulate Confidence intervals for the assessed
sim <- simulate(fit6ts1, 250)
```

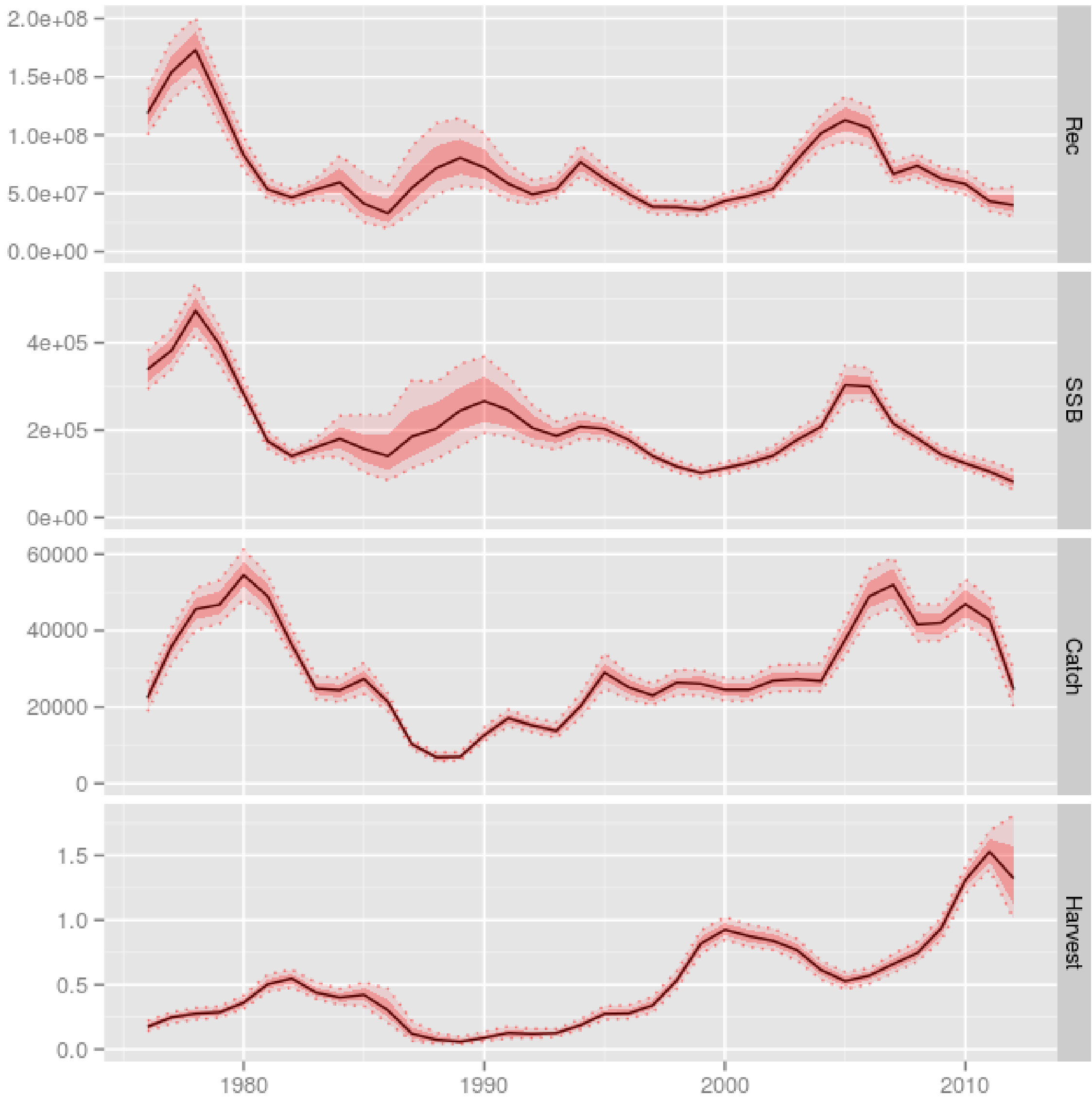


Figure 2.48: Simulated fits with confidence intervals

Estimate fbar out of the 250 iterations and for the last year.

```
fbar(ANC17 + sim)


## An object of class "FLQuant"
## iters:  250
##
## , , unit = unique, season = all, area = unique
##
##      year
## age   1976             1977             1978             1979
##   all 0.174209(0.0335) 0.247655(0.0332) 0.276716(0.0405) 0.286451(0.0324)
##      year
## age   1980             1981             1982             1983
```

```
##   all 0.361396(0.0452) 0.503634(0.0536) 0.548786(0.0583) 0.440458(0.0489)
##      year
## age   1984             1985             1986             1987
##   all 0.400750(0.0470) 0.422264(0.0717) 0.302230(0.0993) 0.120168(0.0596)
##      year
## age   1988             1989             1990             1991
##   all 0.074308(0.0344) 0.058955(0.0223) 0.089294(0.0272) 0.127247(0.0344)
##      year
## age   1992             1993             1994             1995
##   all 0.119625(0.0308) 0.126139(0.0257) 0.186616(0.0329) 0.274683(0.0401)
##      year
## age   1996             1997             1998             1999
##   all 0.278258(0.0382) 0.340442(0.0473) 0.533574(0.0584) 0.818248(0.0720)
##      year
## age   2000             2001             2002             2003
##   all 0.925091(0.0670) 0.875941(0.0709) 0.839918(0.0647) 0.767396(0.0668)
##      year
## age   2004             2005             2006             2007
##   all 0.614161(0.0634) 0.528360(0.0567) 0.570474(0.0471) 0.660883(0.0573)
##      year
## age   2008             2009             2010             2011
##   all 0.744503(0.0642) 0.938043(0.0755) 1.309568(0.0861) 1.529942(0.1241)
##      year
## age   2012
##   all 1.322352(0.3310)
##
## units:  f
```

```
summary(fbar(ANC17 + sim)[, "2012"])
```

```
## An object of class "FLQuant" with:
## dim  :  1 1 1 1 1 250
## quant:  age
## units:  f
##
## Min    :  0.6261
## 1st Qu.:  1.121
## Mean   :  1.381
## Median :  1.322
## 3rd Qu.:  1.572
## Max    :  2.792
## NAs    :  0 %
```

2.6 Compare all a4a models with SAM results

We compare all the model fits, irrespective to the quality of the fit, to the SAM assessment model. In Recruitment and SSB there is a main difference in the models (fit6, fit6ts1 and fitk_surv) picking up an increase in Rec and SSB in the mid eighties till early nighties.

Catch does not change much across models, while Harvest estimates differ mainly in the peak in 1987.

```
res <- FLStocks(SAM = ANC17, FIT1 = ANC17 + fit1, FIT2 = ANC17 +
    fit2, FIT3 = ANC17 + fit3, FIT4 = ANC17 + fit4, FIT5 = ANC17 +
    fit5, FIT6 = ANC17 + fit6, FIT7 = ANC17 + fit7, FIT6ts1 = ANC17 +
    fit6ts1, FITk_surv = ANC17 + fitk)
```

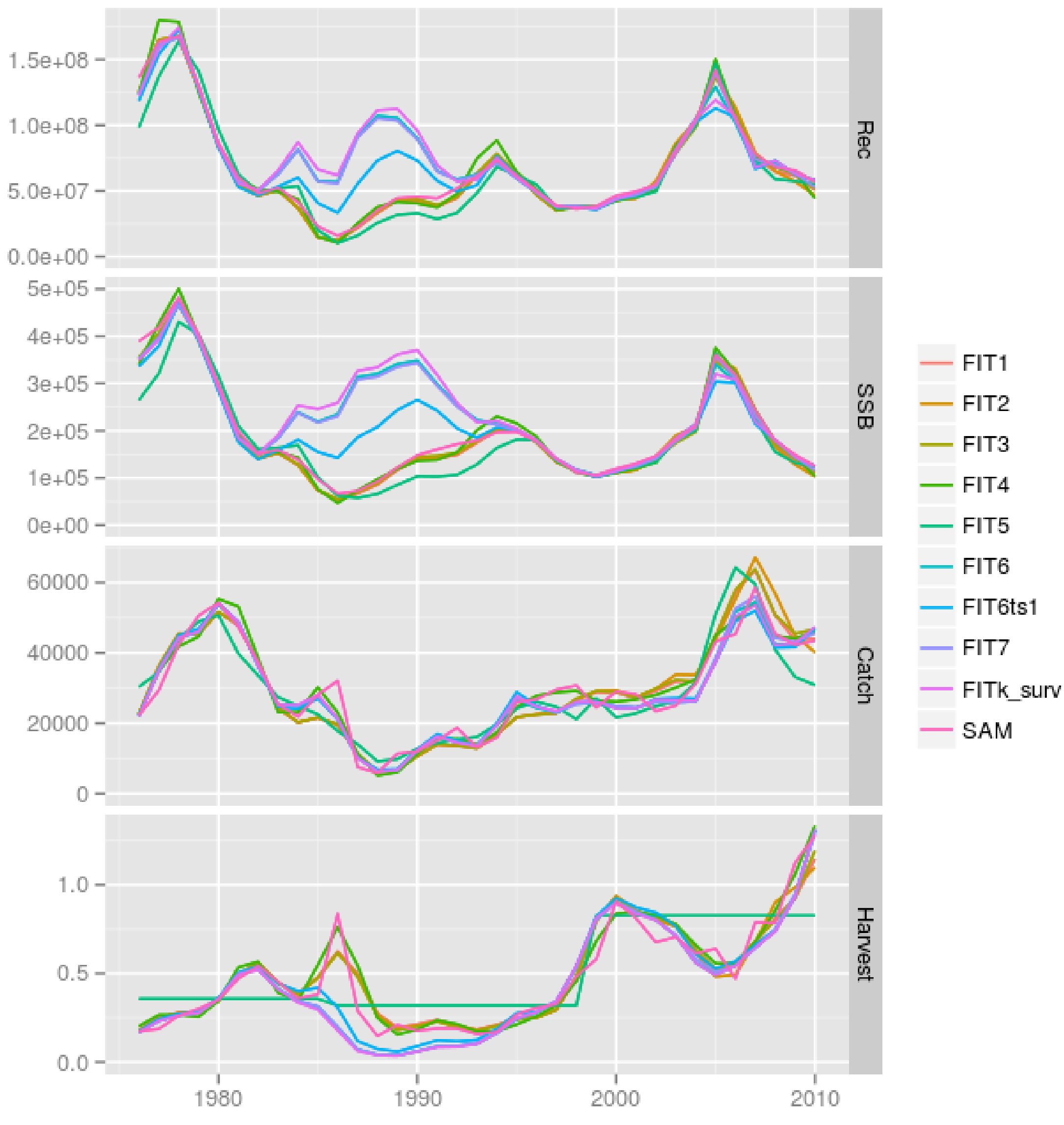


Figure 2.49: Comparison of model a4a fits with SAM fit

2.7 Way forward for the Anchovy assessment in GSA 17

There is a number of pending issues with the EWG 13-12 anchovy SAM assessment7 in GSA 17 and with the a4a models fitted in the report:

- The tuning index West+East (2005-2012) has assumptions about ALK use, it was combined and is short compared to the landing series.
- There are problems of internal consistency between the cohorts which might be generated by the merging process.
- Additionally, the lack of survey data for the early part of the series tends to increase the uncertainty in the fit for SSB and Rec in the period 1980-1990.
- Another potential problem is the catch data comes in as split year.

Possible solutions to the described problems are the following:

- The West+East merged survey should be kept as two separate indexes and modeled accordingly so that separate catchabilities can be included, or variance models.
- The inclusion of the Biomass index from the North and West Central Adriatic seems to stabilize this part of the assessment and highlights the need of including these additional tuning indexes, possibly disaggregated by age, since the data exist disaggregated.
- Since ALK readings from West are applied to East for the production of the combined tuning index, introducing a growth model with some uncertainty could help carry over the uncertainty associated with age slicing.

3 Sardine in the North Adriatic Sea

Iago Mosqueira

3.1 Assessments with the statistical catch-at-age method

3.1.1 Data and input

Data on catch-at-age, mean-weights-at-age, maturity, natural mortality and other biological parameters was extracted from the latest stock assessment conducted for this stock (REF) and recently submitted data.

It is worth noting that the available index of abundance by age only covers the latest part of the fishery (2005-2012), although catch-at-age data strats in 1975. This is likely to complicate matters, as trends in population abundance and fishing mortality prior to 2005 will be estimated based almost entirely on trends in catch-at-age, which are likely to be affected by changes in fishing effort, targeting and other factors not related to stock status.

```
load("data/SardineGSA17.RData")
sar <- SARDINE
tun <- SARDINE.tun
```

Figure 3.1: Time series of catches (in t) and relative abundance at age for Adriatic sardine

3.1.2 Initial model runs

A number of model runs were carried out to explore the influence of various model options in the results, and with a view at approximating the current stock assessment.

R1

The first run (R1) accepted most of the default options in the a4a model to simply explore what inferences could more directly be made form the data. The stock-recruitment relationship is simply a year factor, similar to the random walk used by SAM.

```r
r1 <- sca(sar, tun, srmodel = ~factor(year), fit = "assessment")
show(r1)

## a4a model fit for: Sardine GSA 17
##
## Call:
## .local(stock = stock, indices = indices, fmodel = ..1, qmodel = ..2,
##     srmodel = ..3, n1model = ..4, vmodel = ..5, fit = ..6)
##
## Time used:
##  Pre-processing     Running a4a Post-processing           Total
##          0.4277          0.8402          0.4612          1.7291
##
## Submodels:
##   fmodel: ~te(age, year, k = c(3, 19), bs = "tp")
##  srmodel: ~factor(year)
##  n1model: ~factor(age)
##   qmodel:
##     Echo West + East TrGi SepNov Commercial LFD: ~s(age, k = 4)
##   vmodel:
##     catch:                                      ~s(age, k = 3)
##     Echo West + East TrGi SepNov Commercial LFD: ~1
```

The results (Figure 3.2) appear to indicate a large decrease in abundance for this stock, as the model resorts to a large biomass as the start of the series to explain the sustained increase in catches in the period up to 1983.

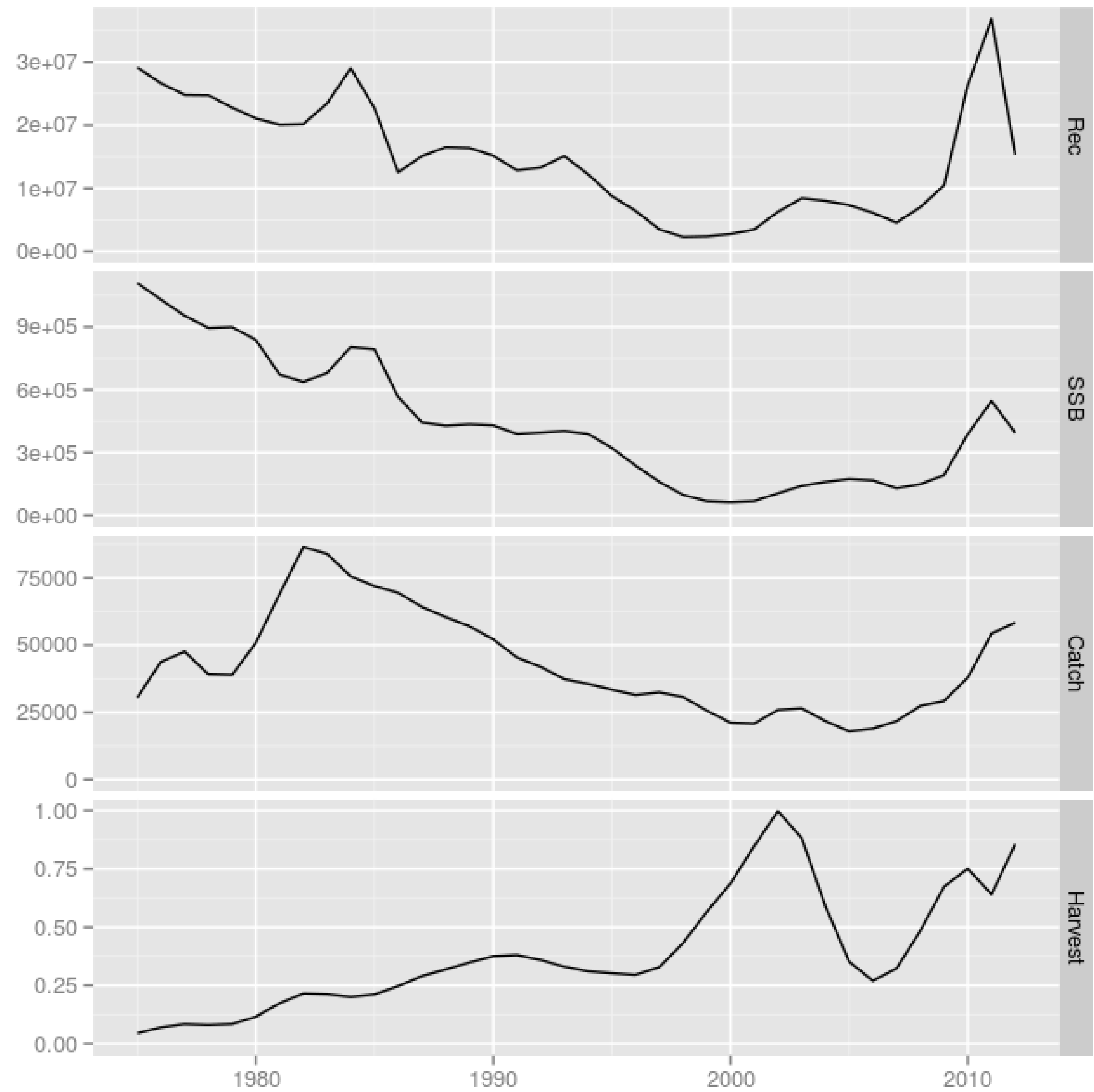


Figure 3.2: Estimated recruitment, SSB, catch and fishing mortality for Adriatic sardine, using model run R1

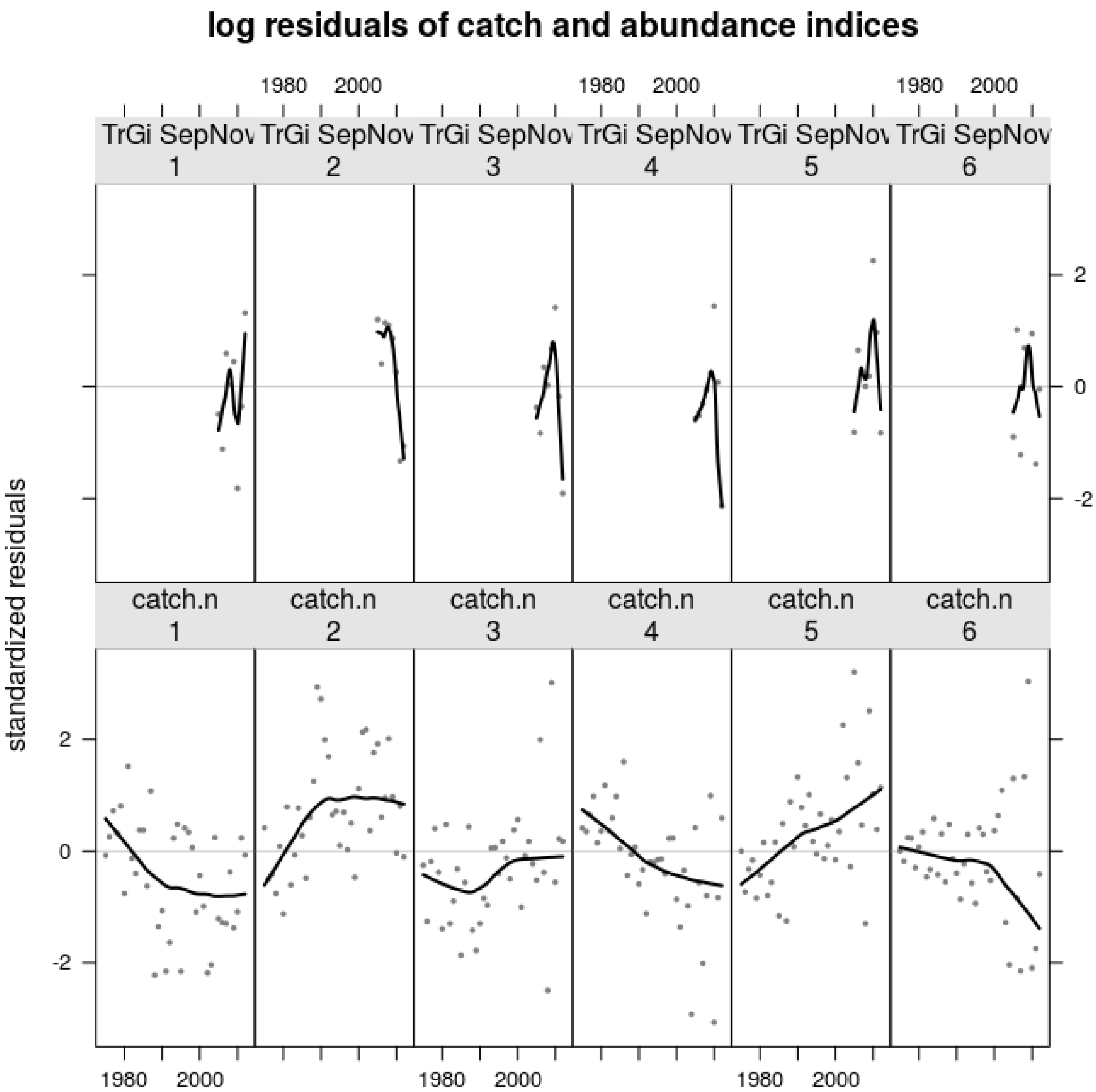


Figure 3.3: Residuals of fit to index of abundance and catch series for model run R1

R2

We now explore the influence of the assumed stock-recruitment relationship in the ability of the stock to withstand the observed levels of catch, by attempting to fit a Beverton & Holt stock recruitment relationship with a moderate level of variability (CV=0.3).

```
r2 <- sca(sar, tun, srmodel = ~bevholt(CV = 0.3), fit = "assessment")
```

Only the estimates over the last few years appear to change (Figure 3.4), while model fit does not improve (AIC=553.942, vs. 517.424 for R1, Figure 3.5).

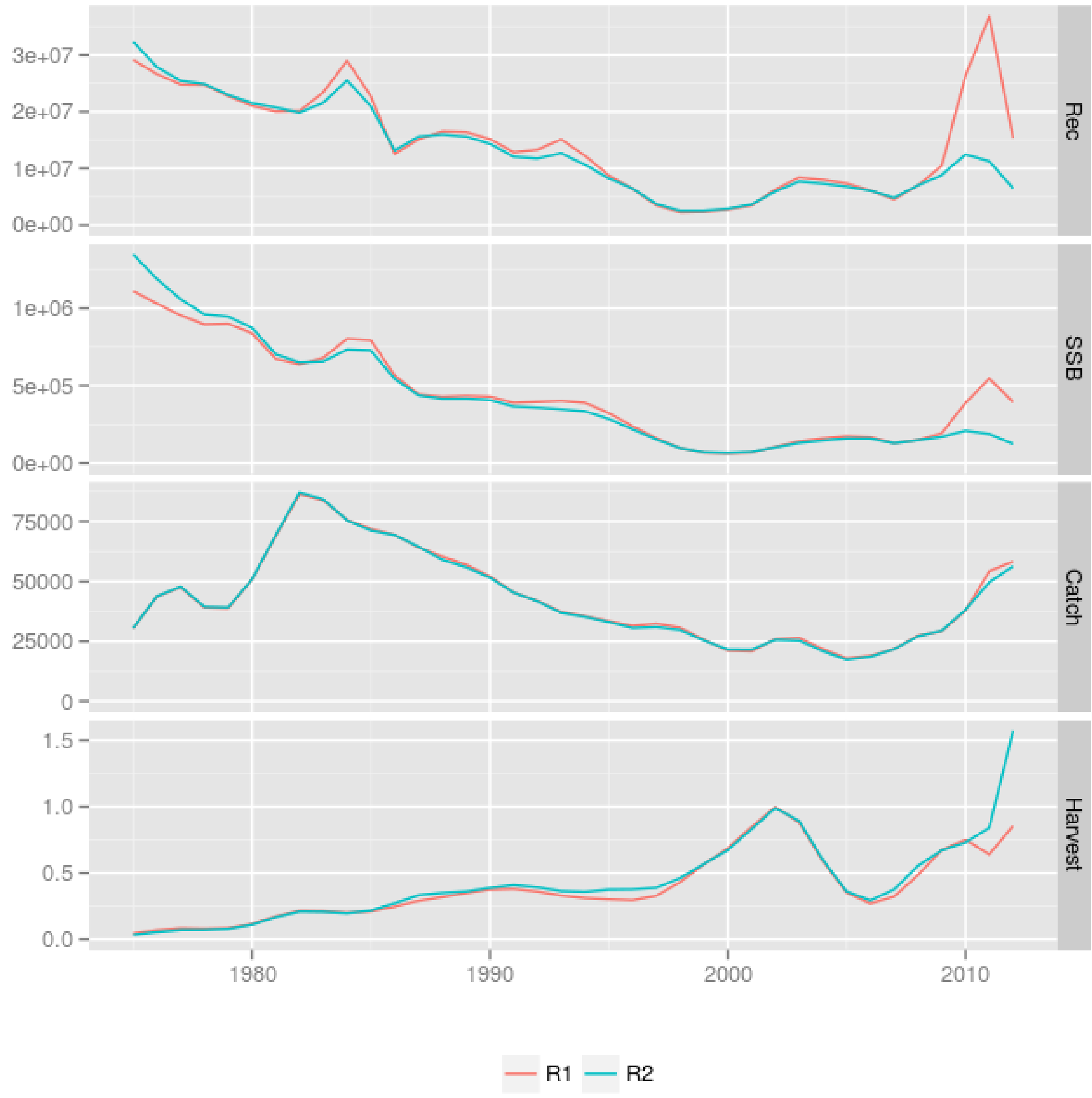


Figure 3.4: Estimated recruitment, SSB, catch and fishing mortality for Adriatic sardine, using model runs R1 and R2

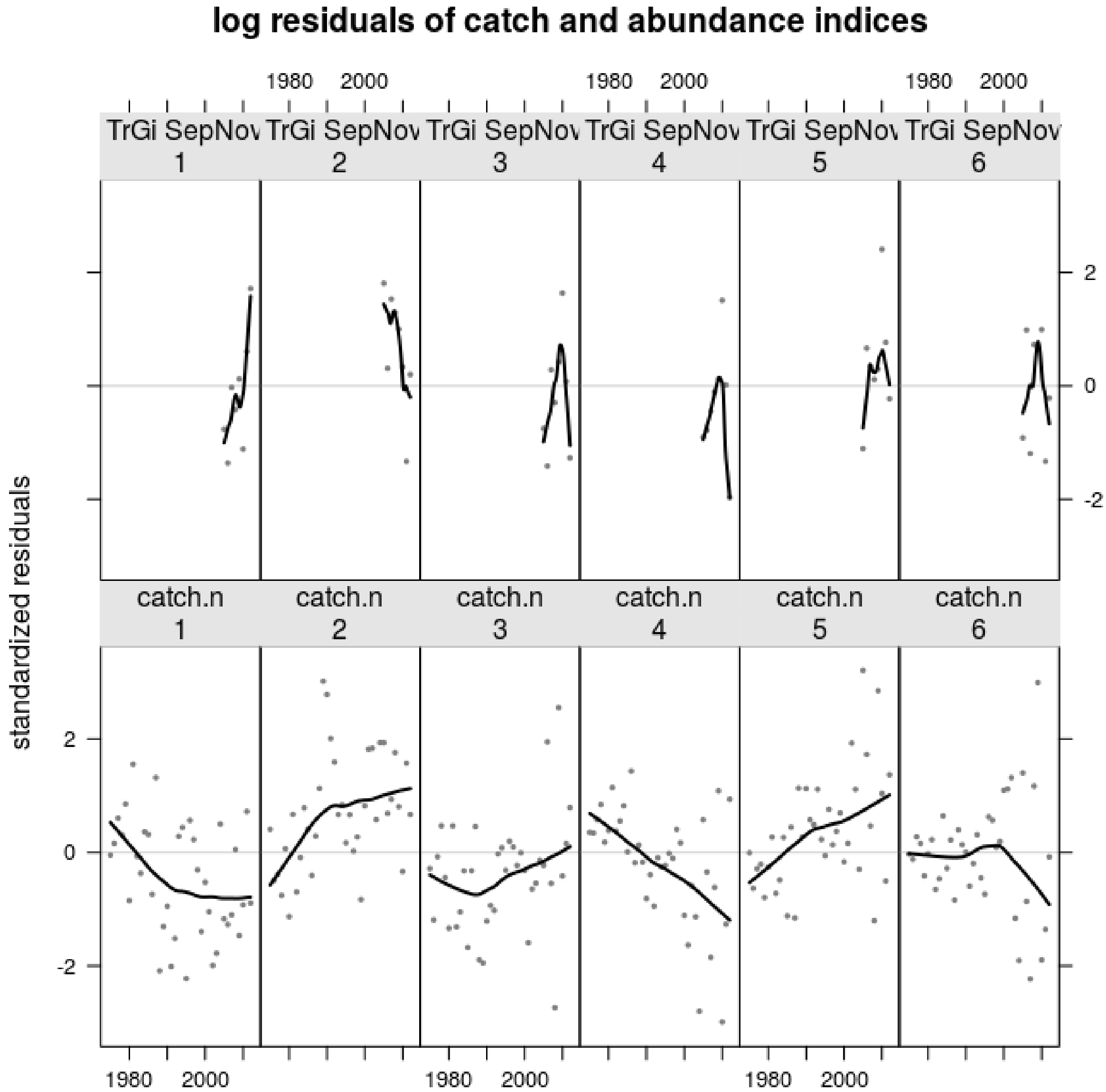


Figure 3.5: Residuals of fit to index of abundance and catch series for model run R2

R3

Given the history of changes in catch, we will now explore different options for the fmodel, away from the default in `sca()`, of a tensor spline. By using `factor(age) + factor(year)`, we give more freedom for the fishing mortality to vary across years and ages, thus reflecting possible changes in targeting and selectivity.

```
srmod <- ~factor(year)
fmod <- ~factor(age) + factor(year)
r3 <- sca(sar, tun, srmodel = srmod, fmodel = fmod, fit = "assessment")
```

But as these changes are mostly informed by the catch data, the model is forced to consider much higher values of biomass and recruitment in the historical period.

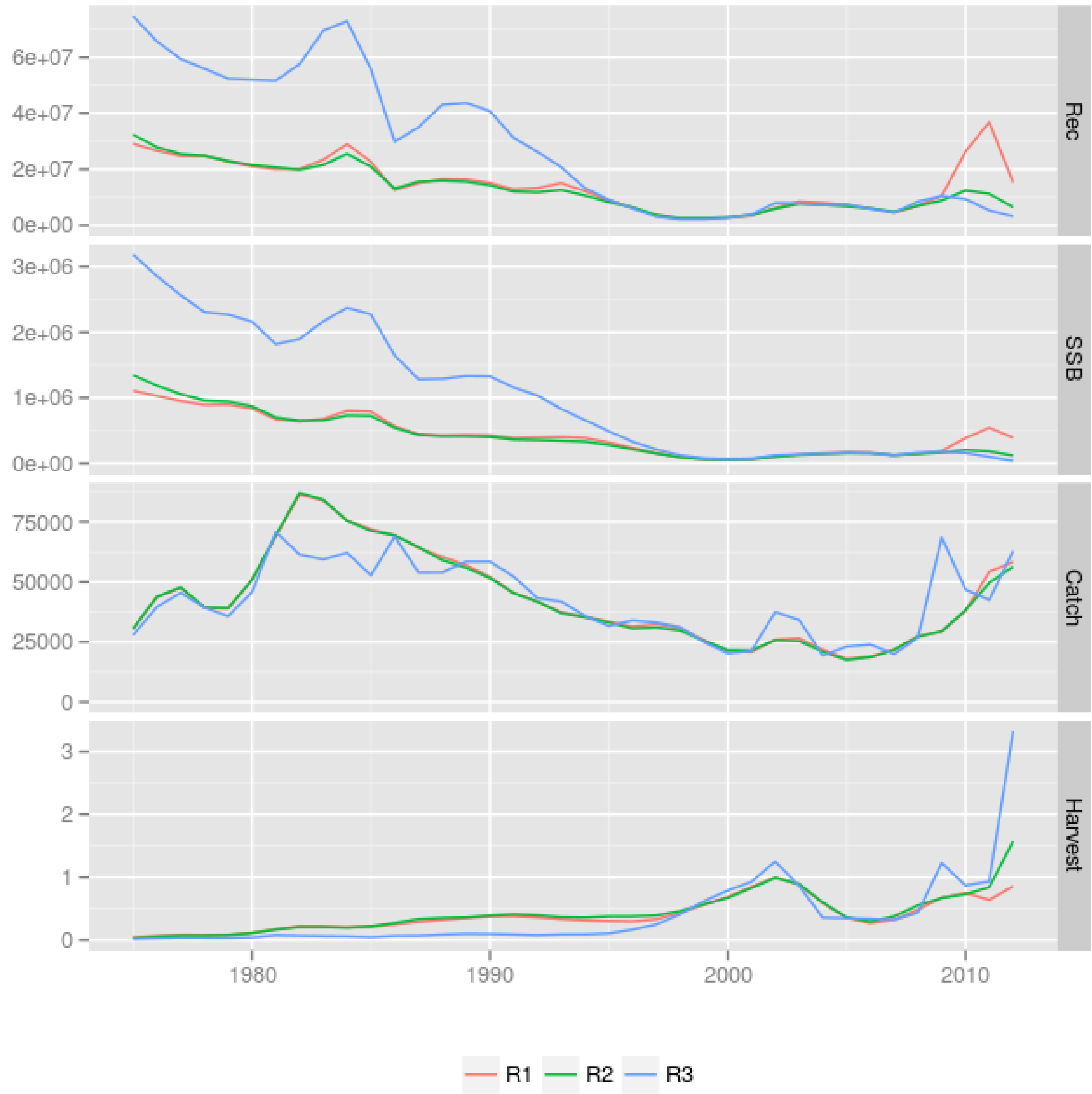


Figure 3.6: Estimated recruitment, SSB, catch and fishing mortality for Adriatic sardine, using model runs R1, R2 and R3

Fit appears to improve, according the the AIC value (AIC=482.694), but the increase in estimated biomass is a reason for concern, as it is the difficulty at fitting the catch series.

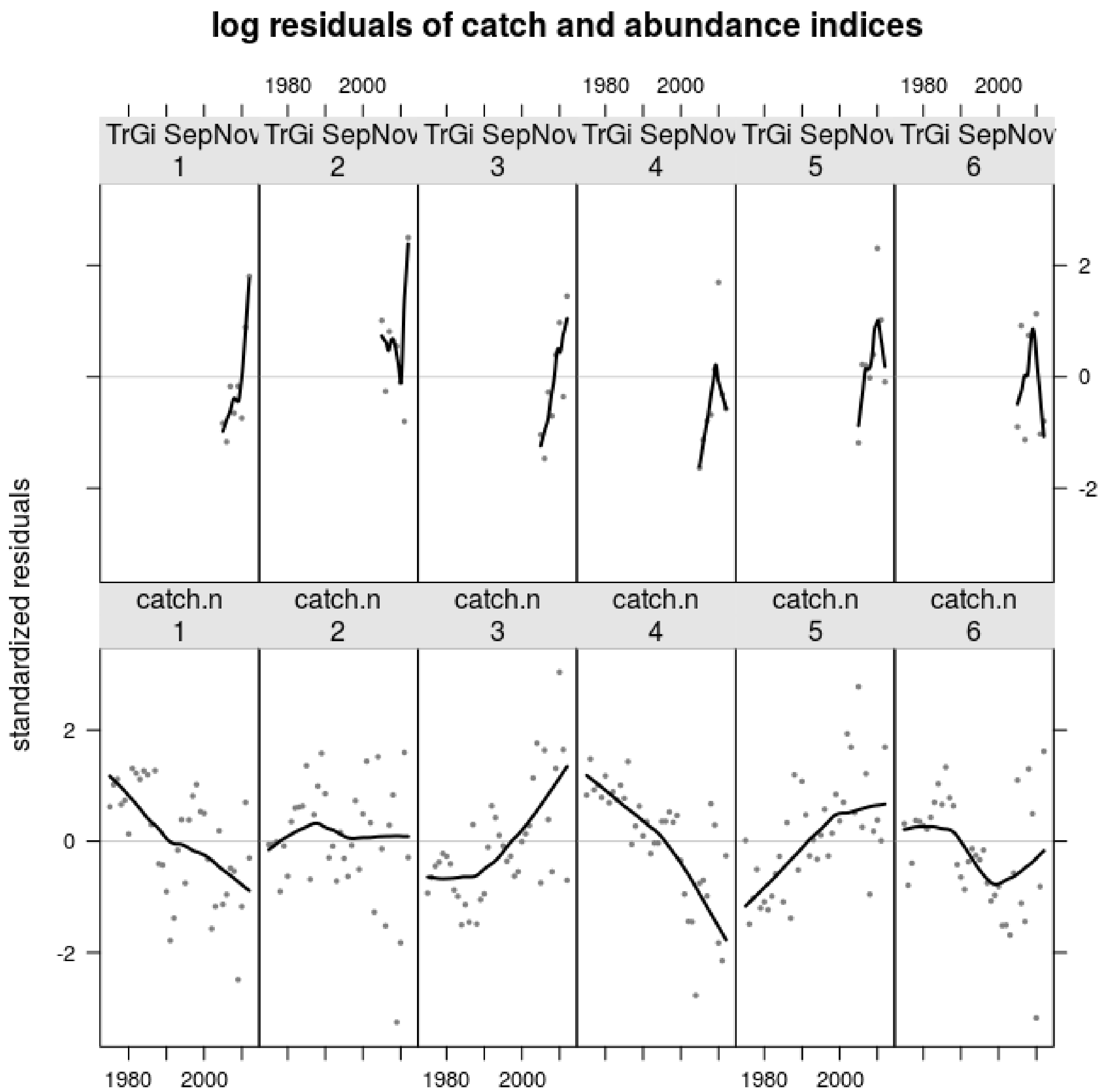


Figure 3.7: Residuals of fit to index of abundance and catch series for model run R3

R4

We now try to capture some possible trends in recruitment by using a spline for the stock-recruitment model, as follows

```
srmod <- ~s(year, k = 20)
fmod <- ~factor(age) + factor(year)
r4 <- sca(sar, tun, srmodel = srmod, fmodel = fmod, fit = "assessment")
```

whoch gives a git that closely matches R2.

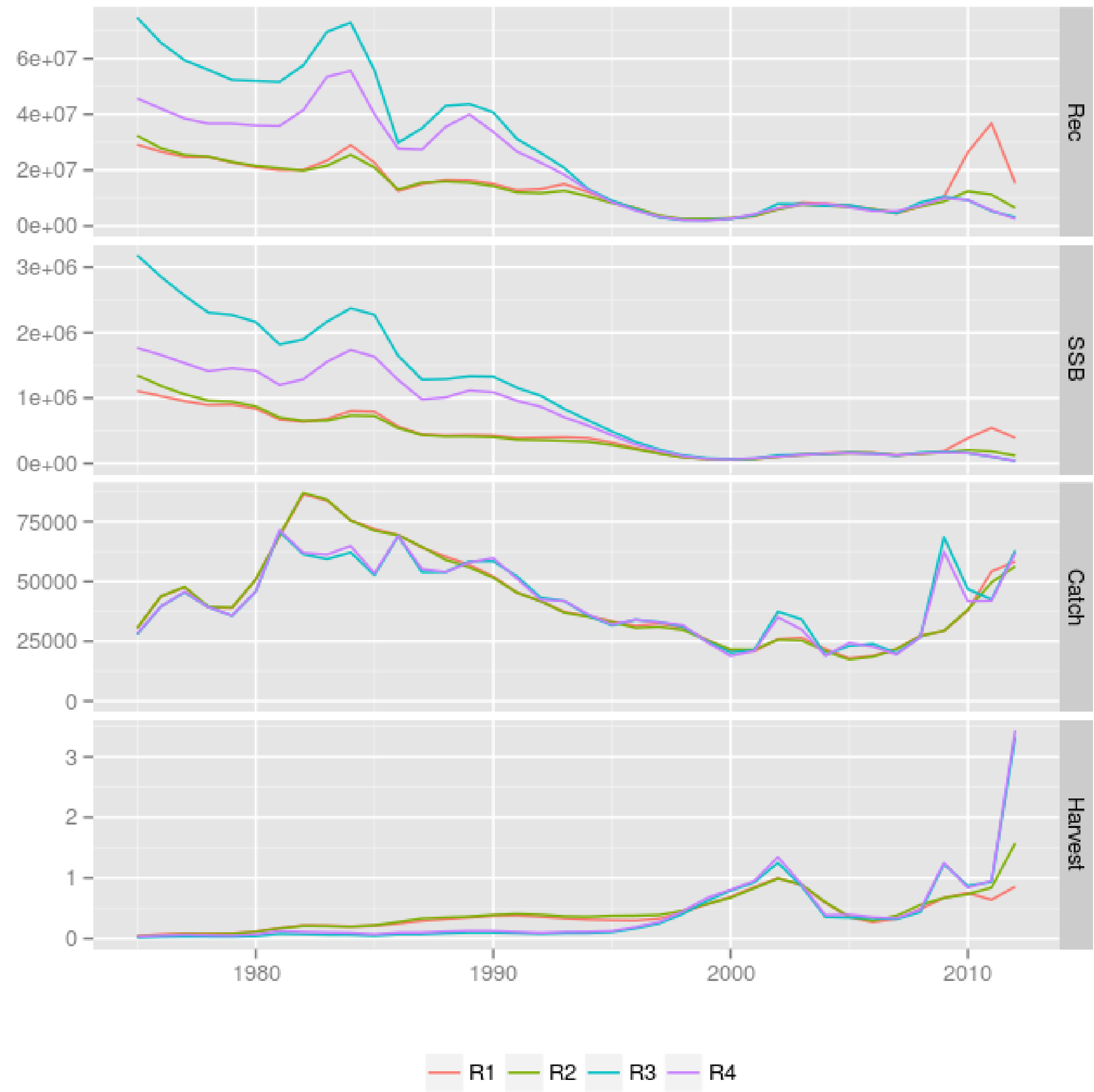


Figure 3.8: Estimated recruitment, SSB, catch and fishing mortality for Adriatic sardine, using model runs R1, R2, R3 and R4

R5

The catch series shows a series of sudden changes, for example a large a increase in the 1980-85 period, which could be related to change in targeting, as the main fleets exploiting this stock do so in combination with anchovy, which tends to be regarded as their preferred target. We introduce a series of breakpoints in the fishing mortality model, by allowing a different set of splines to be applied to that period.

```
srmod <- ~factor(year)
fmod <- ~s(age, k = 3, by = breakpts(year, c(1980, 1985))) +
    s(year, k = 20, by = breakpts(age, c(1.5:4.5)))
r5 <- sca(sar, tun, srmodel = srmod, fmodel = fmod, fit = "assessment")
```

This appears to help explaining that catch increase, for which no cohort effect was observable, without resorting to ever larger estimates of SSB.

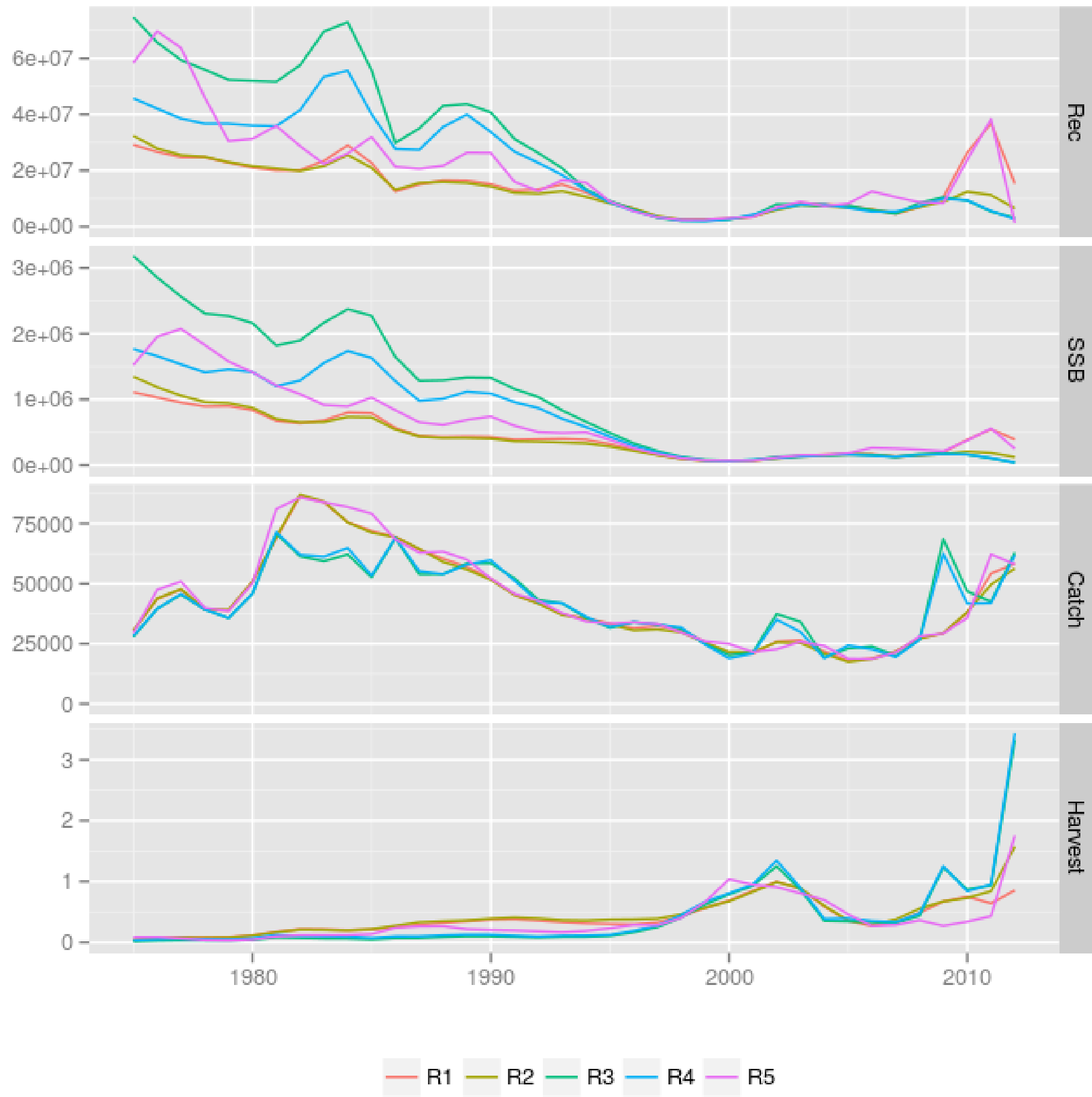


Figure 3.9: Estimated recruitment, SSB, catch and fishing mortality for Adriatic sardine, using model runs R1, R2, R3 and R4

R6

Finally, we pay attention to both the variance and catchability models by, in the first case, employing a simple spline for both catch and index, while for the second a breakpoint is introduced in 2010 to try to understand the sudden increase in relative abundances reported for ages 3 to 6 on that year, which could be an indication of changes in the survey not fully accounted for.

```
srmod <- ~factor(year)
fmod <- ~s(age, k = 3, by = breakpts(year, c(1980, 1985))) +
    s(year, k = 20, by = breakpts(age, c(1.5:4.5)))
vmod <- list(~s(age, k = 3), ~s(age, k = 3))
qmod <- list(~s(age, k = 3, by = breakpts(year, c(2010))))
r6 <- a4aSCA(sar, tun, srmodel = srmod, fmodel = fmod, vmodel = vmod,
    qmodel = qmod)
```

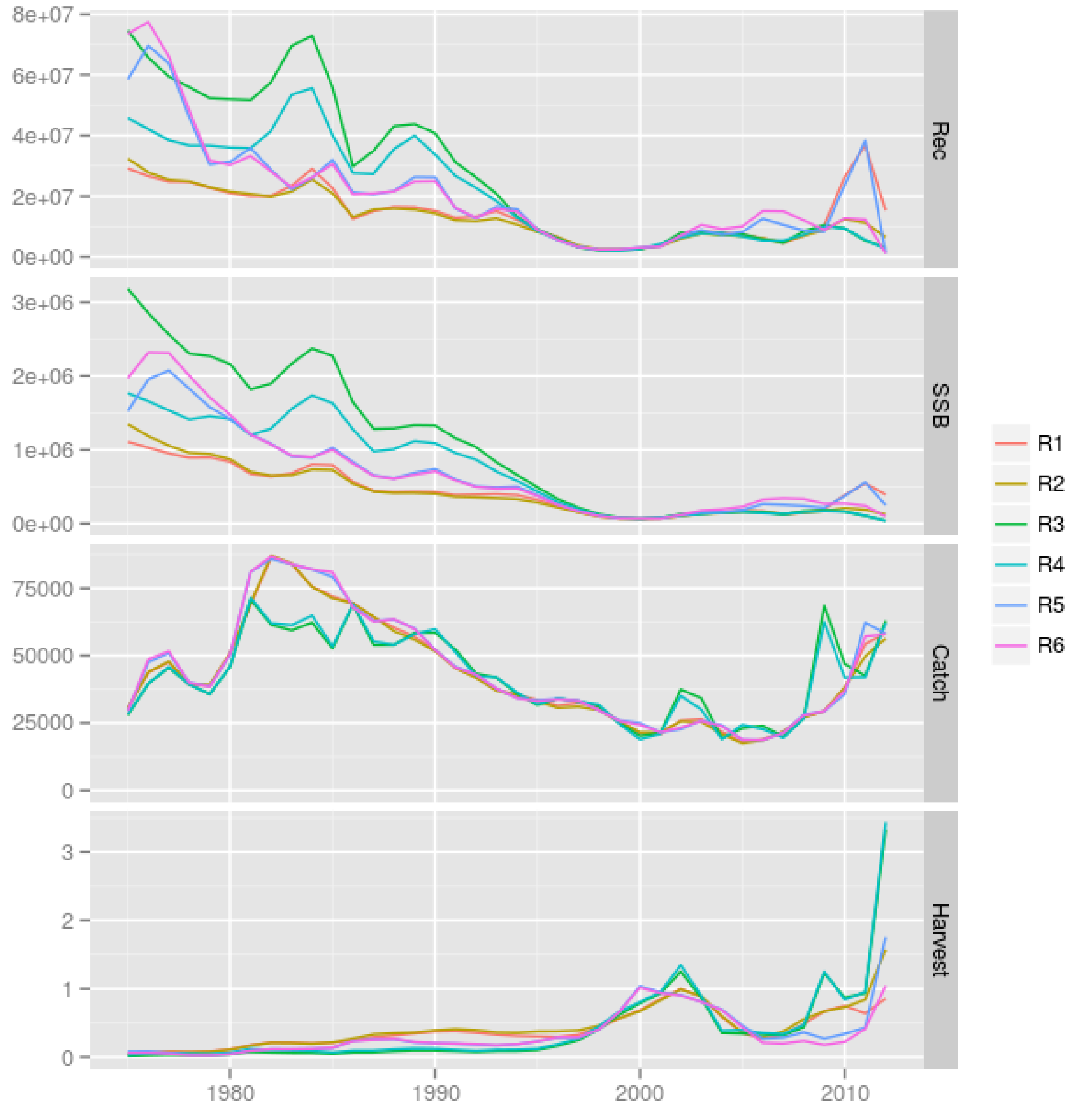


Figure 3.10:

These data points are having a large impact in the recent estimates of fishing mortality (Figure 3.11), that although are matched by recent increases in catch, appear to be out of scale given the history of the stock.

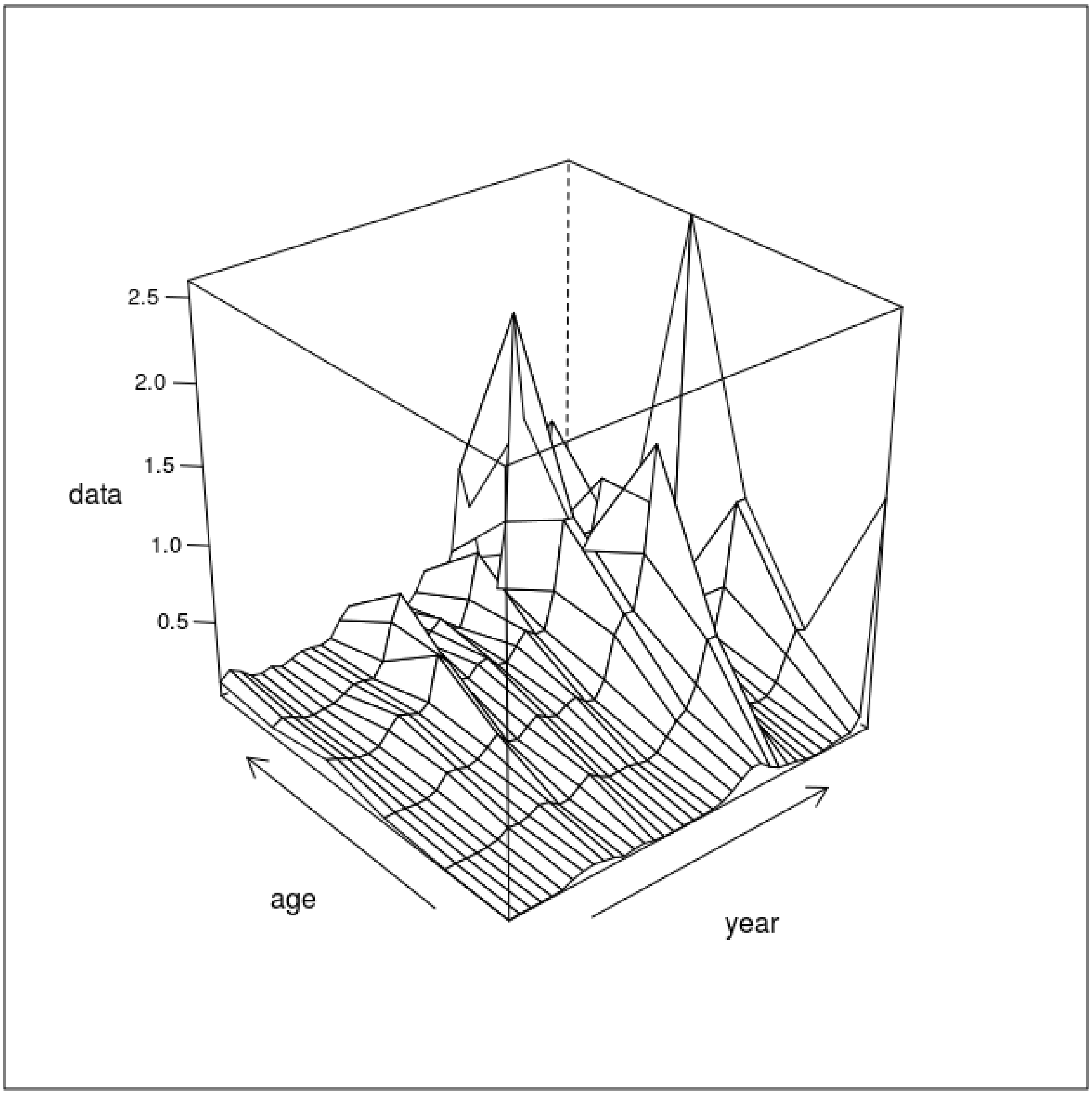


Figure 3.11:

3.2 Comparison of all a4a models with SAM results

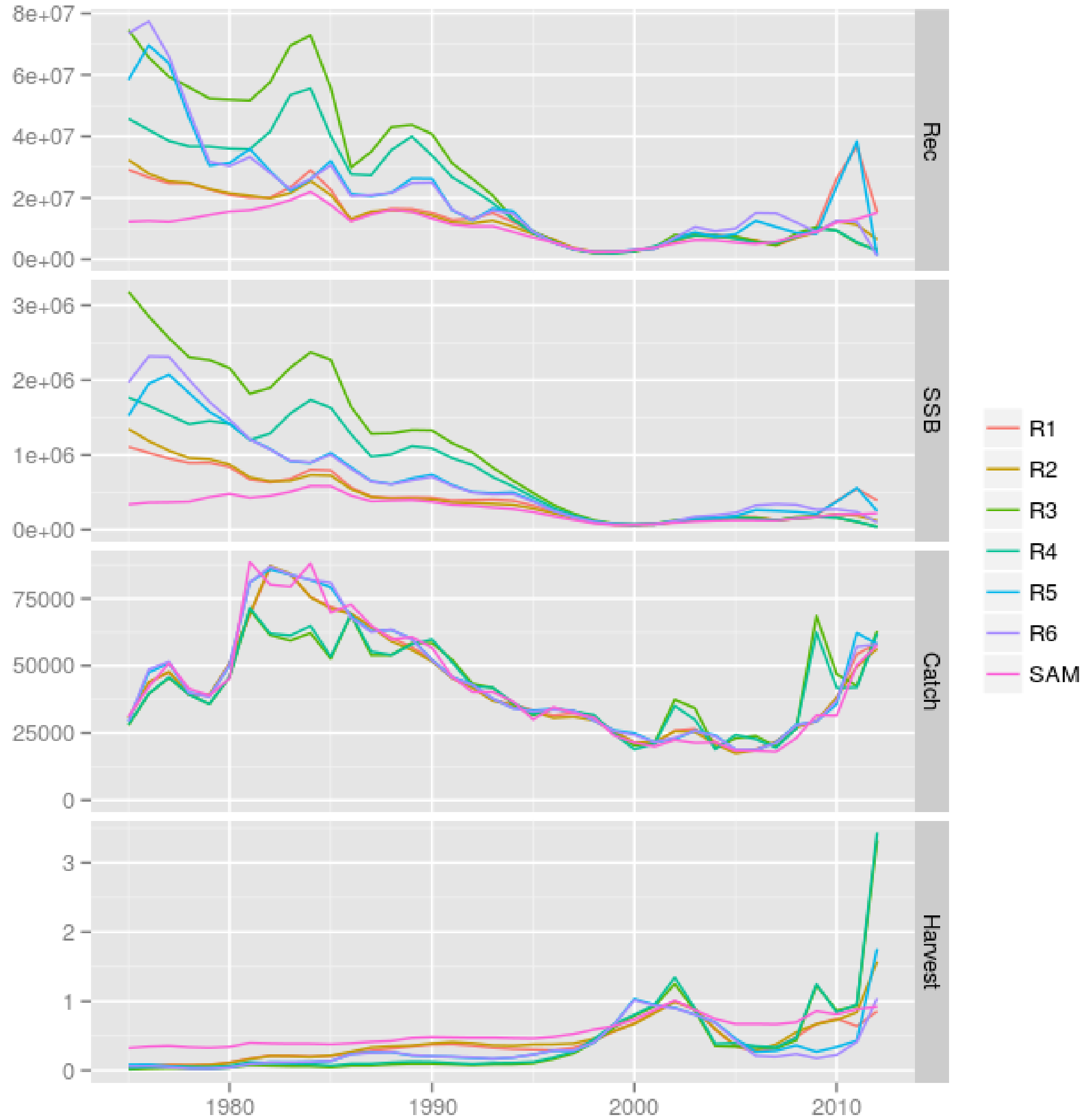


Figure 3.12:

Of the six models presented above (R1 to R6), the last one appears to fit the existing data better, as determined by the AIC value (Table **??**). The comparison of all these runs with the accepted assessment, carried out using SAM, highlights how the lack of abundance indices in the earlier part of the series is dealt with by each model. SAM appears to be able to explain existing catches, and the relatively high natural mortality assumed, with a level of biomass lower than any a4a model run (Figure 3.12) and with correspondingly low recruitments.

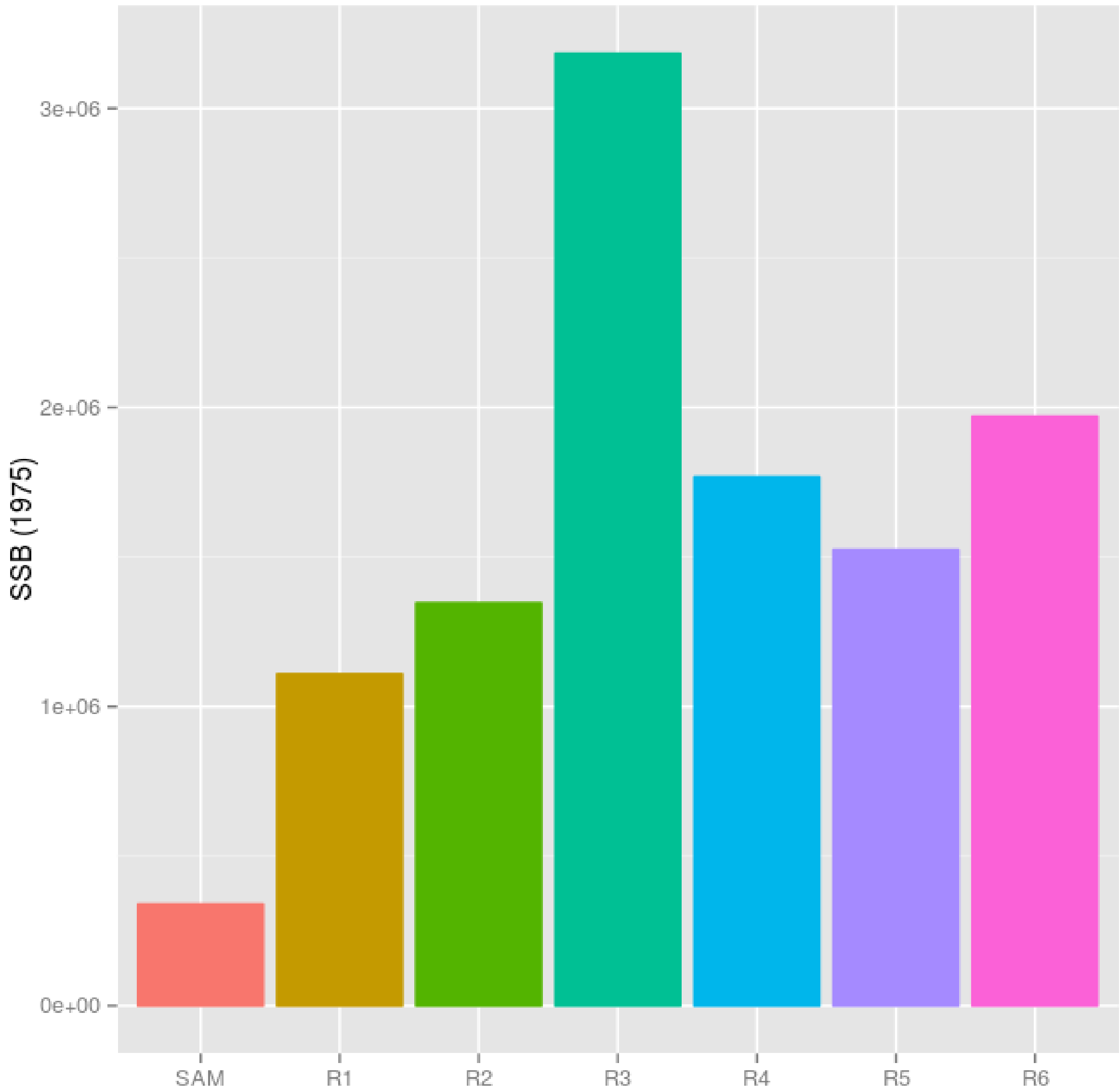


Figure 3.13: Estimateds SSB at start of series (1975) by each model and method.

	df	AIC
r1	108.00	517.42
r2	110.00	553.94
r3	94.00	482.69
r4	76.00	473.51
r5	153.00	438.58
r6	156.00	371.38

Table 3.1: AIC and degrees of freedom (DF) values for the six model runs presented.

This feature will need to be further explored, as this is likely to have a strong impact on the perceived ability of the stock to withstand fishing pressure and of the reference points estimated for it. The recent trends in the main index of abundance also require investigation, as although there is no indication of a strong cohort appearing in the younger ages, and only a sudden increase in reported abundance for four ages.

3.3 Incorporating longer indices of abundance

An attempt was made at filling up the gap in fishery-independent information before 2005 by including in the analysis the relative biomass indices generated from acoustic surveys (see above Section XX).

```
ane_pil <- read.csv("data/ANE_PIL_acoustic.csv", sep = ";")
pil <- ane_pil[ane_pil$specie == "PIL", ]
pilnw <- pil[pil$survey == "nwacoustic_survey", ]
pilcw <- pil[pil$survey == "midadr_acoustic", ]

pilnw_ind <- pilnw$tons_nm
pilcw_ind <- pilcw$tons_nm

dnms <- list(age = "all", year = min(pilnw$year):max(pilnw$year))
nwidx <- FLIndexBiomass(index = FLQuant(pilnw_ind, dimnames = dnms))

dnms2 <- list(age = "all", year = min(pilcw$year):max(pilcw$year))
cwidx <- FLIndexBiomass(index = FLQuant(pilcw_ind, dimnames = dnms2))

range(nwidx)[c("startf", "endf")] <- c(0, 0)
range(cwidx)[c("startf", "endf")] <- c(0, 0)

sar.tun <- FLIndices(northwest = nwidx, centralwest = cwidx,
    west_east = tun[[1]])
```

The model run using thee indices (R7), is similarly specified to the previous one (R6), but with individual catchability models for each survey (see `qmod` below).

```
fmod <- ~s(age, k = 5, by = breakpts(year, c(1980, 1985))) +
    s(year, k = 20, by = breakpts(age, c(1.5:4.5)))
qmod <- list(~s(year, k = 10), ~s(year, k = 10), ~s(age, k = 4,
    by = breakpts(year, c(2009, 2010))))
vmod <- list(~s(age, k = 3), ~1, ~1, ~s(age, k = 3))
r7 <- a4aSCA(sar, sar.tun, srmodel = srmod, fmodel = fmod, vmodel = vmod,
    qmodel = qmod, fit = "assessment")
```

Model results are not greatly dissimilar from those in R6, but the sudden jump in fishing mortality that other models needed to explain the most recent survey trends are somehow mitigated but the extra information.

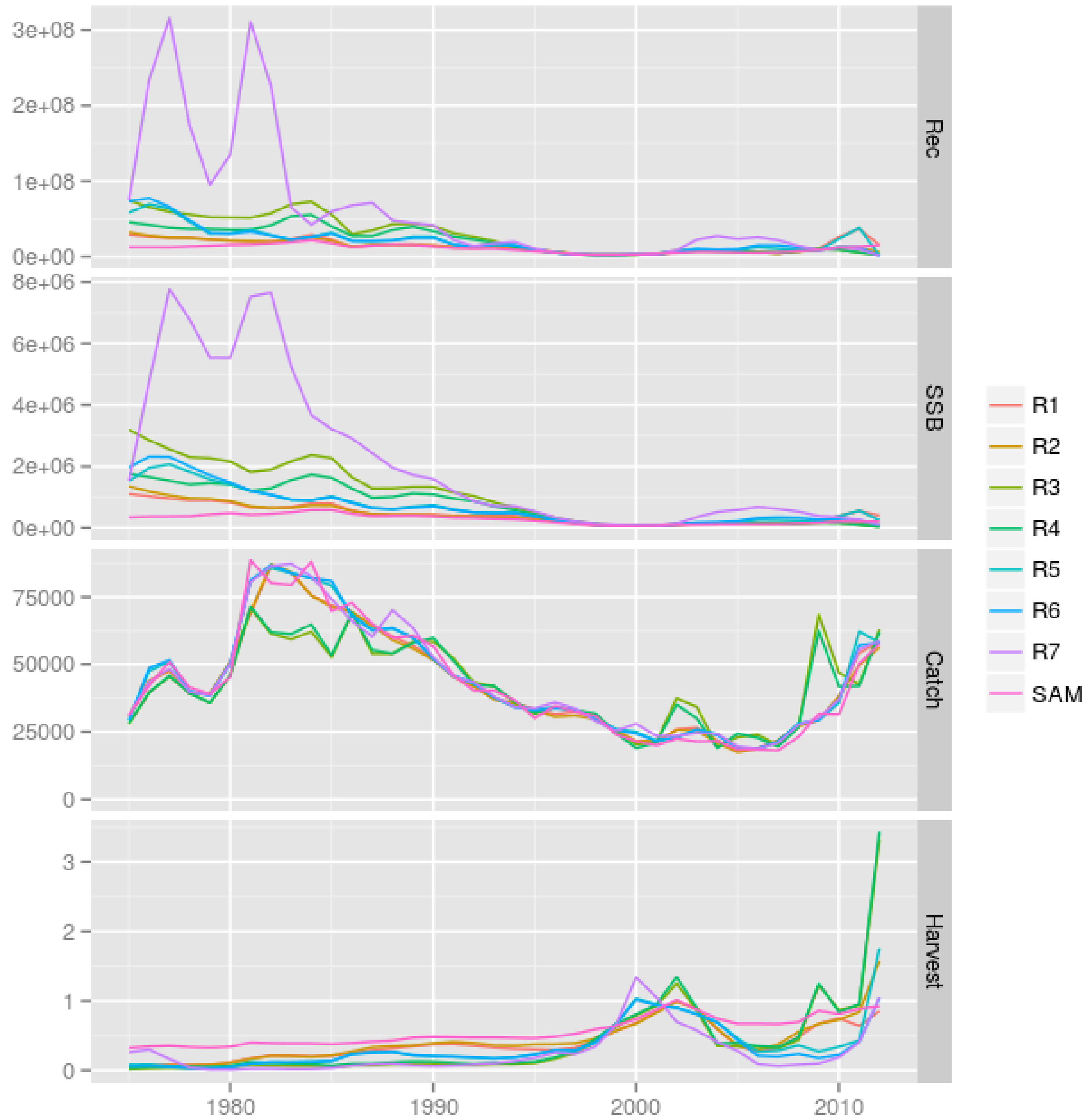

3e+08
2e+08
1e+08
0e+00
Rec
8e+06
6e+06
4e+06
2e+06
0e+00
SSB
75000
50000
25000
0
Catch
3
2
1
0
Harvest
1980
1990
2000
2010
R1
R2
R3
R4
R5
R6
R7
SAM

Figure 3.14:

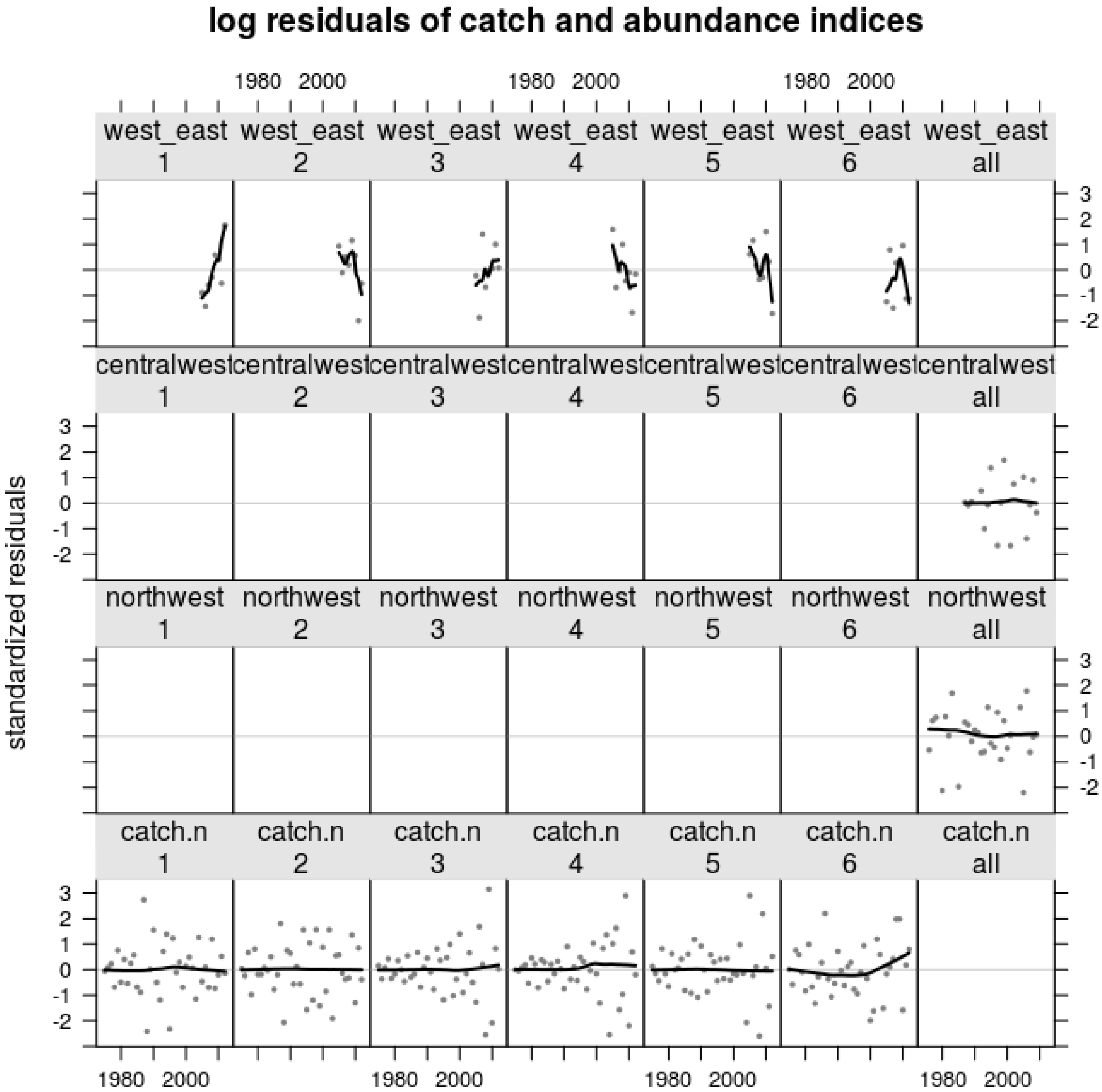


Figure 3.15: Residual plot for R7.

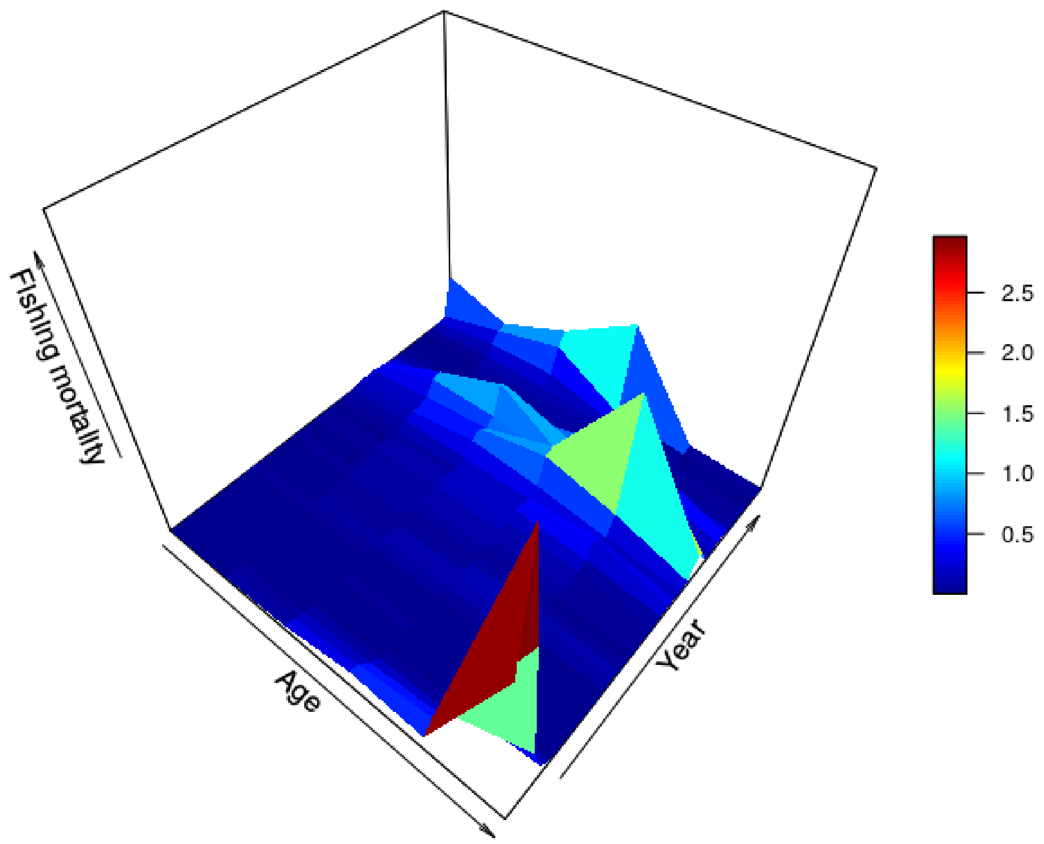


Figure 3.16: Trends in fishing mortality at age by year in model run R by year in model run R7.

4 Anchovy and Sardine interactions

Chato Osio & Iago Mosqueira

Since the anchovy stock is fished with purse seiners and pelagic trawlers also targeting sardine stocks, it is important to understand if over time there have been switches in targeting between the two stocks. For example it could be hypothesized that there was lower targeting in some periods like the early nineties. To check we import the assessment results from Sardine stock and compare trends in F, SSB and Catches.

```
fmod <- ~s(age, k = 5, by = breakpts(year, c(1980, 1985))) +
    s(year, k = 20, by = breakpts(age, c(1.5:4.5)))
qmod <- list(~s(year, k = 10), ~s(year, k = 10), ~s(age, k = 4,
    by = breakpts(year, c(2009, 2010))))
vmod <- list(~s(age, k = 3), ~1, ~1, ~s(age, k = 3))
r7 <- a4aSCA(sar, sar.tun, srmodel = srmod, fmodel = fmod, vmodel = vmod,
    qmodel = qmod, fit = "assessment")
```

```
res2 <- FLStocks(ANC = ANC17 + fit7, SAR = sar + r7)
```

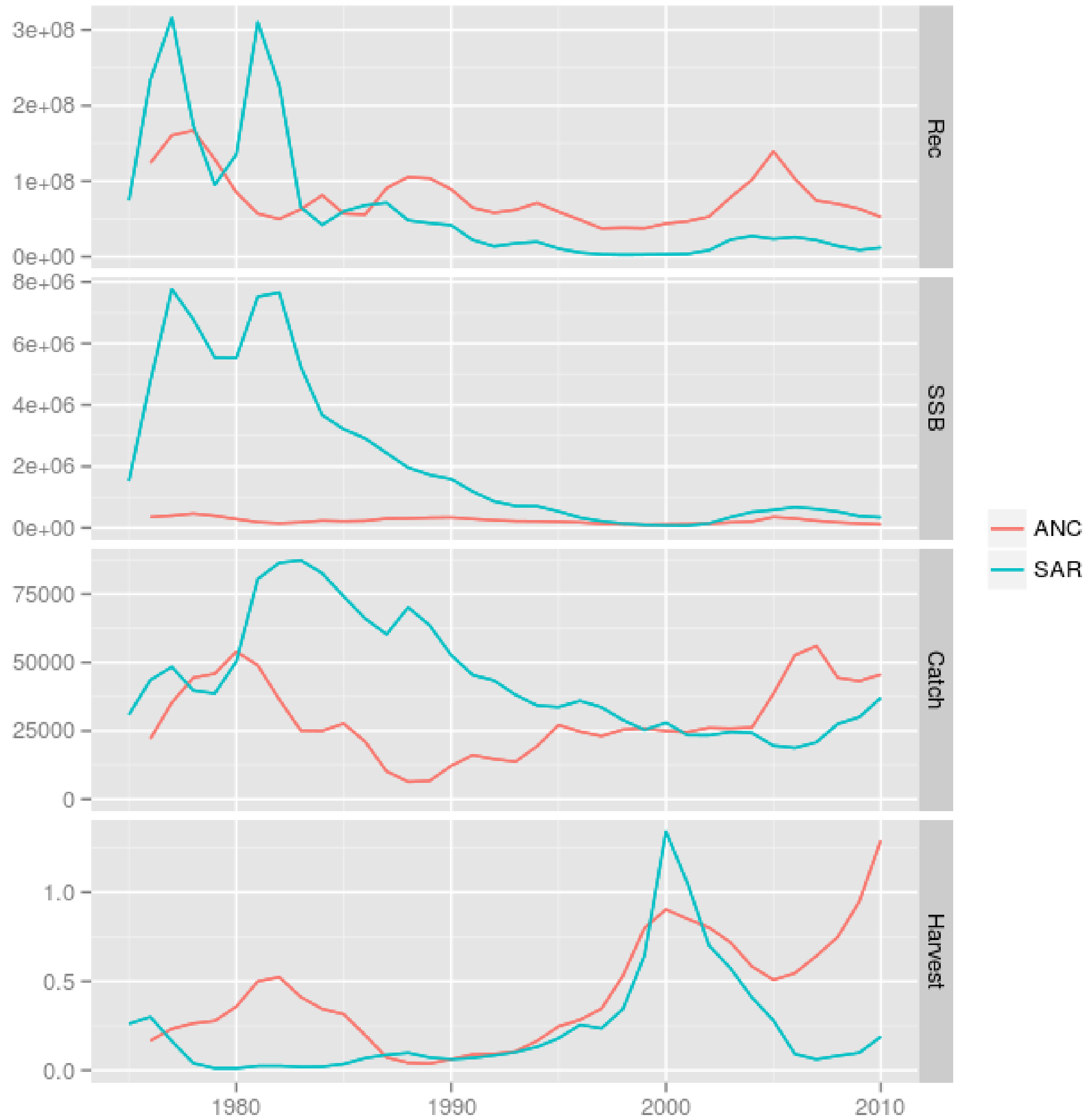


Figure 4.1: Comparative plots for Anchovy and Sardine

The plots show that there is an almost symmetrical trend for the two stocks: when catches and SSB go down for Anchovy there is switching to Sardine and vice versa. This is important to corroborate the results of fit6 and fit6ts1 that pick up an increase in SSB in the latter part of the 1980's.

5 Hake in Gulf of Lions

Tristan Rouyer

5.1 Replicating accepted assessments

We will try to replicate the XSA assessment using a4a. We start by reading the XSA assessment and visualizing the object, which can be seen in figure 5.1.

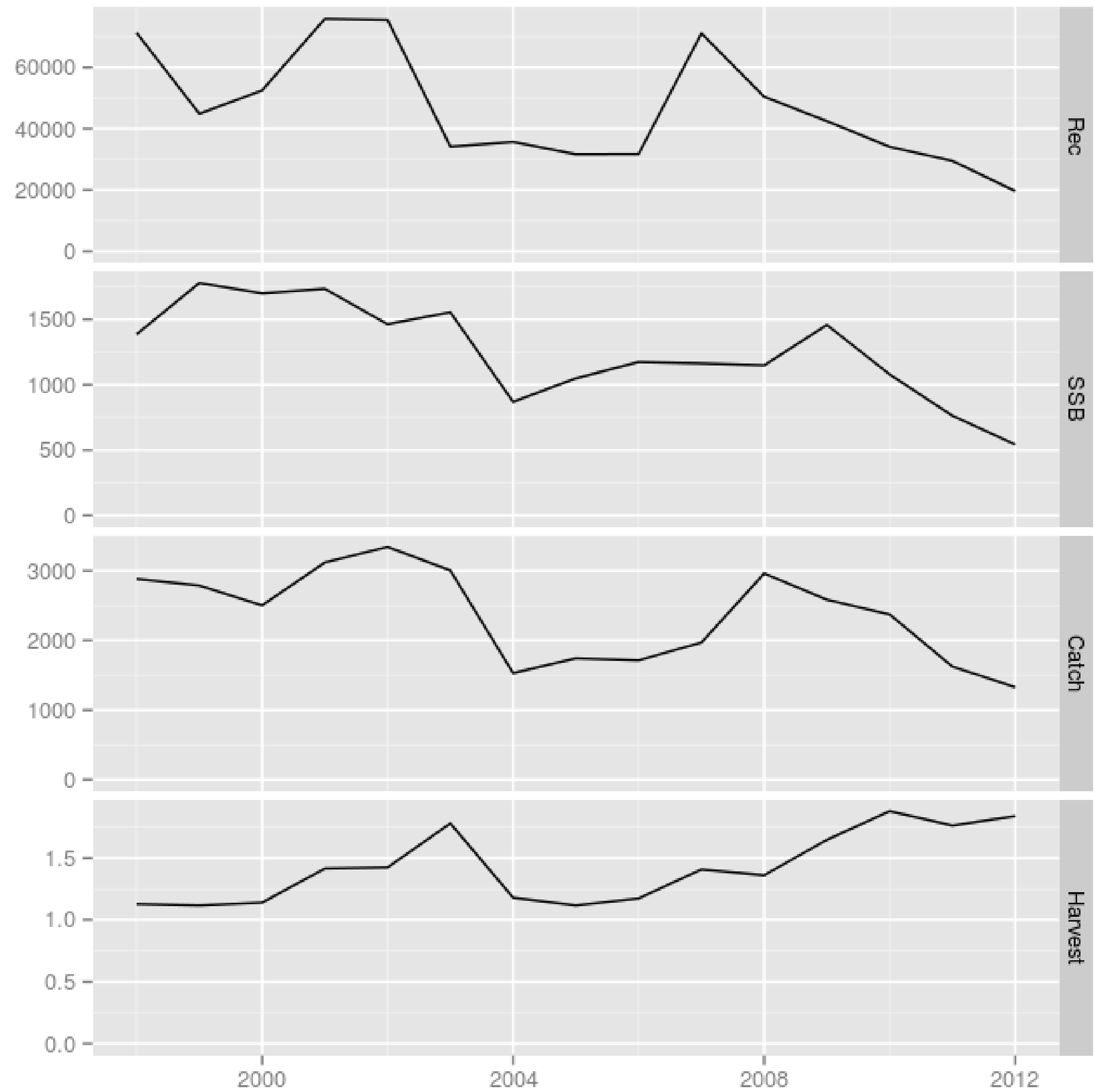


Figure 5.1: Results of the XSA assessment

We will then use a4a to try to reproduce these results. We start by reading the data and setting up a new stock object.

```
hkesca <- readFLStock("data/HakeIND.dat", no.discards = TRUE)
# Set the harvest units, fbar range and plus group age
units(harvest(hkesca)) <- "f"
range(hkesca)["minfbar"] <- 0
range(hkesca)["maxfbar"] <- 3
range(hkesca)["plusgroup"] <- 5
```

Then we read the abundance index data, which is here the MEDITS survey.

```
## Read tuning data
hke.idx <- readFLIndices("data/HakeTUN.dat")
```

The a4a model is fitted with submodels as close as possible from the XSA assumptions. As it can be seen on figure 5.2, the results are quite close from XSA in terms of trend and absolute value for the different variables. Particularly, the fishing mortality estimated by the a4a model is really close from the one obtained with XSA.

```
index <- hke.idx
qmod <- list(~factor(age))
fmod <- ~factor(replace(age, age > 5, 5)) + factor(year)
srmod <- ~factor(year)
fit <- sca(stock = hkesca, indices = index, fmodel = fmod, qmodel = qmod,
    srmodel = srmod, fit = "MP")
stk <- hkesca + fit
## we make an FLStocks for display
z <- FLStocks(a4a = stk, XSA = hke)
```

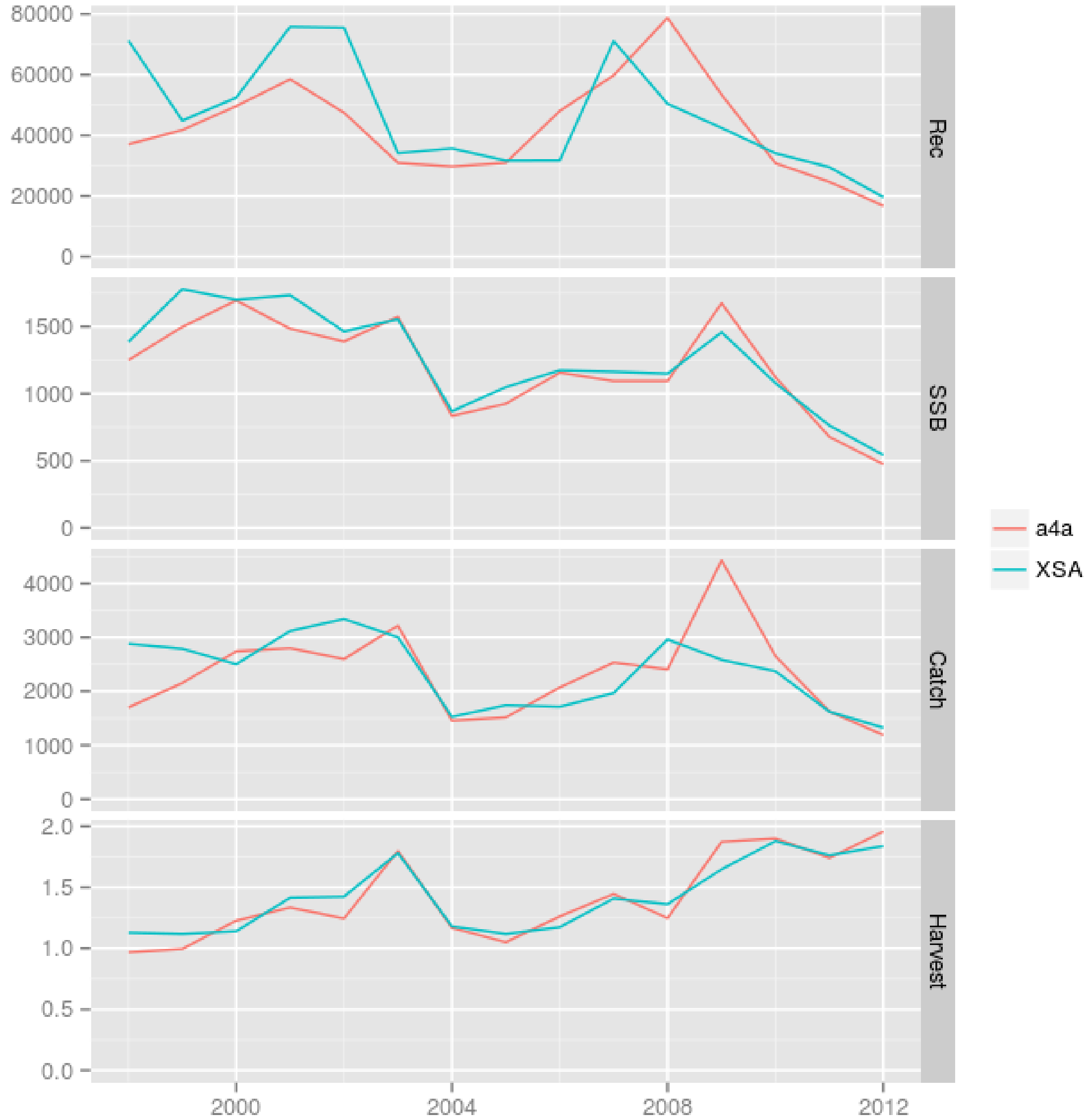


Figure 5.2: Replicating the XSA assessment

5.2 Assessments with the statistical catch-at-age method

We will now try to get an improved assessment using the statistical catch at age routine. Then we read the tuning data for different age groups:

```
## Read tuning data
hke.idx <- readFLIndices("data/HakeTUN.dat")
## Read tuning data: only ages 0 to 2
hke.idx2 <- hke.idx
hke.idx2[[1]] <- trim(hke.idx2[[1]], age = 0:2)
## Read tuning data: only ages 0 to 3
hke.idx3 <- hke.idx
```

```
hke.idx3[[1]] <- trim(hke.idx3[[1]], age = 0:3)
```

We will now do some specifications for the submodels but keep it simple to start with. However, we will only work with ages 0 to 2 as we are not very confident that the MEDITS data for ages 3 and 4 are very reliable. MEDITS is indeed not believed to be a very good sampler for bigger hake individuals.

```
index <- hke.idx2
qmod <- list(~s(age, k = 3))
fmod <- ~s(year, k = 10) + s(age, k = 3)
srmod <- ~s(year, k = 10)
fit <- sca(stock = hke, indices = index, fmodel = fmod, srmodel = srmod,
    qmodel = qmod, fit = "MP")
stk <- hke + fit
```

The trends of the time series are consistent with what is known for the stock (Figure 5.3). The residuals for the survey behave nicely (Figure 5.4) and the fishing mortality is not taking off as it was previously (Figure 5.5). However, it is also weird that the last year of fishing mortality is changing so radically. It is probably due to the effect of smoothers. The catch residuals do not look good, particularly for age 0.

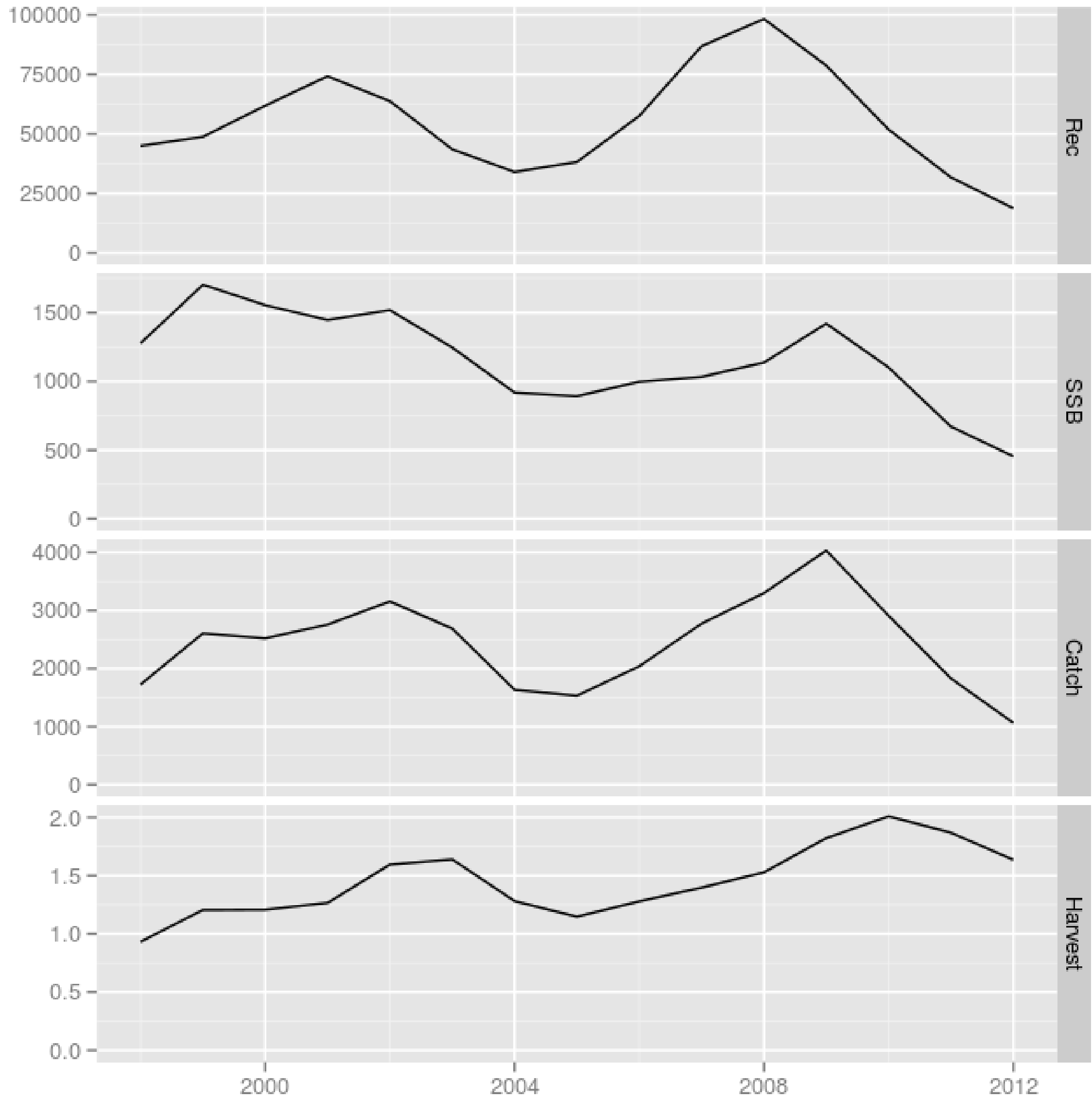


Figure 5.3: Stock plot for simple model specifications

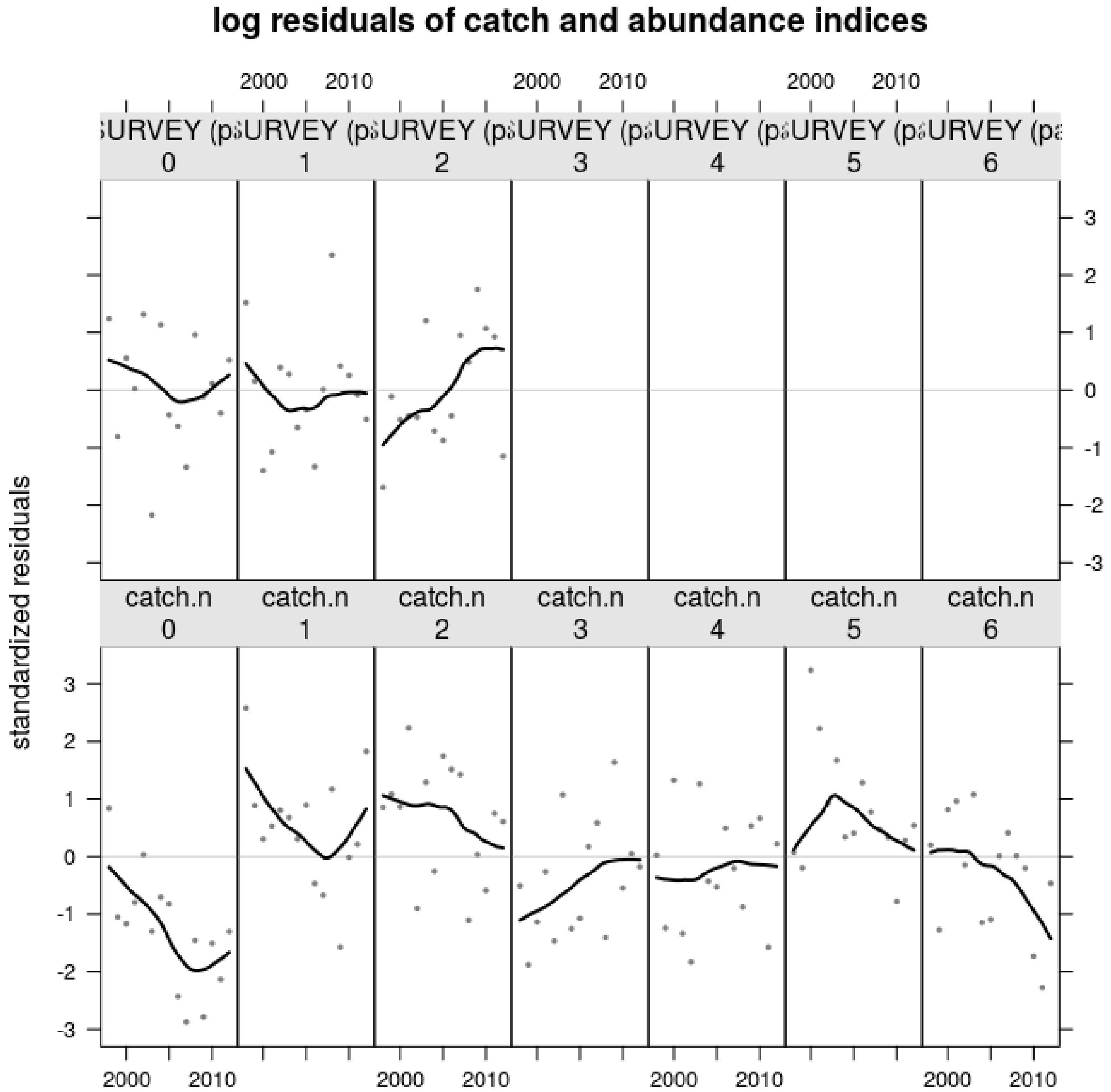


Figure 5.4: Residual plot for simple specifications

We look at the surface plot for fishing mortality:

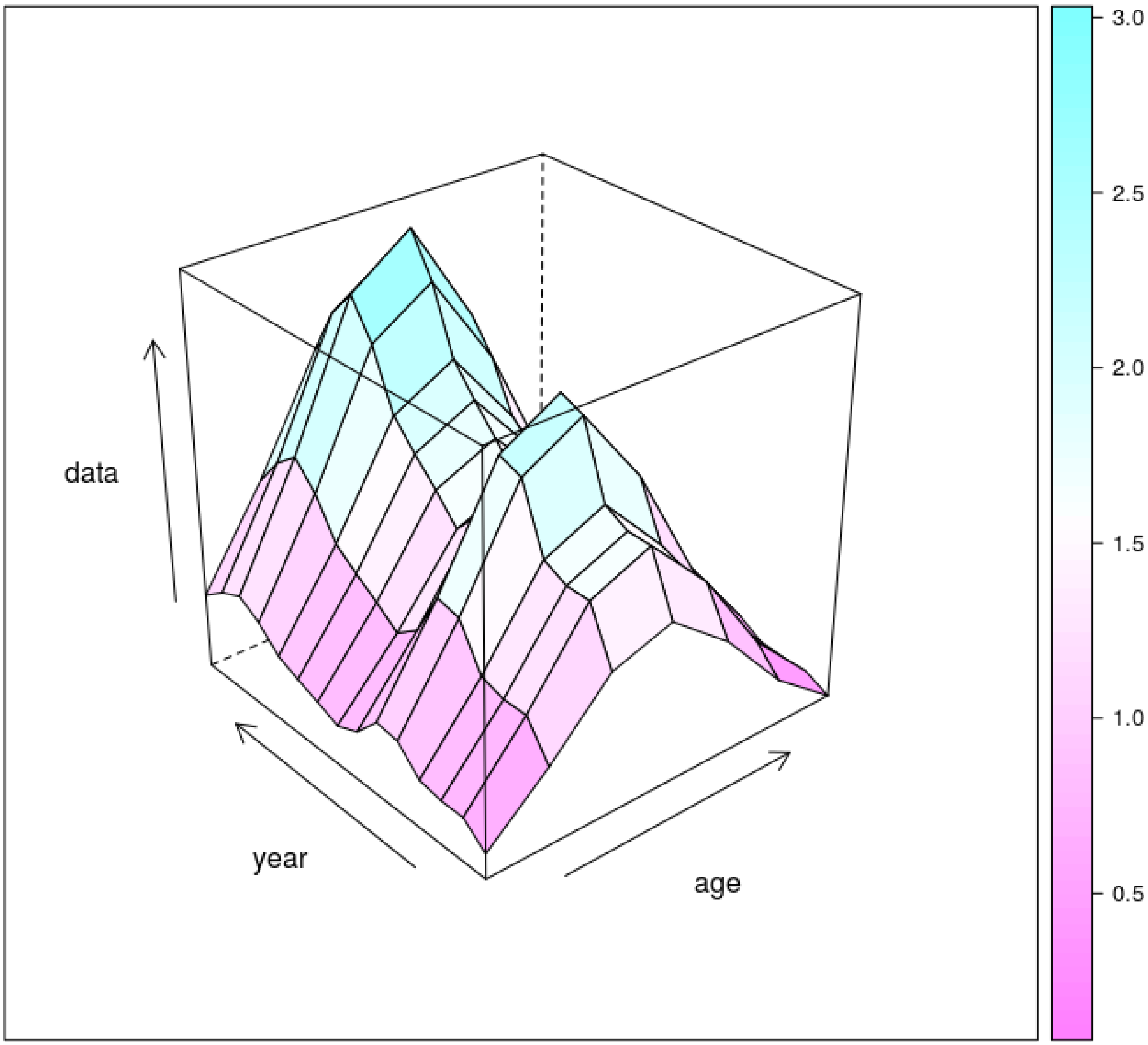


Figure 5.5: 3D plot of the fitted fishing mortality

Improving the assessment: more complex specifications

We go on with adding a bit of complexity using a logistic shape on age to constrain the fishing mortality on the last ages, so that this behaviour is more consistent with the trawler catchability.

```
index <- hke.idx2
qmod <- list(~s(age, k = 3))
fmod <- ~I(1/(1 + exp(age))) + as.factor(year) + s(age, k = 3)
srmod <- ~s(year, k = 10)
fit <- sca(stock = hke, indices = index, fmodel = fmod, srmodel = srmod,
    qmodel = qmod, fit = "MP")
stk <- hke + fit
```

As previously, the trends of the time series are in line with the previous results (Figure 5.6). The residuals for the survey behave nicely and residuals for fishing mortaly are better (Figure 5.7, Figure 5.8). However, there are still some nasty trends.We can also notice that the fishing mortality is taking off the last year (Figure 5.9).

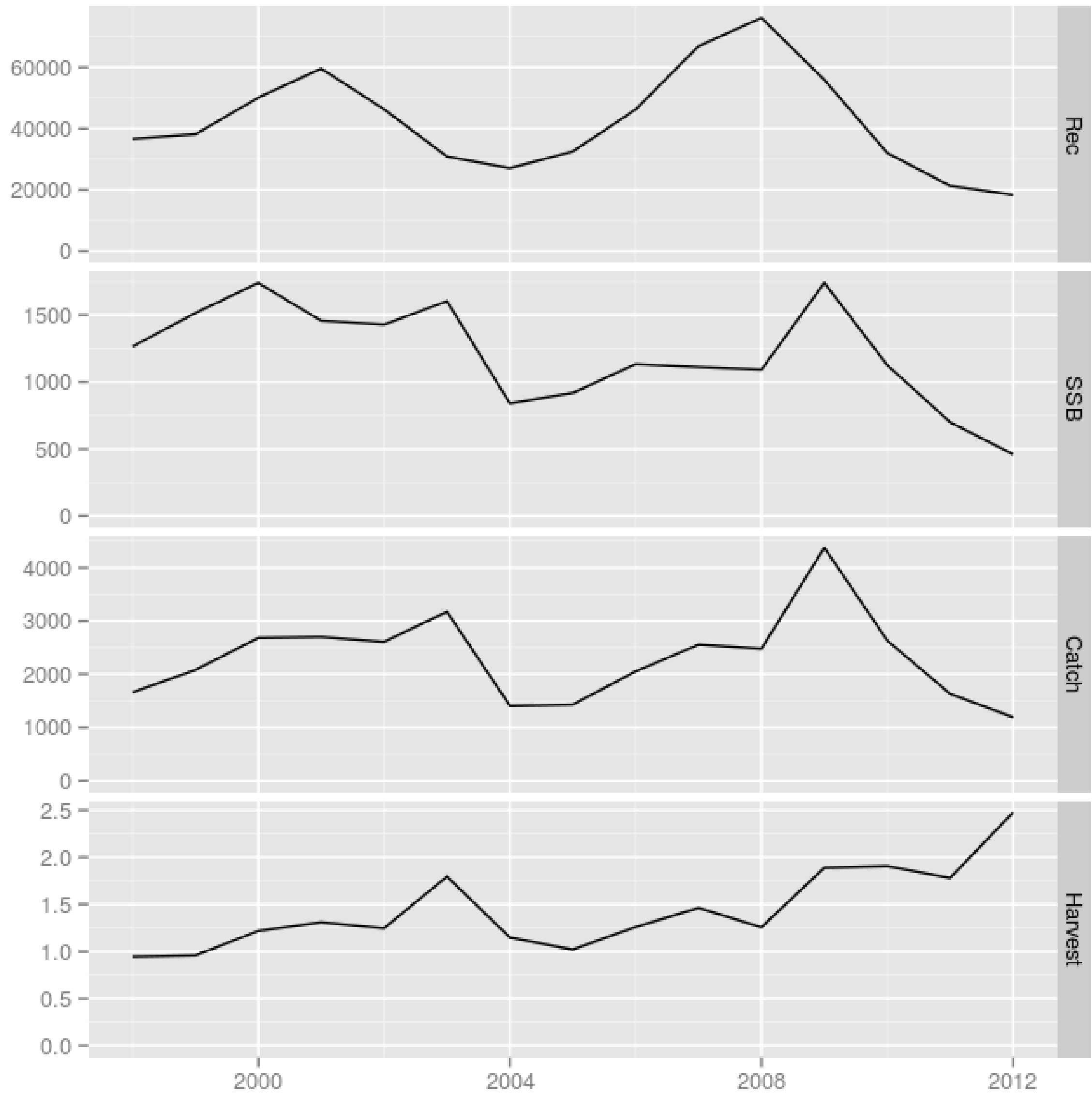


Figure 5.6: Stock plot for more complex specifications

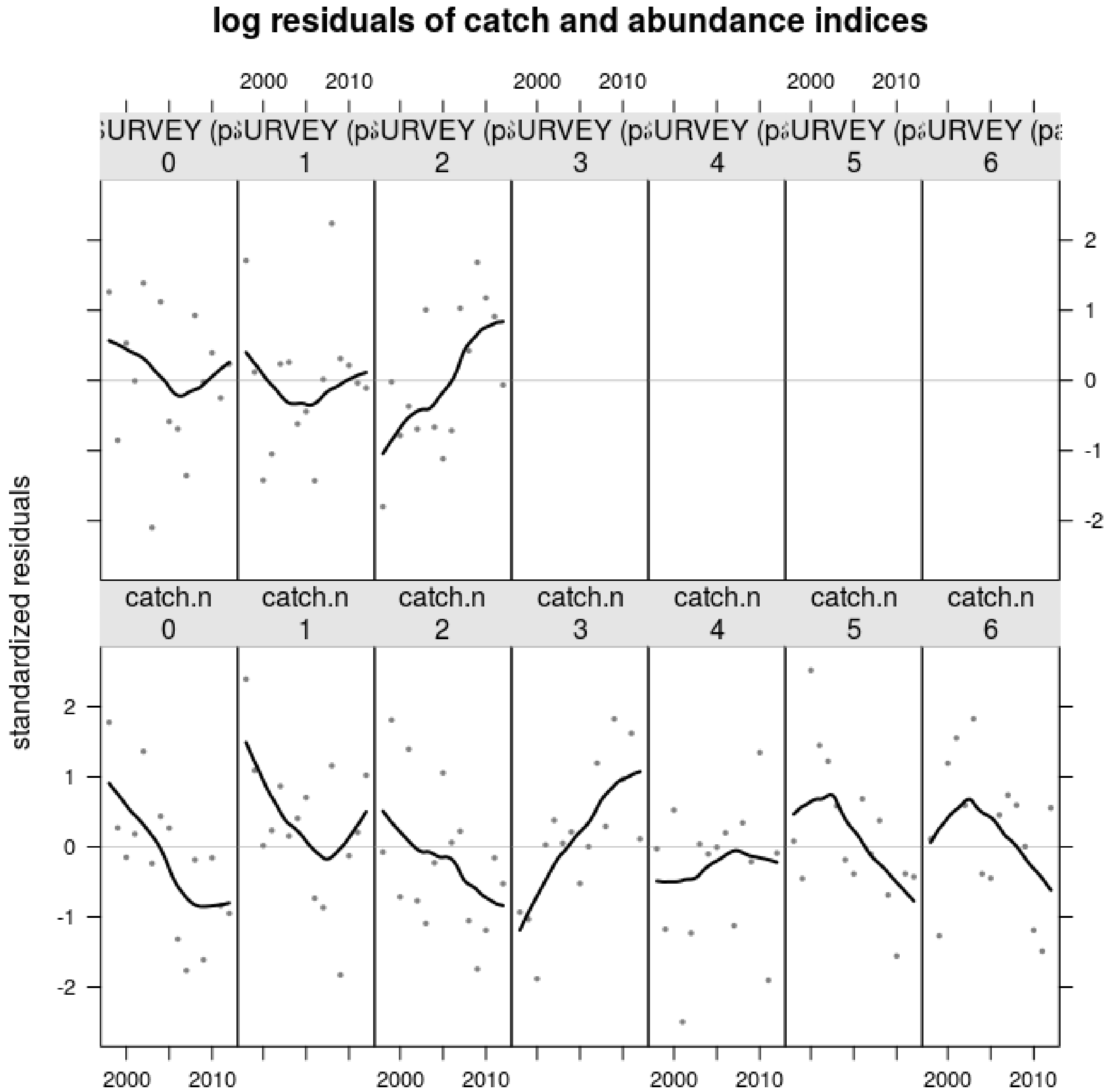


Figure 5.7: Residual plot for more complex specifications

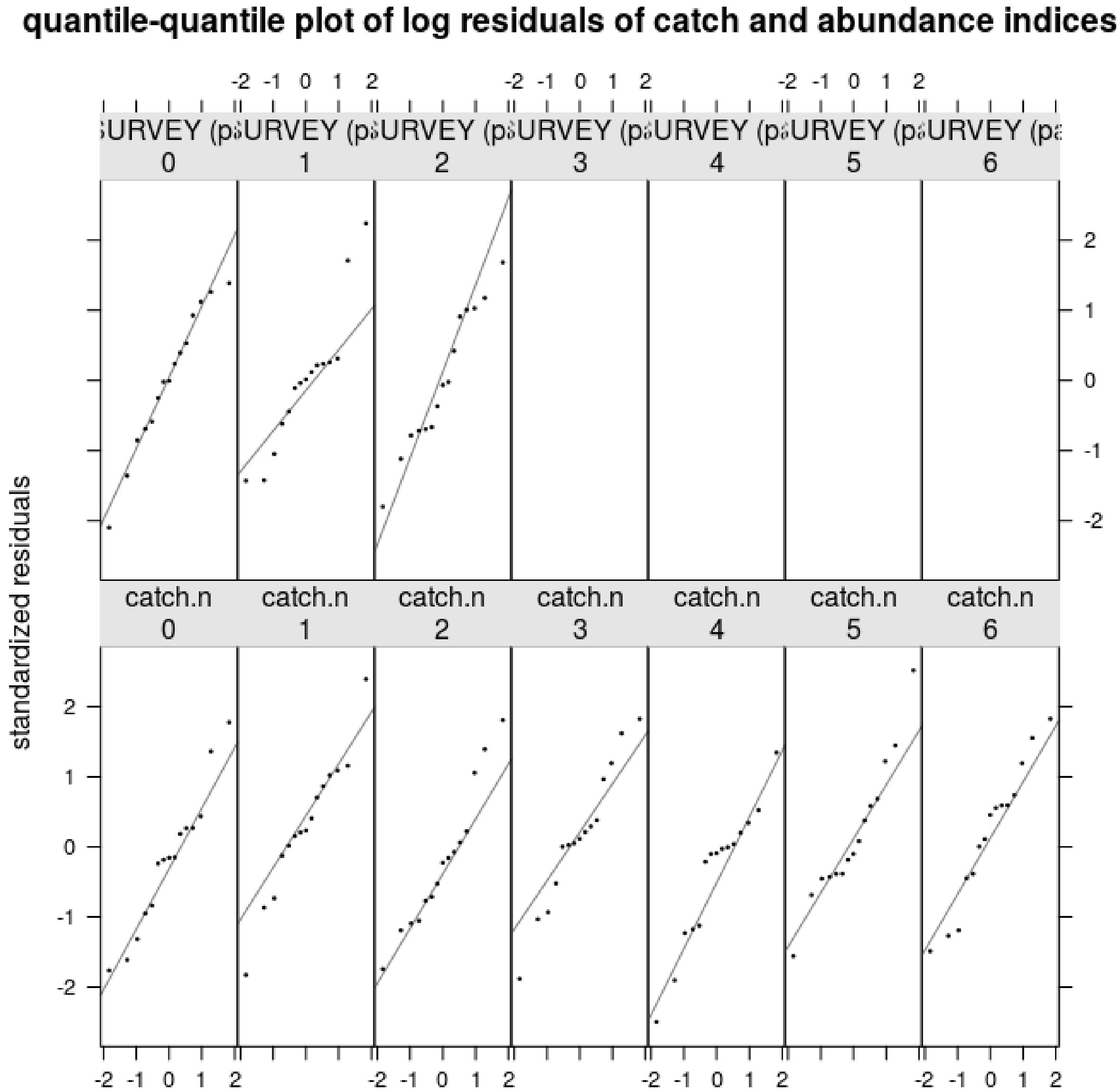


Figure 5.8: Residual plot for more complex specifications with qqplots

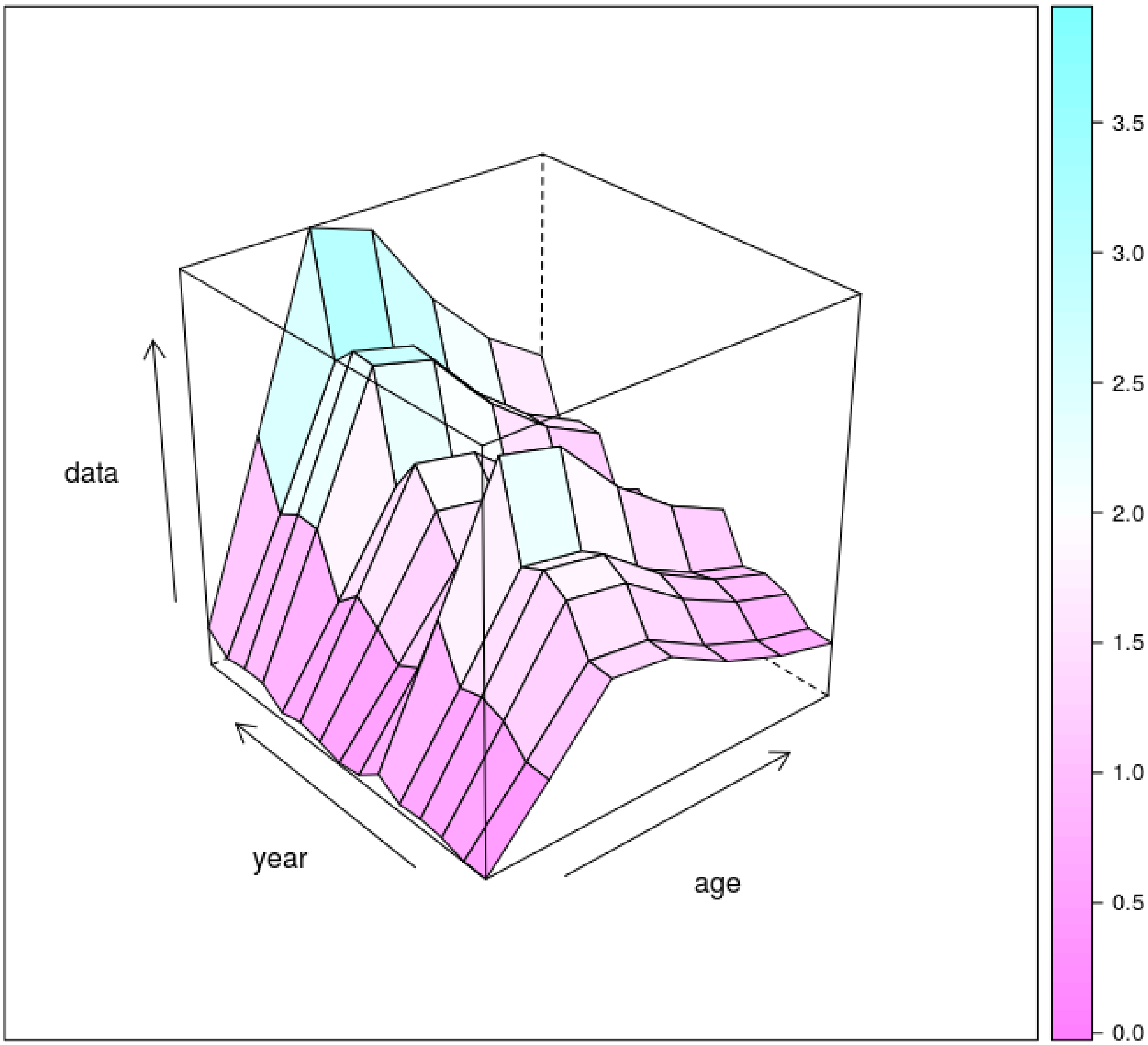


Figure 5.9:

0	1	2	3	4	5	6
0.218	0.805	1.412	1.343	1.052	0.847	0.714
0.221	0.817	1.433	1.363	1.067	0.859	0.725
0.281	1.038	1.821	1.732	1.356	1.092	0.921
0.301	1.116	1.957	1.861	1.458	1.173	0.990
0.287	1.062	1.864	1.772	1.388	1.117	0.943
0.413	1.530	2.684	2.553	1.999	1.609	1.358
0.264	0.979	1.716	1.632	1.278	1.029	0.868
0.235	0.870	1.526	1.452	1.137	0.915	0.772
0.290	1.074	1.883	1.791	1.403	1.129	0.953
0.337	1.246	2.185	2.078	1.627	1.310	1.105
0.289	1.071	1.879	1.787	1.400	1.127	0.951
0.435	1.610	2.824	2.686	2.104	1.693	1.429
0.439	1.624	2.849	2.709	2.122	1.708	1.441

0.410	1.517	2.661	2.530	1.982	1.595	1.346
0.570	2.110	3.702	3.521	2.757	2.219	1.873

Table 5.1: Fishing mortality

We obtain kind of similar trends in residuals for the catch, but the fishing mortality is not in control for the last year. We will try to add a smoother to the year term in the fishing mortality submodel for the year effect.

Improving the assessment: sensitivity

We will now add a logistic shape on age to constrain the fishing mortality and a smoother to control the year effect on the f model. We will try to sort of test the sensitivity of the results to the k. This is not a sensitivity test sensu stricto, but it will help to understand the effect of this parameter.

```
## create a sequence of ks to test
kind <- seq(6, 14)
## create a list for the FLSTocks
STK <- vector(mode = "list")
FIT <- vector(mode = "list")
for (i in 1:length(kind)) {
    index <- hke.idx2
    qmod <- list(~s(age, k = 3))
    fmod <- ~I(1/(1 + exp(age))) + s(year, k = kind[i]) + s(age,
        k = 3)
    srmod <- ~s(year, k = 10)
    fit <- sca(stock = hke, indices = index, fmodel = fmod, srmodel = srmod,
        qmodel = qmod, fit = "assessment")
    STK[[i]] <- hke + fit
    FIT[[i]] <- fit
}
stks <- FLStocks(STK)
names(stks) <- paste("k=", kind, sep = "")
names(FIT) <- paste("k=", kind, sep = "")
```

We see that the f estimates are definitely sensitive to the k we give for the year effect (Figure 5.11). The fishing mortality flying away is not necessarily realistic either. We can choose a middle value, 9 or 10, but the best approach would certainly be to do some model averaging, which will be shown a bit further in the document. For the moment we go for a value of 9 for the k and we look at the residuals (Figure 5.11), which are much better. The residuals for the survey behave nicely and residuals for fishing mortality are not bad either, excepted a few nasty trends for ages 0 and 3. The fishing mortality seems to be in control (Figure 5.12).

We look at the surface plot for fishing mortality: We feel much more comfortable with these results. The last thing that is not completely on are the trends in the residuals of the catches for ages 0 and 3.

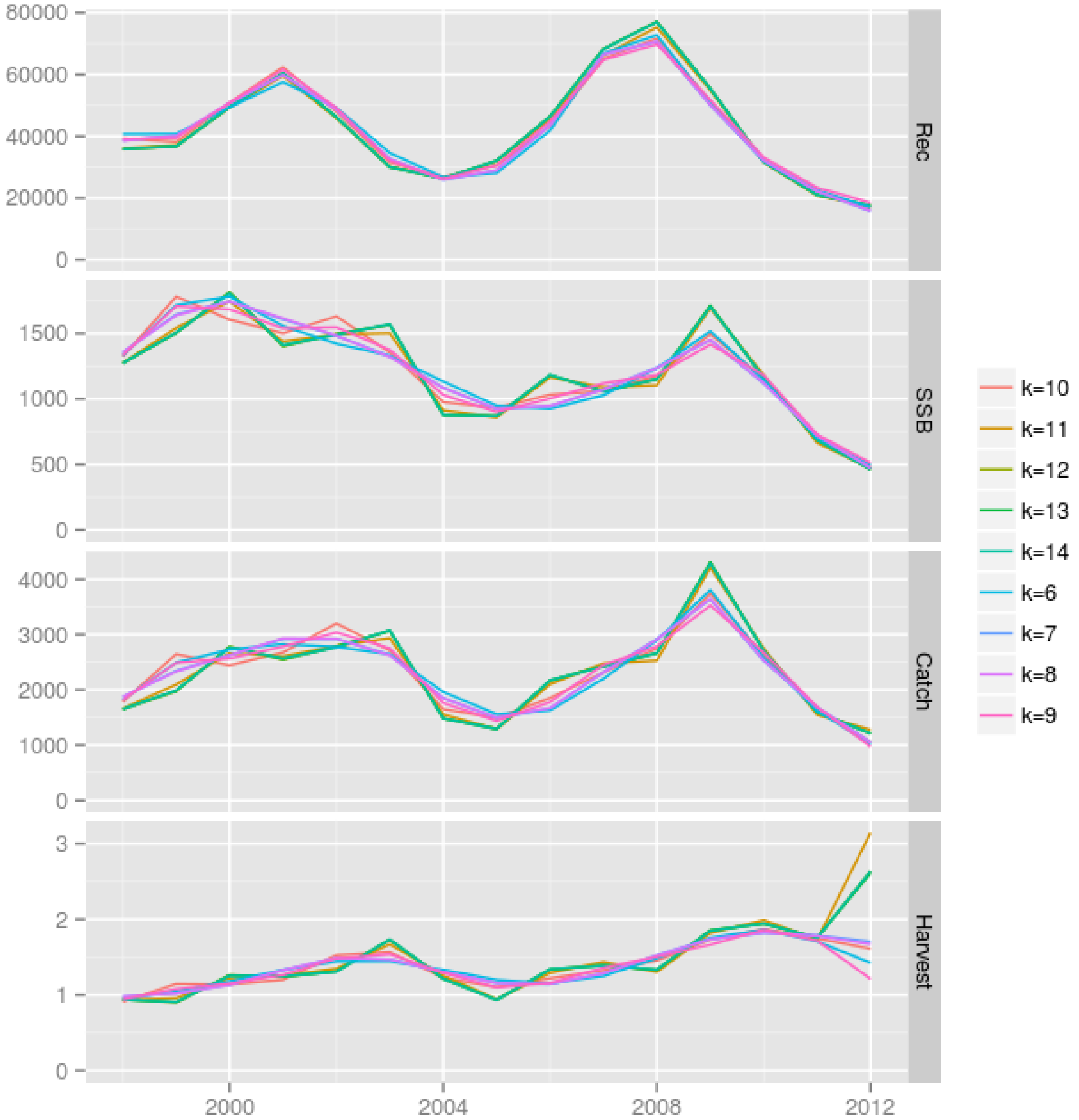


Figure 5.10: Stock plot with sensitivities to different k

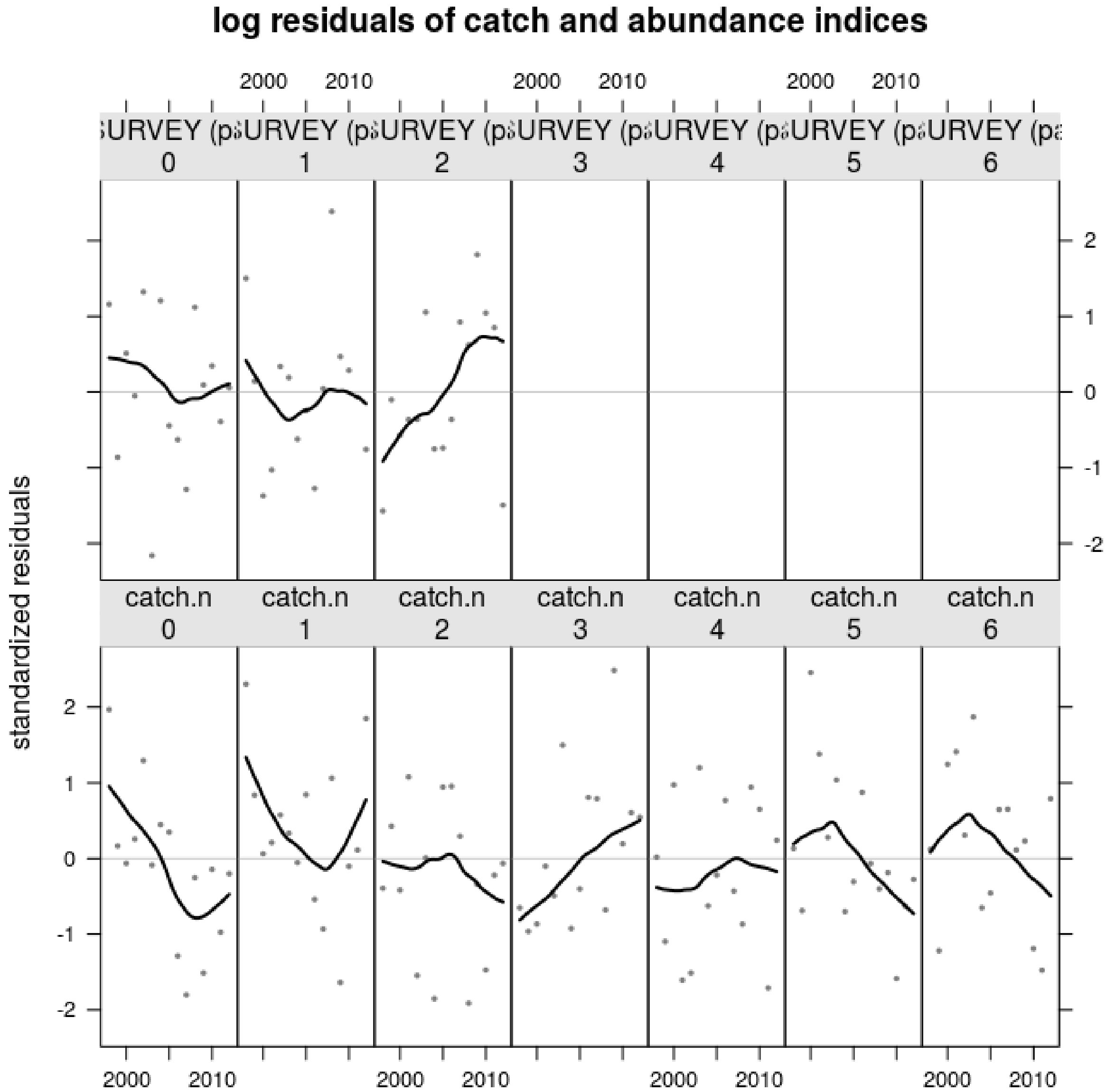


Figure 5.11: Dirty fit

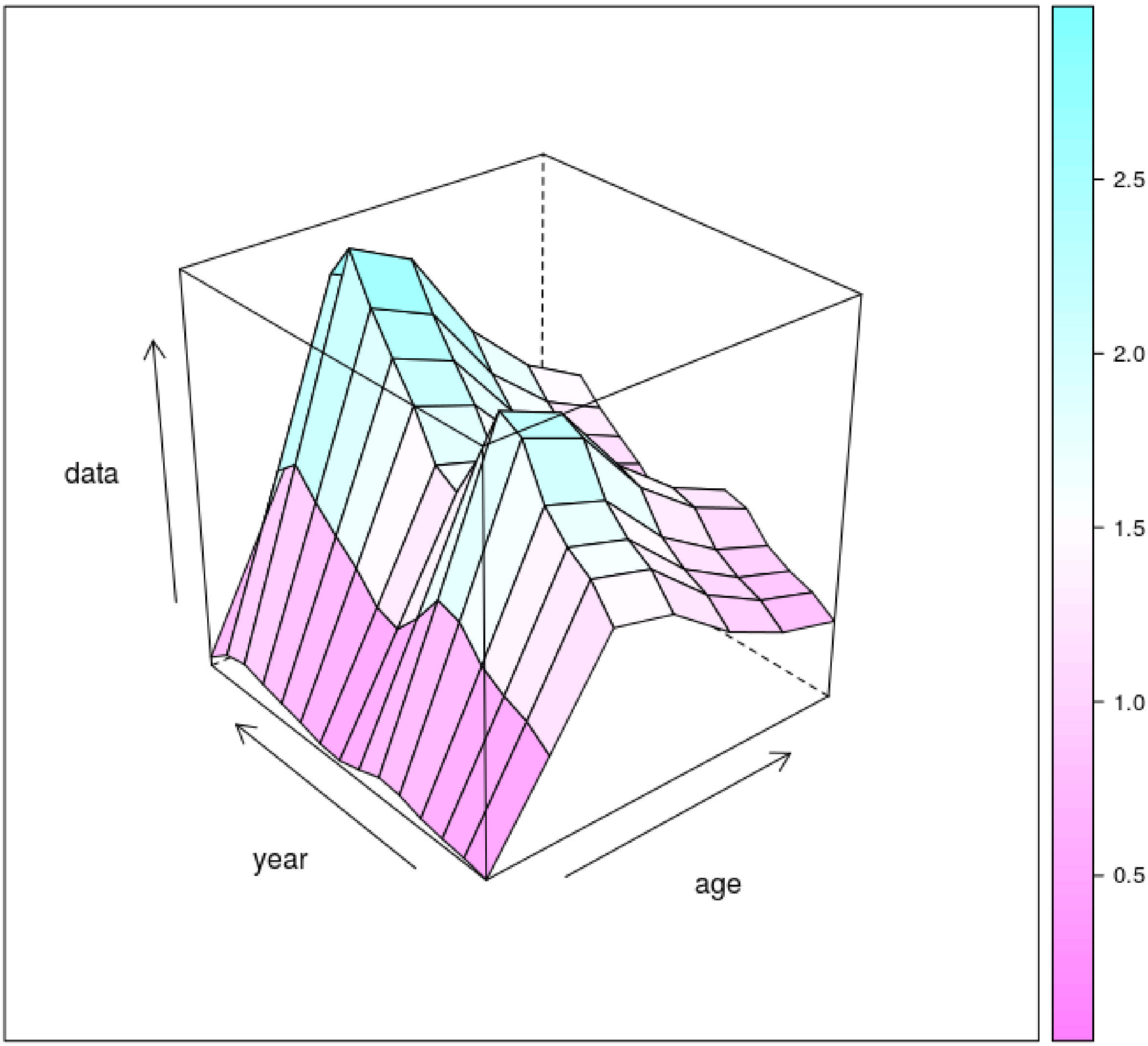


Figure 5.12: Dirty fit

0	1	2	3	4	5	6
0.208	0.809	1.425	1.328	1.030	0.843	0.735
0.238	0.928	1.634	1.522	1.182	0.966	0.842
0.254	0.989	1.742	1.623	1.260	1.030	0.898
0.279	1.085	1.911	1.781	1.382	1.130	0.985
0.327	1.271	2.240	2.087	1.619	1.324	1.154
0.339	1.320	2.325	2.166	1.681	1.375	1.198
0.285	1.109	1.953	1.820	1.412	1.155	1.007
0.242	0.944	1.663	1.549	1.202	0.983	0.857
0.256	0.996	1.754	1.634	1.268	1.037	0.904
0.299	1.165	2.052	1.912	1.484	1.213	1.058
0.332	1.291	2.275	2.120	1.645	1.345	1.173
0.367	1.429	2.517	2.345	1.820	1.488	1.297
0.410	1.597	2.815	2.622	2.035	1.664	1.450

0.379	1.473	2.596	2.418	1.877	1.535	1.338
0.267	1.040	1.832	1.706	1.324	1.083	0.944

Table 5.2: Fishing mortality

Improving the assessment: even more complex specifications

We will now add a bidimensional smoother for the f model. We will also test the effect of the k on the results.

```
## create a sequence of ks to test
kind <- seq(5, 9)
## create a list for the FLSTocks
STK <- vector(mode = "list")
FIT <- vector(mode = "list")
RESC <- vector(mode = "list")
RESS <- vector(mode = "list")
for (i in 1:length(kind)) {
    index <- hke.idx2
    qmod <- list(~s(age, k = 3))
    fmod <- ~I(1/(1 + exp(age))) + te(age, year, k = c(3, kind[i]))
    srmod <- ~s(year, k = 10)
    fit <- sca(stock = hke, indices = index, fmodel = fmod, srmodel = srmod,
        qmodel = qmod, fit = "MP")
    STK[[i]] <- hke + fit
    FIT[[i]] <- fit
    res <- residuals(fit, hke, index)
    RESC[[i]] <- res[[1]]
    RESS[[i]] <- res[[2]]
}
stks <- FLStocks(STK)
names(stks) <- paste("k=", kind, sep = "")
resc <- FLQuants(RESC)
ress <- FLQuants(RESS)
```

We plot the results:

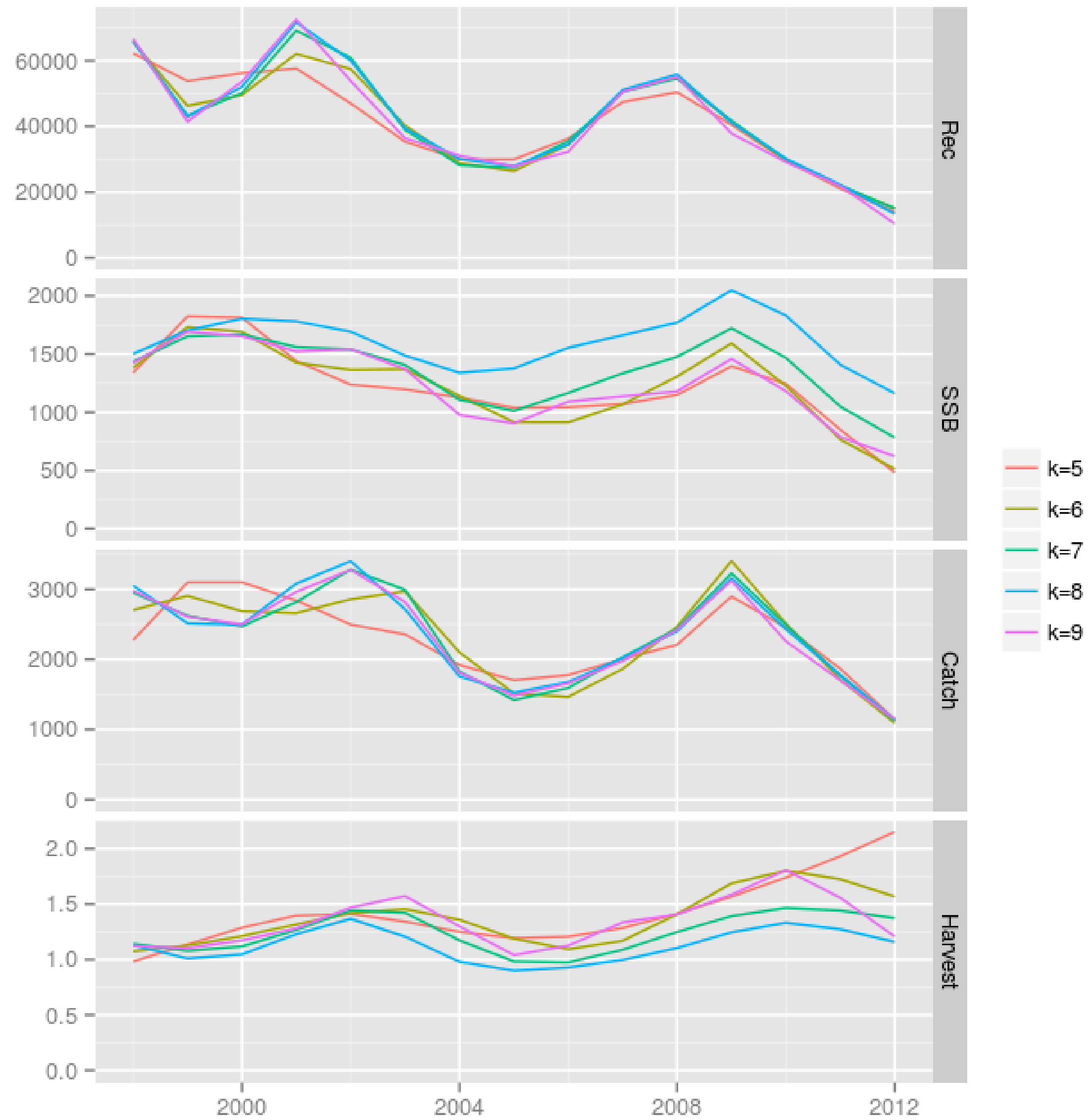


Figure 5.13: Dirty fit

Looking one by one we see that k set to 6 or 7 is the best. We choose 6. We look at the residuals:

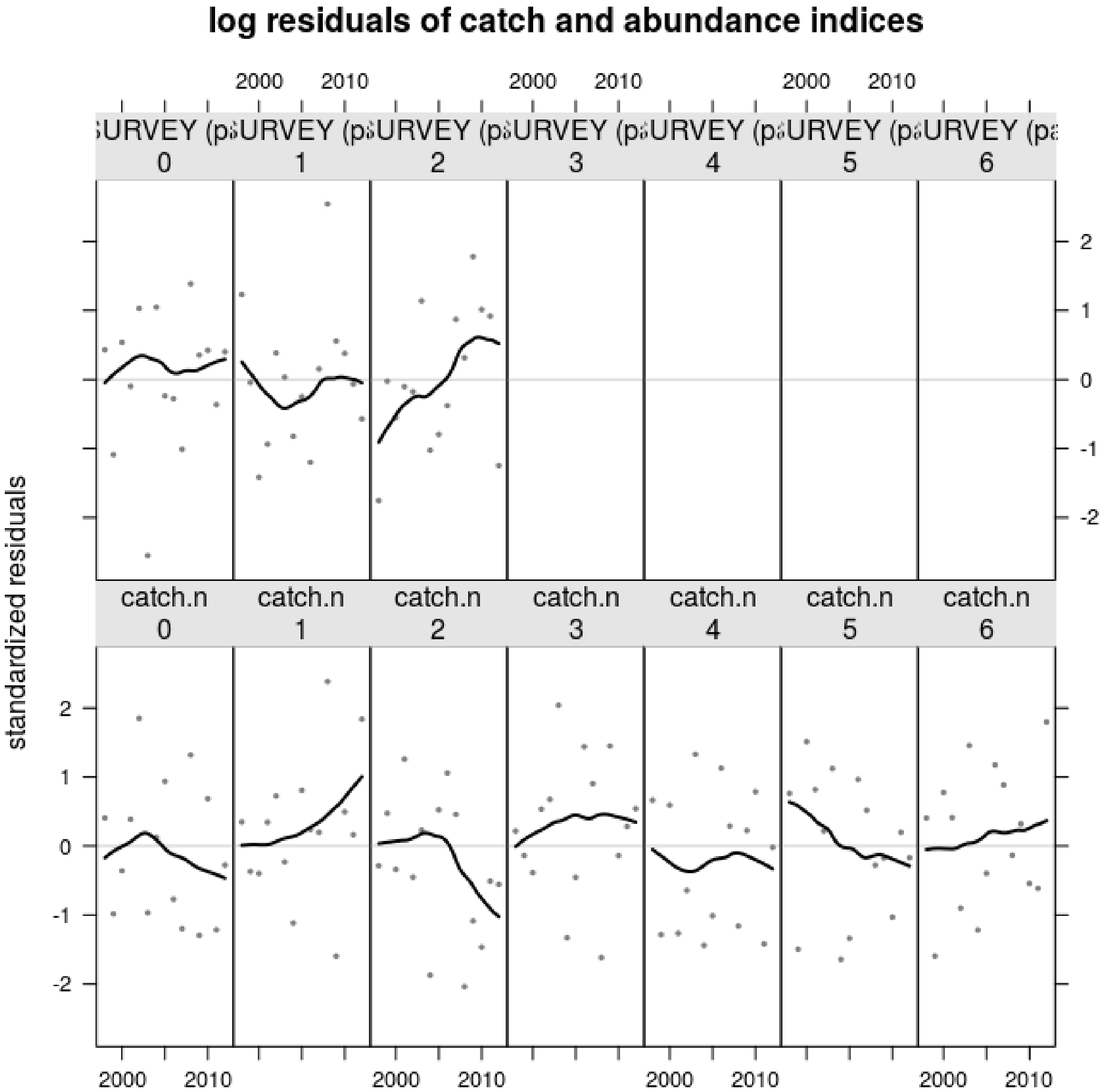


Figure 5.14: Dirty fit

This is much better. The residuals for the survey behave nicely and residuals for fishing mortality are better as well.

We look at the surface plot for fishing mortality:

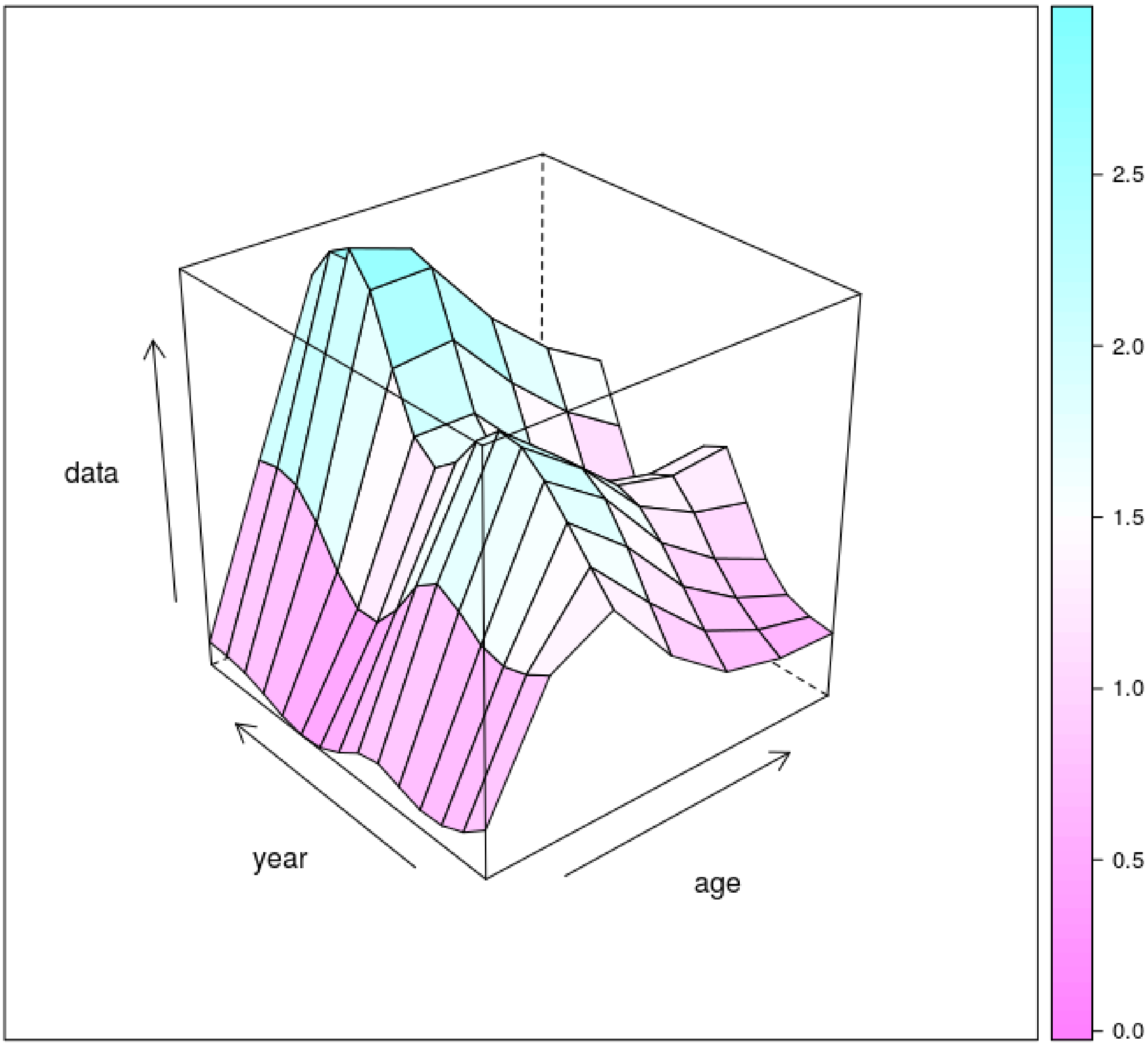


Figure 5.15: Dirty fit

We notice that we still have very high fishing mortalities.

0	1	2	3	4	5	6
0.490	1.282	1.505	1.022	0.721	0.622	0.606
0.362	1.188	1.683	1.274	0.903	0.722	0.628
0.293	1.142	1.867	1.543	1.109	0.849	0.689
0.285	1.180	2.039	1.765	1.311	1.028	0.850
0.334	1.303	2.164	1.872	1.464	1.258	1.161
0.385	1.390	2.183	1.862	1.520	1.422	1.461
0.349	1.283	2.044	1.762	1.445	1.352	1.386
0.246	1.033	1.827	1.649	1.308	1.117	1.014
0.174	0.854	1.710	1.635	1.258	0.975	0.777
0.160	0.853	1.832	1.833	1.426	1.089	0.845
0.192	1.018	2.192	2.234	1.810	1.464	1.212

0.250	1.261	2.614	2.625	2.166	1.827	1.597
0.298	1.427	2.805	2.676	2.099	1.685	1.402
0.320	1.465	2.721	2.395	1.679	1.177	0.845
0.326	1.432	2.508	2.007	1.220	0.715	0.422

Table 5.3: Fishing mortality

We are kind of happy so far with this model specification. We can now look at how the model can predict the data.

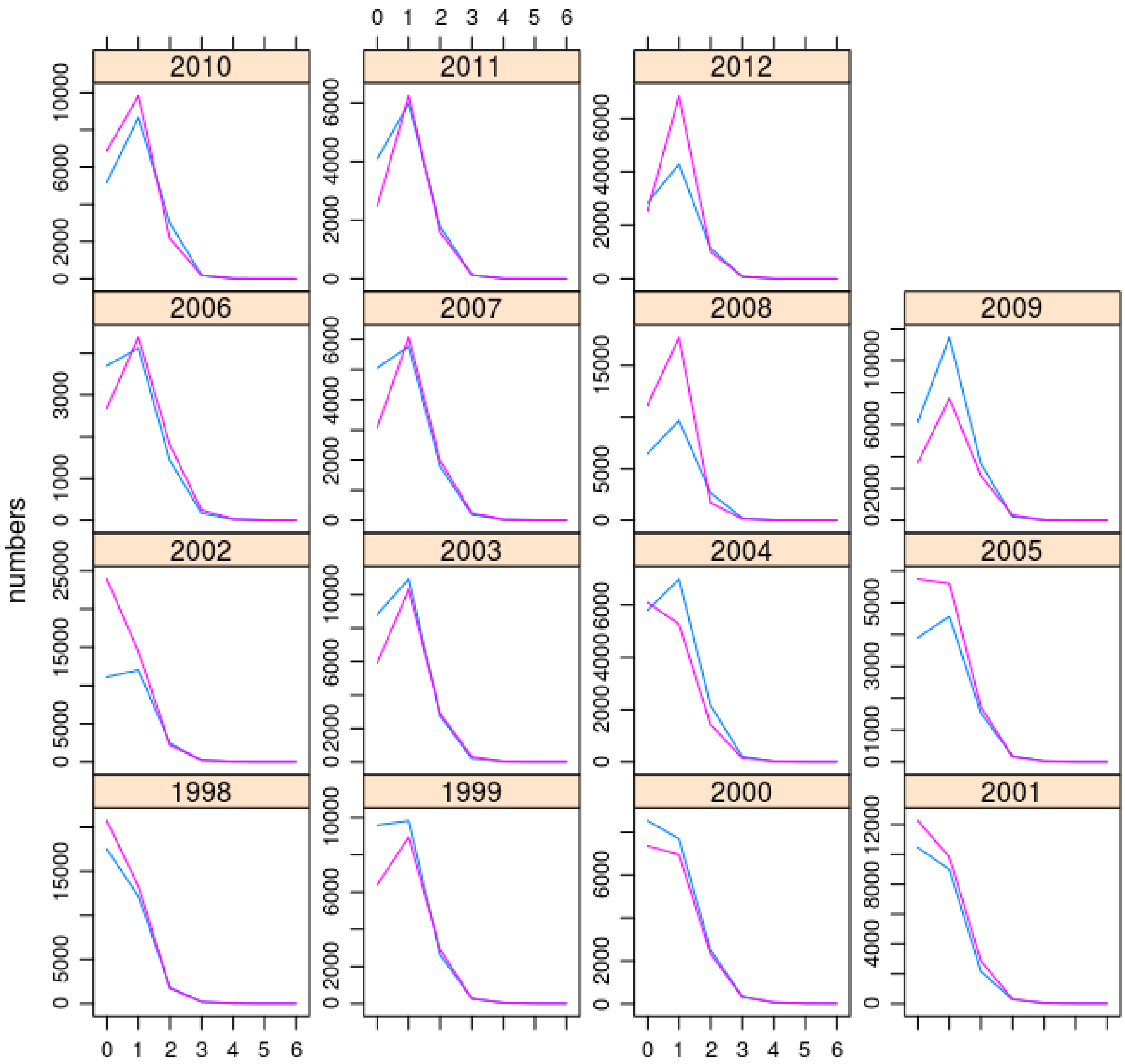


Figure 5.16: Predicting the catch

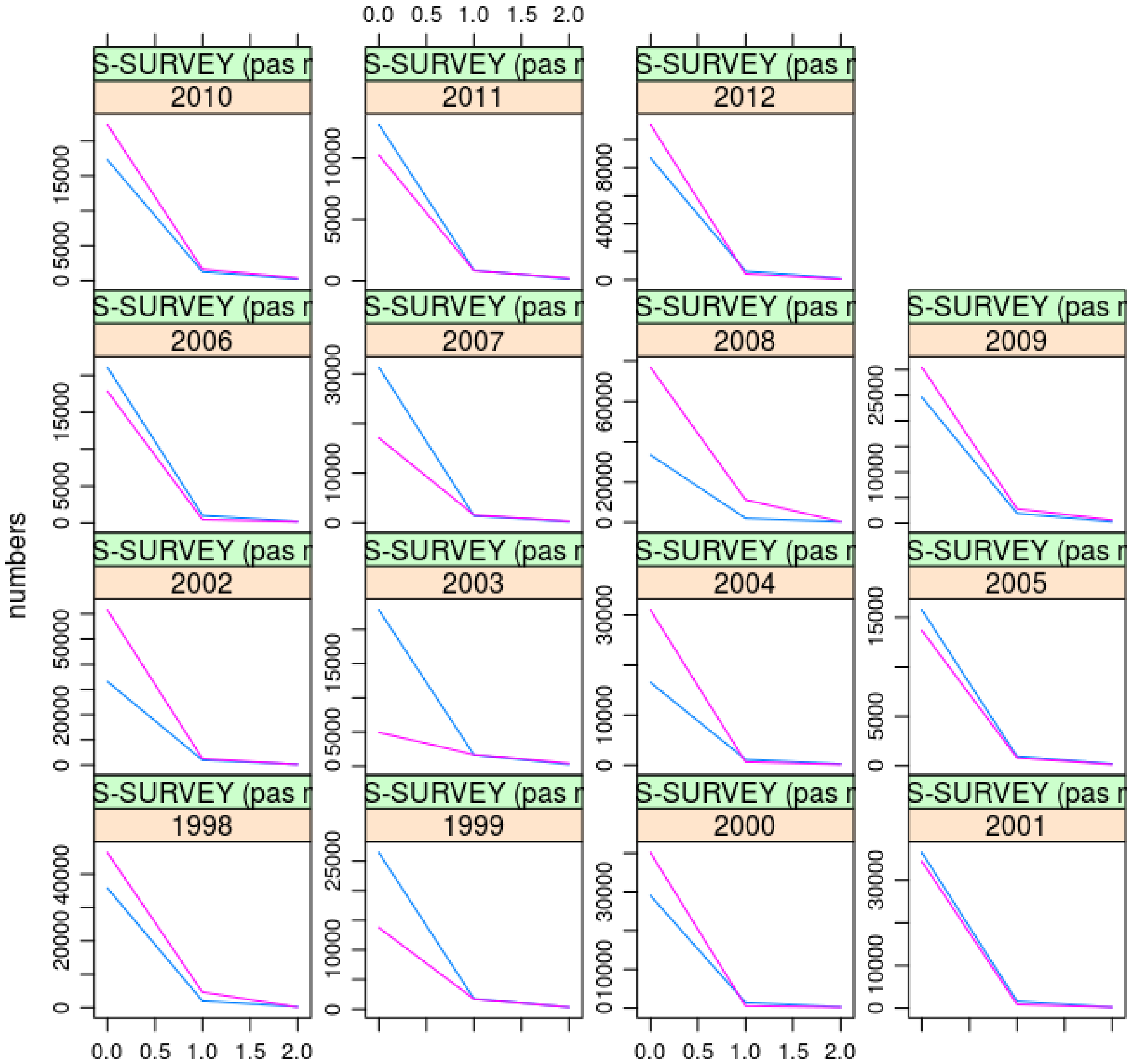


Figure 5.17: Predicting the survey

5.2.1 Model averaging

We start from a previous model and we will play with the different k values for

```
## create a sequence of ks to test
kind <- seq(5, 9)
## create a list for the FLSTocks
STK <- vector(mode = "list")
FIT <- vector(mode = "list")
for (i in 1:length(kind)) {
    index <- hke.idx2
    qmod <- list(~s(age, k = 3))
    fmod <- ~I(1/(1 + exp(age))) + te(age, year, k = c(3, kind[i]))
```

```
    srmod <- ~factor(year)
    fit <- sca(stock = hke, indices = index, fmodel = fmod, srmodel = srmod,
        qmodel = qmod, fit = "assessment")
    STK[[i]] <- hke + fit
    FIT[[i]] <- fit
}
names(STK) <- paste("k=", kind, sep = "")
names(FIT) <- paste("k=", kind, sep = "")
stks <- FLStocks(STK)
stock.sim <- ma(a4aFitSAs(FIT), hke, BIC, nsim = 1000)
stkss <- FLStocks(stks, stock.sim)
```

We plot the stock average:

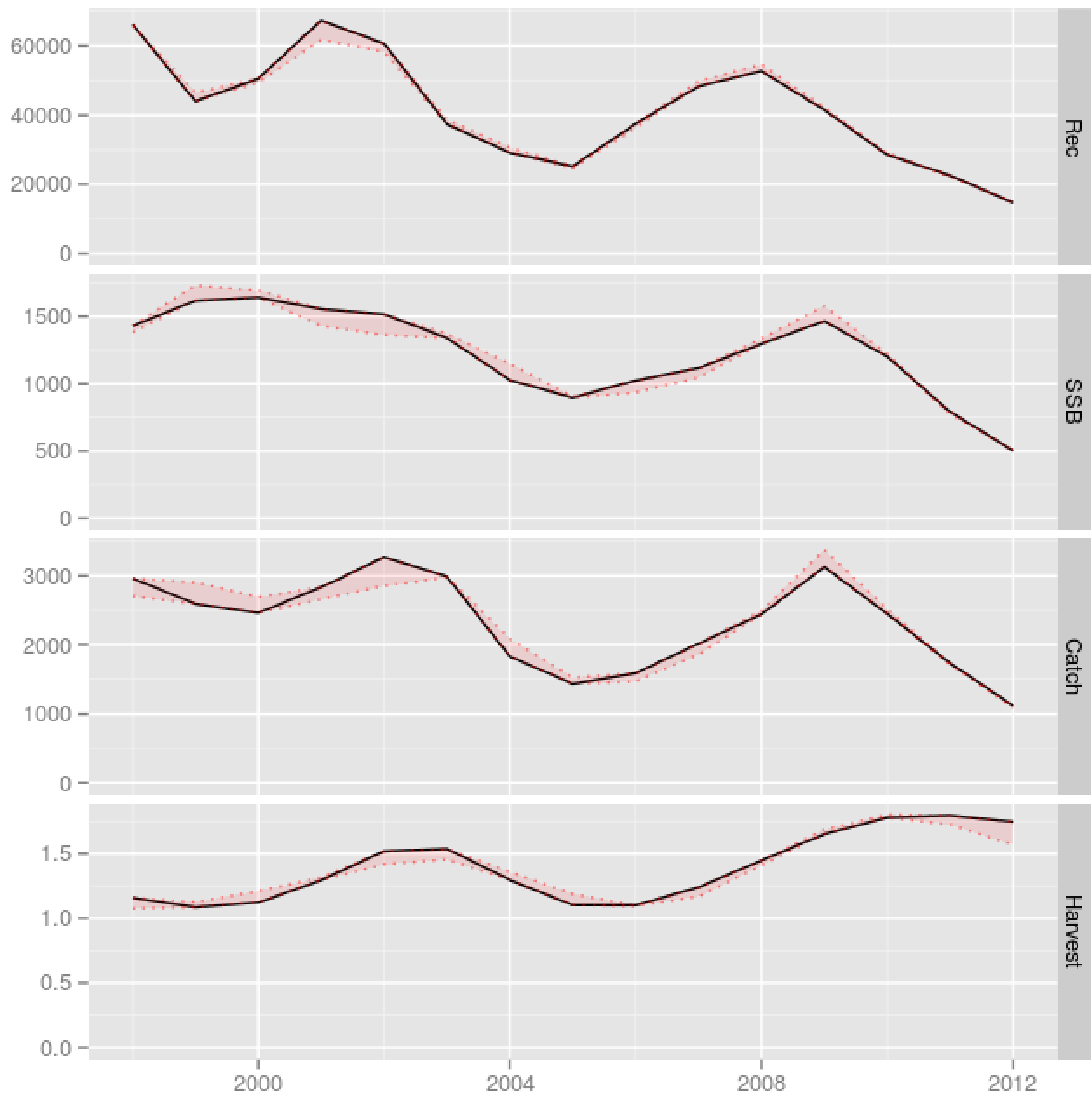


Figure 5.18: Simulations for the average model

We plot all the stocks together:

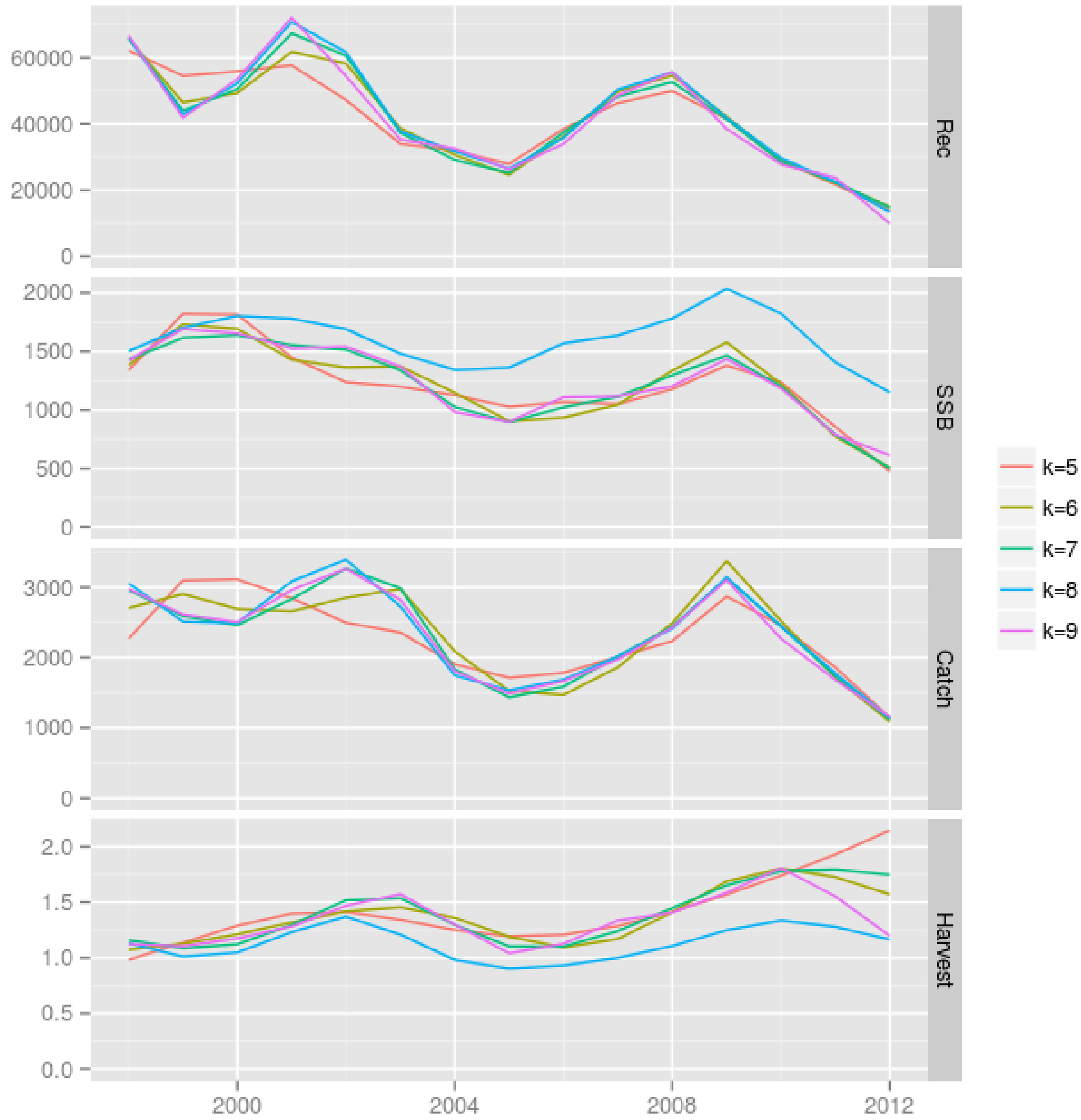


Figure 5.19: Results of all the stocks

Play with the variance model

We start from the best model we have. We will now use the function a4aSCA that allows to adjust the variance of surveys and catch. We start by giving a shape for ages.

```
index <- hke.idx2
qmod <- list(~s(age, k = 3))
fmod <- ~I(1/(1 + exp(age))) + te(year, age, k = c(6, 4))
srmod <- ~factor(year)
vmod <- list(~s(year, k = 3) + s(age, k = 3), ~age)
fit <- a4aSCA(stock = hke, indices = index, fmodel = fmod, srmodel = srmod,
```

```
    qmodel = qmod, vmodel = vmod)
stk <- hke + fit
```

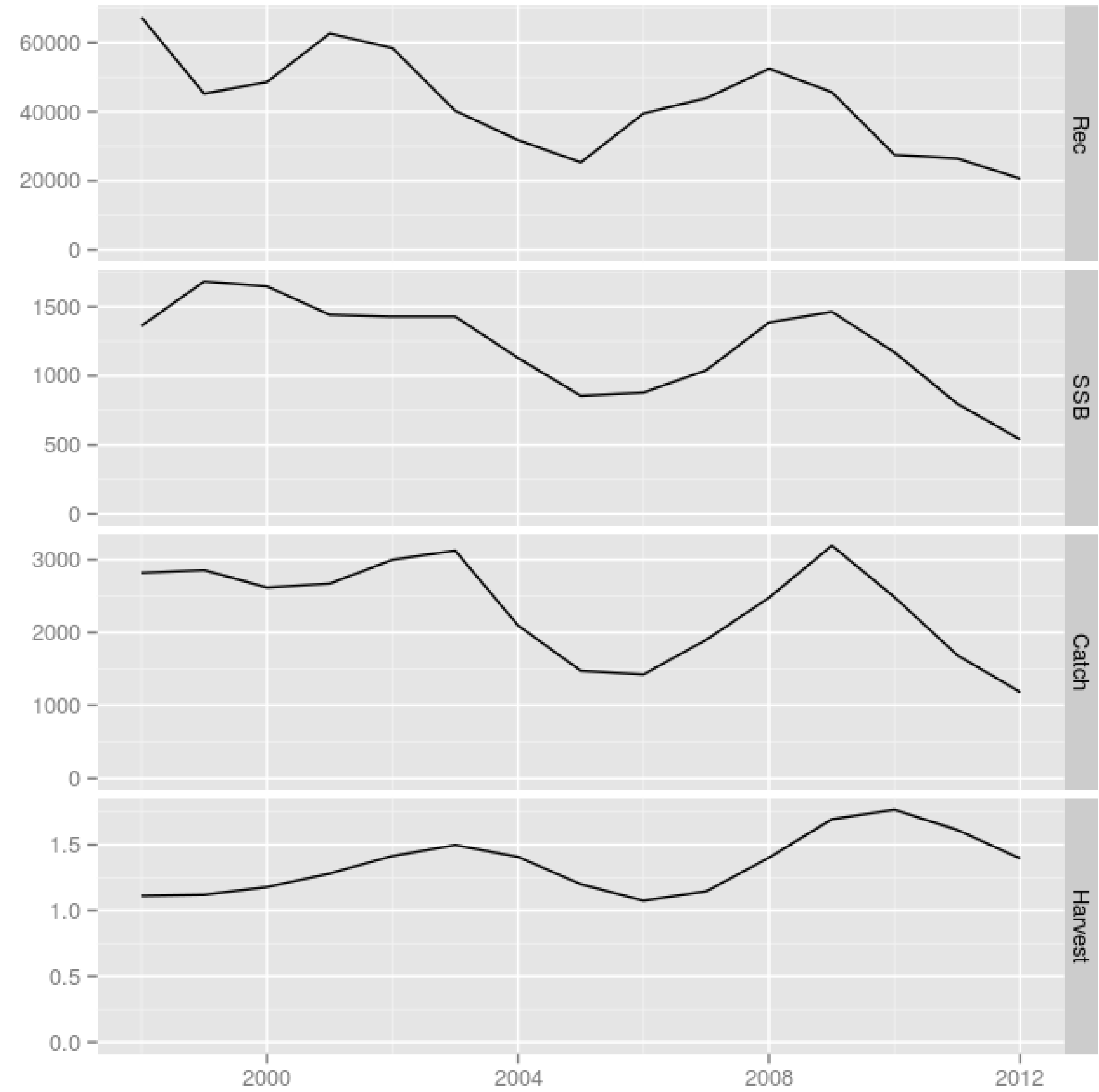


Figure 5.20: Stock plot for the model with variance submodel

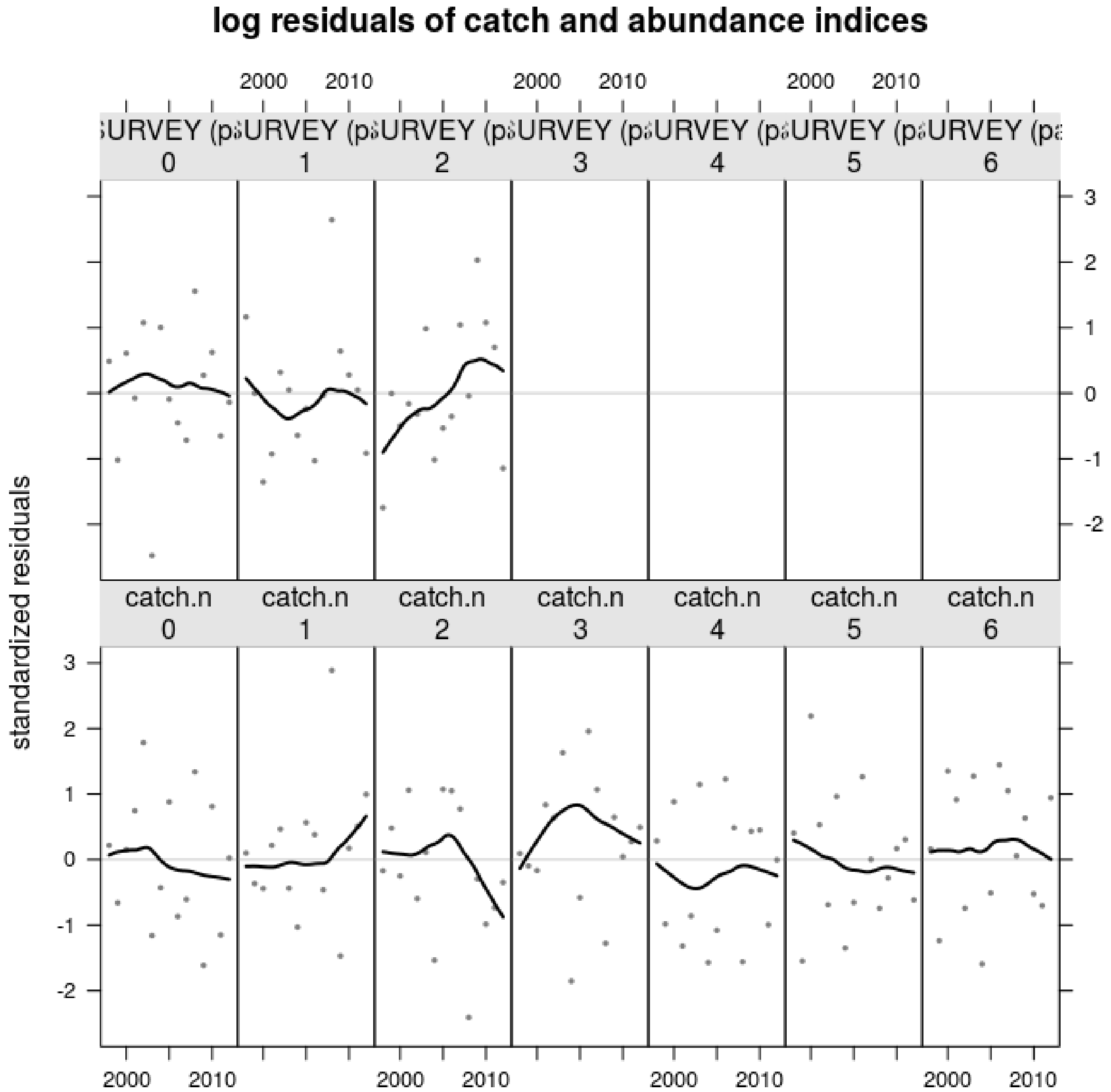


Figure 5.21: Residual plot for the model with variance submodel

We are quite happy with what we end up with. We can now look at the shapes of the fitted submodels for variance and catchability:

```
## [1] "stkmodel" "qmodel"   "vmodel"
```

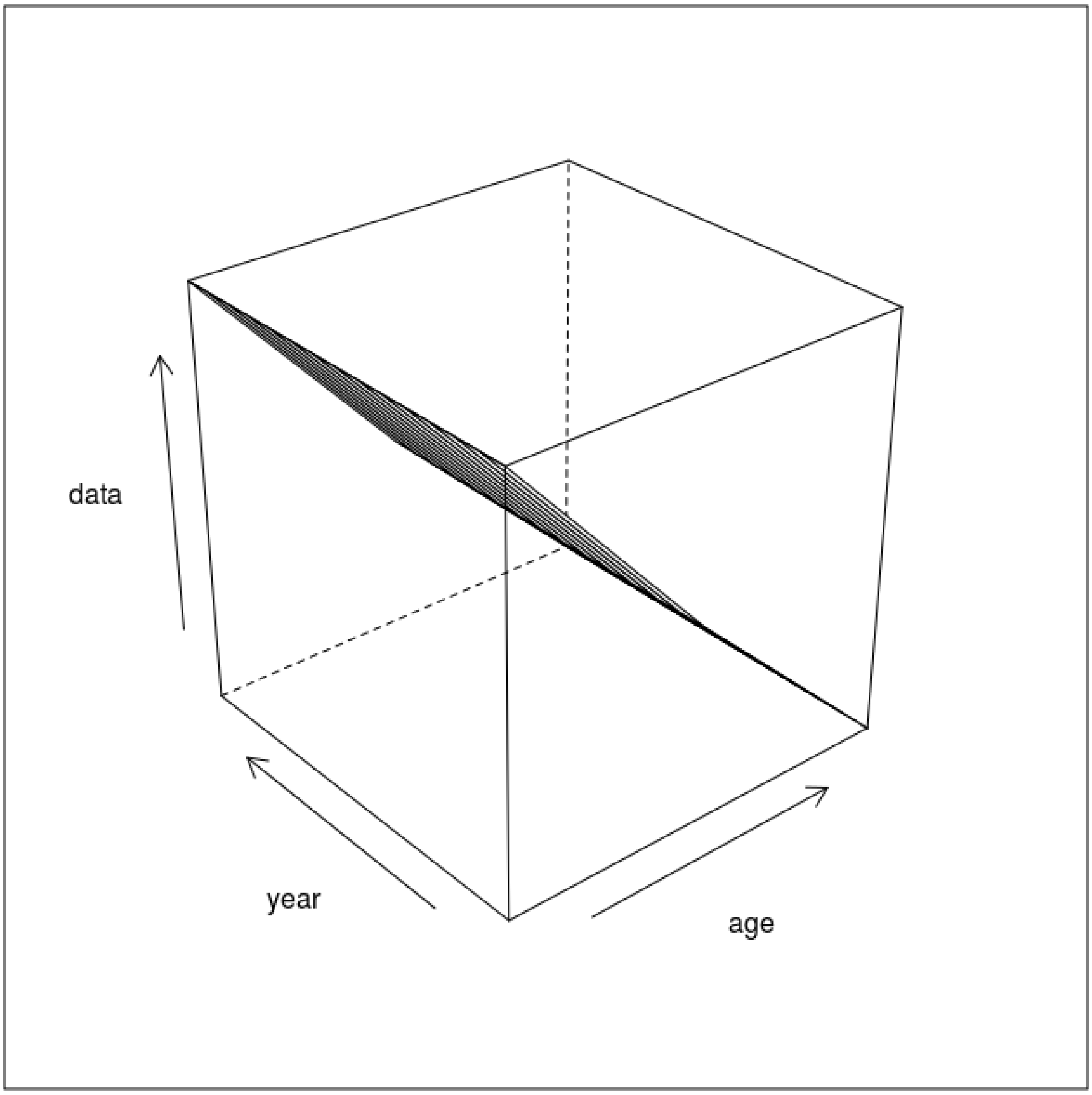


Figure 5.22: Submodels for variance and catchability

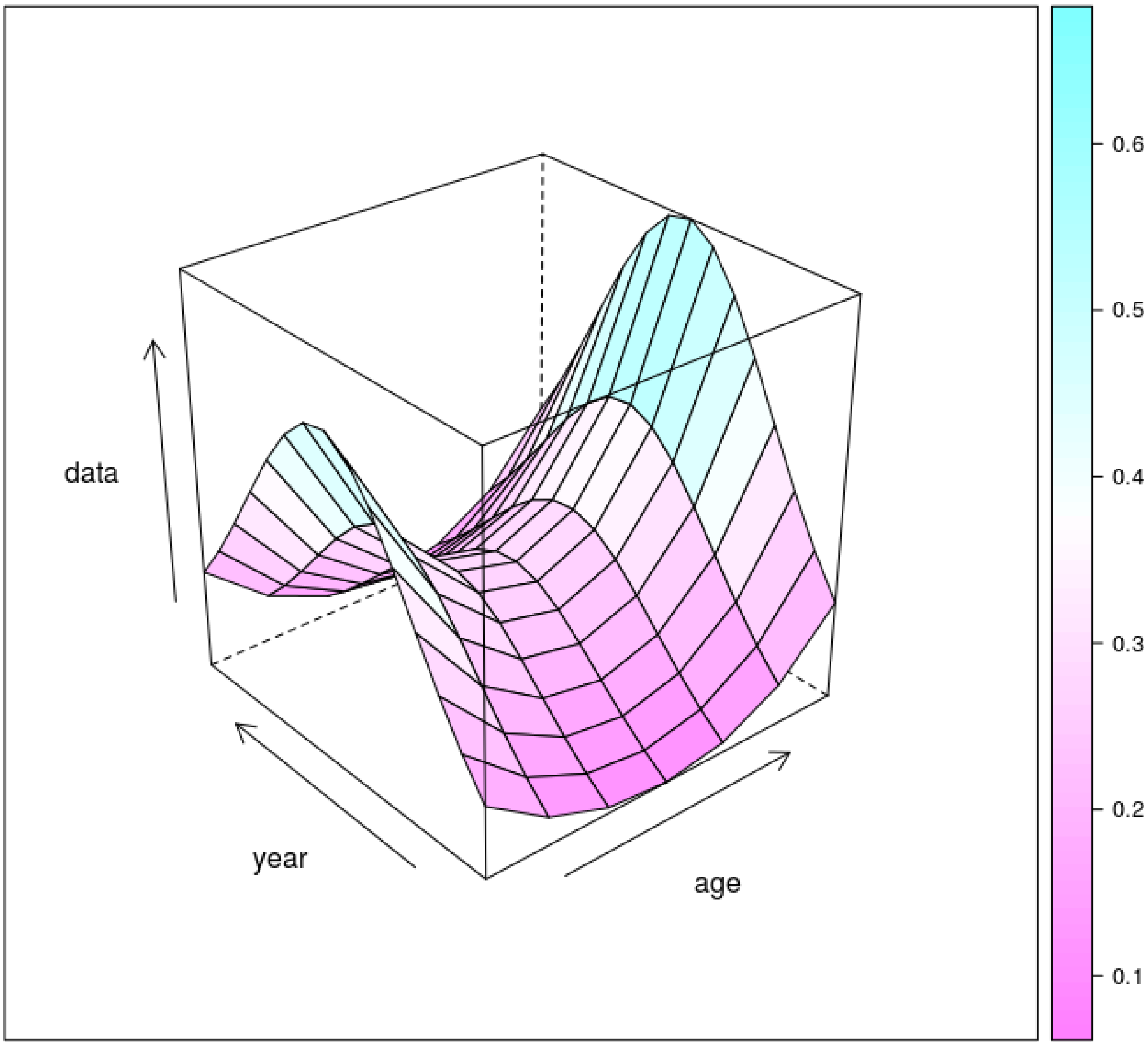


Figure 5.23: Submodels for variance and catchability

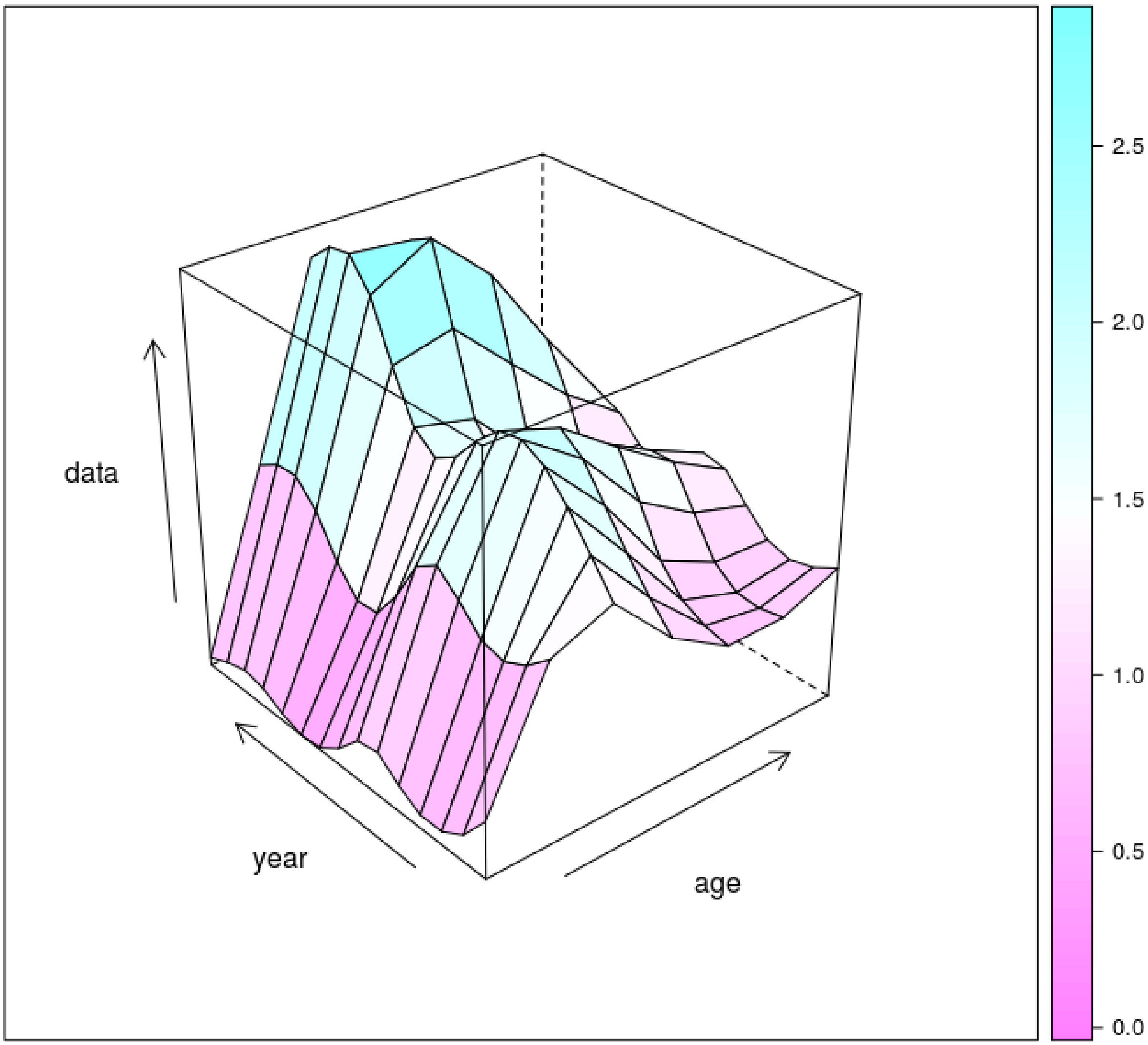


Figure 5.24: Submodels for variance and catchability

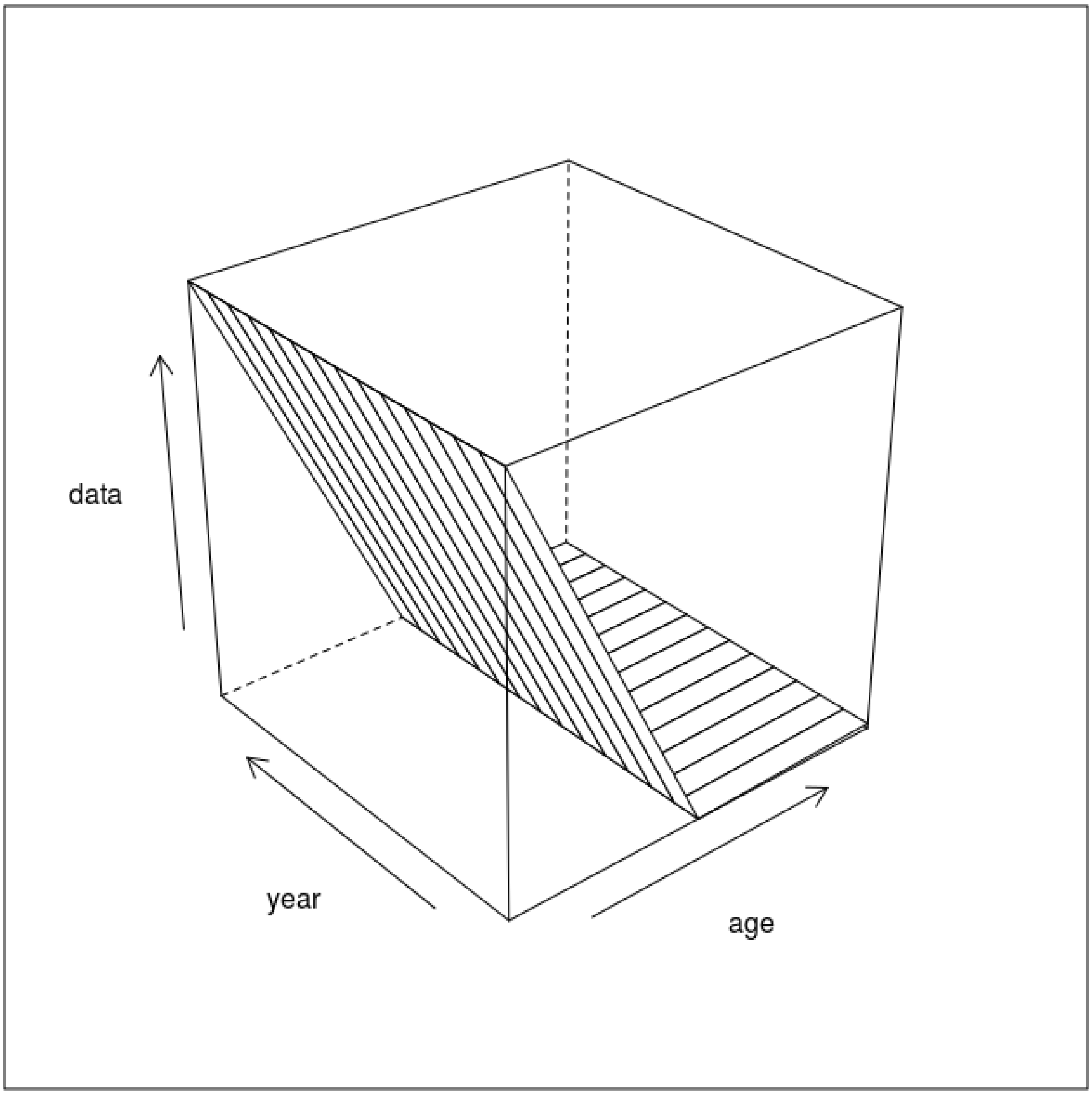


Figure 5.25: Submodels for variance and catchability

Average across k values

Now we average the models across some assumptions for k:

```
## create a sequence of ks to test
kind <- seq(5, 9)
## create a second sequence
kfage <- seq(3, 5)
## create a list for the FLSTocks
STK <- vector(mode = "list")
FIT <- STK
SIM <- FIT
compt <- 1
```

```
n <- NA
Nit <- 100
for (i in 1:length(kind)) {
    for (j in 1:length(kfage)) {
        index <- hke.idx2
        qmod <- list(~s(age, k = 3))
        fmod <- ~I(1/(1 + exp(age))) + te(year, age, k = c(kind[i],
            kfage[j]))
        srmod <- ~factor(year)
        vmod <- list(~s(year, k = 3) + s(age, k = 3), ~age)
        fit <- a4aSCA(stock = hke, indices = index, fmodel = fmod,
            srmodel = srmod, qmodel = qmod, vmodel = vmod)
        STK[[compt]] <- hke + fit
        FIT[[compt]] <- fit
        SIM[[compt]] <- hke + simulate(fit, Nit)
        n <- c(n, paste("k.f.year=", kind[i], ",k.f.age=", kfage[j],
            sep = ""))
        compt <- compt + 1
    }
}
names(STK) <- n[-1]
names(FIT) <- names(STK)
stks <- FLStocks(STK)
## We extract the BIC lapply(FIT,BIC) stock.sim <-
## ma(a4aFitSAs(FIT), hke, BIC, nsim = Nit) we manually put
## the simulations in an object to avoid the weights given by
## the BIC or AIC make an object with many iterations
hke.ma <- propagate(hke, (compt - 1) * Nit)
## fill the iterations with the simulations
for (i in 1:(compt - 1)) {
    hke.ma[, , , , , (((i - 1) * Nit) + 1):(i * Nit)] <- SIM[[i]]
}
```

Using this technique gives equal weight to all the models, whereas the AIC or BIC often gives a lot of weight to a few models.

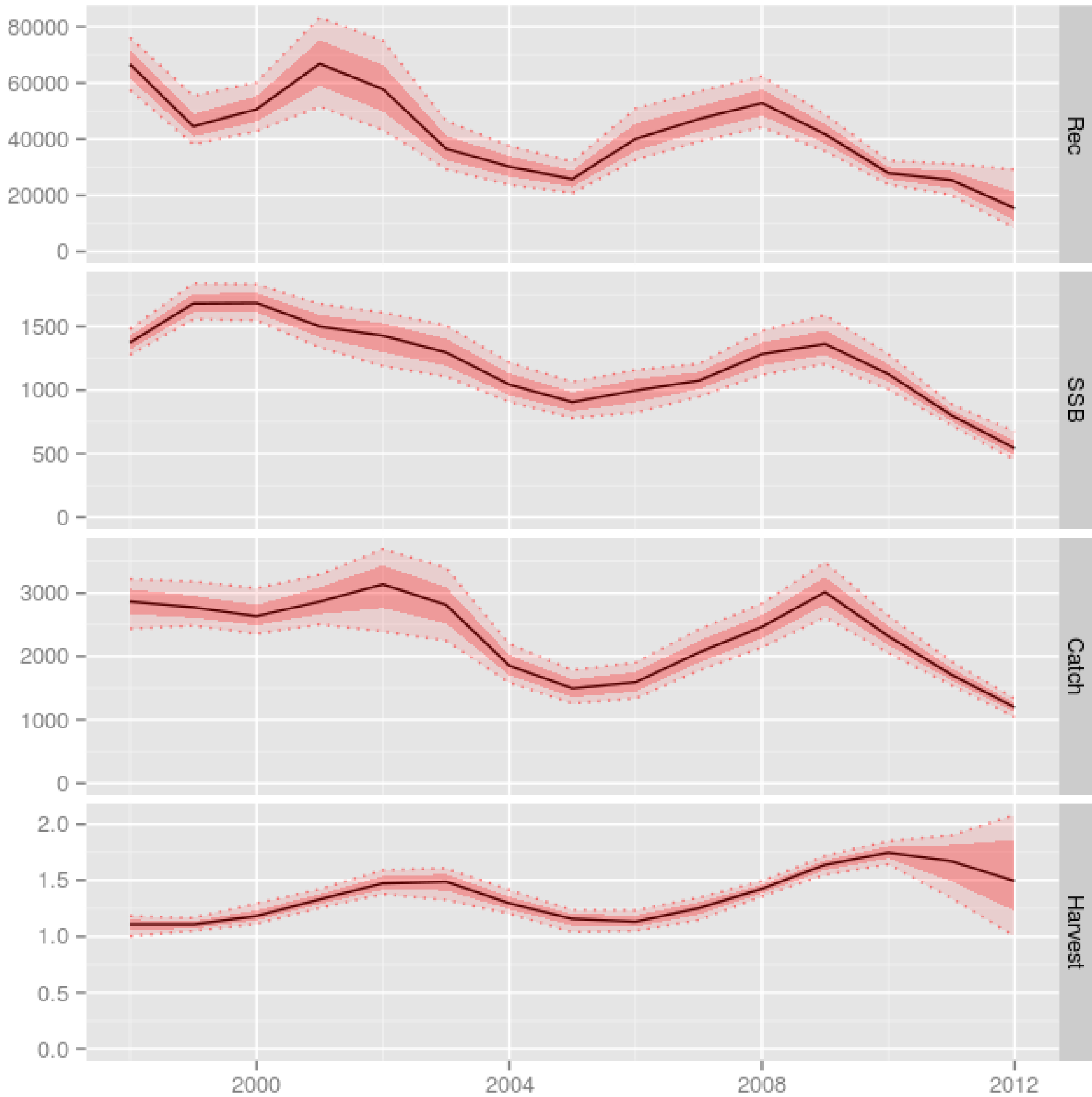


Figure 5.26: Stock averaging

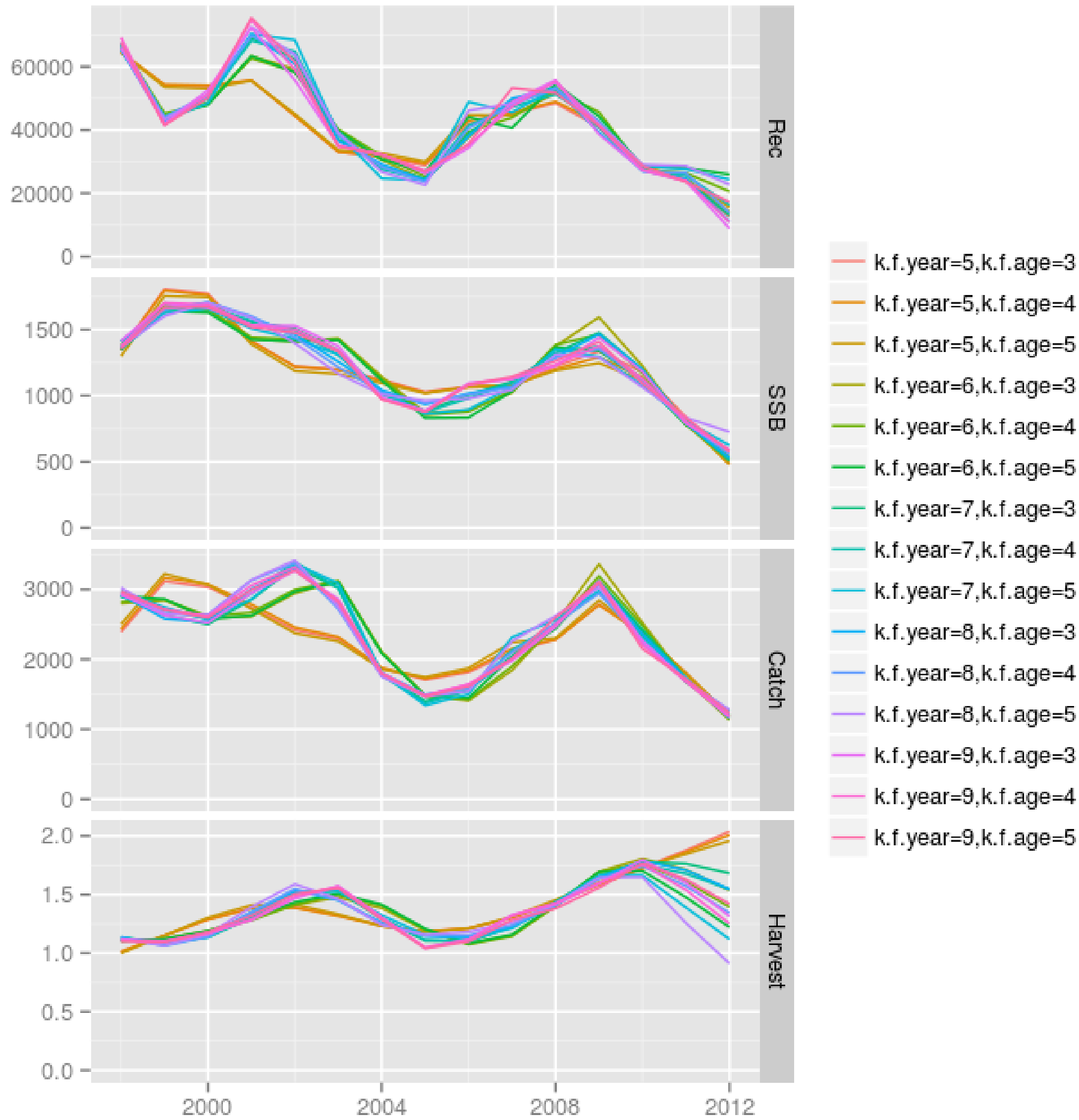


Figure 5.27: Different models

5.3 Introducing uncertainty in growth and natural mortality using length data

We will now read abundance at length and specify a growth model with uncertainty to spread the uncertainty on age decomposition through the assessment. We start with reading the number at length data and make and FLQuant with it:

```
## READ LENGTH DATA
d <- read.table("data/N_hke_lgth_1998_2012.txt", header = TRUE)
d2 <- as.matrix(d[, -1])/1000
## MAKE FLQUANT
```

```
dnms <- dimnames(hke@catch.n)
names(dnms) <- names(dimnames(FLQuant()))
names(dnms)[1] <- "len"
dnms$len <- as.character(d$Size_cm)
cth.n <- FLQuant(d2, dimnames = dnms)
```

Then we create a growth object with it, using the data from Mellon et al 2010:

```
# Growth both sexes Mellon 2010
vbObj <- a4aGr(grMod = ~linf * (1 - exp(-k * (t - t0))), grInvMod = ~t0 -
    1/k * log(1 - len/linf), params = FLPar(linf = 110, k = 0.178,
    t0 = 0))
# trial: predict from the length
predict(vbObj, len = seq(5, 70, length = 10))

##     iter
##          1
##   1  0.2613
##   2  0.6617
##   3  1.0928
##   4  1.5597
##   5  2.0690
##   6  2.6291
##   7  3.2513
##   8  3.9511
##   9  4.7507
##   10 5.6832

# trial: predict from the ages
predict(vbObj, t = seq(0, 10, length = 10))

##     iter
##         1
##   1   0.00
##   2  19.74
##   3  35.94
##   4  49.23
##   5  60.13
##   6  69.08
##   7  76.42
##   8  82.45
##   9  87.39
##   10 91.45
```

We will now introduce uncertainty with a variance covariance matrix between the parameters:

```
# Make an empty cor matrix
cm <- diag(c(1, 1, 1))
```

```
# k and linf are negatively correlated while t0 is
# independent
cm[1, 2] <- cm[2, 1] <- -0.5
# scale cor to var using CV=0.2
cv <- 0.1
p <- c(linf = 110, k = 0.178, t0 = 0.001)
vc <- matrix(1, ncol = 3, nrow = 3)
## For K it comes from Mellon 2010
cv1 <- 0.04
cv2 <- 0.005/0.178
cv3 <- 1e-08
l <- vc
l[1, ] <- l[, 1] <- p[1] * cv1
k <- vc
k[, 2] <- k[2, ] <- p[2] * cv2
t <- vc
t[3, ] <- t[, 3] <- p[3] * cv3
mm <- t * k * l
diag(mm) <- diag(mm)^2
mm <- mm * cm
```

We now create a new growth object with the uncertainty:

```
# new growth object
vbObj <- a4aGr(grMod = ~linf * (1 - exp(-k * (t - t0))), grInvMod = ~t0 -
    1/k * log(1 - len/linf), params = FLPar(linf = p["linf"],
    k = p["k"], t0 = p["t0"]), vcov = mm)
vbObj@params

## An object of class "FLPar"
## params
##    linf       k      t0
## 110.000   0.178   0.001
## units:  NA

dim(vbObj@params)

## [1] 3 1
```

We then generate simulations for the catch at age matrix. Actually the best might be to do it for every single variable and to use l2a on the stock. But we just have the length data for catch here, so we keep it to that:

```
Nit <- 10  ## only 10 because otherwise it is long
vbNorm <- mvrnorm(Nit, vbObj)
ages <- predict(vbNorm, len = as.vector(d[, 1]))
# catch at age
c.n1 <- l2a(cth.n, vbNorm)
# problem with older ages, so we trim
```

```
c.n2 <- trim(c.n1, age = 0:6)
c.n2["6"] <- quantSums(trim(c.n1, age = 6:round(max(ages, na.rm = TRUE))))
dim(c.n2)

## [1]  7 15  1  1  1 10

hke2 <- hke
hke2@catch.n <- c.n2
```

We will use the last model fitted:

```
index <- hke.idx2
qmod <- list(~s(age, k = 3))
fmod <- ~I(1/(1 + exp(age))) + te(year, age, k = c(6, 4))
srmod <- ~factor(year)
vmod <- list(~s(year, k = 3) + s(age, k = 3), ~age)
fit <- a4aSCA(stock = hke2, indices = index, fmodel = fmod, srmodel = srmod,
    qmodel = qmod, vmodel = vmod)
stk <- hke2 + fit
```

We plot the results:

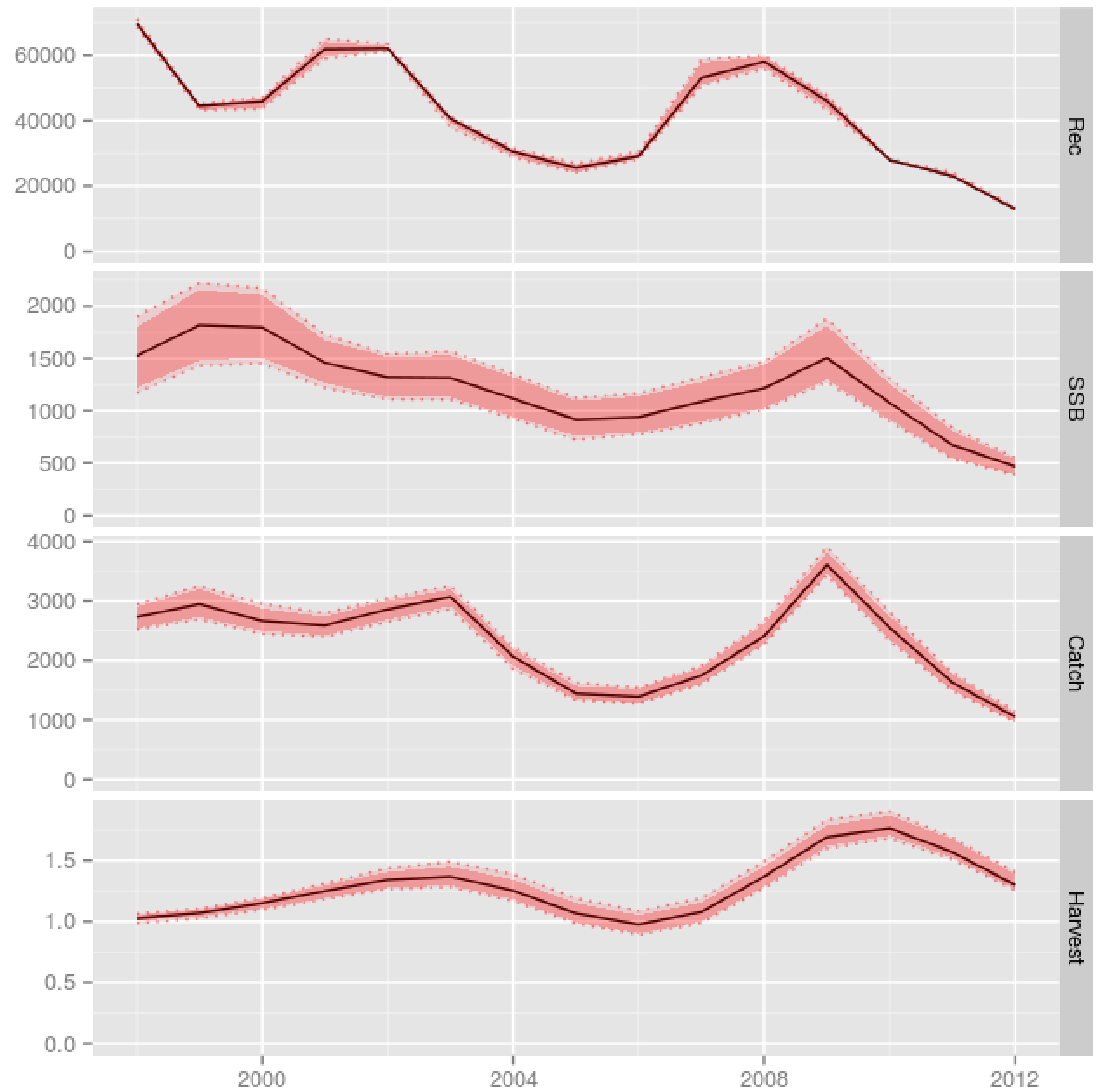


Figure 5.28: Dirty fit

Here we will try to use length-based data to set the natural mortality model, that will afterwards be used to build the M matrix for stock assessment.

```
# Make an empty cor matrix
cm <- diag(c(1, 1))
# k and linf are negatively correlated while t0 is
# independent
cm[1, 2] <- cm[2, 1] <- -0.5
p <- c(linf = 110, k = 0.178)
vc <- matrix(1, ncol = 2, nrow = 2)
## For K it comes from Mellon 2010
cv1 <- 0.04
cv2 <- 0.005/0.178
l <- vc
```

```
l[1, ] <- l[, 1] <- p[1] * cv1
k <- vc
k[, 2] <- k[2, ] <- p[2] * cv2
mm <- k * l
diag(mm) <- diag(mm)^2
mm <- mm * cm
all.equal(cm, cov2cor(mm))

## [1] TRUE

## we make the object with the variance covariance
shapeGis <- FLModelSim(model = ~k * (linf/len)^1.5, params = FLPar(linf = 110,
    k = 0.178), vcov = mm)
m <- a4aM(shape = shapeGis)
# one needs to set the range for the object which will be
# used by the m method
range(m) <- c(0, 110, NA, 2000, 2003, 15, 30)
range(m)

##       min       max plusgroup   minyear   maxyear   minmbar   maxmbar
##         0       110        NA      2000      2003        15        30

# now simulate
Nit <- 100
msim <- mvrnorm(Nit, m)
rngyear(msim) <- c(2000, 2003)
# and compute natural mortality
m.sim <- m(msim)
# note that this one is by length, before adding to the stock
# object it must be transformed into ages
```

5.4 Short term forecast

```
# future recruitment
futureRec <- mean(rec(hke.ma)[, ac(2010:2012)])
hke.srr <- list(model = "mean", params = FLPar(futureRec))
hke.sr <- as.FLSR(hke.ma, model = geomean)
hke.sr <- fmle(hke.sr, control = list(trace = 0))

hke.stf <- stf(hke.ma, nyears = 3)
fwd.ctrl <- fwdControl(data.frame(year = 2013:2015, val = 1.16,
    quantity = "f"))
hke.fwd <- fwd(hke.stf, ctrl = fwd.ctrl, sr = hke.sr, maxF = 10)

# CHOICE FOR Fstatusquo
Fstatusquo <- mean(fbar(hke.ma)[, ac(2010:2012)])
```

```
# OTHER SCENARIOS
F01 <- 0.148  #as.numeric(refpts(hkebrp)['f0.1','harvest'])
Fsc <- cbind(rep(Fstatusquo, length(seq(0, 2, by = 0.1))), seq(0,
    2, by = 0.1) * Fstatusquo, seq(0, 2, by = 0.1) * Fstatusquo)
Fsc <- rbind(c(Fstatusquo, F01, F01), Fsc)
Ffactor <- c(NA, seq(0, 2, by = 0.1))
Results.sc <- matrix(NA, length(Ffactor), 10)
hke.stf <- stf(hke.ma, nyears = 3)
for (i in 1:length(Ffactor)) {
    fwd.ctrl <- fwdControl(data.frame(year = 2013:2015, val = Fsc[i,
        ], quantity = "f"))
    hke.fwd <- fwd(hke.stf, ctrl = fwd.ctrl, sr = hke.srr, maxF = 10)
    ### BUILD TABLE
    Results.sc[i, 1] <- Ffactor[i]
    Results.sc[i, 2] <- median(fbar(hke.fwd)[, ac(2015)])
    Results.sc[i, 3] <- median(catch(hke.fwd)[, ac(2012)])
    Results.sc[i, 4] <- median(catch(hke.fwd)[, ac(2013)])
    Results.sc[i, 5] <- median(catch(hke.fwd)[, ac(2014)])
    Results.sc[i, 6] <- median(catch(hke.fwd)[, ac(2015)])
    Results.sc[i, 7] <- median(ssb(hke.fwd)[, ac(2014)])
    Results.sc[i, 8] <- median(ssb(hke.fwd)[, ac(2015)])
    Results.sc[i, 9] <- (median(ssb(hke.fwd)[, ac(2015)]) - median(ssb(hke.fwd)[,
        ac(2014)]))/median(ssb(hke.fwd)[, ac(2014)]) * 100
    Results.sc[i, 10] <- (median(catch(hke.fwd)[, ac(2014)]) -
        median(catch(hke.fwd)[, ac(2012)]))/median(catch(hke.fwd)[,
        ac(2012)]) * 100
}
# GIVE NAMES TO COLUMNS
colnames(Results.sc) <- c("Ffactor", "Fbar", "Catch_2012", "Catch_2013",
    "Catch_2014", "Catch_2015", "SSB_2014", "SSB_2015", "Change_SSB_2014-2015(%)",
    "Change_Catch_2012-2014(%)")
# VISUALIZE DATABASE
Results.sc
```

```
##       Ffactor   Fbar Catch_2012 Catch_2013 Catch_2014 Catch_2015 SSB_2014
##  [1,]      NA 0.1480       1197      947.2      162.1      483.4      348
##  [2,]     0.0 0.0000       1197      947.2        0.0        0.0      348
##  [3,]     0.1 0.1642       1197      947.2      178.6      524.4      348
##  [4,]     0.2 0.3285       1197      947.2      330.5      841.9      348
##  [5,]     0.3 0.4927       1197      947.2      460.1     1024.2      348
##  [6,]     0.4 0.6570       1197      947.2      571.6     1121.2      348
##  [7,]     0.5 0.8212       1197      947.2      668.0     1166.8      348
##  [8,]     0.6 0.9854       1197      947.2      752.1     1179.4      348
##  [9,]     0.7 1.1497       1197      947.2      825.7     1171.6      348
## [10,]     0.8 1.3139       1197      947.2      890.7     1152.6      348
## [11,]     0.9 1.4781       1197      947.2      946.2     1127.6      348
## [12,]     1.0 1.6424       1197      947.2      994.8     1100.3      348
## [13,]     1.1 1.8066       1197      947.2     1039.4     1072.5      348
## [14,]     1.2 1.9709       1197      947.2     1080.1     1045.4      348
## [15,]     1.3 2.1351       1197      947.2     1115.9     1019.9      348
```

```
## [16,]     1.4 2.2993       1197      947.2     1148.7      995.4       348
## [17,]     1.5 2.4636       1197      947.2     1178.6      973.0       348
## [18,]     1.6 2.6278       1197      947.2     1205.7      951.4       348
## [19,]     1.7 2.7921       1197      947.2     1230.1      931.8       348
## [20,]     1.8 2.9563       1197      947.2     1252.9      913.8       348
## [21,]     1.9 3.1205       1197      947.2     1274.6      897.4       348
## [22,]     2.0 3.2848       1197      947.2     1294.7      882.6       348
##        SSB_2015 Change_SSB_2014-2015(%) Change_Catch_2012-2014(%)
##  [1,]   1551.5                  345.810                  -86.4518
##  [2,]   1799.2                  416.961                 -100.0000
##  [3,]   1525.7                  338.384                  -85.0807
##  [4,]   1298.4                  273.075                  -72.3860
##  [5,]   1109.3                  218.744                  -61.5536
##  [6,]    951.1                  173.290                  -52.2373
##  [7,]    817.9                  134.999                  -44.1834
##  [8,]    706.3                  102.943                  -37.1544
##  [9,]    613.3                   76.212                  -31.0074
## [10,]    534.5                   53.571                  -25.5790
## [11,]    466.9                   34.144                  -20.9409
## [12,]    410.1                   17.823                  -16.8741
## [13,]    360.1                    3.471                  -13.1522
## [14,]    319.0                   -8.348                   -9.7478
## [15,]    283.4                  -18.574                   -6.7585
## [16,]    252.8                  -27.363                   -4.0179
## [17,]    226.6                  -34.878                   -1.5237
## [18,]    203.8                  -41.443                    0.7436
## [19,]    184.7                  -46.941                    2.7799
## [20,]    167.7                  -51.812                    4.6847
## [21,]    153.4                  -55.932                    6.4990
## [22,]    140.7                  -59.569                    8.1833
```

6 Hake in the Southern Adriatic

Dimitrios Damalas

Our goal was to attempt a replication of last years stock assessment results as conducted during 15-19 July 2013 at the 13-09 EWG. Input data were identical to the ones used therein.

6.1 Replicating accepted assessments

6.1.1 Load data from recent assessments

```
load("data/HKE18.RData")

# keep the stocks in temp objs to avoid possible overwriting
hketemp <- hke.stk
hketemp2 <- hke.stk_2
```

The final approach selected by the EWG 13-09 was an XSA named "shrinkage 2" (Figures 6.1 and 6.2). Data from official Data Call 2013 - index from MEDITS survey. The details of the final XSA approach below and on pages 284-292 of the STECF 13-22 Report.

```
FLXSA.control.hke_2 <- FLXSA.control(x = NULL, tol = 1e-09, maxit = 30,
    min.nse = 0.3, fse = 2, rage = 0, qage = 4, shk.n = TRUE,
    shk.f = TRUE, shk.yrs = 2, shk.ages = 2, window = 100, tsrange = 20,
    tspower = 3, vpa = FALSE)
hke.xsa_2 <- FLXSA(hke.stk, hke.idx, FLXSA.control.hke_2)
```

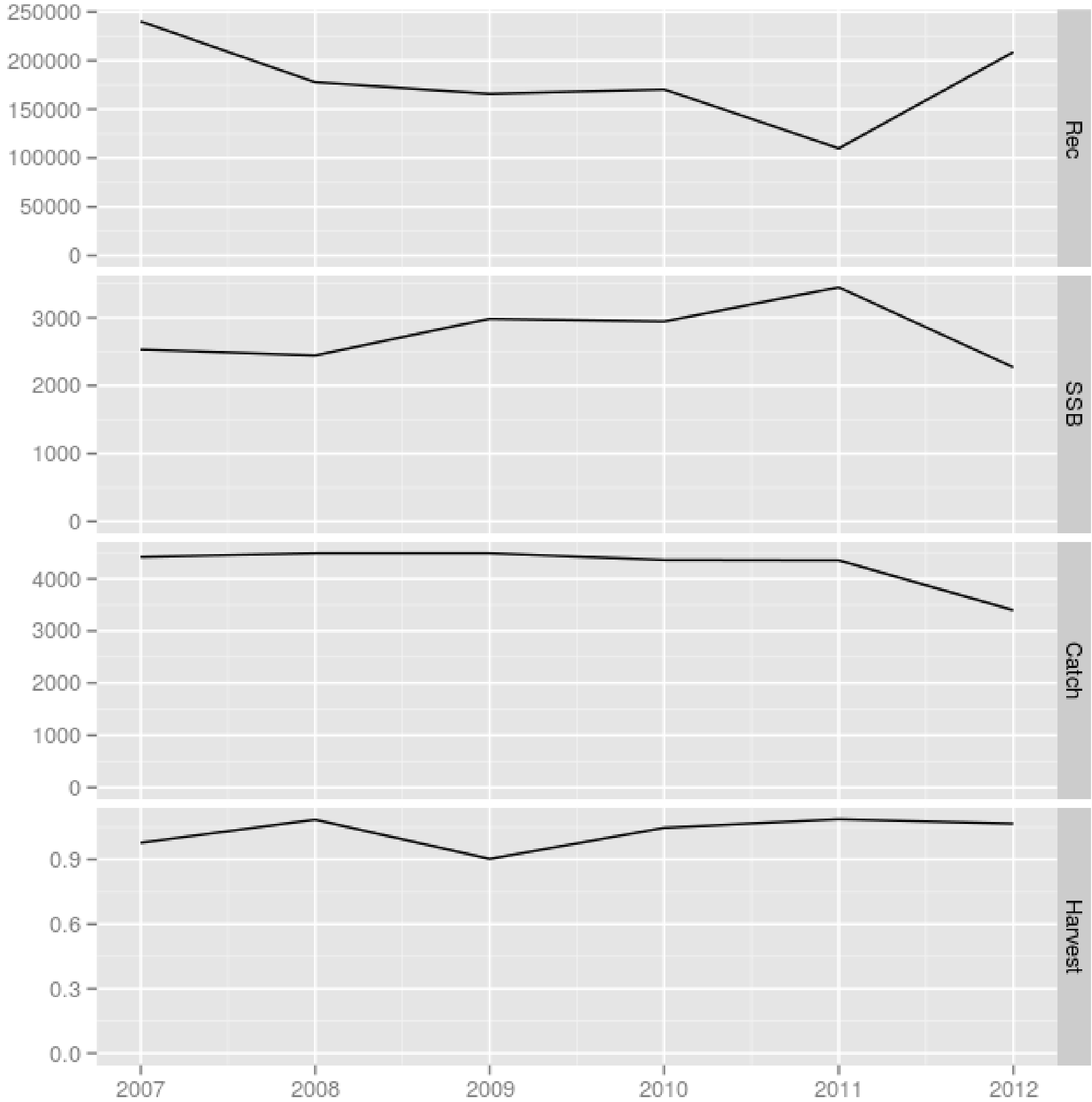


Figure 6.1: XSA results

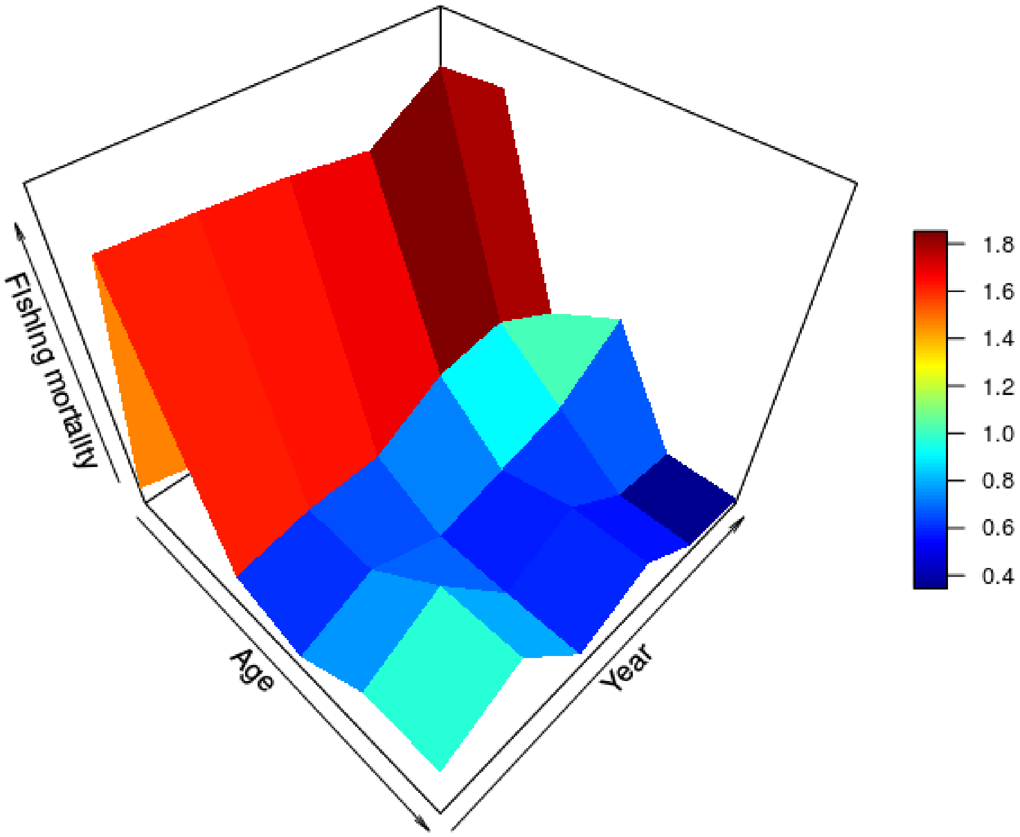


Figure 6.2: XSA estimates of fishing mortality

6.1.2 Building a4a models on the same data used in recent assessessments

Default model

```
fit0 <- sca(stock = hke.stk, indices = hke.idx)
hke.stk.a4a <- hke.stk + fit0
plot(hke.stk.a4a)
```

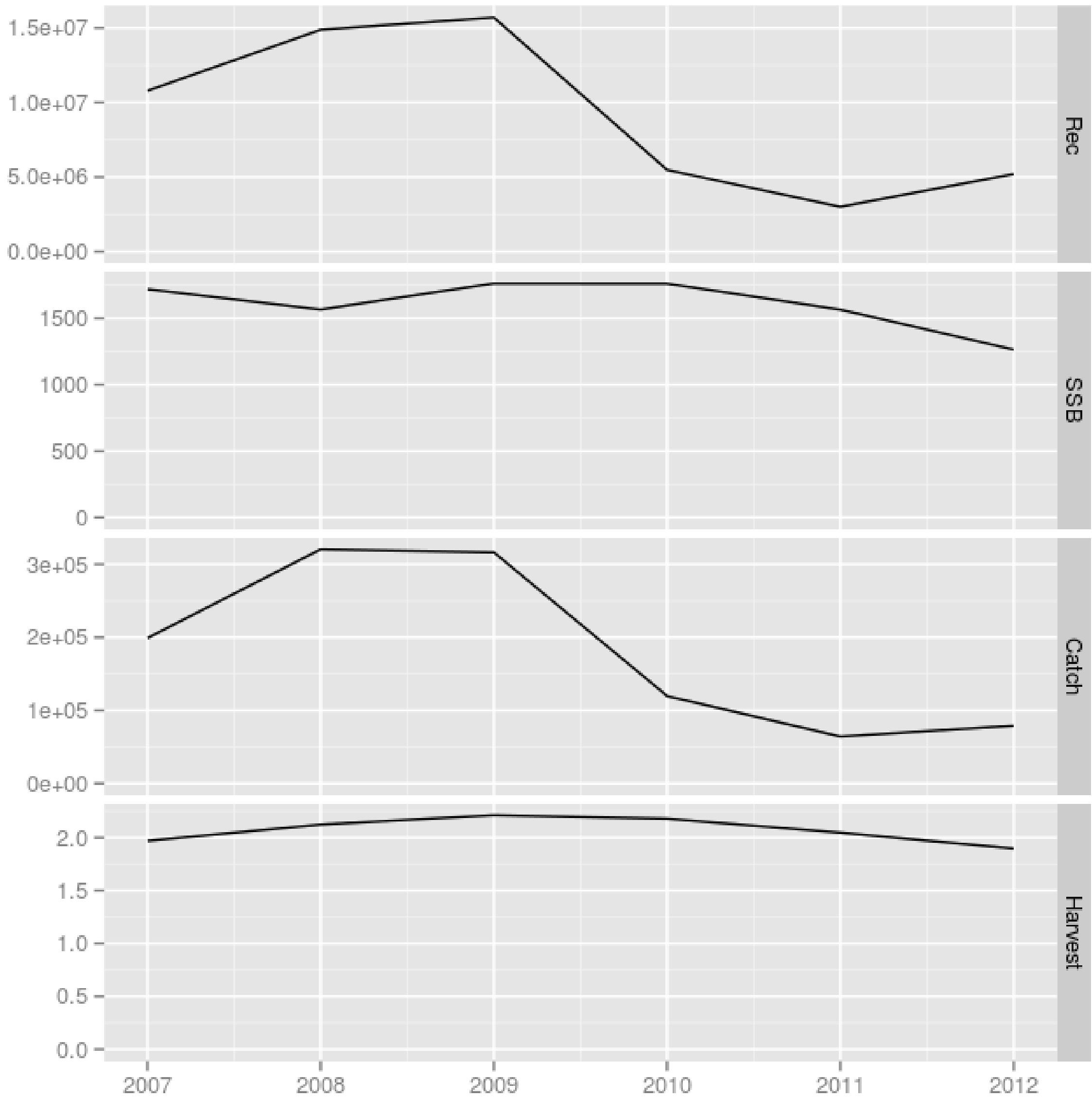


Figure 6.3: Assessment summary - default model

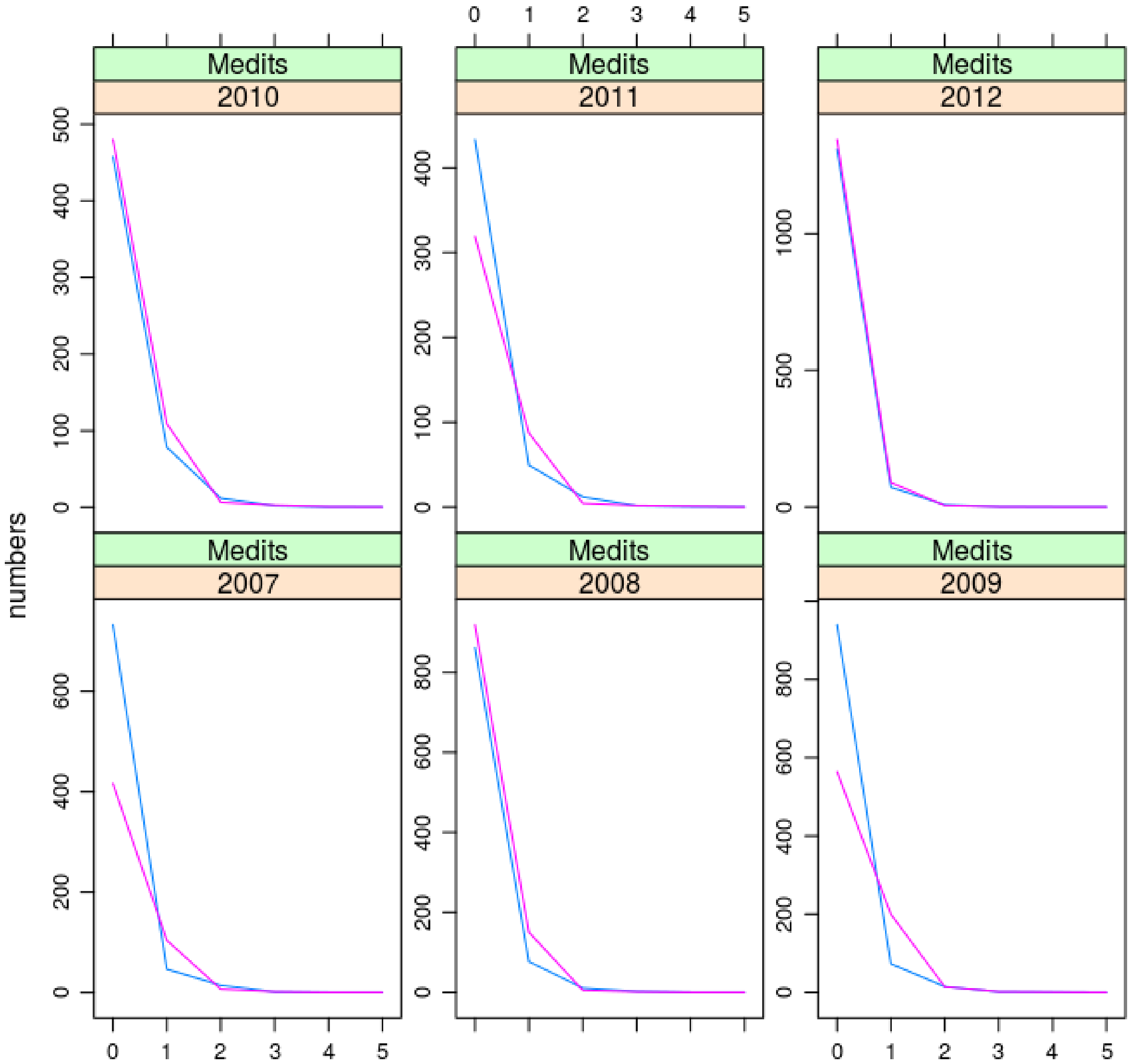


Figure 6.4: Plot fitted against observed for survey

Plotting fitted against observed catch at age , as well as against the survey index, it is obvious that the fits deviate largely from the observations (fitted = blue, observed = magenta).

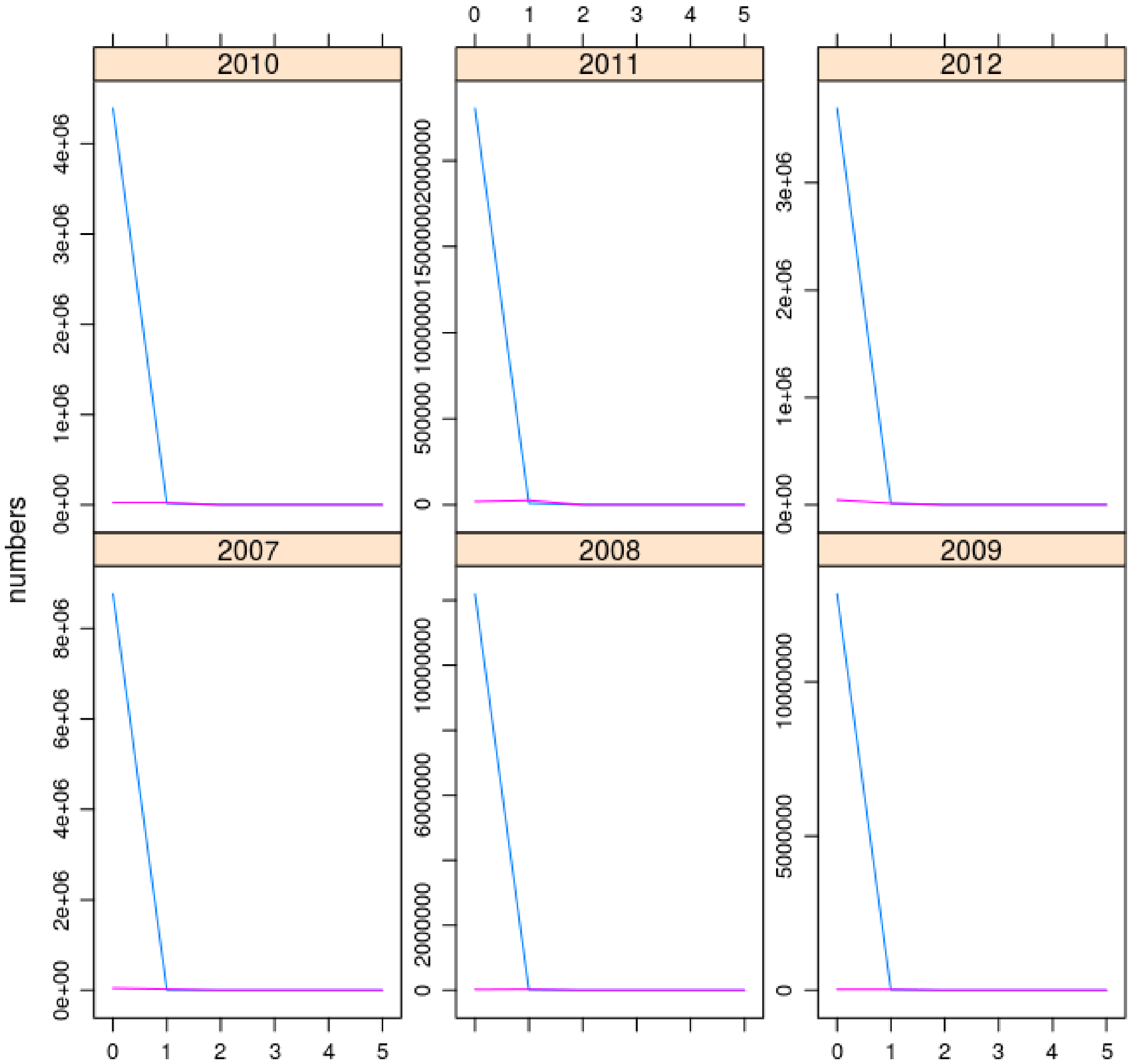


Figure 6.5: Plot fitted agains observed for catch at age

```
res <- residuals(fit0, hke.stk, hke.idx)
```

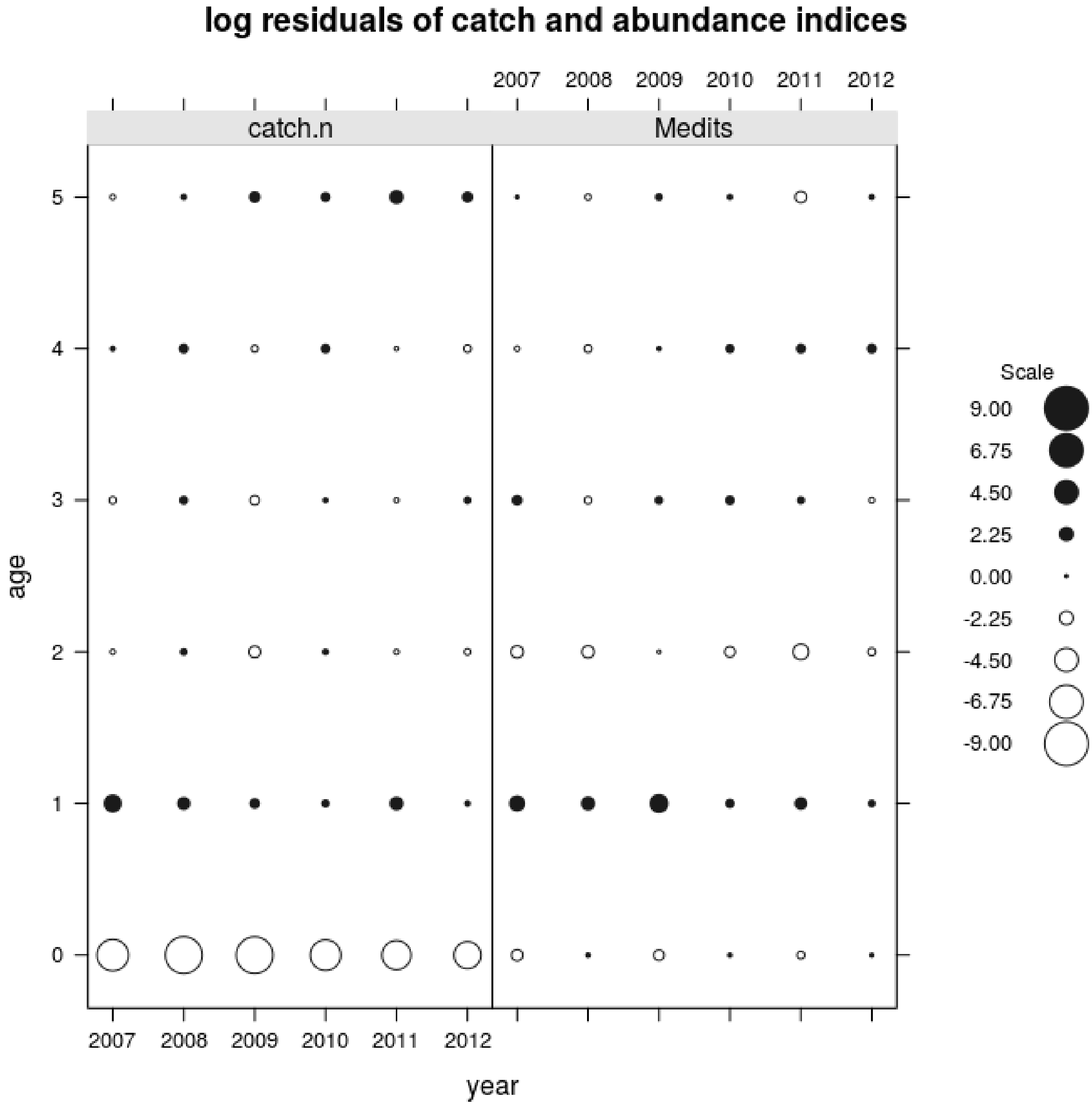


Figure 6.6: Residuals by age and year

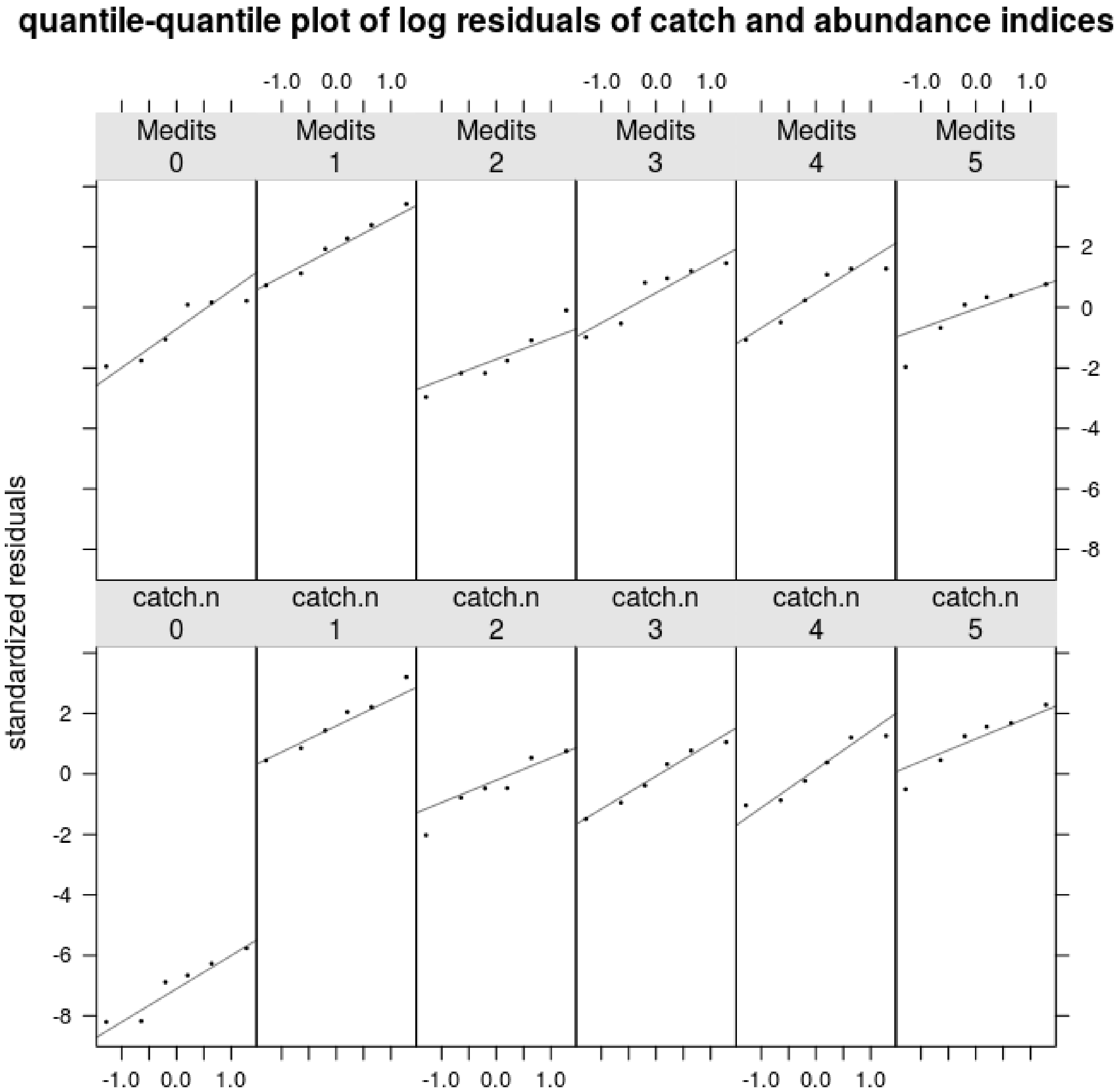


Figure 6.7: Quantile-quantile plot of residuals

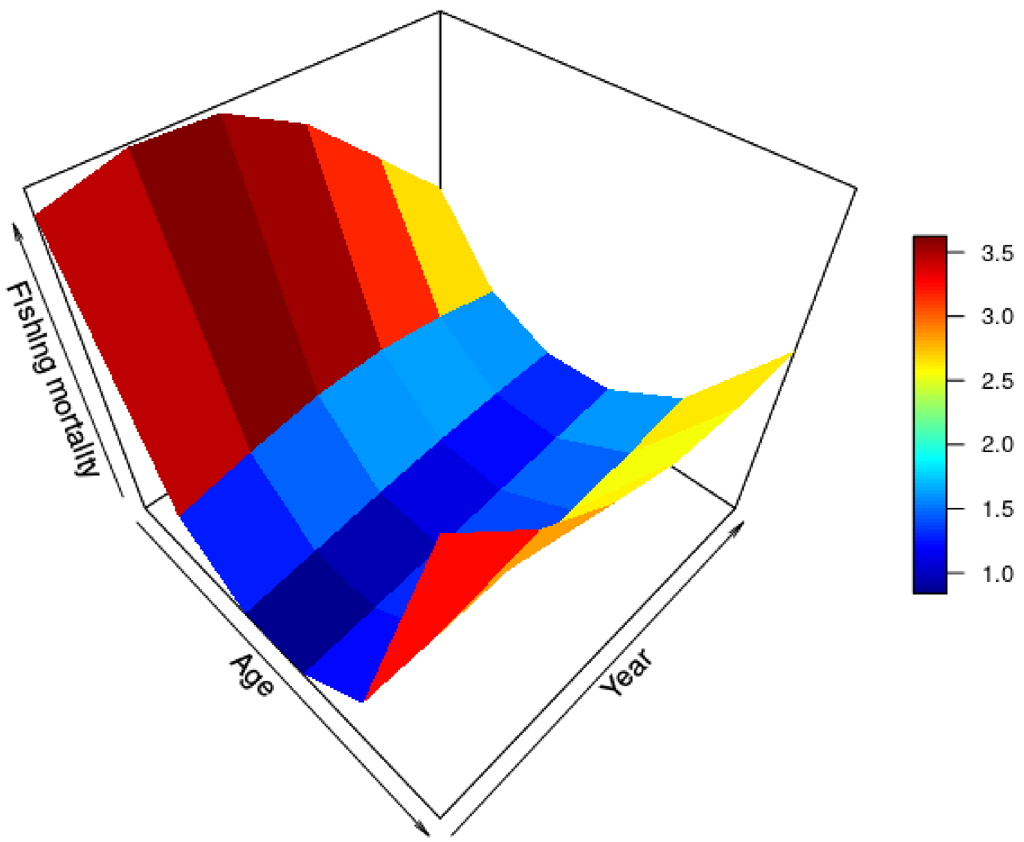


Figure 6.8: Fishing mortality

Due to the poor fit of the selected model, F values are unrealistically high. A better approach should be investigated.

Separable model

Trying to improve the fit, we now investigate a quite simple model, treating 'age' and 'year' as categorical factors. If this fails then probably non-linearities are inherent in the data set and the use of smoothers is required to deal with them through a more complex model.

```
qmod1 <- list(~factor(age))
fmod1 <- ~factor(age) + factor(year)
srmod1 <- ~factor(year)
```

```
hke.stk <- hketemp
fit1 <- sca(stock = hke.stk, indices = hke.idx, fmodel = fmod1,
    qmodel = qmod1, srmodel = srmod1)
hke.stk.a4a.1 <- hke.stk + fit1
plot(hke.stk.a4a.1)
```

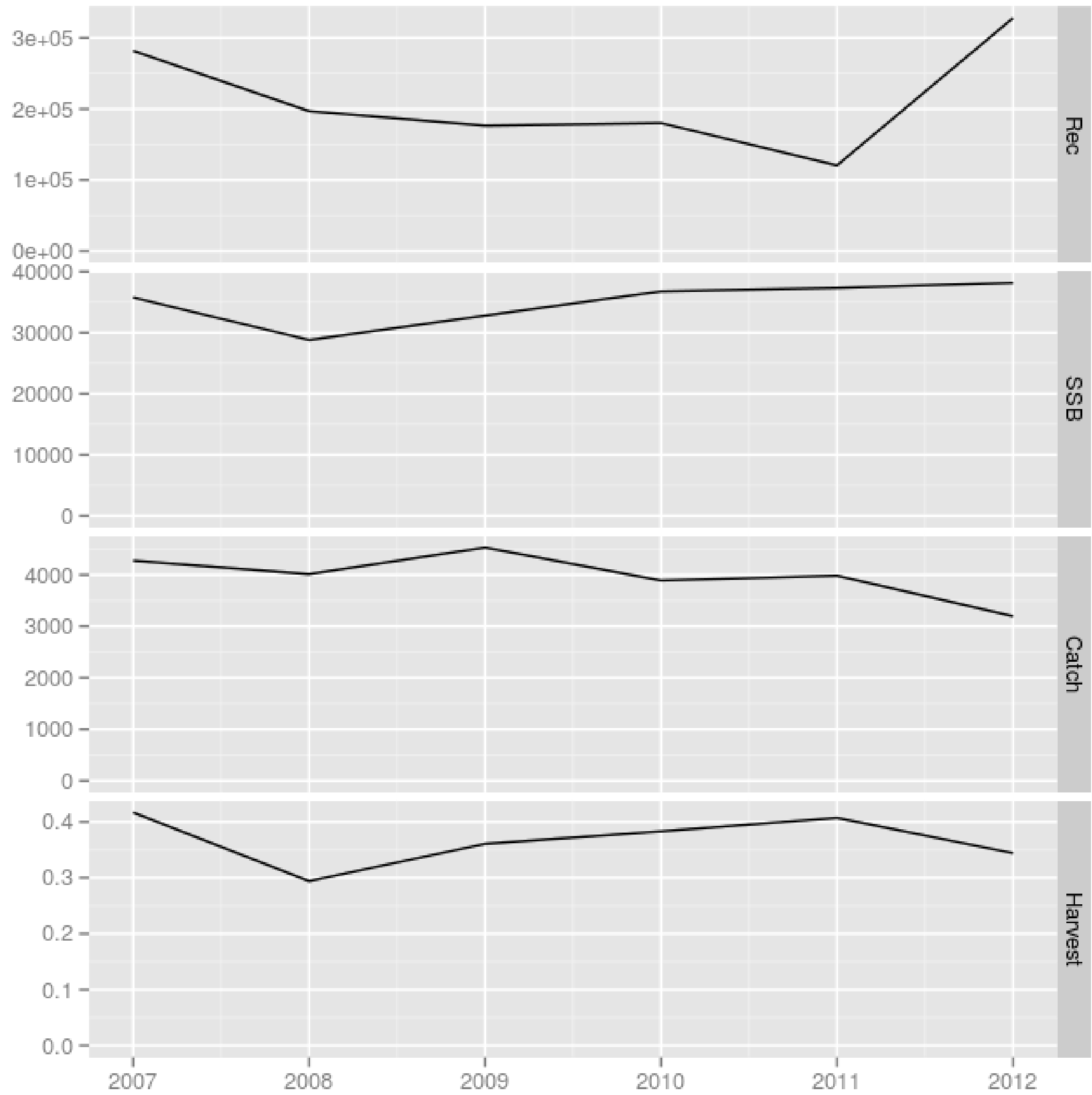


Figure 6.9: Assessment summary

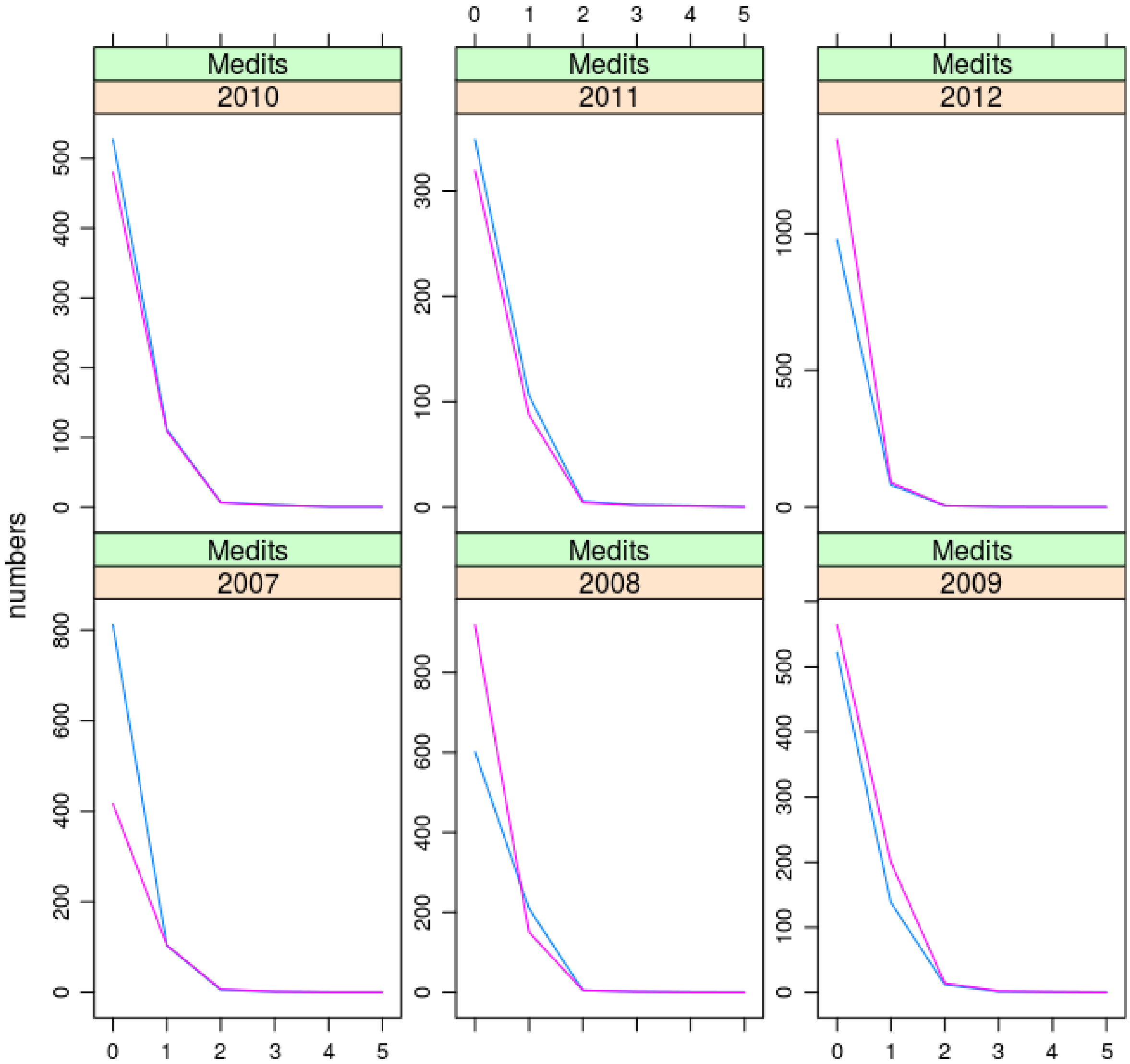


Figure 6.10: Plot fitted against observed for survey

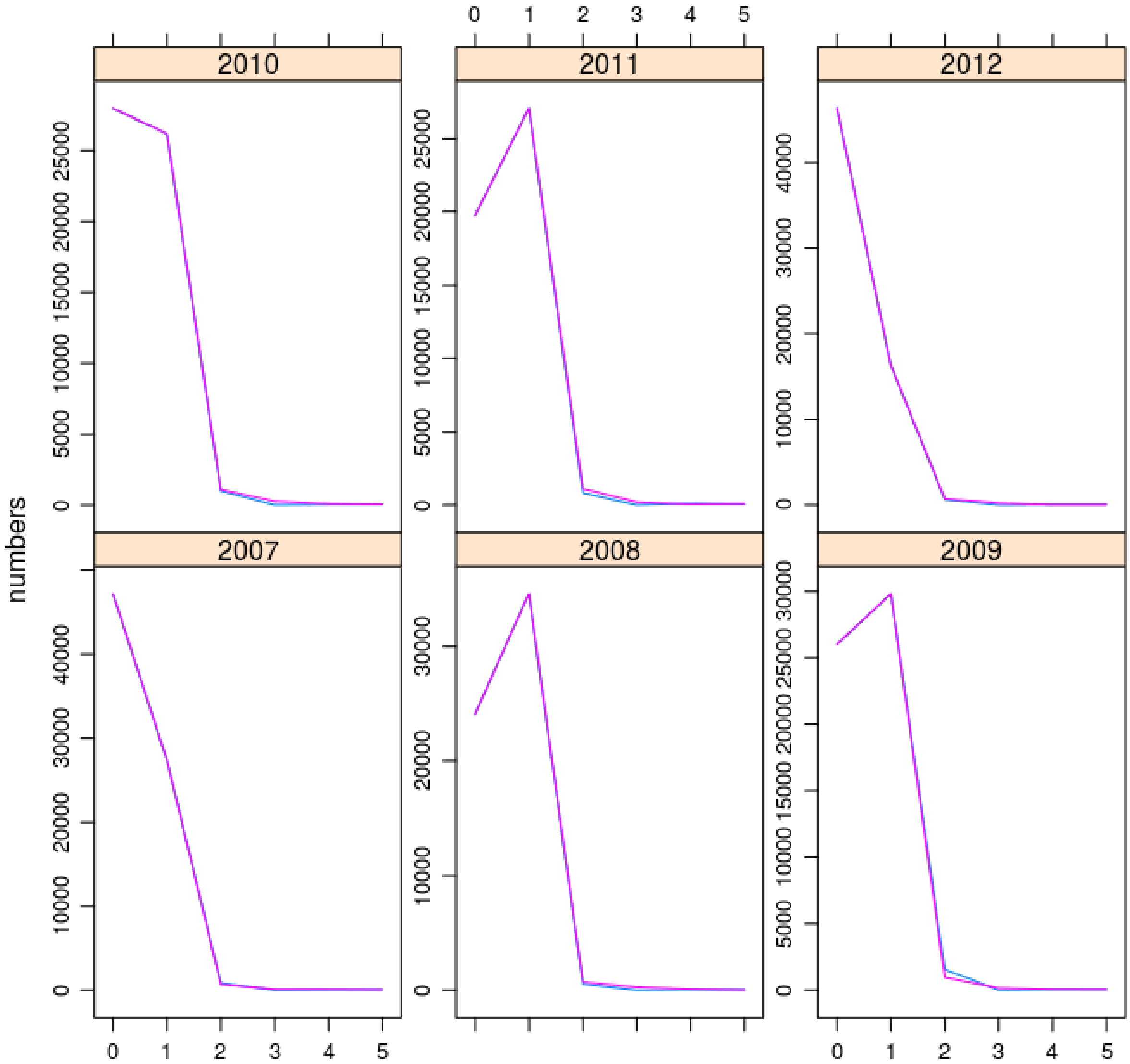


Figure 6.11: Plot fitted agains observed for catch at age

Fits are much closer to the observed (compared to the simple model), however the survey data are still not fitted adequately (fitted = blue, observed = magenta).

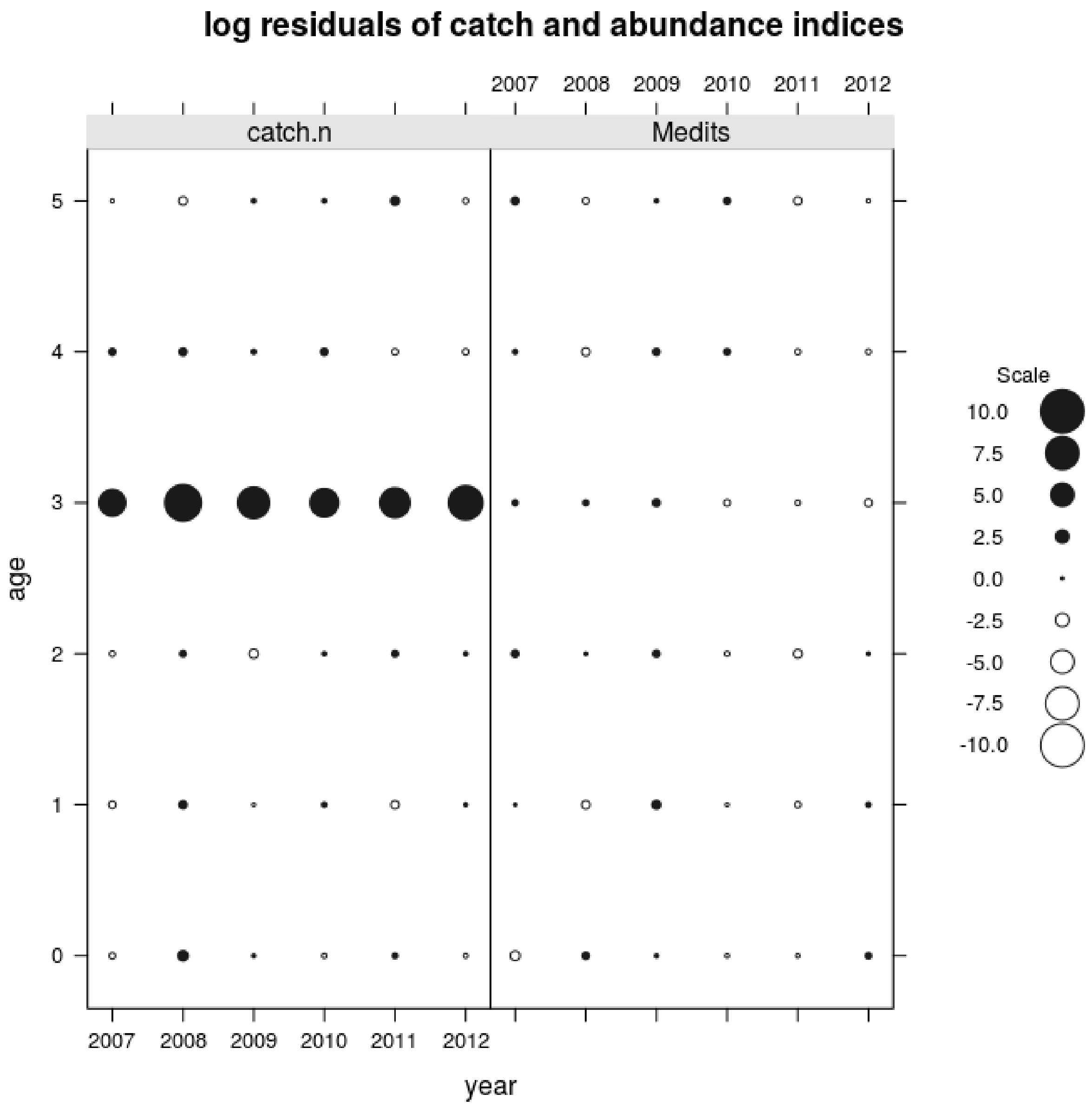


Figure 6.12: Residuals by age and year

Through the residuals plot above, obviously the predictions are quite far from the observed for specific age classes (age 3) and will need more flexibility in either the harvest modelling (fmod) or the catchability (qmod).

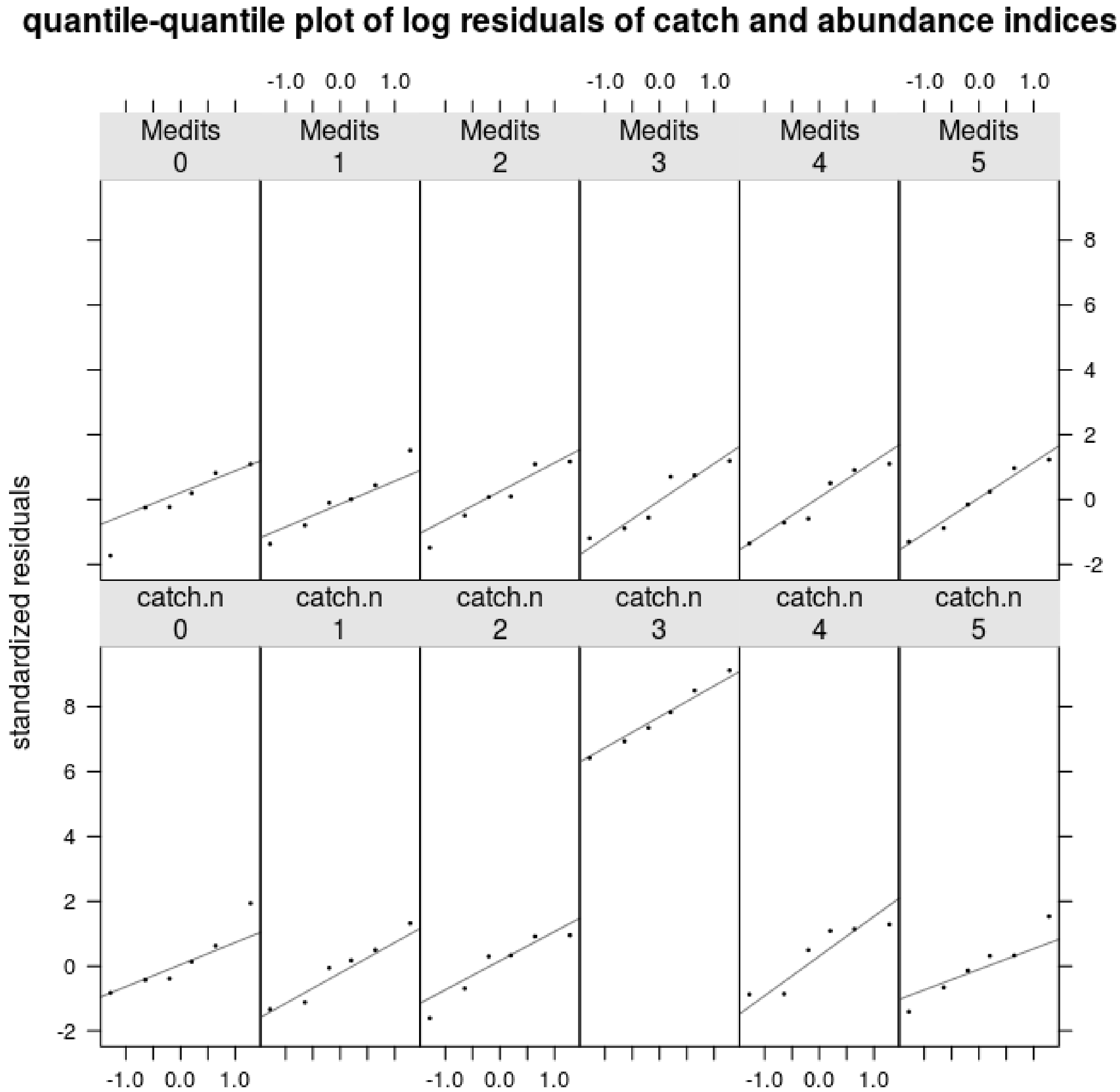


Figure 6.13: Quantile-quantile plot of residuals

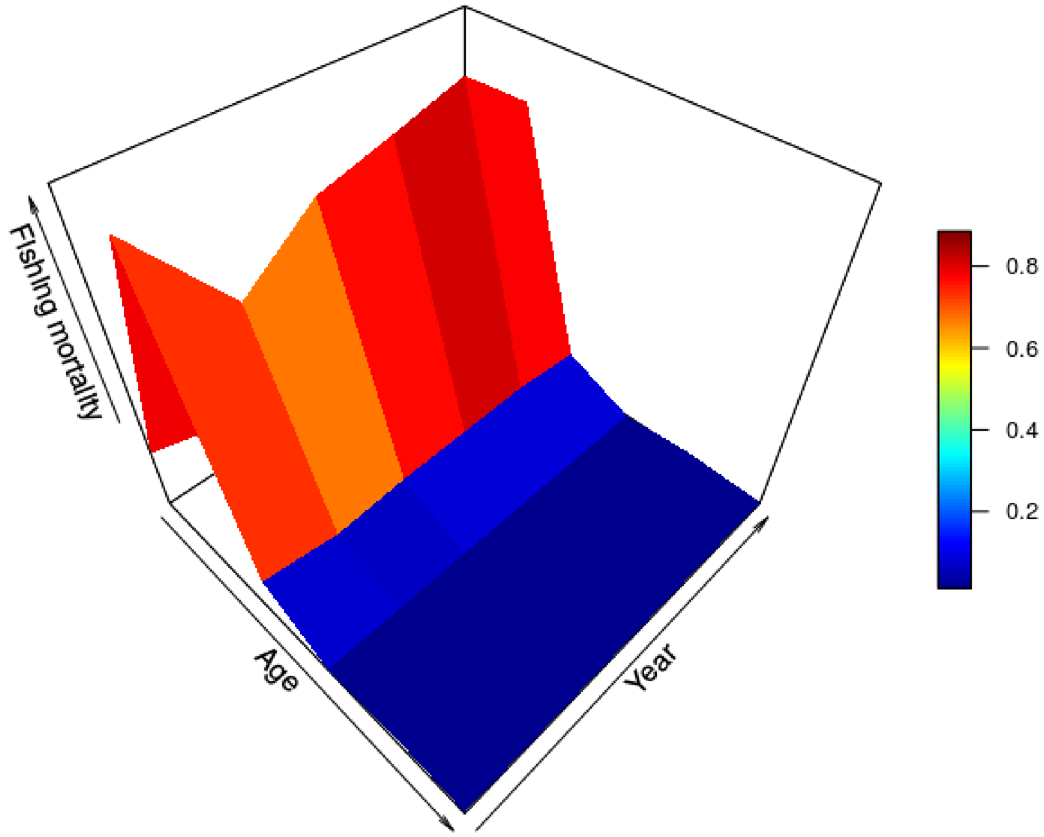


Figure 6.14: Fishing mortality

In this case now, F values are unrealistically low (at least for a Mediterranean stock). It seems that the problem is mainly in the modelling of fishing mortality (fmod).

Separable model with smoothers

We now investigate a more complex approach, treating 'age' and 'year' through smoothers. Since we already have some indications that the problem resides mainly in fishing mortality, we model F through smoothers with a maximum of k=6 nodes, when we have 7 years of data in total.

```
qmod2 <- list(~s(age, k = 3))
fmod2 <- ~s(age, k = 6) + s(year, k = 6)
srmod2 <- ~s(year, k = 2)
```

```
hke.stk <- hketemp
fit2 <- sca(stock = hke.stk, indices = hke.idx, fmodel = fmod2,
    qmodel = qmod2, srmodel = srmod2)
hke.stk.a4a.2 <- hke.stk + fit2
plot(hke.stk.a4a.2)
```

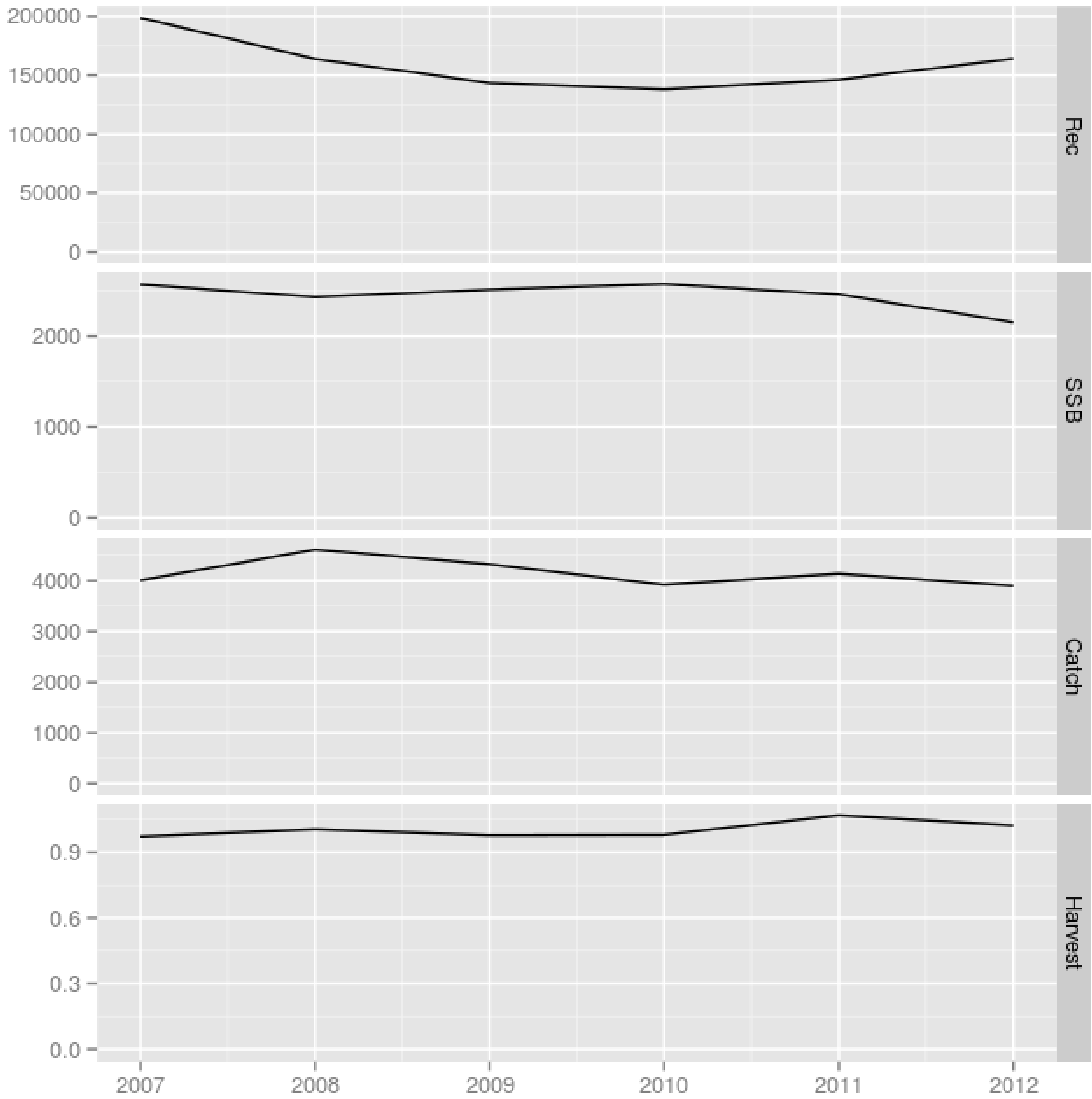


Figure 6.15: Assessment summary

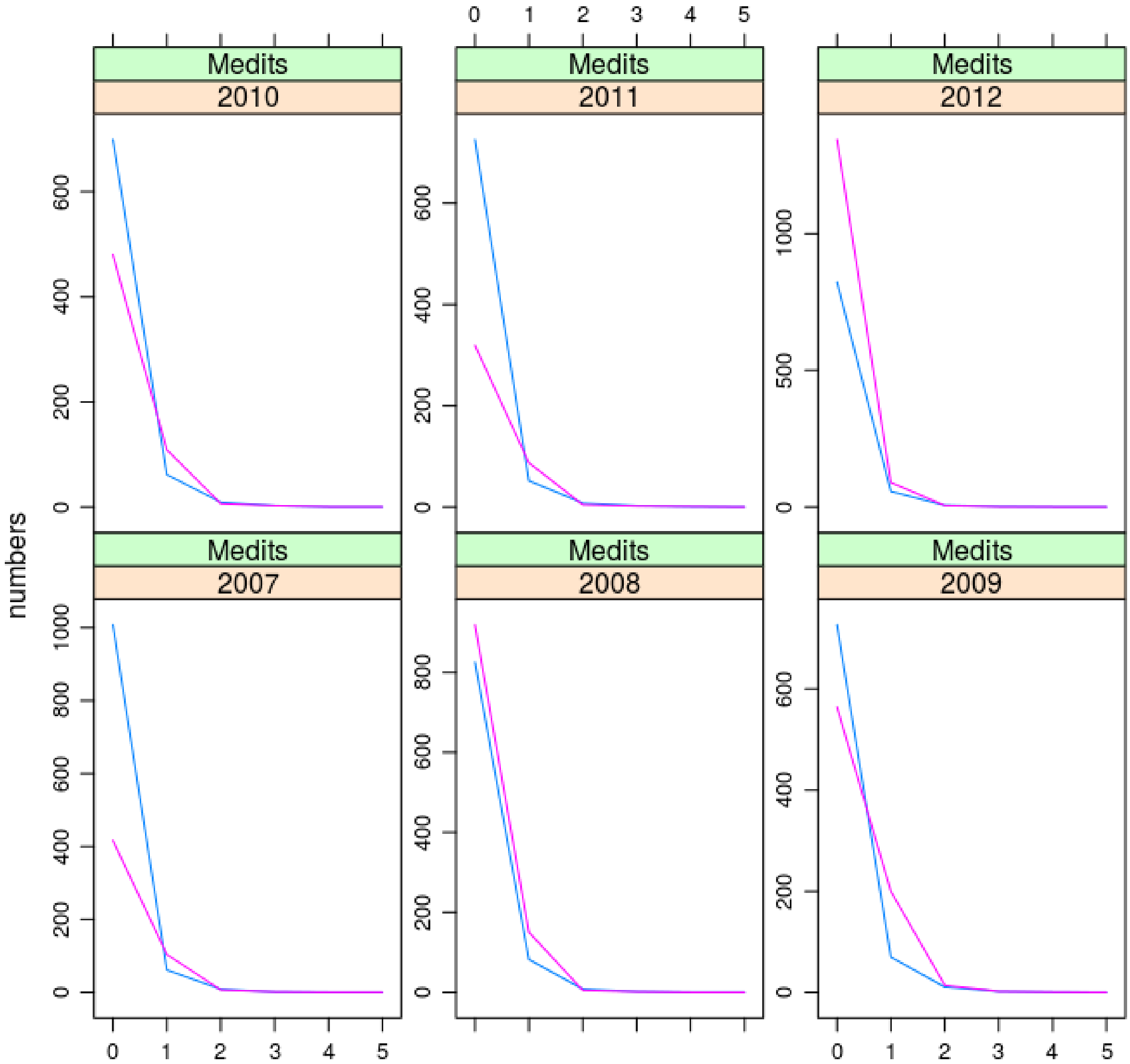


Figure 6.16: Plot fitted against observed for survey

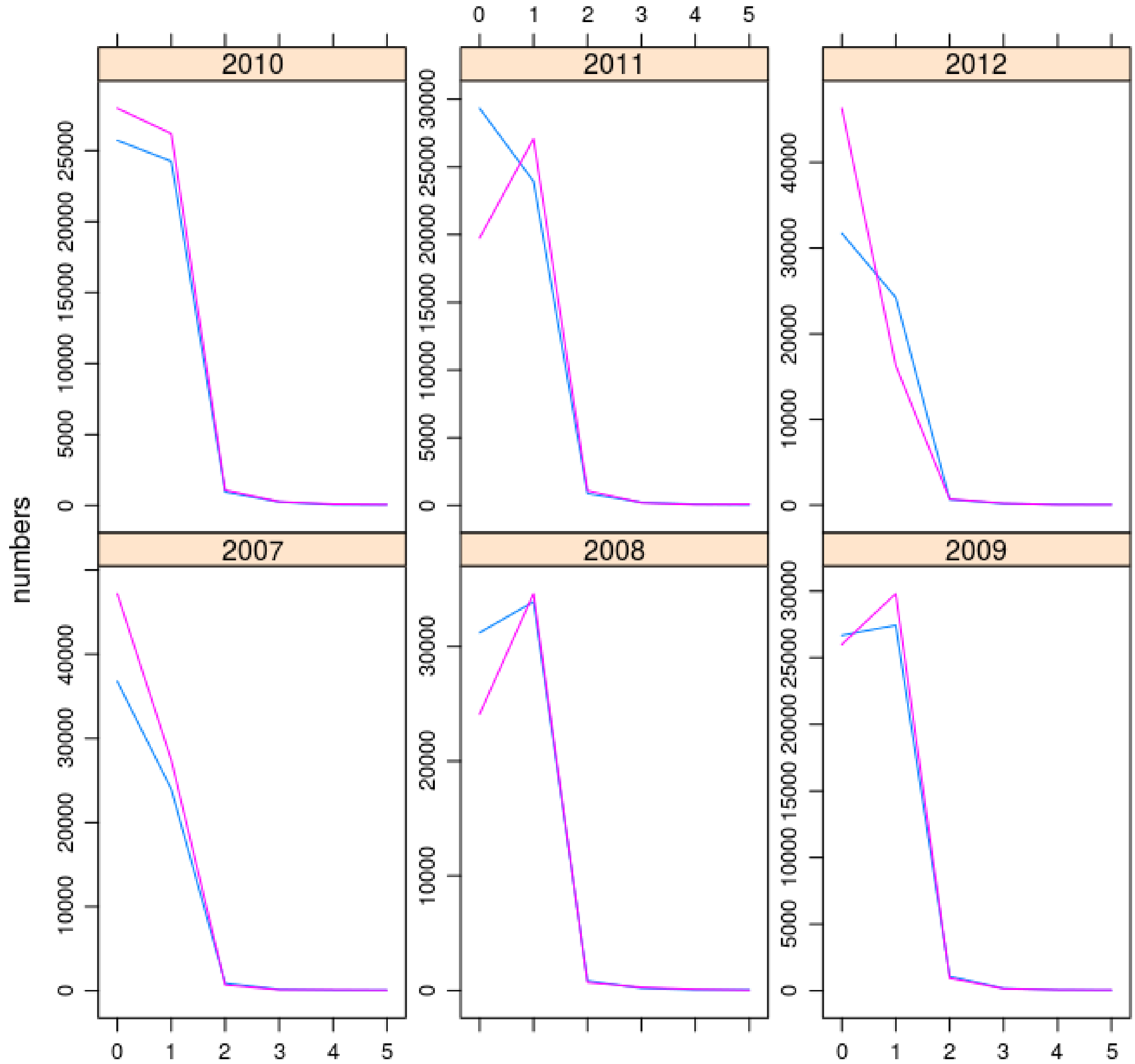


Figure 6.17: Plot fitted agains observed for catch at age

Fits do net get any better than the previous simple separable model (fitted = blue, observed = magenta).

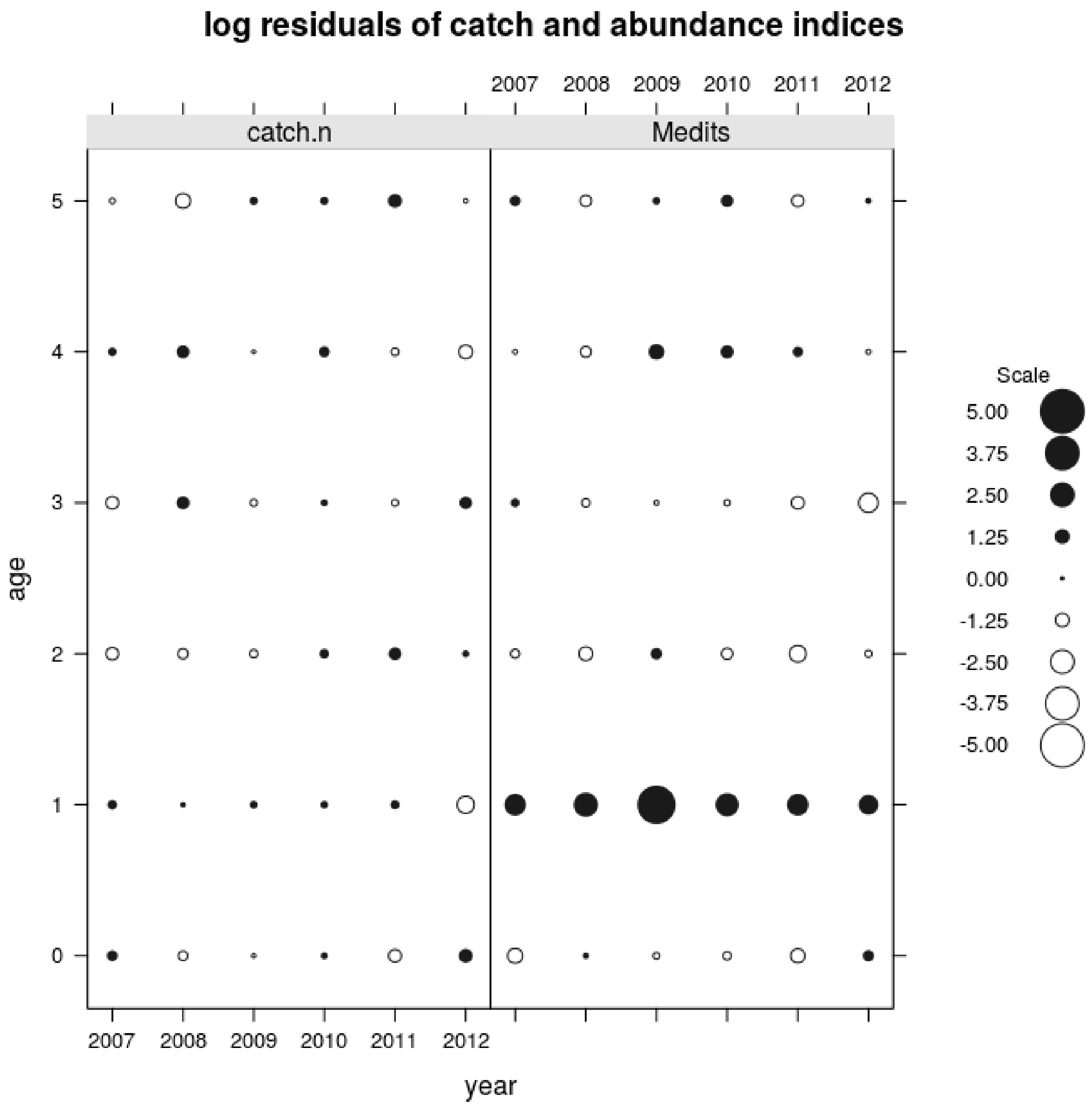


Figure 6.18: Residuals by age and year

Now the problem seems to be shifted to age 1, which is an age class facing the highest pressure and consists a great part of the catches.

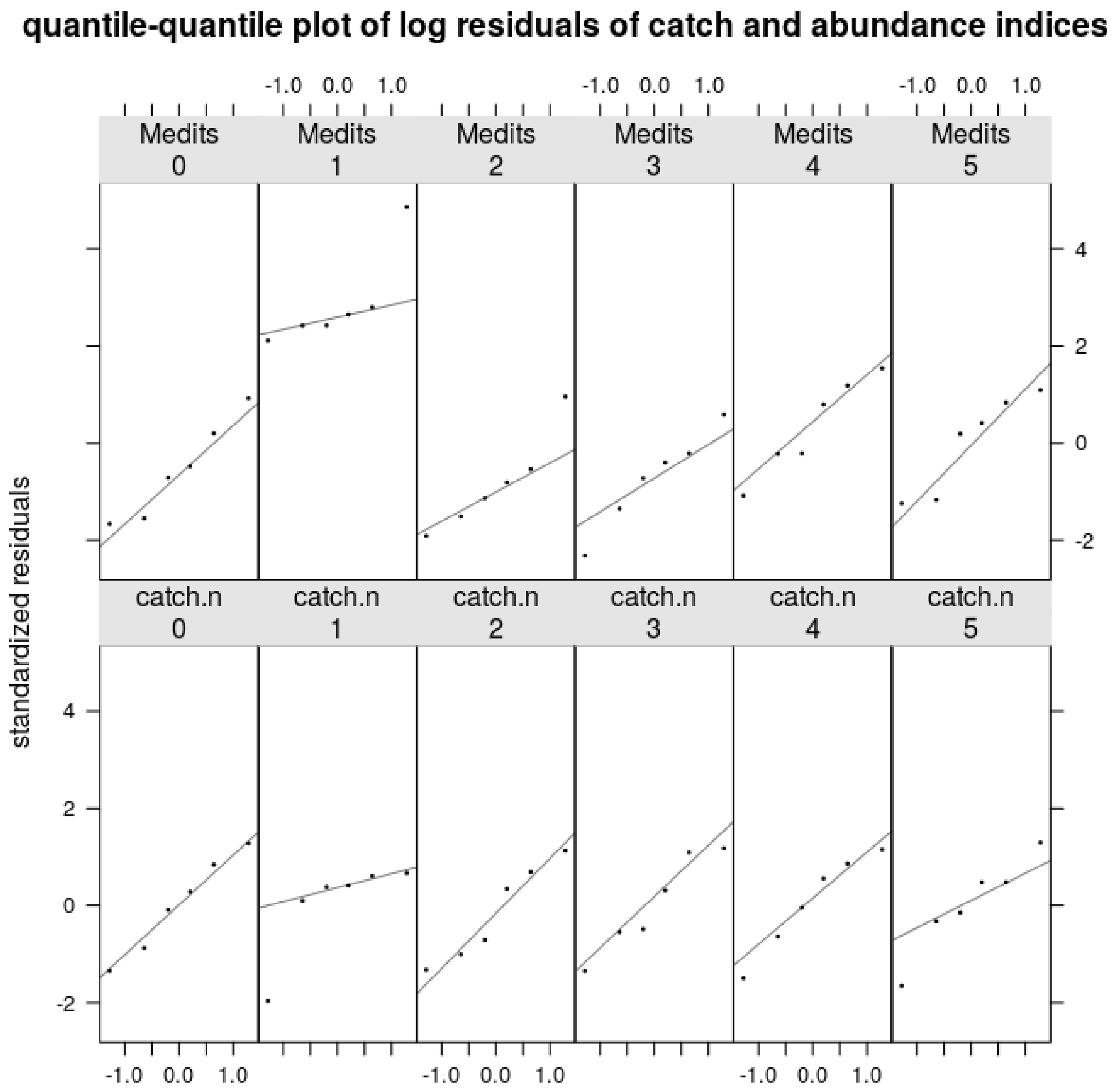


Figure 6.19: Quantile-quantile plot of residuals

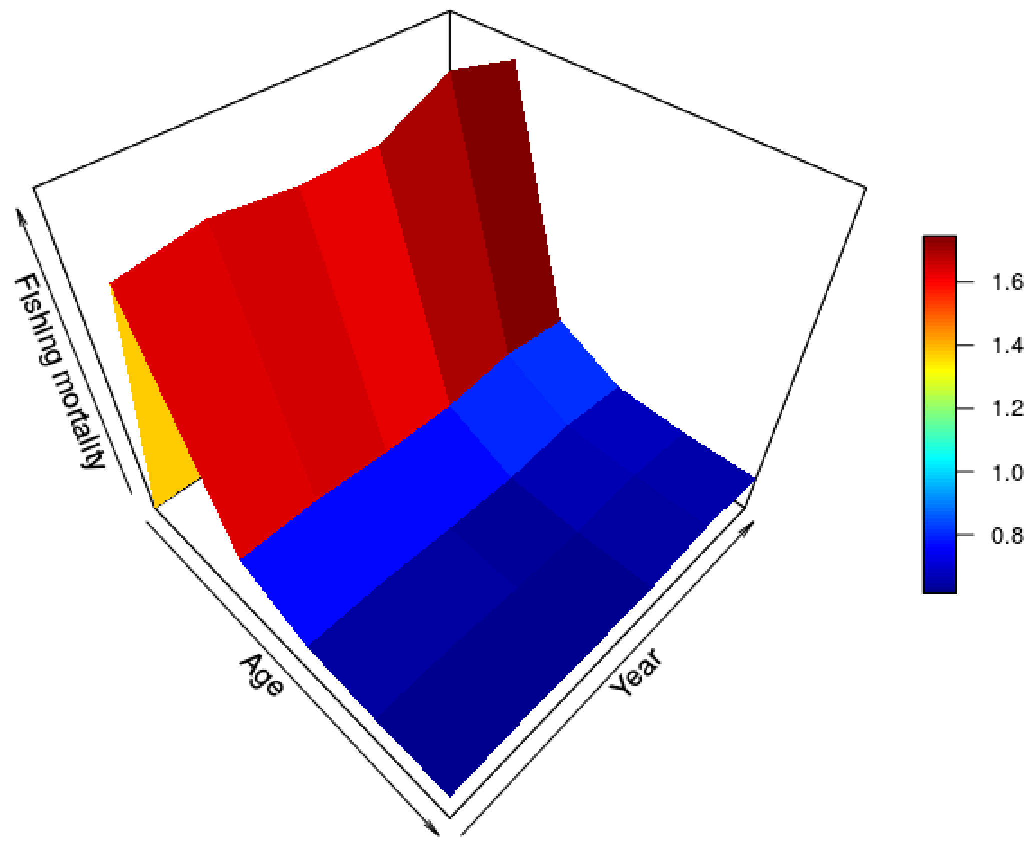


Figure 6.20: Fishing mortality

F values (and the age-year shape) are very similar to the ones from the last years XSA assessment.

Separable model with smoothers 2

```
qmod3 <- list(~s(age, k = 6))
fmod3 <- ~s(age, k = 6) + s(year, k = 6)
srmod3 <- ~s(year, k = 6)
hke.stk <- hketemp
fit3 <- sca(stock = hke.stk, indices = hke.idx, fmodel = fmod3,
    qmodel = qmod3, srmodel = srmod3)
hke.stk.a4a.3 <- hke.stk + fit3
plot(hke.stk.a4a.3)
```

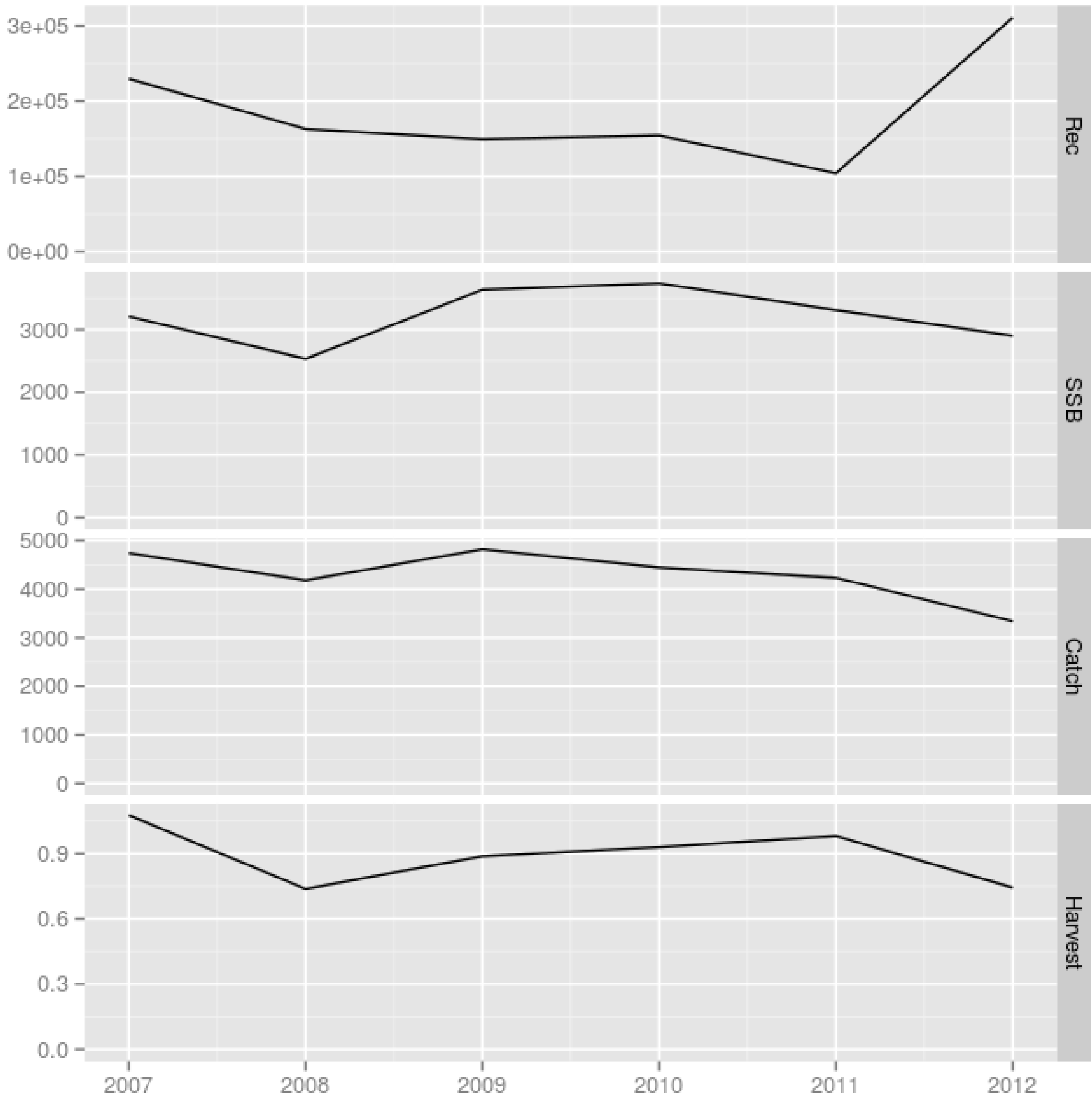


Figure 6.21: Assessment summary

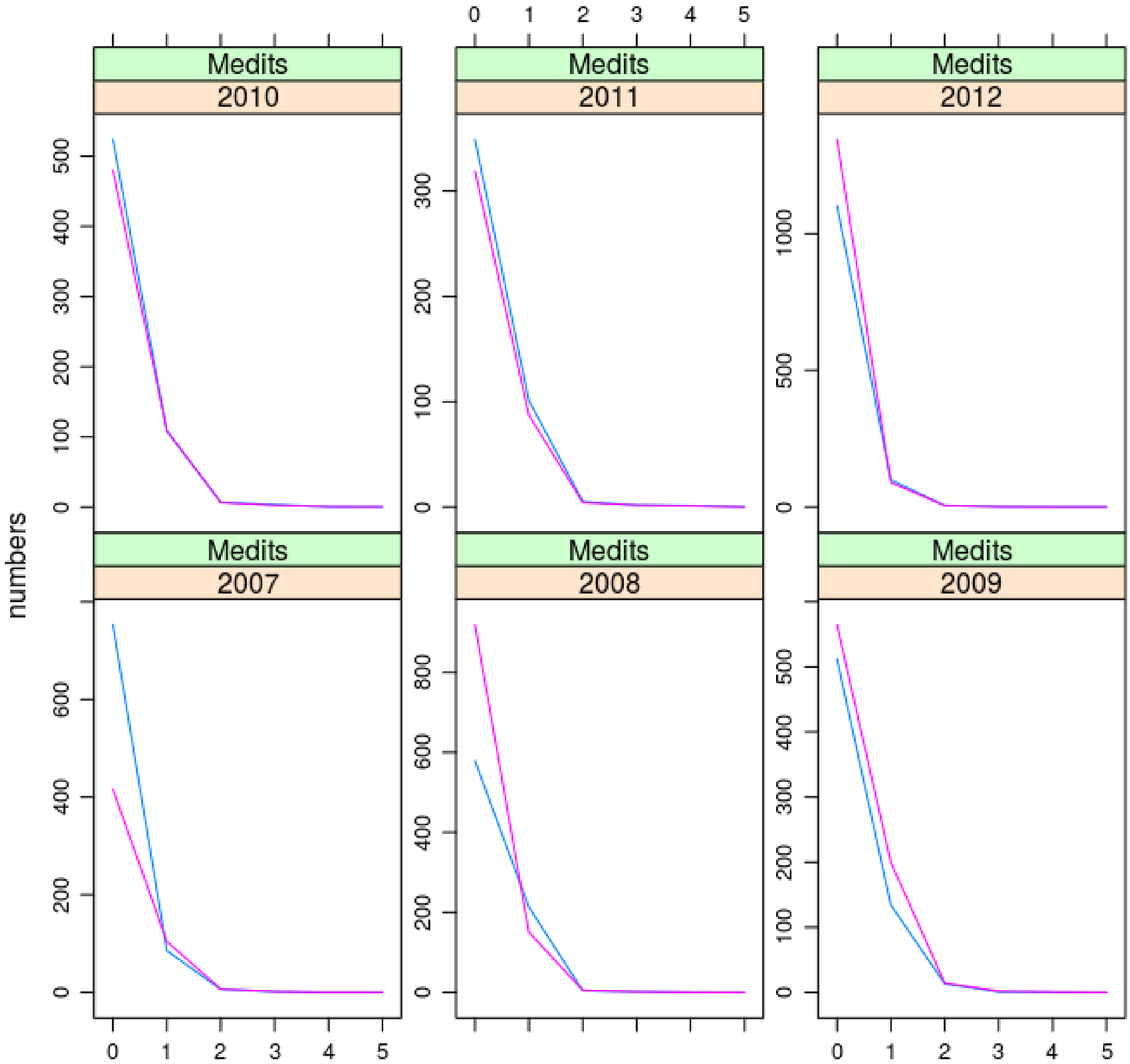


Figure 6.22: Plot fitted against observed for survey

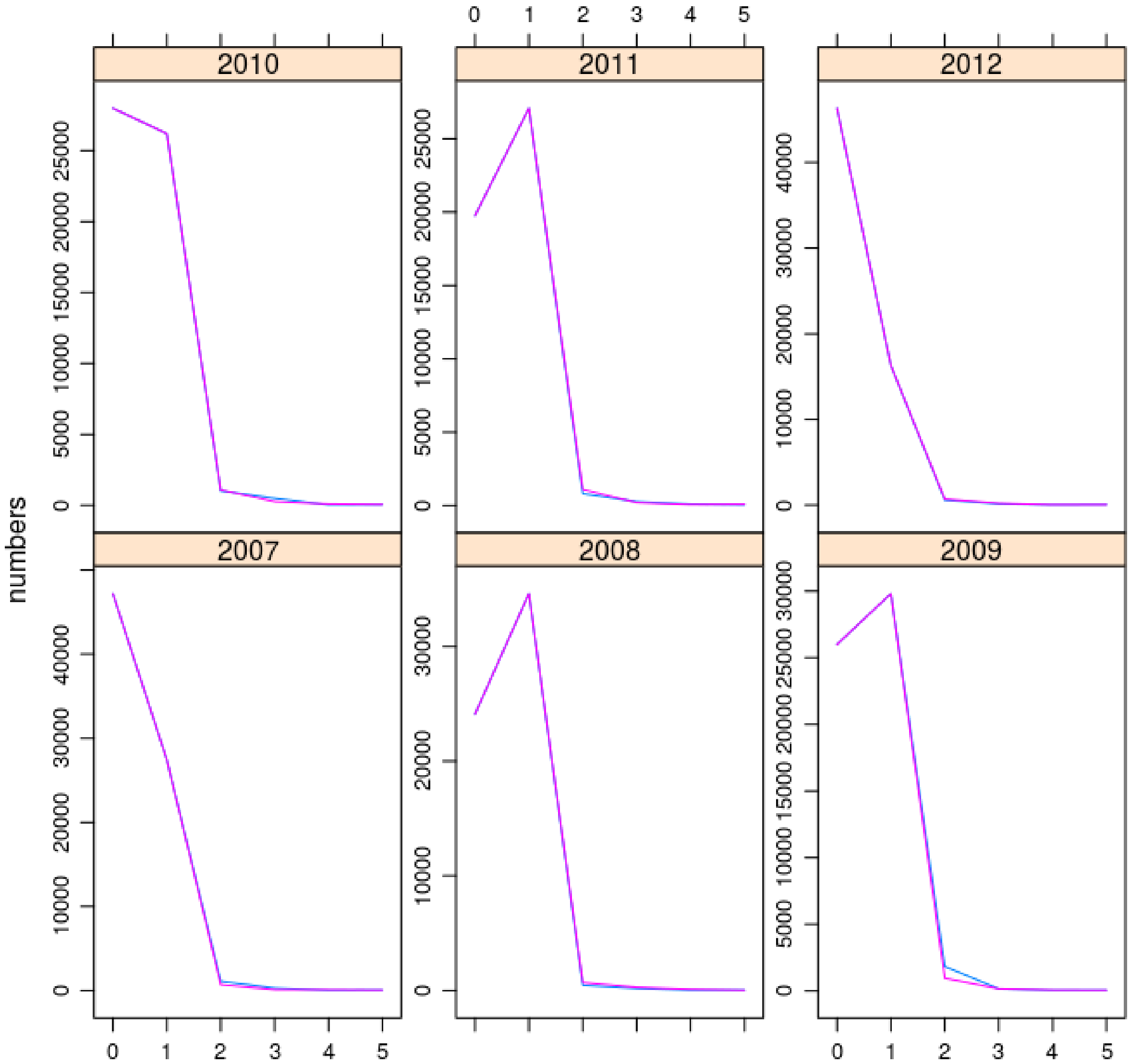


Figure 6.23: Plot fitted agains observed for catch at age

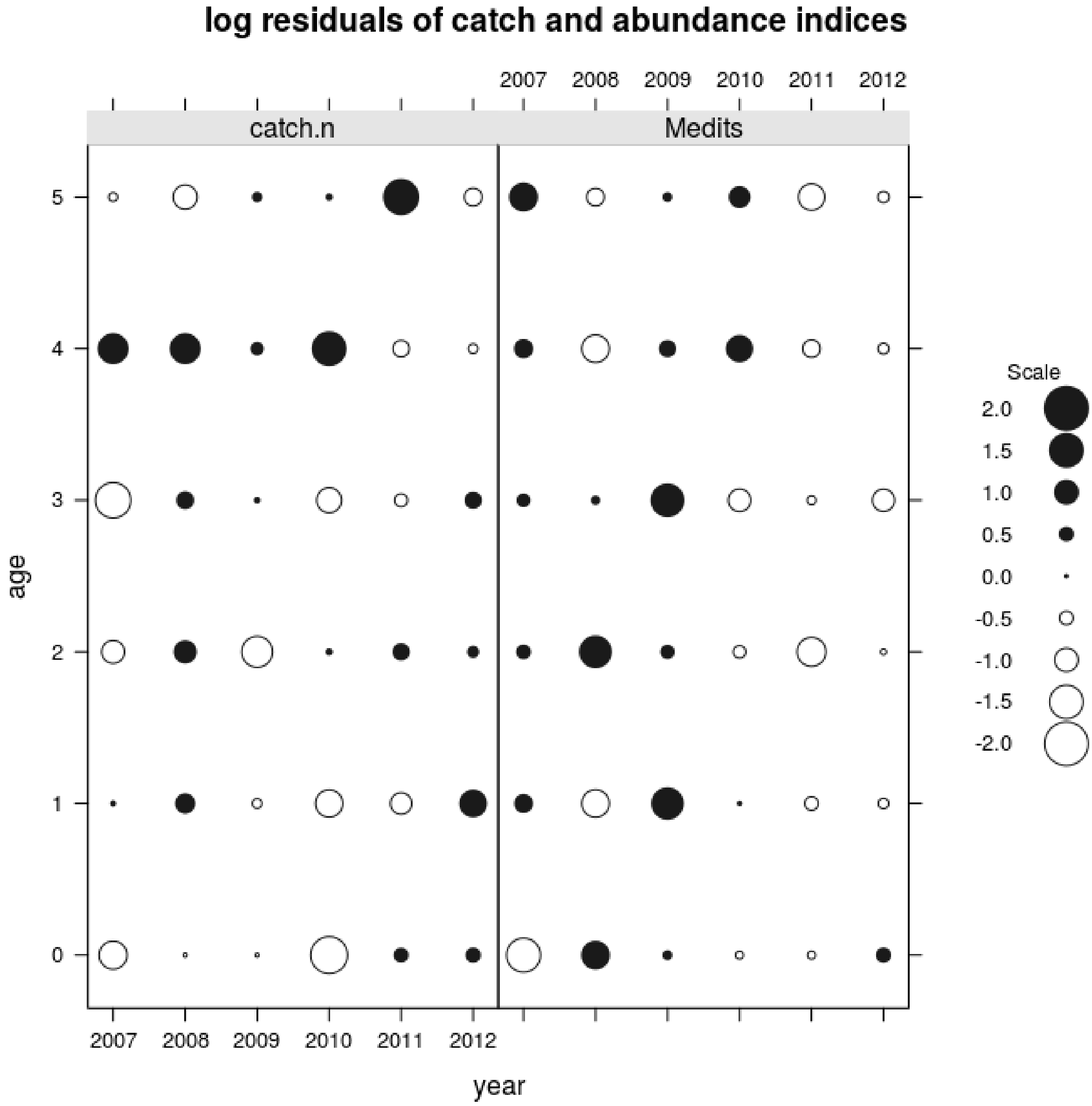


Figure 6.24: Residuals by age and year

Although the model is not a good fit, the variability of fitted vs predicted is now scattered more or less uniformly among years, and is not age specific anymore.

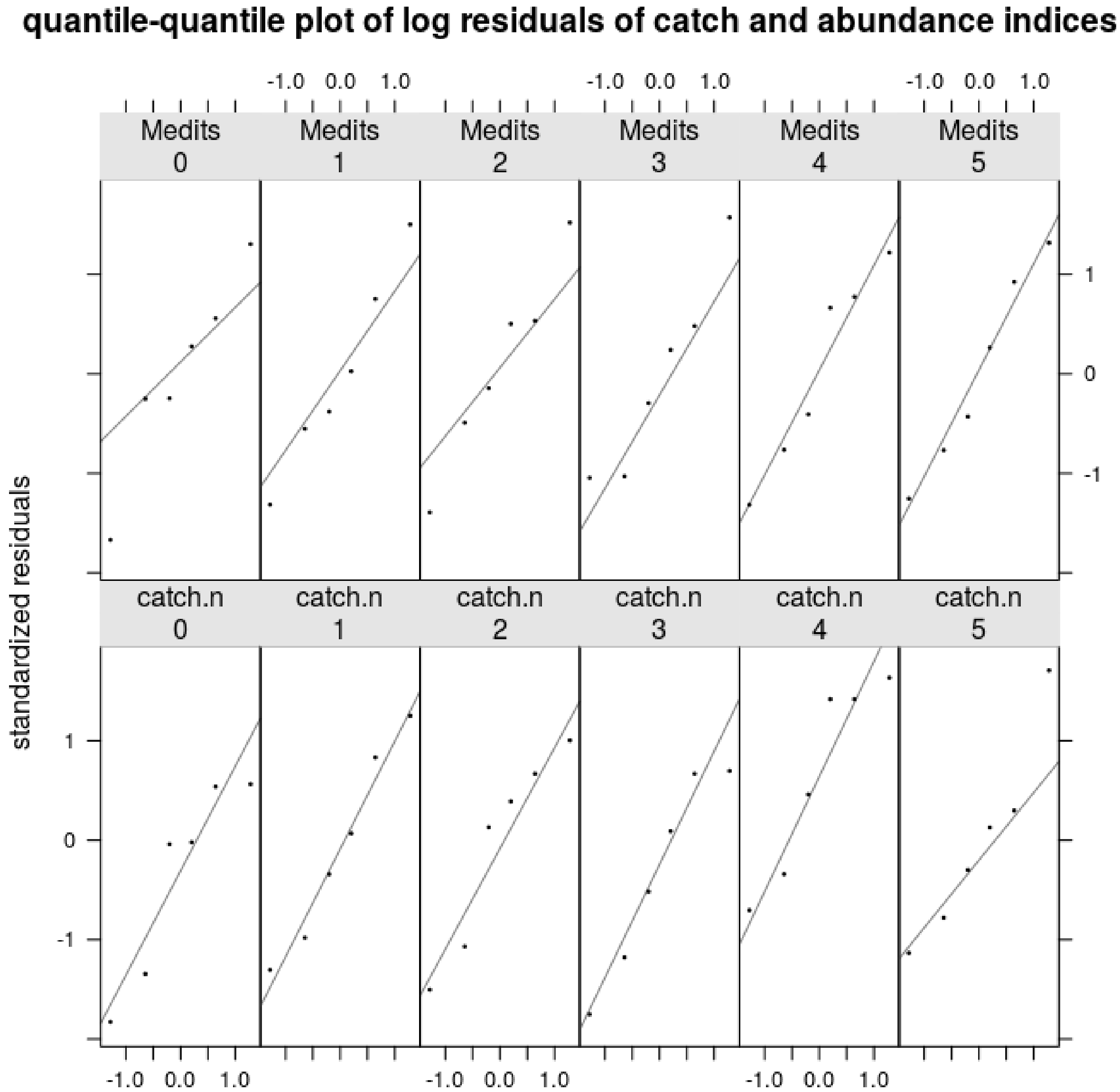


Figure 6.25: Quantile-quantile plot of residuals

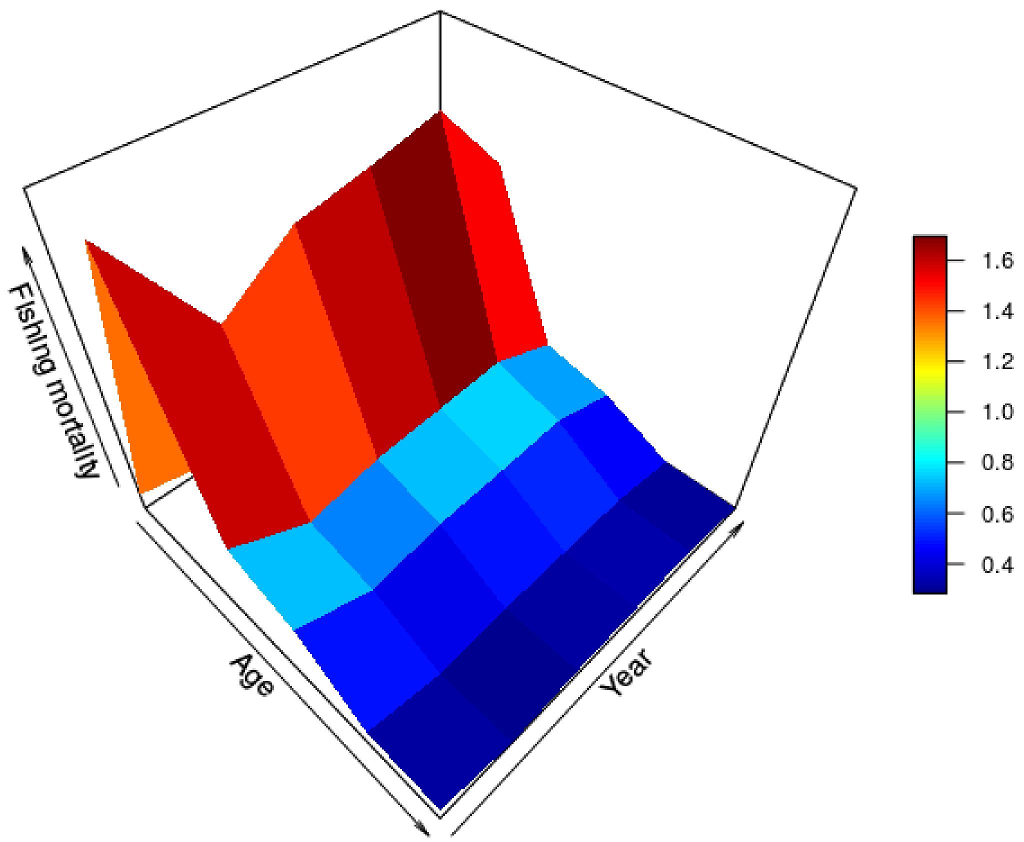


Figure 6.26: Fishing mortality

Perhaps in our effort to fit an adequate model, we ended up with "overfitting" and it would be better to use smoothers only for 'year' and model 'age' as a categorical factor. We also assume a very variable, annually fluctuating, recruitment pattern.

Separable model with smoothers 3

```
qmod4 <- list(~factor(age))
fmod4 <- ~factor(age) + s(year, k = 6)
srmod4 <- ~s(year, k = 6)
hke.stk <- hketemp
fit4 <- sca(stock = hke.stk, indices = hke.idx, fmodel = fmod4,
    qmodel = qmod4, srmodel = srmod4)
hke.stk.a4a.4 <- hke.stk + fit4
```

```
plot(hke.stk.a4a.4)
```

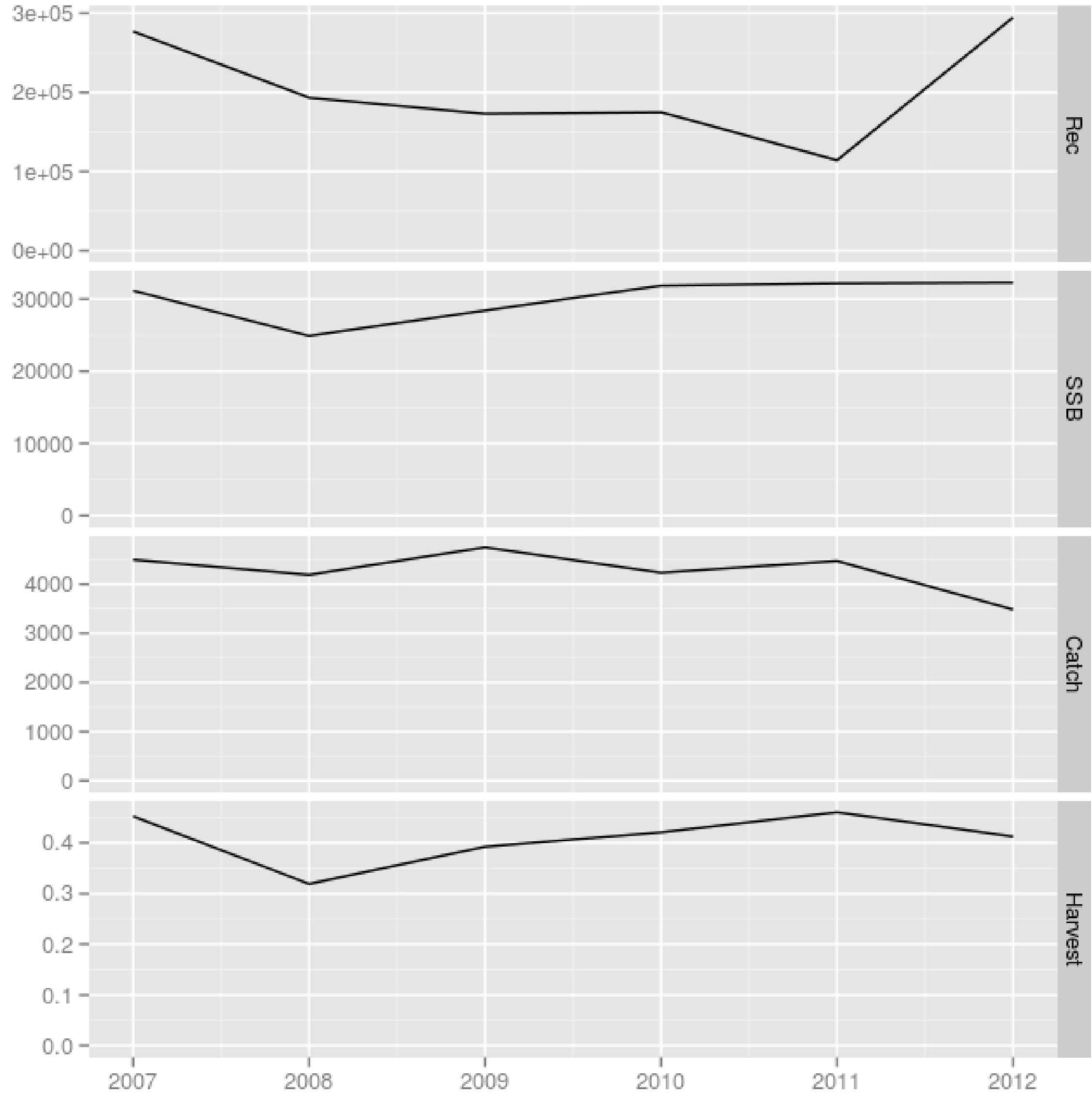


Figure 6.27: Assessment summary

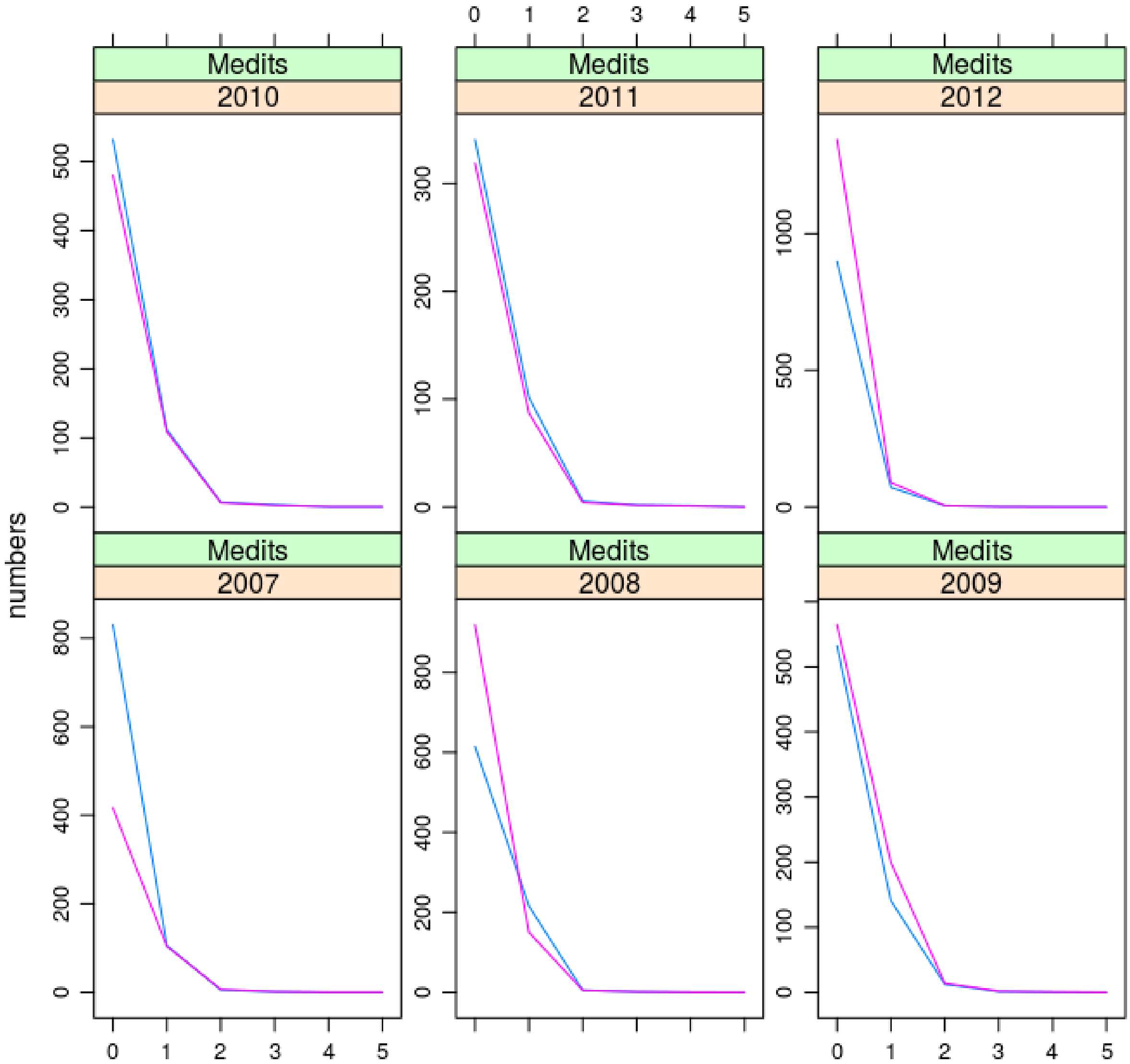


Figure 6.28: Plot fitted against observed for survey

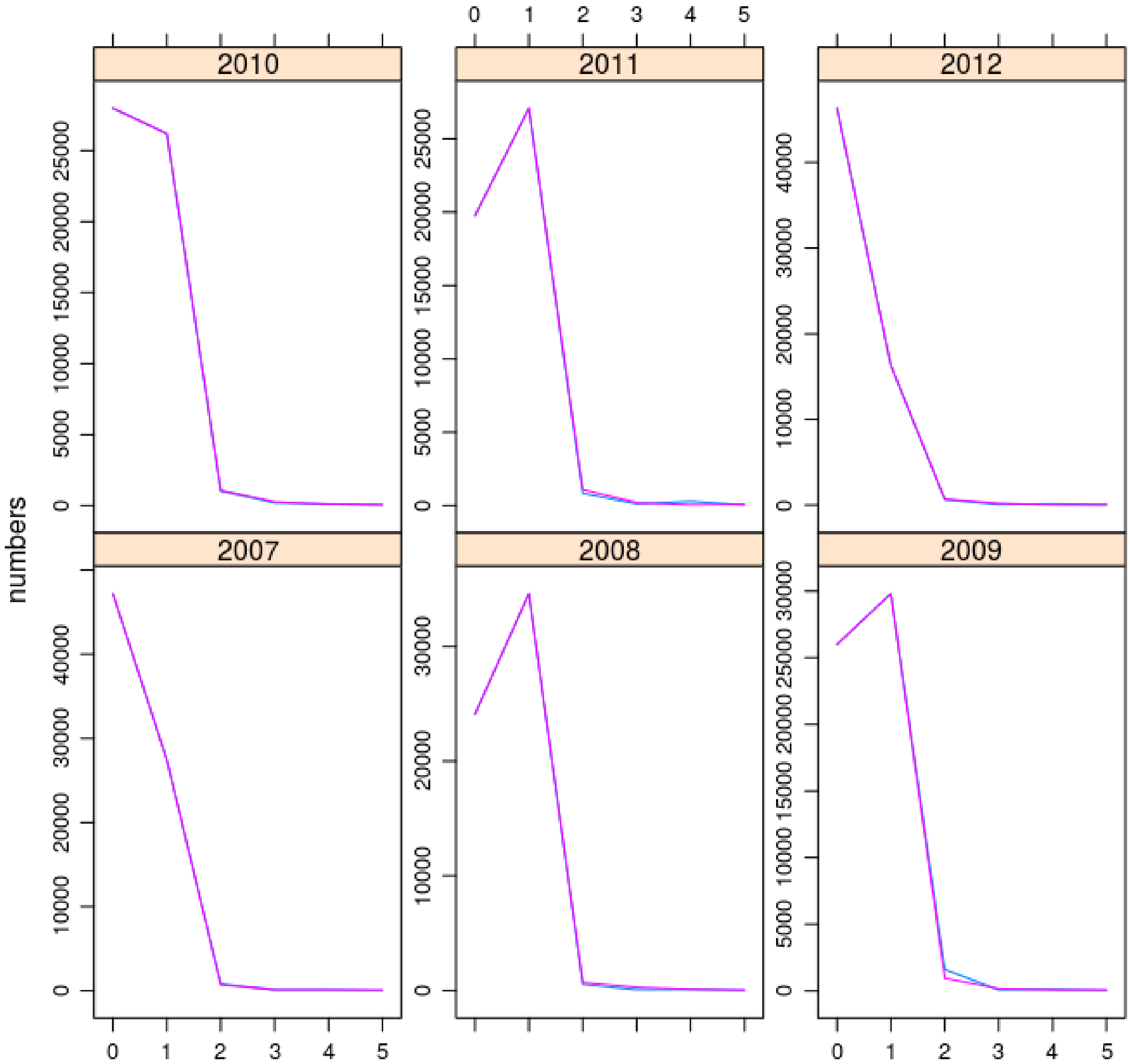


Figure 6.29: Plot fitted agains observed for catch at age

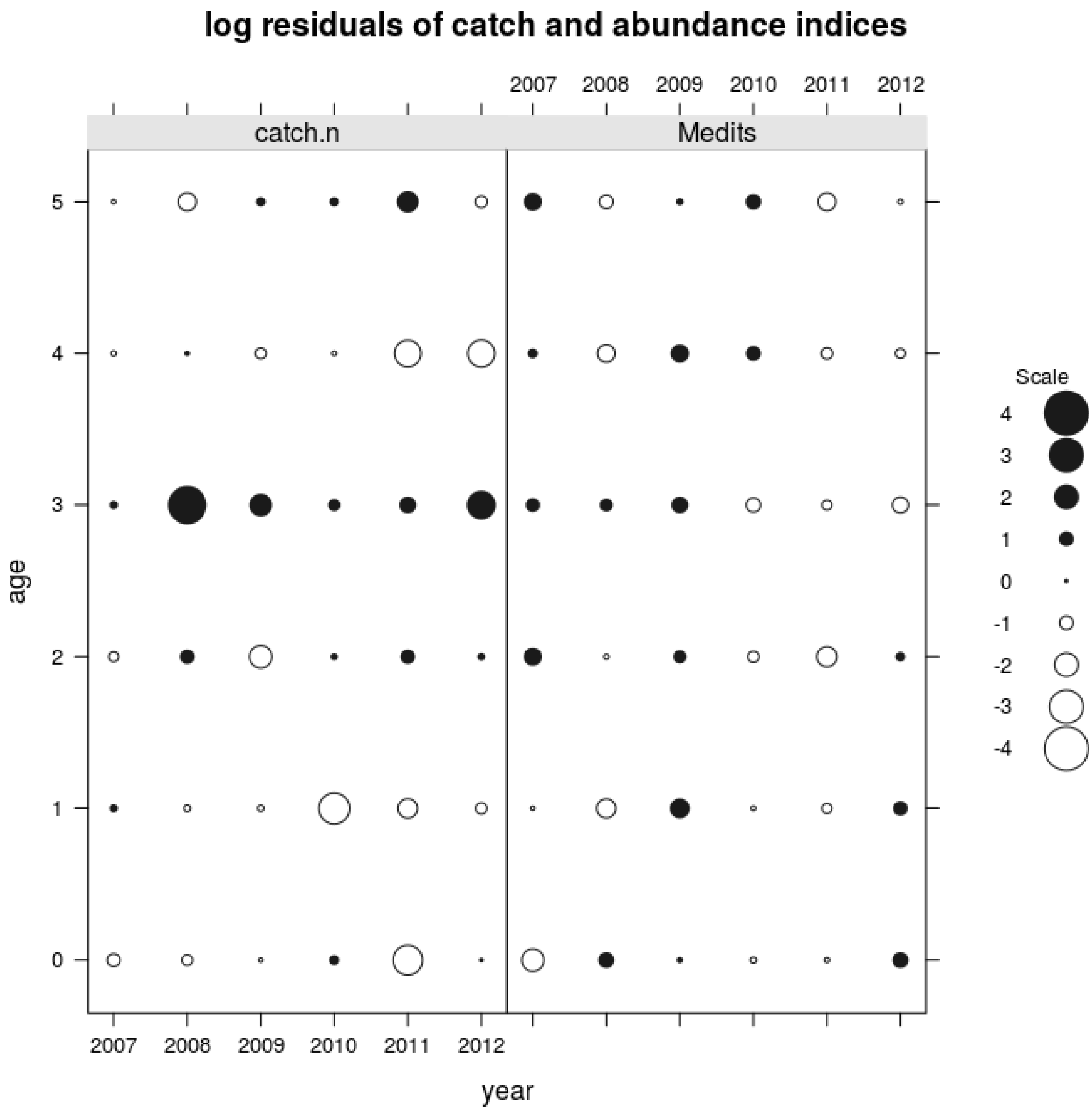


Figure 6.30: Residuals by age and year

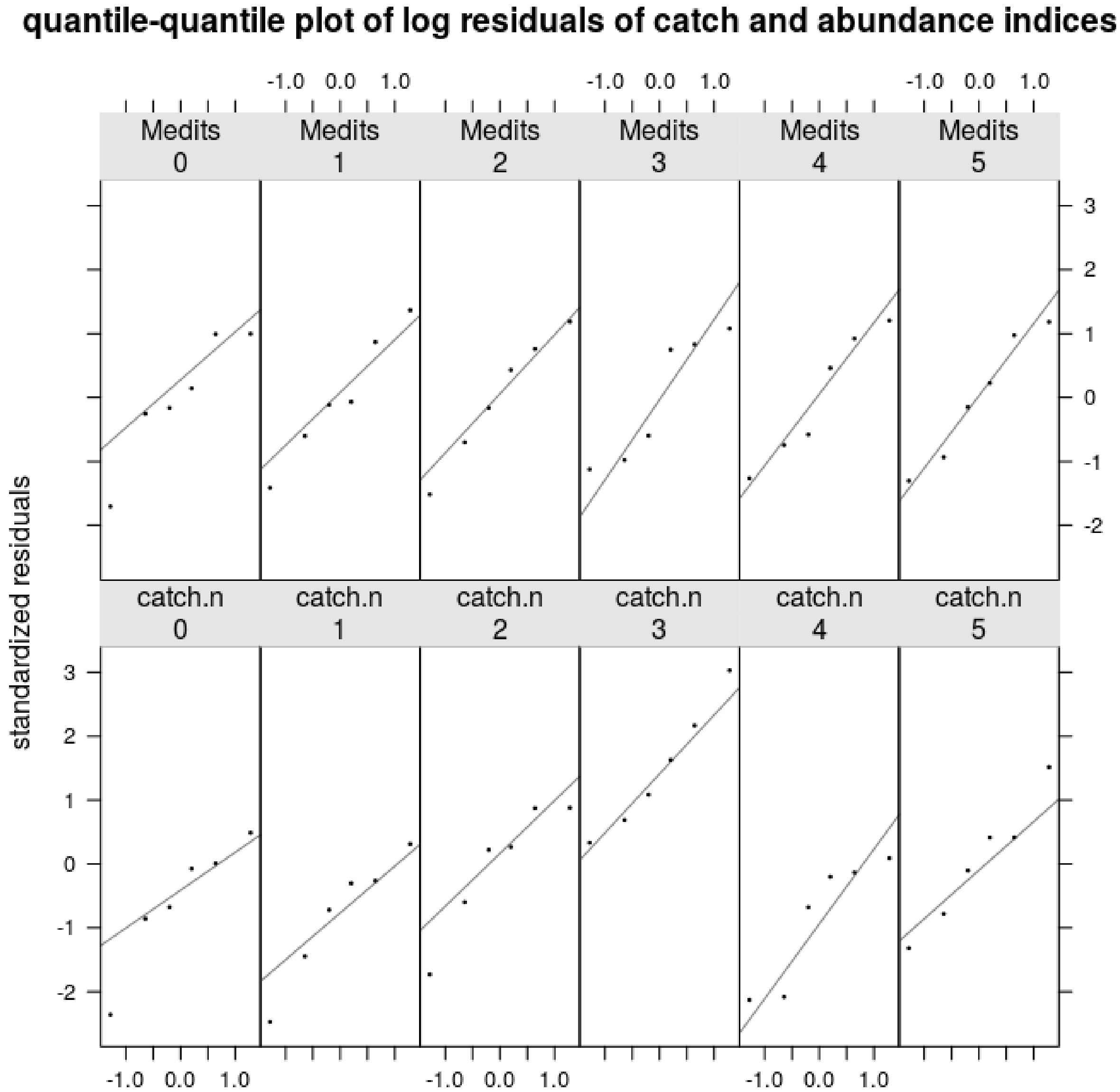


Figure 6.31: Quantile-quantile plot of residuals

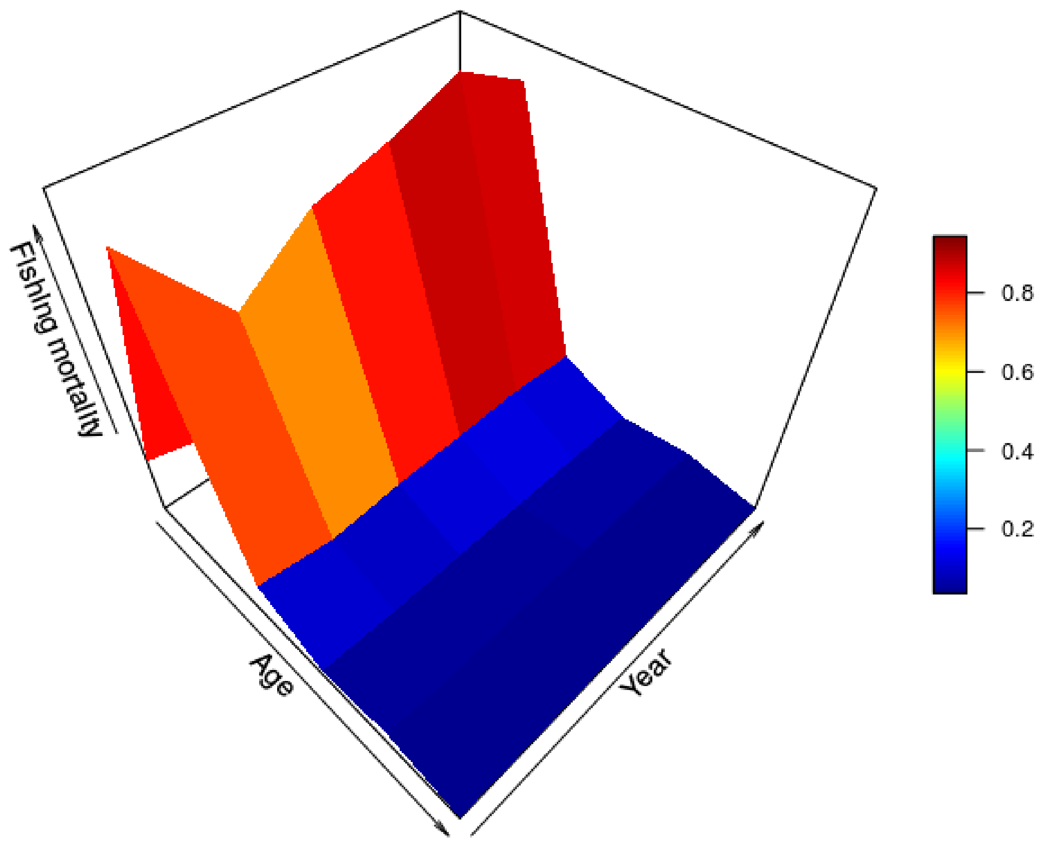


Figure 6.32: Fishing mortality

Not any better than the previous model. However, so far, all evaluation of assessments was done through visual inspection of residuals. Let's now assess the models we've built through a statistically sound criterion like the AIC.

Compare models

```
AIC(fit0, fit1, fit2, fit3, fit4)

##      df    AIC
## fit0 28  163.8
## fit1 32 -334.5
## fit2 26  113.5
## fit3 32 -311.9
## fit4 32 -343.7
```

Age treated as factor and year through a wiggly smoother gives the 'best' model from an AIC point of view. However, non is convincing through residuals inspection.

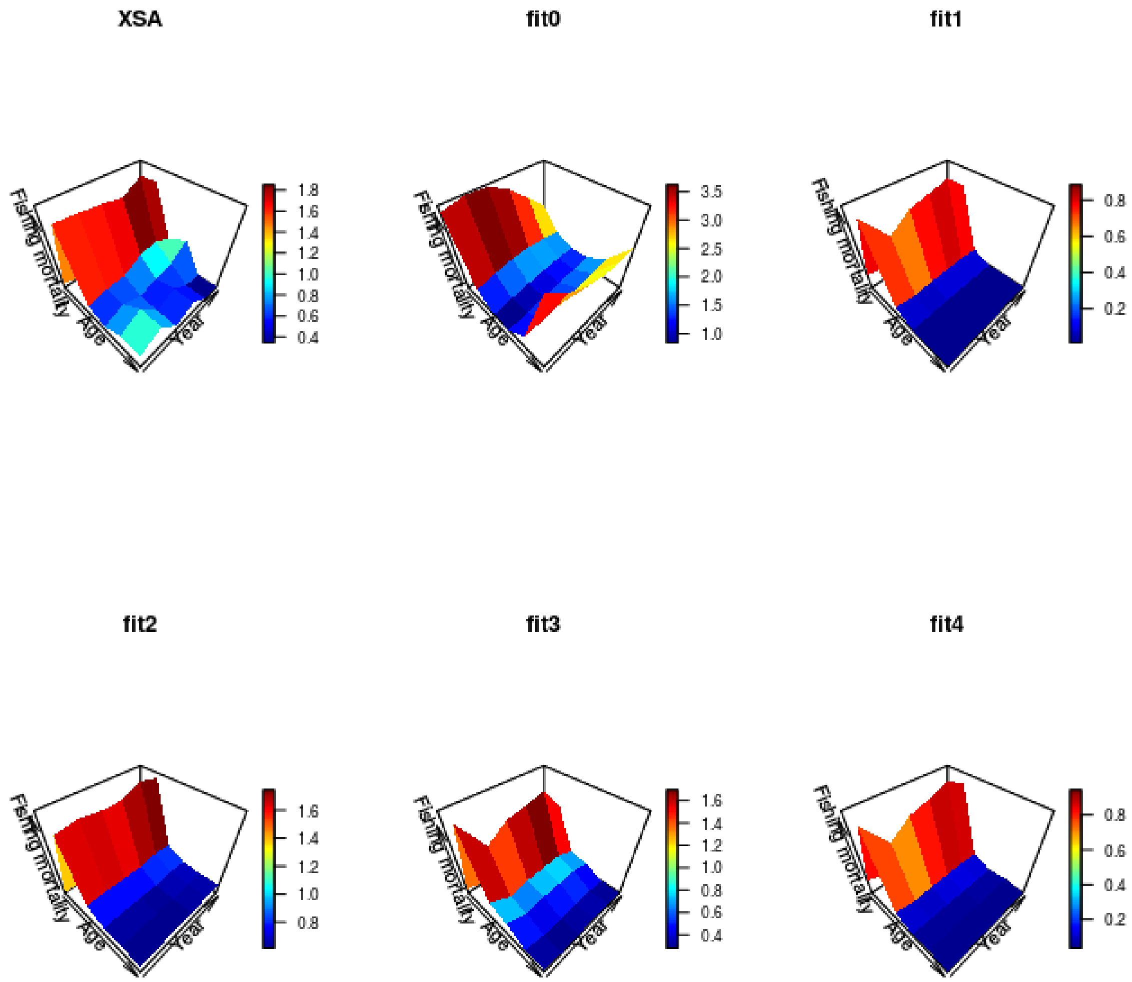


Figure 6.33: Compare a4a fits with XSA results

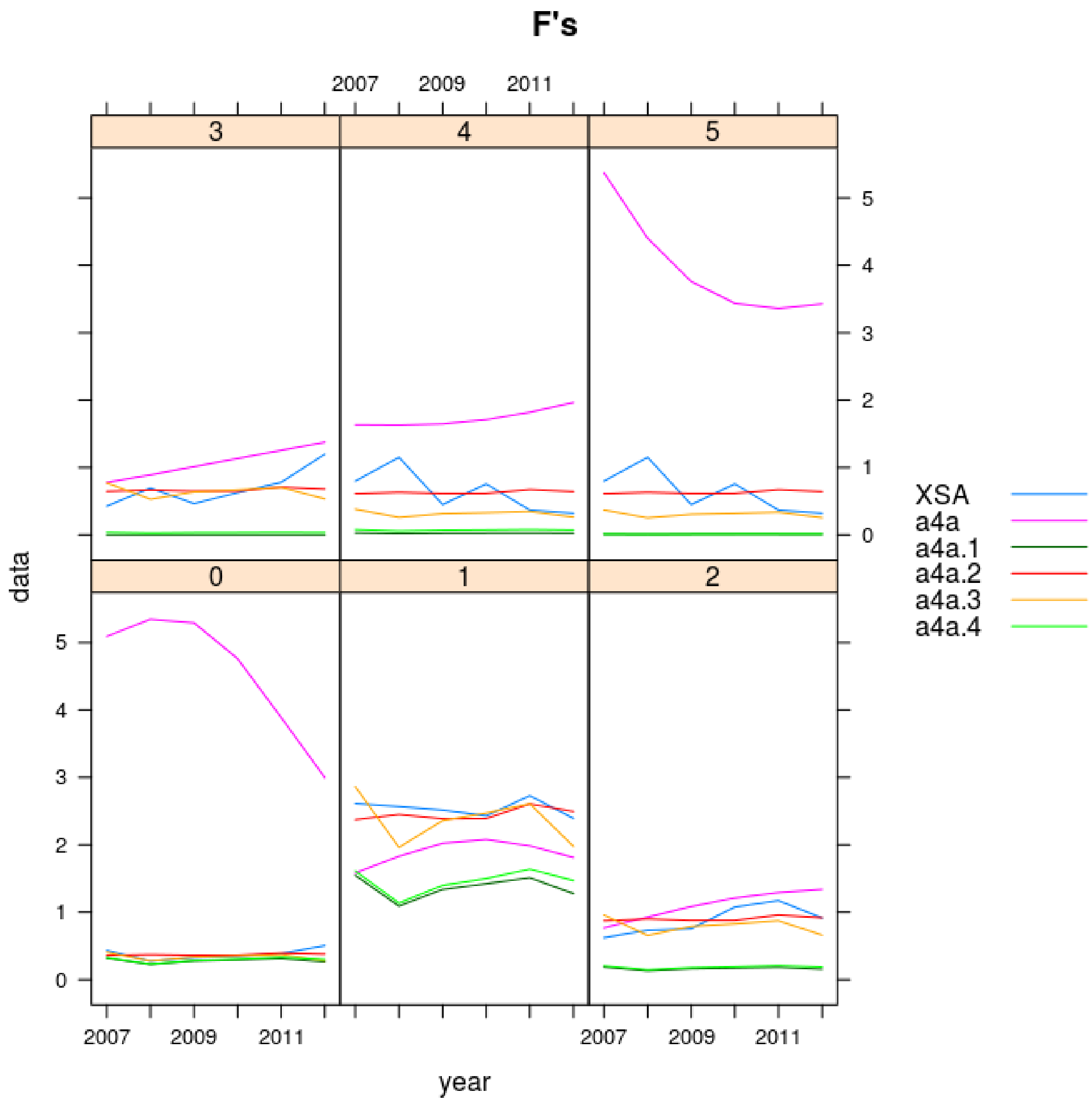


Figure 6.34: Compare F's

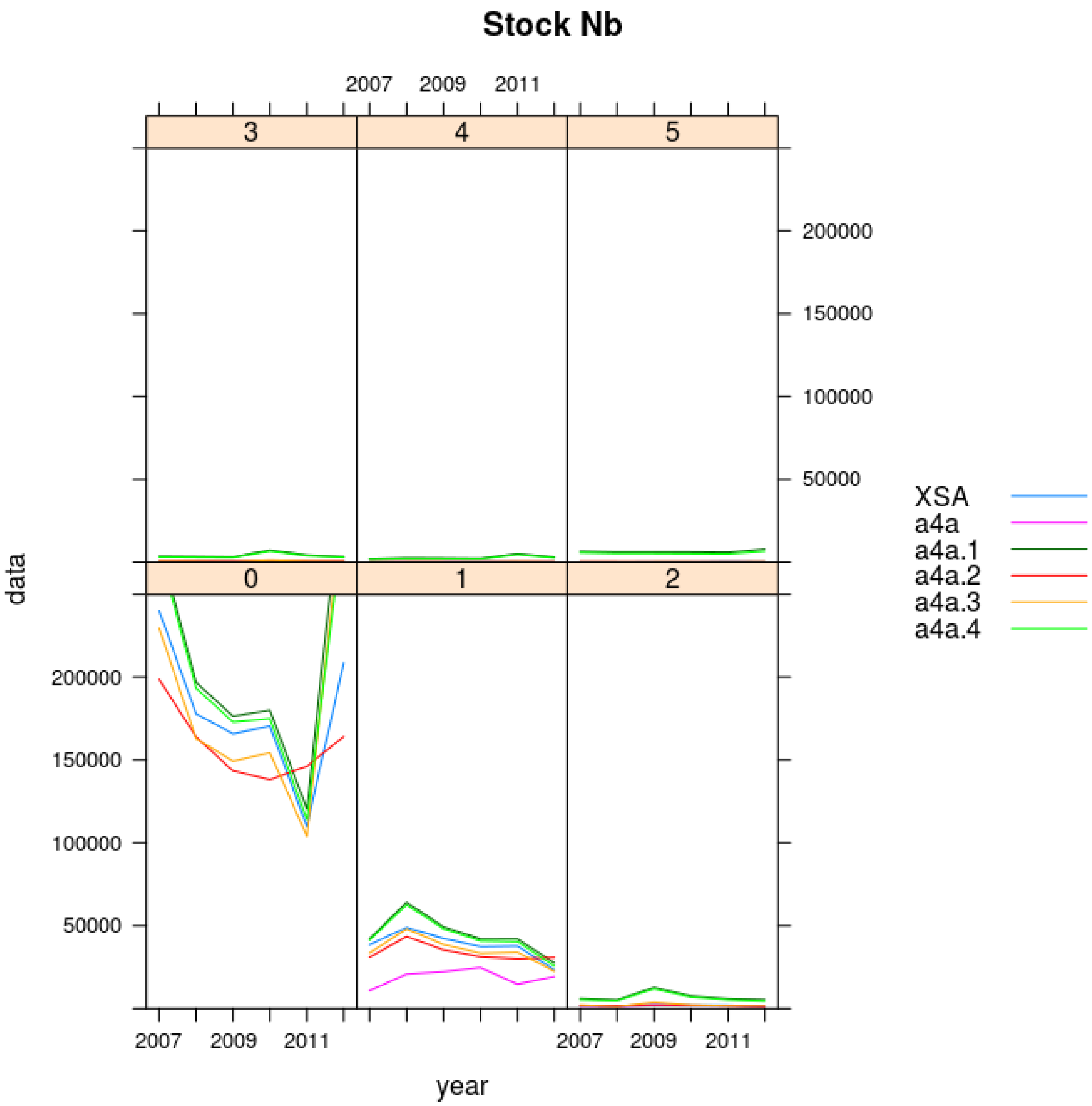


Figure 6.35: Compare abundance

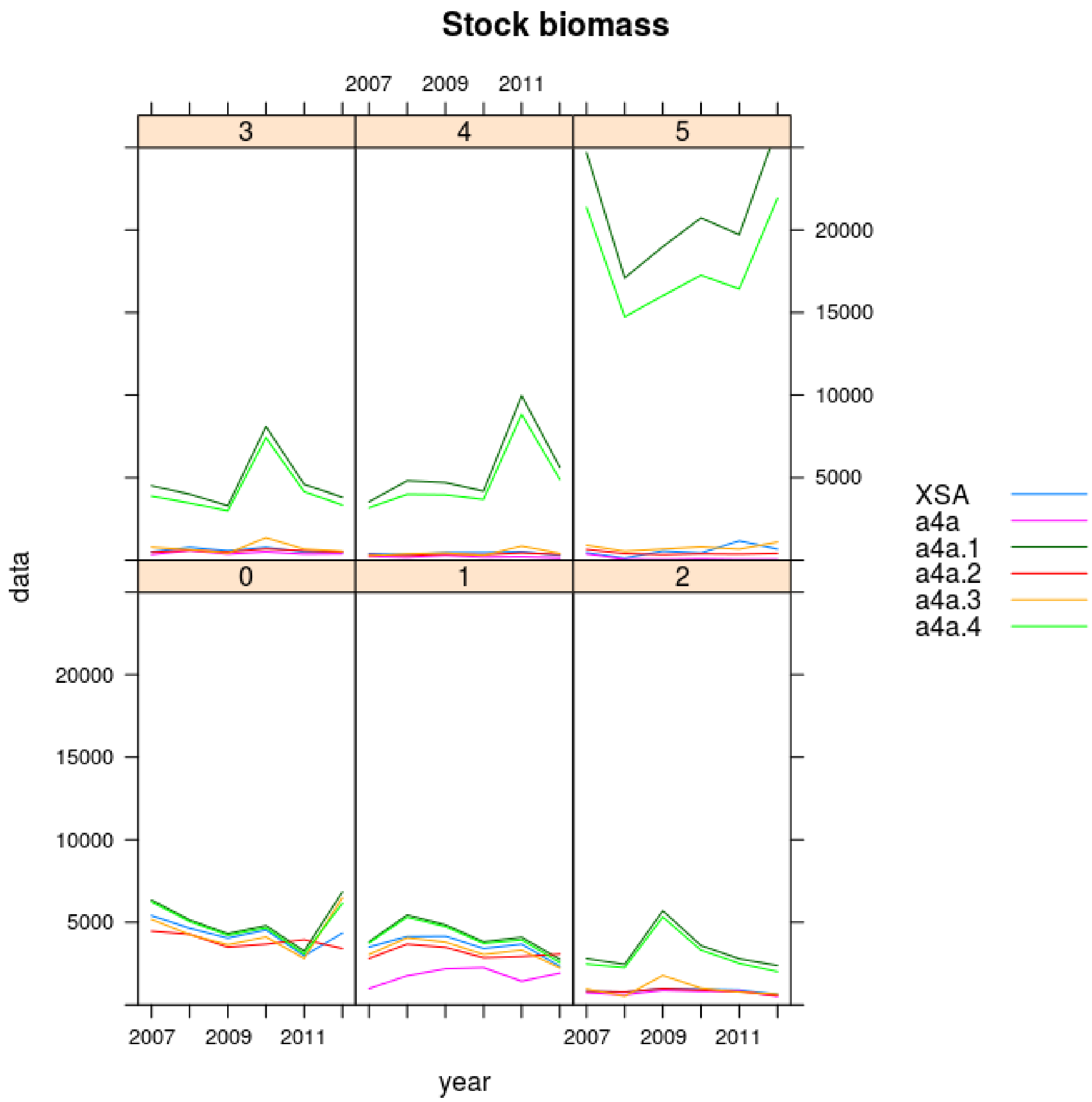


Figure 6.36: Compare Stock biomass

Final comments

None of the 5 investigated models could reproduce XSA's results. 'fit3' with qmod3 <- list($s(age, k = 6)$), fmod3 <- $s(age, k = 6)+s(year, k = 6)$ and srmod3 <- $s(year, k = 6)$ seems to give F's and stock values closer to last years' XSA assessments.

However model evaluation through AIC, ranks 'fit3' as the third best out of all five. The AIC selected 'best' model (fit4) gives extremely low F values. This might be an indication that we failed to model fishing mortality adequately and efforts should concentrate towards fitting a better 'fmodel'.

6.2 Assessments with the statistical catch-at-age method

6.2.1 Readind data

```
# READ THE LANDINGS AT LENGTH DATA
HKE18.lnd <- read.table("data/HKE GSA18 LND_LEN.csv", header = T,
    sep = ",")
# we convert the object to a matrix
HKE18.lnd.matrix <- as.matrix(HKE18.lnd)
dim(HKE18.lnd.matrix)

## [1] 101   7

# We need to specify the dimnames
HKE18.lnd.flq <- FLQuant(HKE18.lnd.matrix[, -1], dimnames = list(len = 0:100,
    year = 2007:2012), unit = "numbers")

# READ THE CATCH AT LENGTH DATA
HKE18.orig <- read.table("data/HKE GSA18 CA_LEN.csv", header = T,
    sep = ",")
# HKE18.orig <- t(HKE18.orig)
class(HKE18.orig)

## [1] "data.frame"

# we convert the object to a matrix
HKE18.matrix <- as.matrix(HKE18.orig)
dim(HKE18.matrix)

## [1] 101   7

# We need to specify the dimnames
HKE18.flq <- FLQuant(HKE18.matrix[, -1], dimnames = list(length = 0:100,
    year = 2007:2012), unit = "numbers")

# READ THE CATCH WEIGHT DATA
HKE18.cwt <- read.table("data/HKE GSA18 CA_WT.csv", header = T,
    sep = ",")
class(HKE18.cwt)

## [1] "data.frame"

# we convert the object to a matrix
HKE18.cwt.matrix <- as.matrix(HKE18.cwt)
dim(HKE18.cwt.matrix)

## [1] 6 7
```

```
# We need to specify the dimnames
HKE18.cwt.flq <- FLQuant(HKE18.cwt.matrix[, -1], dimnames = list(age = 0:5,
    year = 2007:2012), unit = "kg")
HKE18.cwt.flq <- setPlusGroup(HKE18.cwt.flq, 5)

# READ THE MATURITY mat DATA
HKE18.mat <- read.table("data/HKE GSA18 mat.csv", header = T,
    sep = ",")
class(HKE18.mat)

## [1] "data.frame"

# we convert the object to a matrix
HKE18.mat.matrix <- as.matrix(HKE18.mat)
dim(HKE18.mat.matrix)

## [1] 6 7

# We need to specify the dimnames
HKE18.mat.flq <- FLQuant(HKE18.mat.matrix[, -1], dimnames = list(age = 0:5,
    year = 2007:2012), unit = "prop")
HKE18.mat.flq <- setPlusGroup(HKE18.mat.flq, 5)

# save.image(file='data/HKE.GSA18.raw.data2.RData')

# OR if all the above is useless just load the FLquants
# created from tha dumped file
# load('HKE.GSA18.raw.data2.RData')
```

Converting catches and landings from lengths o ages

```
# ----------------------------------------------------------------------------
# Growth params are needed - use the ones suggested in the
# STECF EWG 13-09
# ----------------------------------------------------------------------------
# von Bertalanffy equation params
vbObj <- a4aGr(grMod = ~linf * (1 - exp(-k * (t - t0))), grInvMod = ~t0 -
    1/k * log(1 - len/linf), params = FLPar(linf = 104, k = 0.2,
    t0 = -0.01))
# Check vonBer results
lc = 20
predict(vbObj, len = lc)

##    iter
##         1
##   1 1.058
```

```
# Lengths to Ages Slicing

# Catches data - length to age slicing
HKE18.CA <- l2a(HKE18.flq, vbObj)
HKE18.CA <- setPlusGroup(HKE18.CA, 5)

# Landings data - length to age slicing
HKE18.LAND <- l2a(HKE18.lnd.flq, vbObj)
HKE18.LAND <- setPlusGroup(HKE18.LAND, 5)
```

Model natural mortality

```
# Simple m=0.2
mod0 <- FLModelSim(model = ~a, params = FLPar(a = 0.2))
m0 <- a4aM(level = mod0)
rngquant(m0) <- c(0, 5)  # set the quant range
rngyear(m0) <- c(2007, 2012)  # set the year range
HKE18.m <- m(m0)
# We can do better...

# Try modelling m Jensen's estimator
shape1 <- FLModelSim(model = ~exp(-age - 0.2))
level1 <- FLModelSim(model = ~5 * k, params = FLPar(k = 0.2))
m1 <- a4aM(shape = shape1, level = level1)
rngquant(m1) <- c(0, 5)  # set the quant range
rngyear(m1) <- c(2007, 2012)  # set the year range
HKE18.m <- m(m1)
HKE18.m <- setPlusGroup(HKE18.m, 5)
HKE18.m


## An object of class "FLQuant"
## , , unit = unique, season = all, area = unique
##
##    year
## age 2007      2008      2009      2010      2011      2012
##   0 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
##   1 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794 0.3678794
##   2 0.1353353 0.1353353 0.1353353 0.1353353 0.1353353 0.1353353
##   3 0.0497871 0.0497871 0.0497871 0.0497871 0.0497871 0.0497871
##   4 0.0183156 0.0183156 0.0183156 0.0183156 0.0183156 0.0183156
##   5 0.0067379 0.0067379 0.0067379 0.0067379 0.0067379 0.0067379
##
## units:  NA
```

Create FLStock and FLIndex objects

```r
# Load MEDITS survey index
load("data/hke.medits.RData")
Hake18.S.Ind.new <- hke.idx
range(Hake18.S.Ind.new)

##     min     max minyear maxyear
##       0       5    2007    2012

validObject(Hake18.S.Ind.new)

## [1] TRUE

# Set when the survey is taking place in the year
range(Hake18.S.Ind.new[[1]], "startf") <- 0.5
range(Hake18.S.Ind.new[[1]], "endf") <- 0.75

# Pass stock related FLQuants in an FLStock
Hake18.stk <- FLStock(catch.wt = HKE18.cwt.flq, catch.n = HKE18.CA,
    mat = HKE18.mat.flq, m = HKE18.m, landings.n = HKE18.LAND)

name(Hake18.stk) <- "GSA 18 HAKE"
desc(Hake18.stk) <- "Data from 2013 Data Call"

# when spawning takes place
m.spwn(Hake18.stk) <- 0
# harvest spawning
harvest.spwn(Hake18.stk) <- 0
# stock wt is catch.wt
Hake18.stk@stock.wt <- Hake18.stk@catch.wt
Hake18.stk@landings.wt <- Hake18.stk@catch.wt

# investigate object Hake18.stk
range(Hake18.stk)

##       min       max plusgroup   minyear   maxyear   minfbar   maxfbar
##         0         5         5      2007      2012         0         5

validObject(Hake18.stk)

## [1] TRUE

# keep the stock in temp obj to avoid possible overwriting
hketemp <- Hake18.stk
# save.image(file='HKE.GSA18.a4a.data2.RData')
```

6.2.2 Run assessments with a4a

Model 0

Can't get any simpler than that. Horrible outputs...

```
Hake18.stk <- hketemp
qmod <- list(~1)
fmod <- ~1
srmod <- ~1
fit0 <- sca(stock = Hake18.stk, indices = Hake18.S.Ind.new, fmodel = fmod,
    qmodel = qmod, srmodel = srmod)

Hake18.stk.a4a.0 <- Hake18.stk + fit0
landings(Hake18.stk.a4a.0) <- computeLandings(Hake18.stk.a4a.0)
```

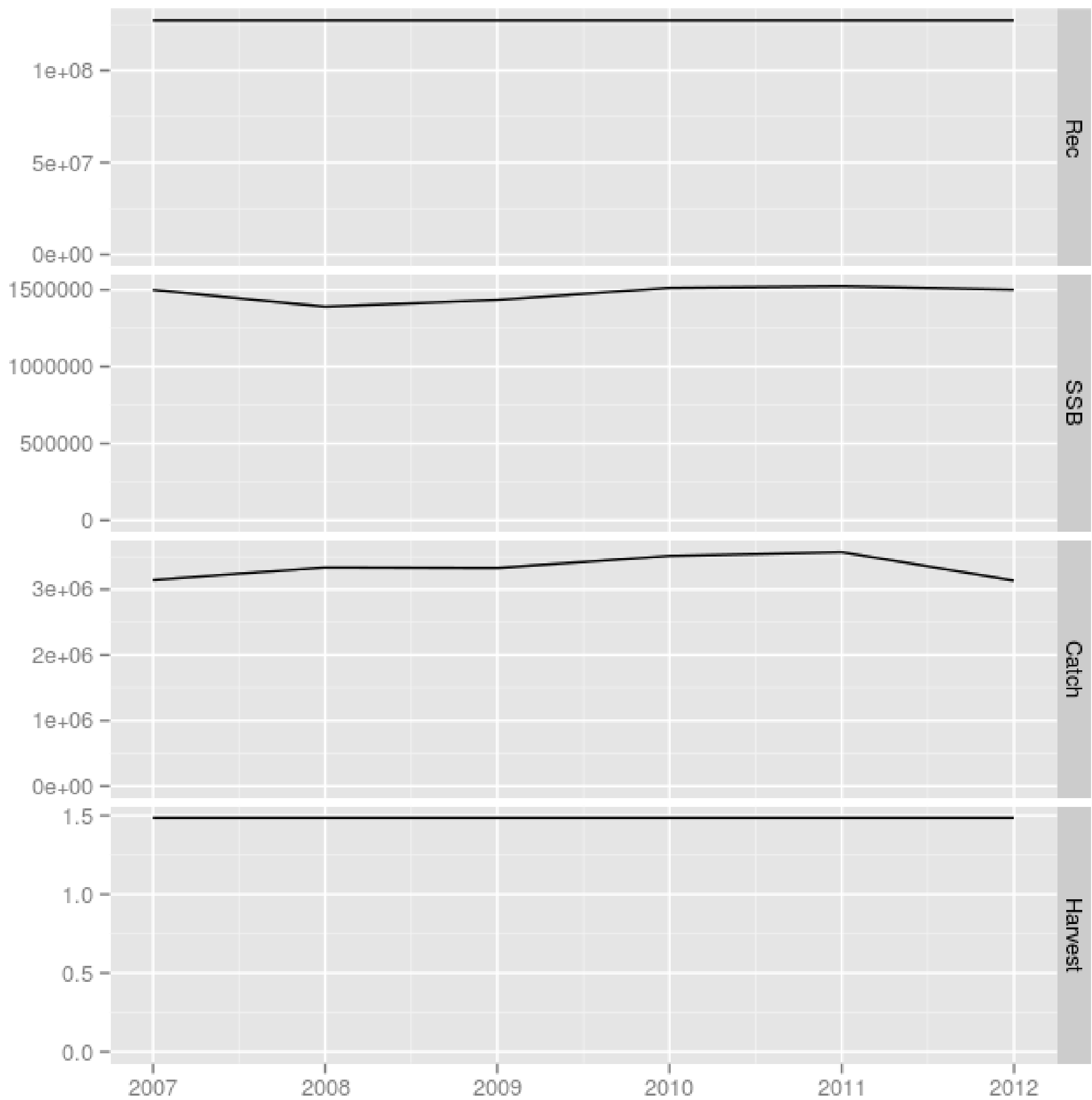


Figure 6.37: Stock summary

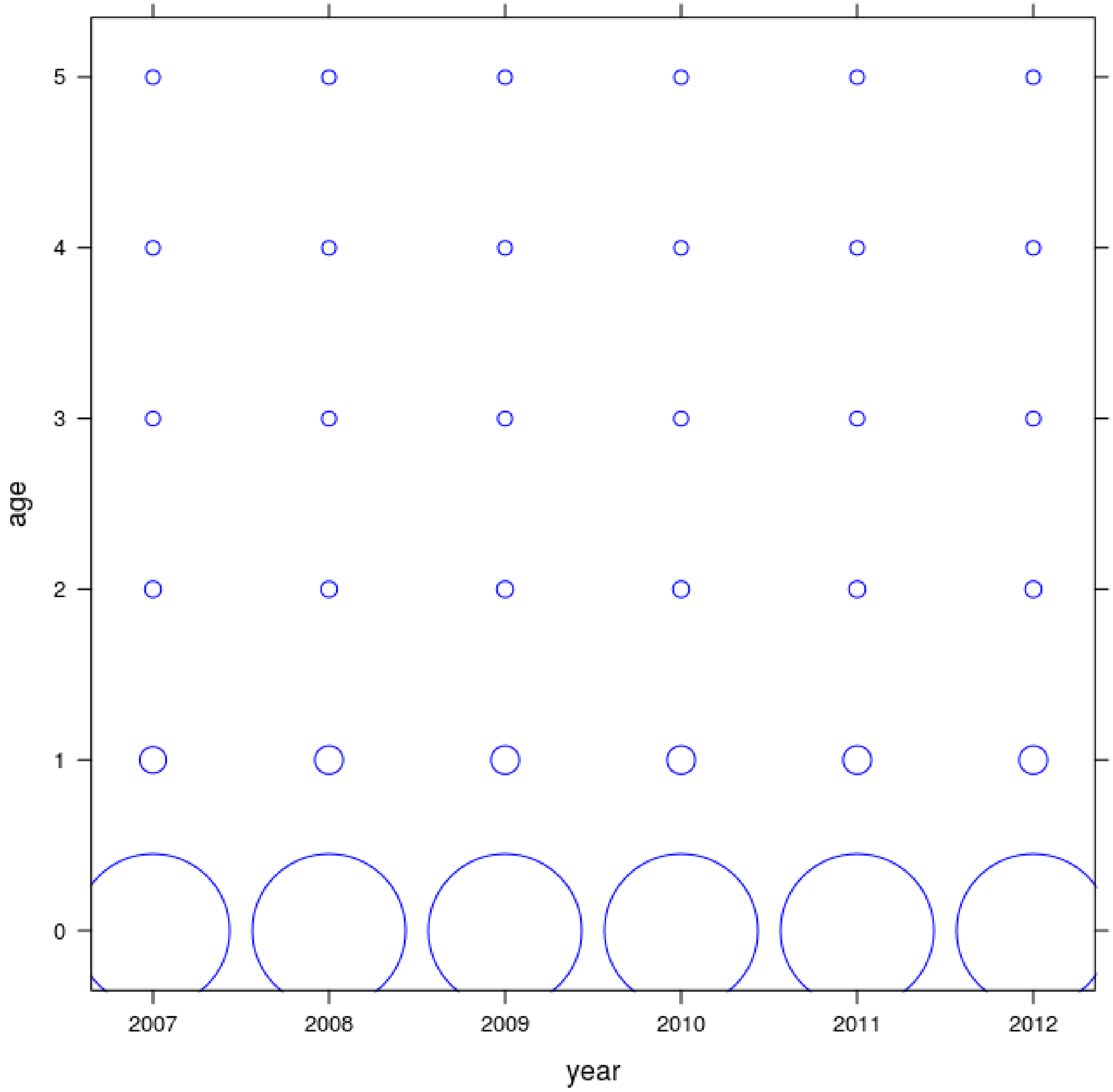


Figure 6.38: The catch matrix

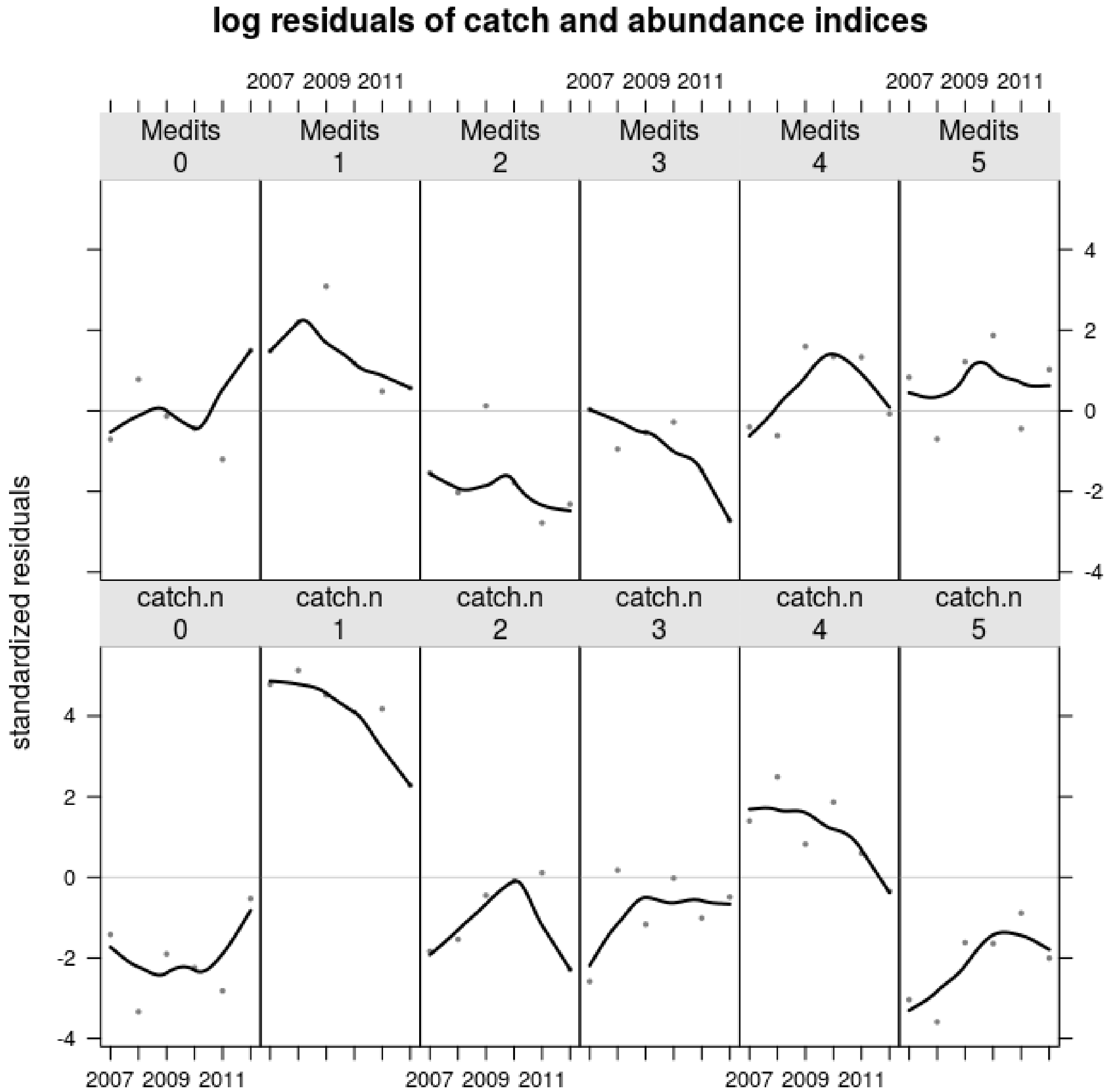


Figure 6.39: Residuals

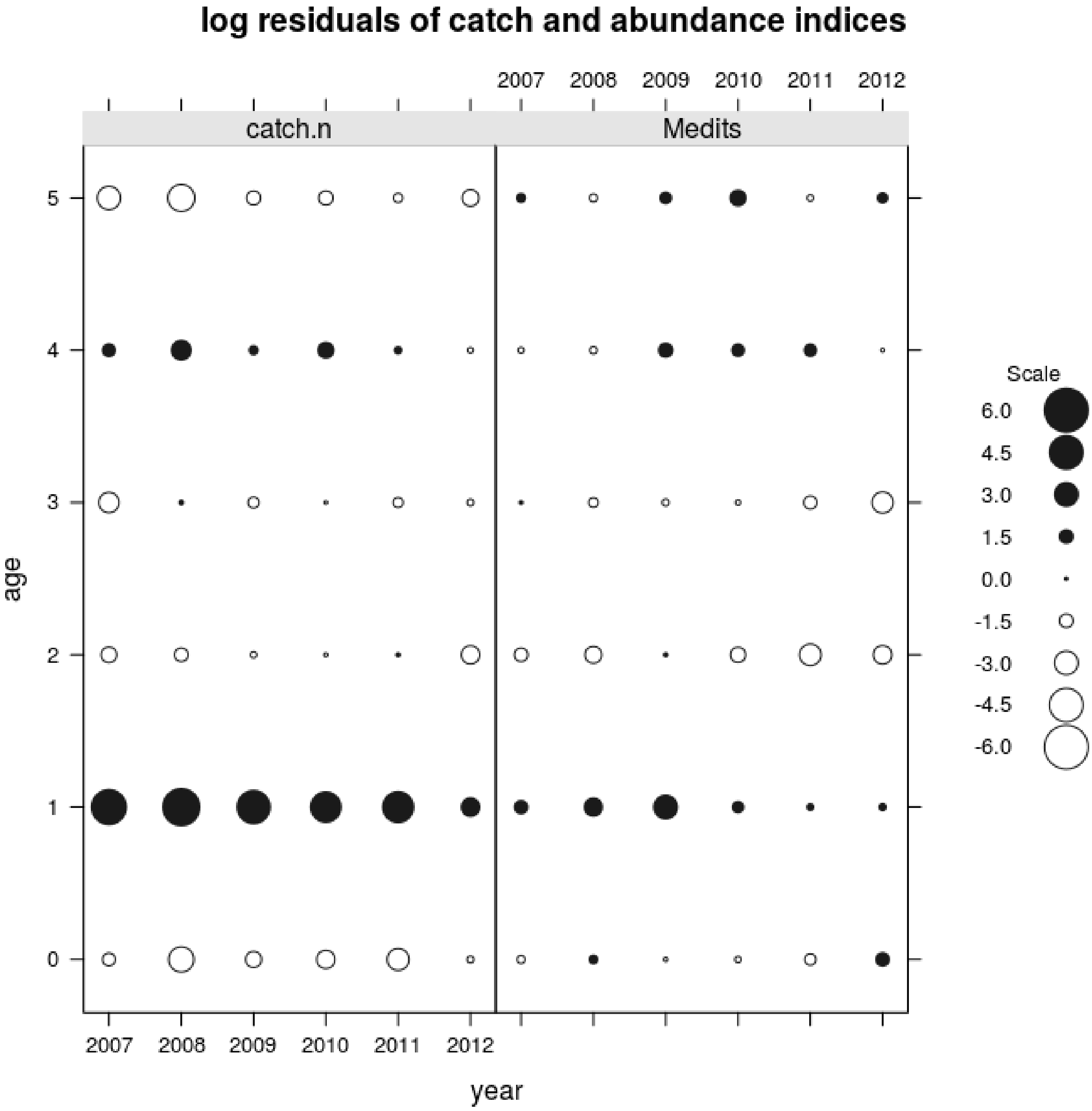


Figure 6.40: Residuals by age and year

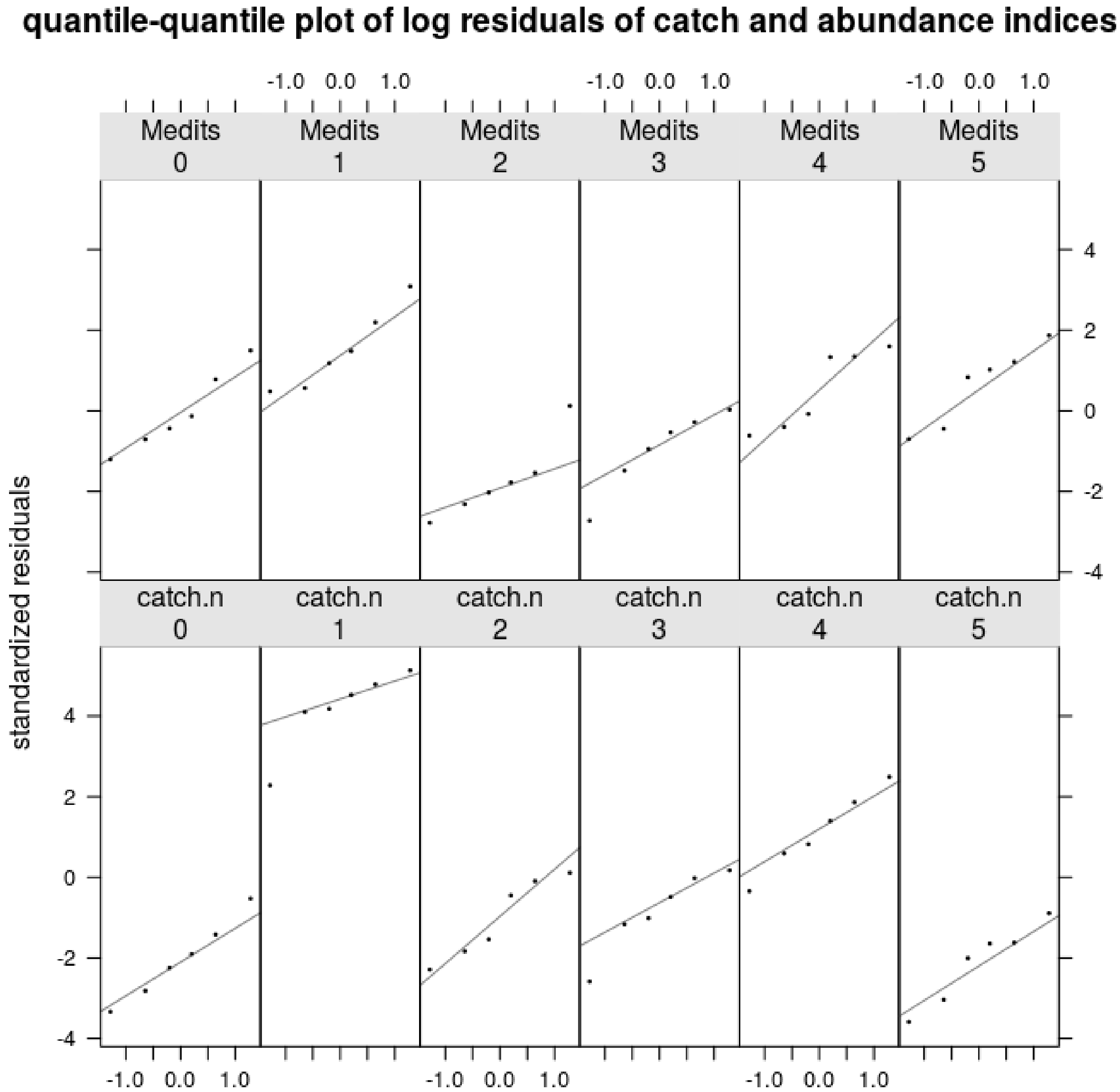


Figure 6.41: Quantile-quantile plot of residuals

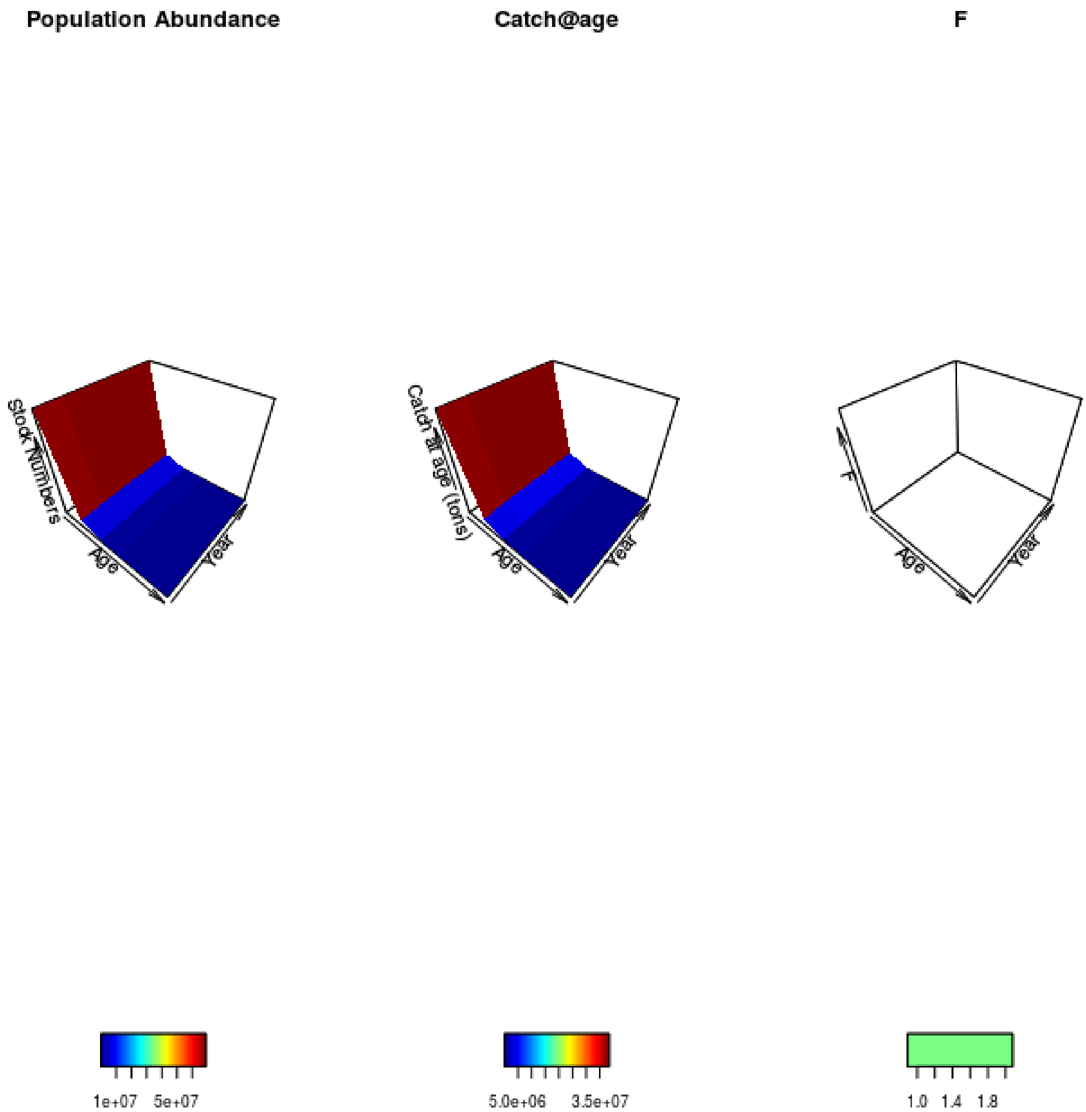


Figure 6.42: Assessment summary

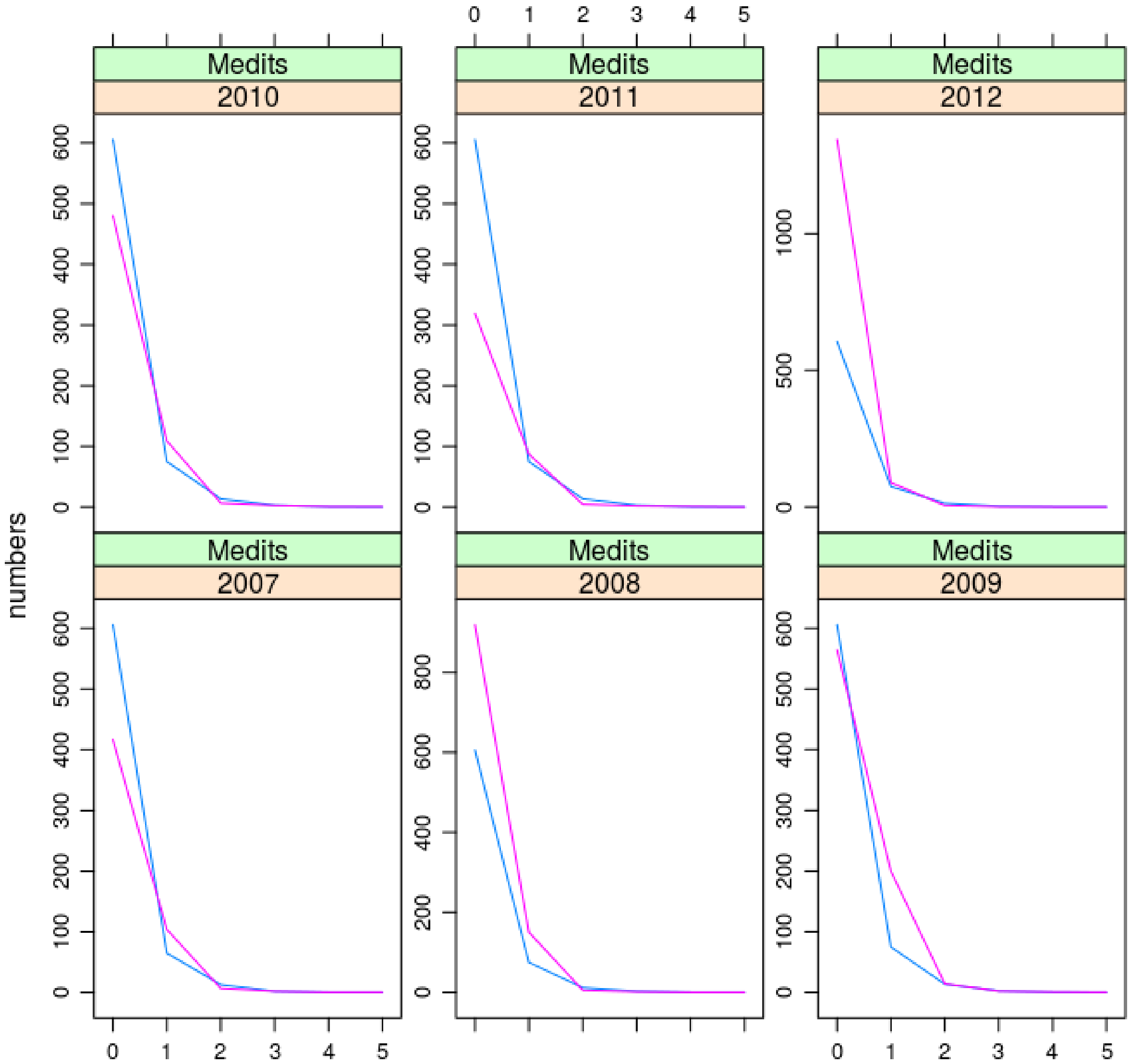


Figure 6.43: Plot fitted against observed for survey

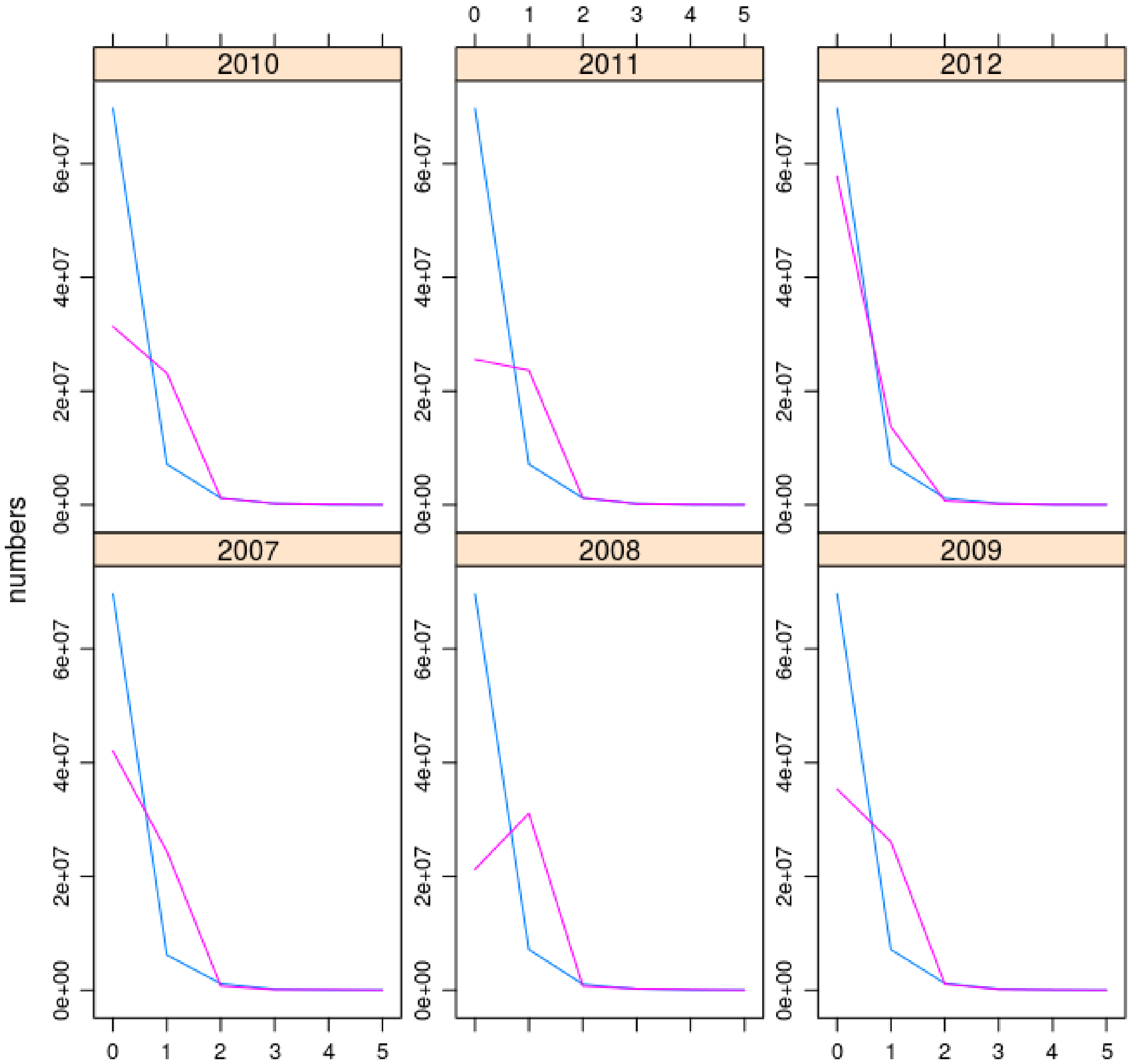


Figure 6.44: Plot fitted agains observed for catch at age

Model 1

Simple modelling with categorical factors. Gives very high F's...

```
Hake18.stk <- hketemp

qmod1 <- list(~factor(age))
fmod1 <- ~factor(age) + factor(year)
srmod1 <- ~factor(year)
fit1 <- sca(stock = Hake18.stk, indices = Hake18.S.Ind.new, fmodel = fmod1,
    qmodel = qmod1, srmodel = srmod1)

Hake18.stk.a4a.1 <- Hake18.stk + fit1
```

```
landings(Hake18.stk.a4a.1) <- computeLandings(Hake18.stk.a4a.1)
```

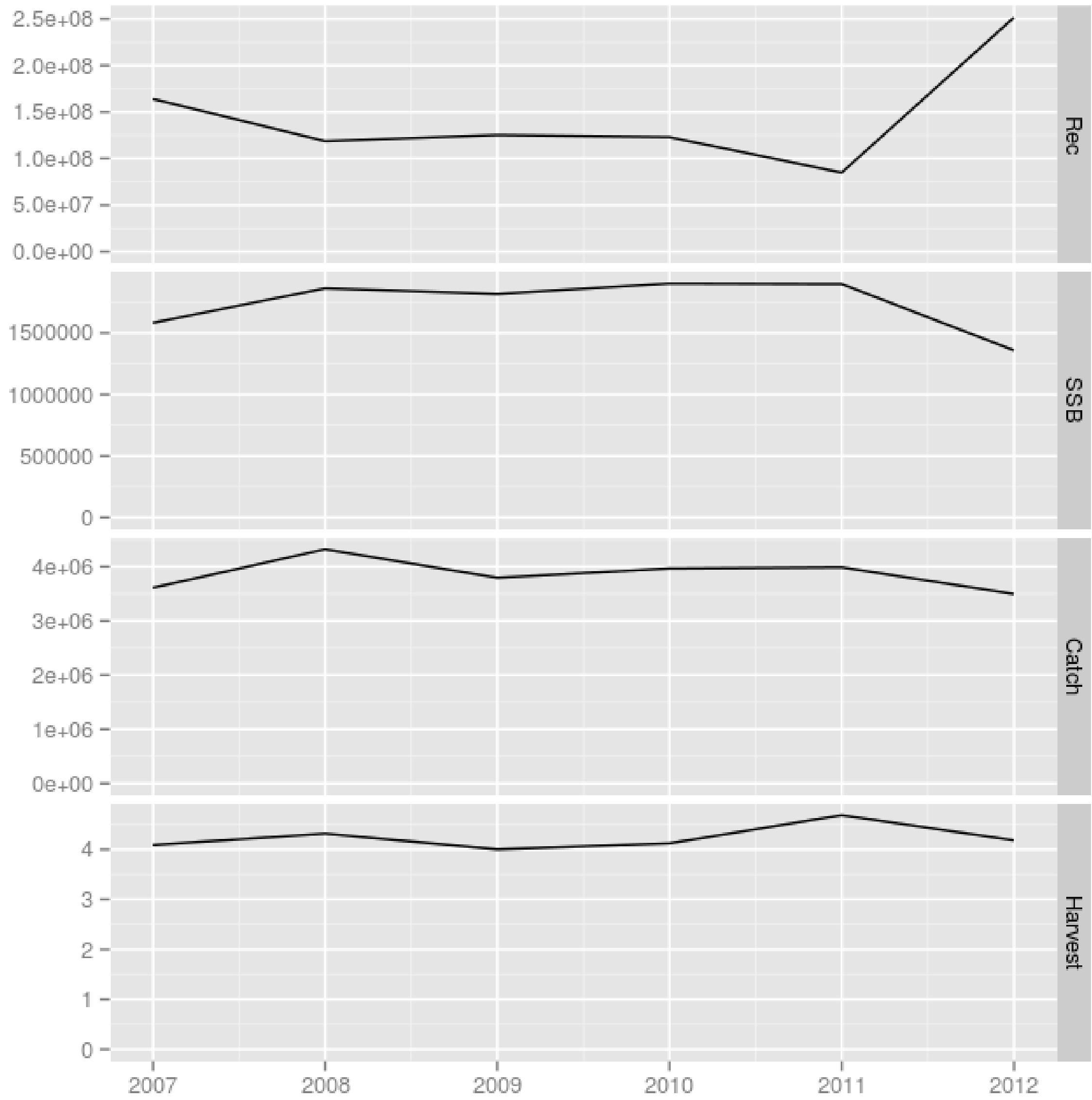


Figure 6.45: Stock summary

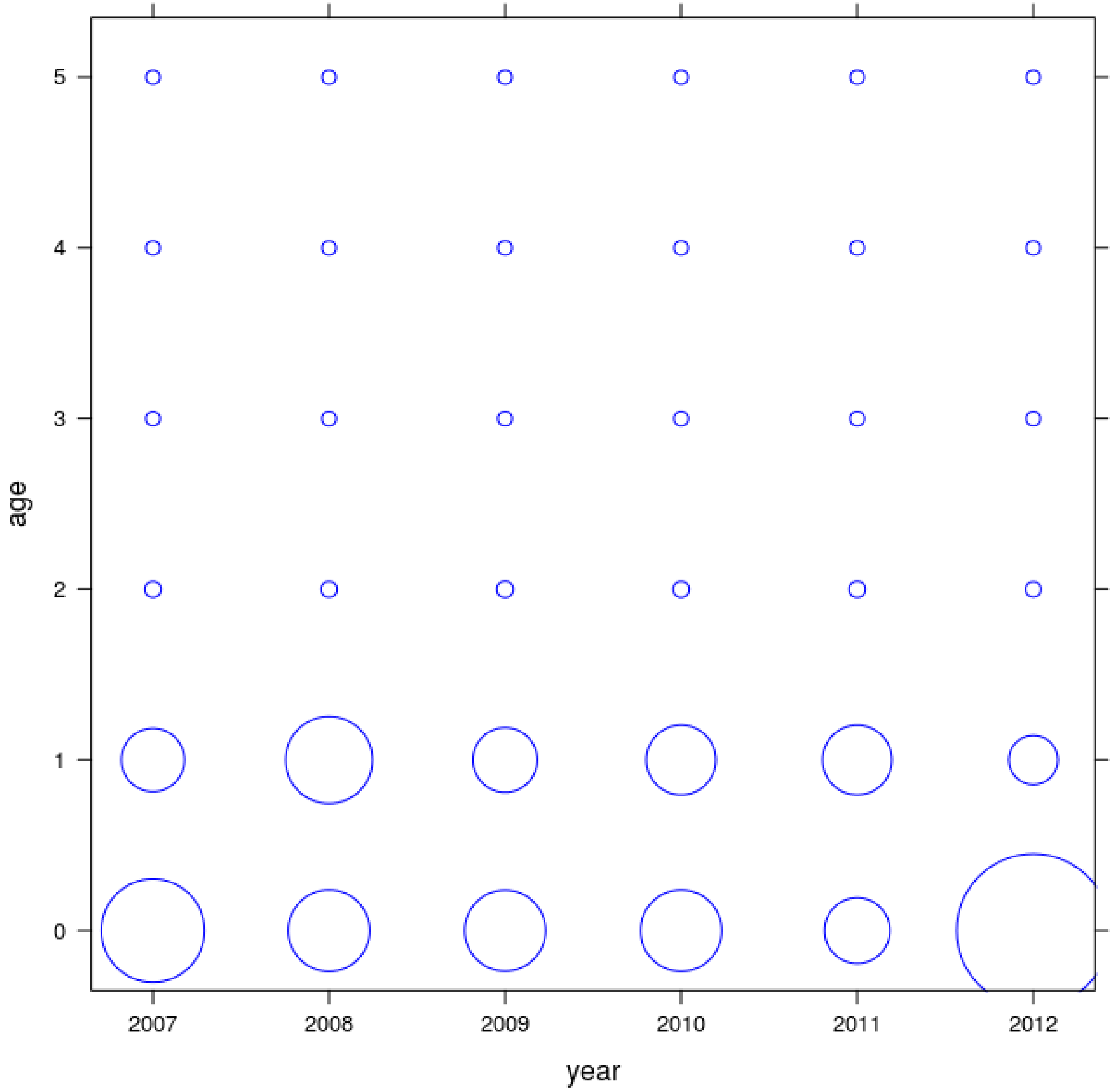


Figure 6.46: The catch matrix

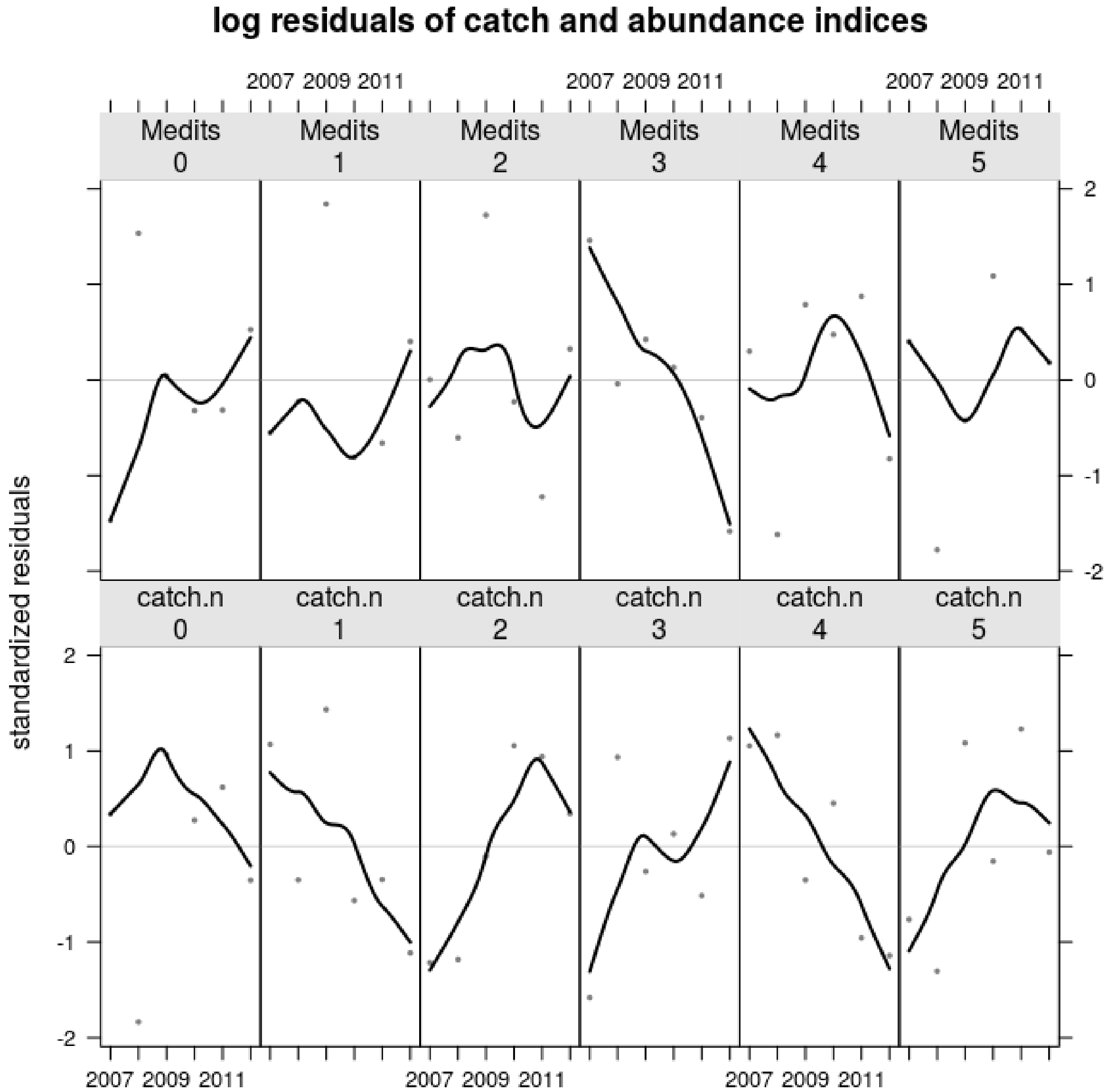


Figure 6.47: Residuals

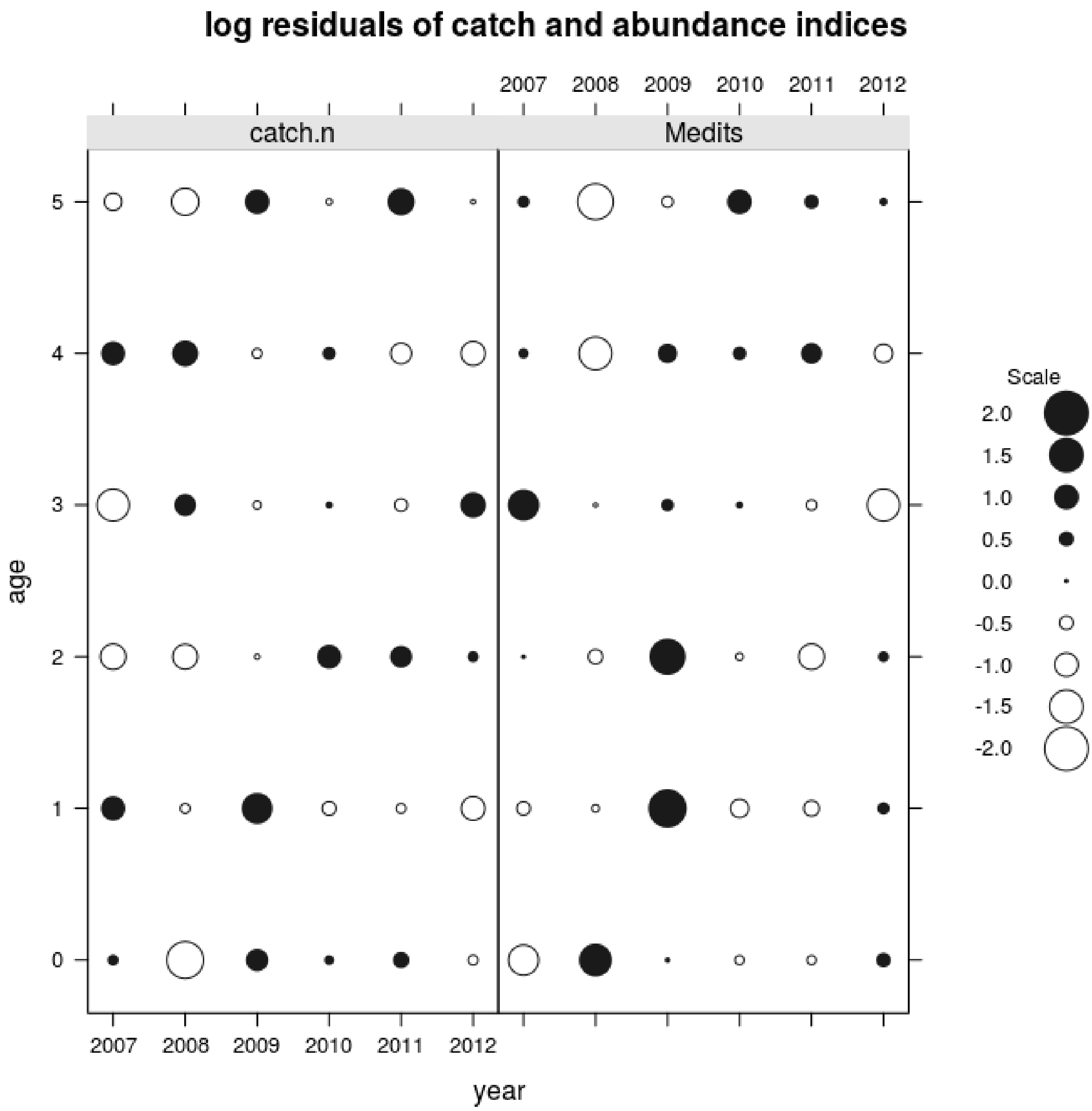


Figure 6.48: Residuals by age and year

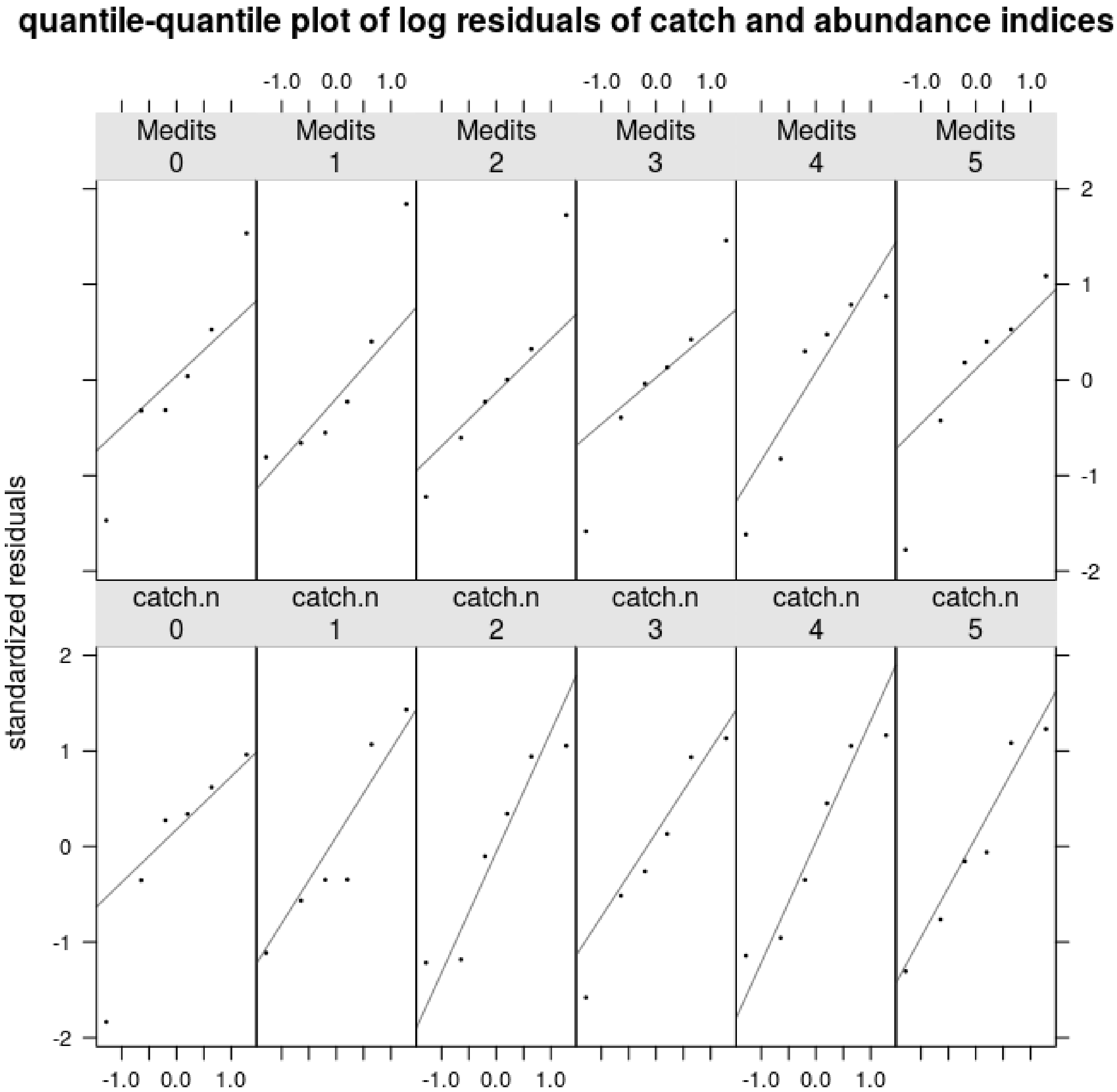


Figure 6.49: Quantile-quantile plot of residuals

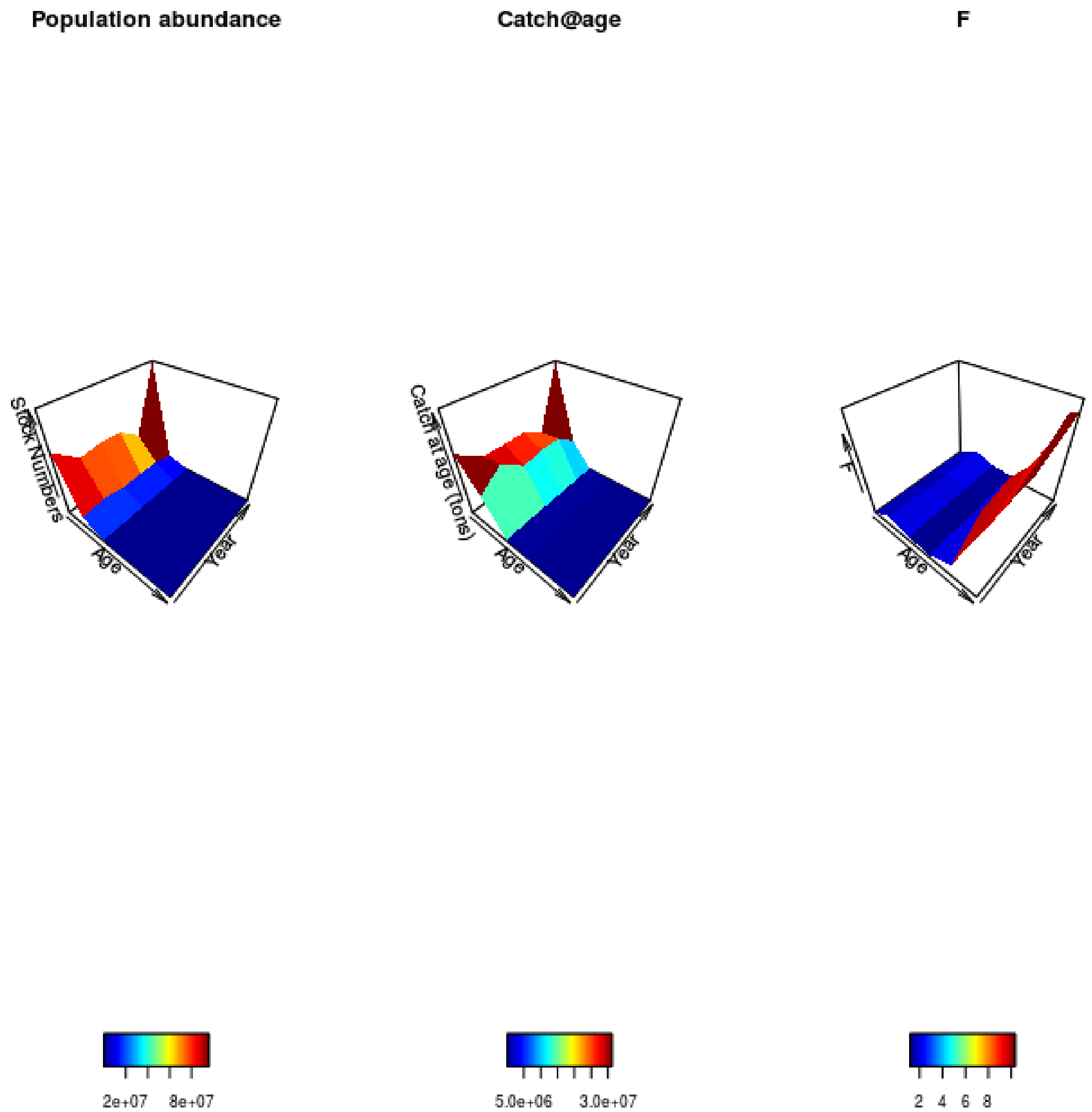


Figure 6.50: Assessment summary

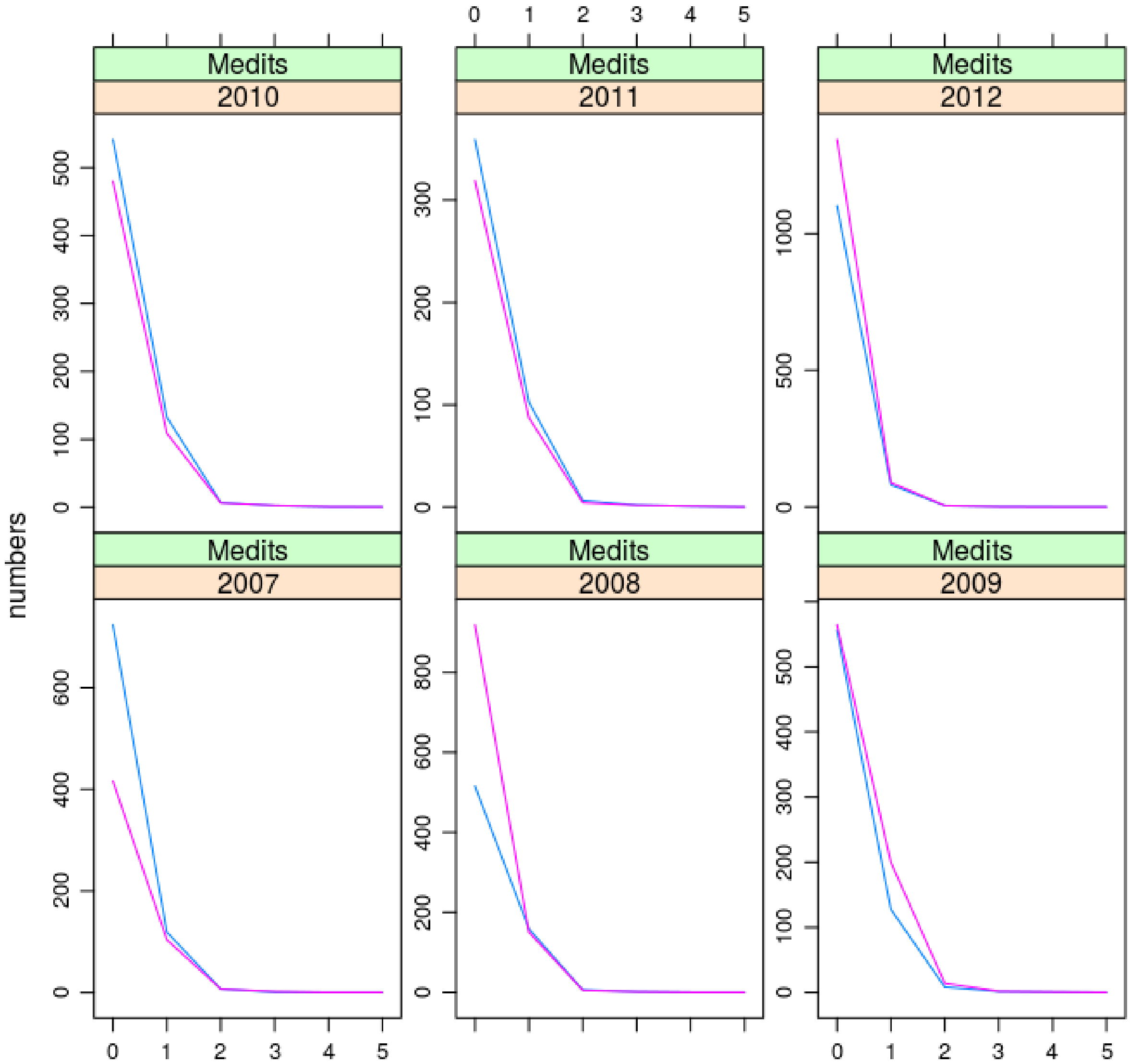


Figure 6.51: Plot fitted against observed for survey

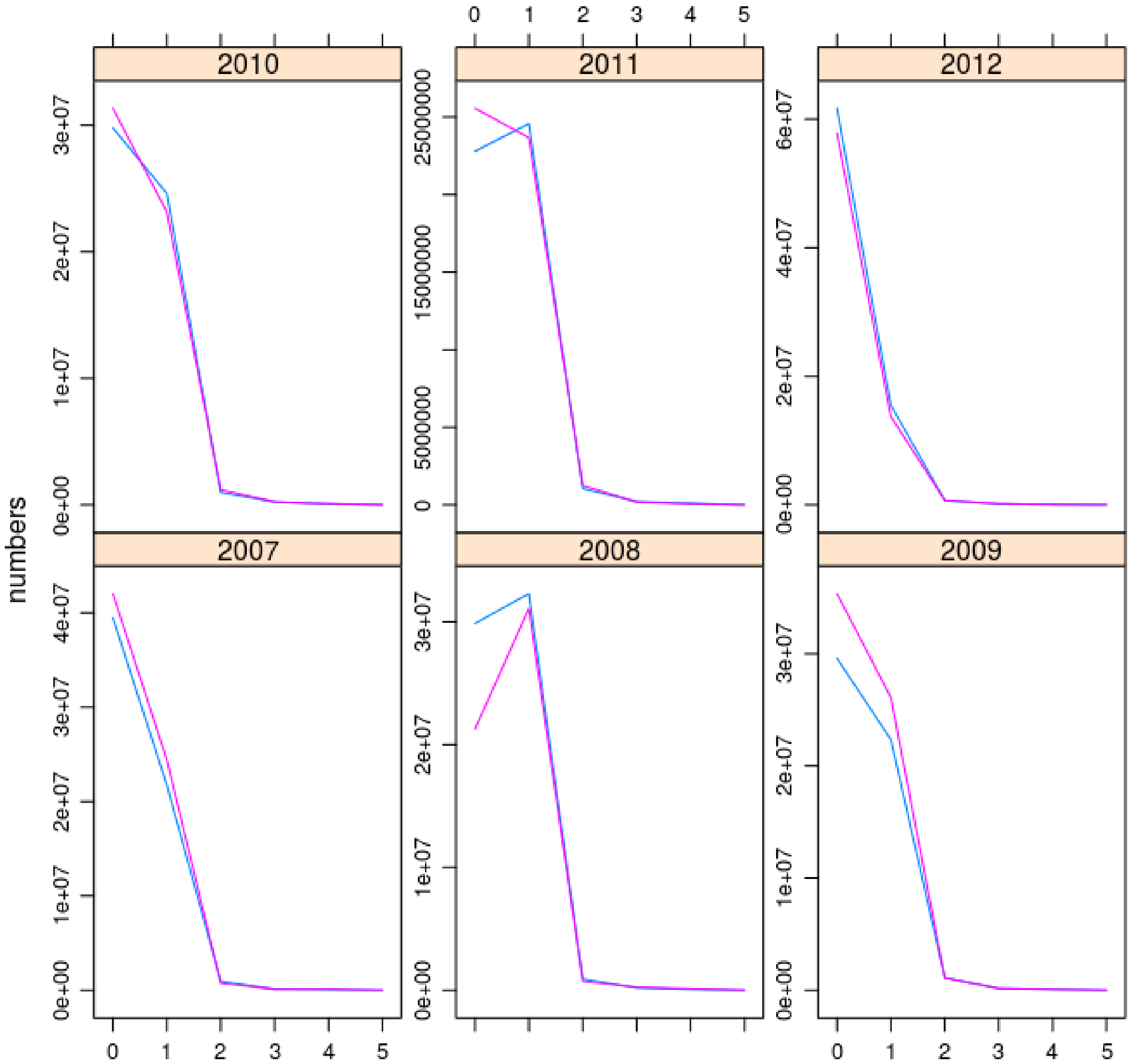


Figure 6.52: Plot fitted agains observed for catch at age

Model 2

Modelling with smoothers. Why not? (Reasonable trends & F values). Worst from AIC-BIC point of view...

```
Hake18.stk <- hketemp
qmod2 <- list(~s(age, k = 3))
fmod2 <- ~s(age, k = 6) + s(year, k = 6)
srmod2 <- ~s(year, k = 2)
fit2 <- sca(stock = Hake18.stk, indices = Hake18.S.Ind.new, fmodel = fmod2,
    qmodel = qmod2, srmodel = srmod2)

Hake18.stk.a4a.2 <- Hake18.stk + fit2
```

```
landings(Hake18.stk.a4a.2) <- computeLandings(Hake18.stk.a4a.2)
```

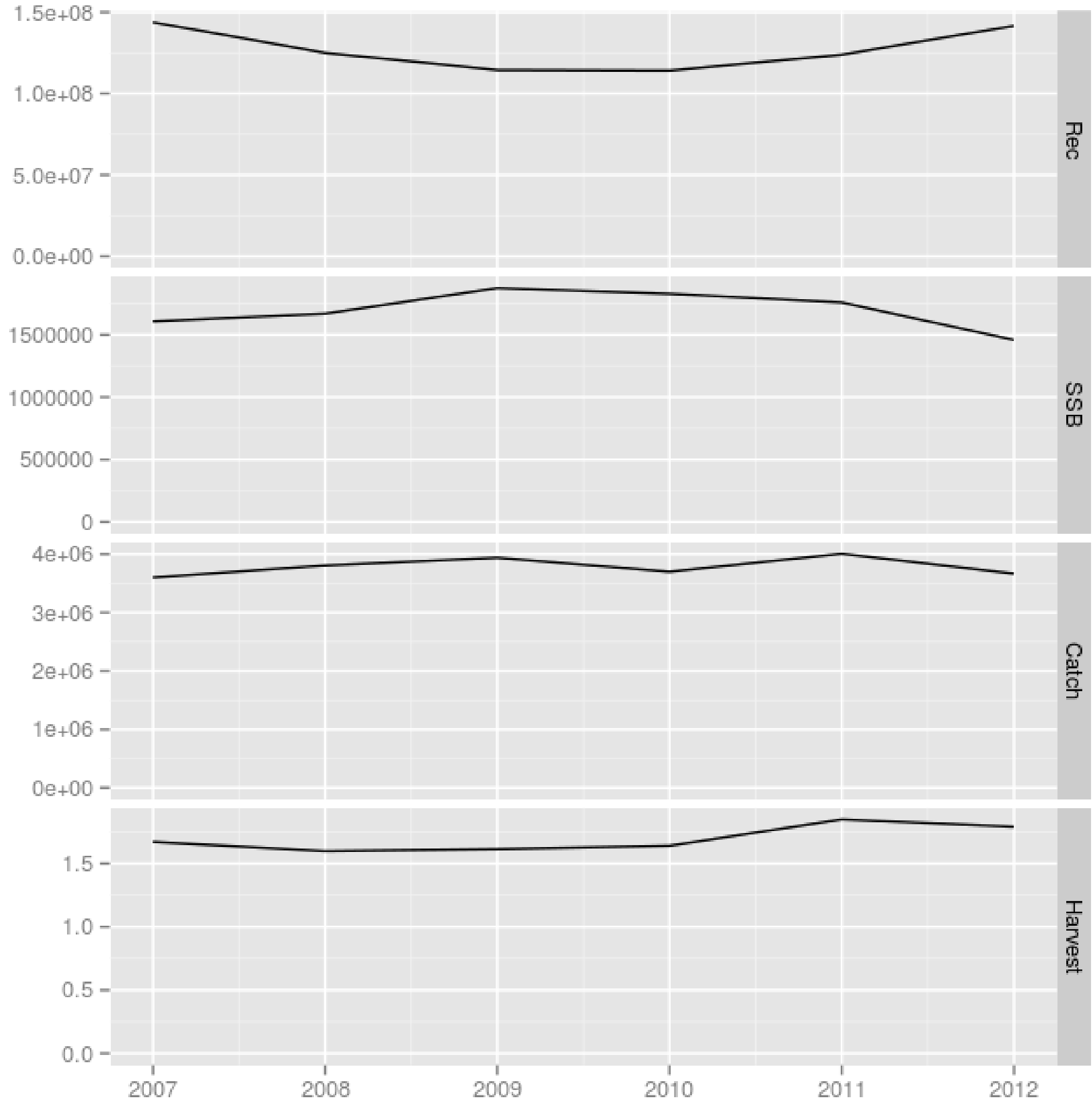


Figure 6.53: Stock summary

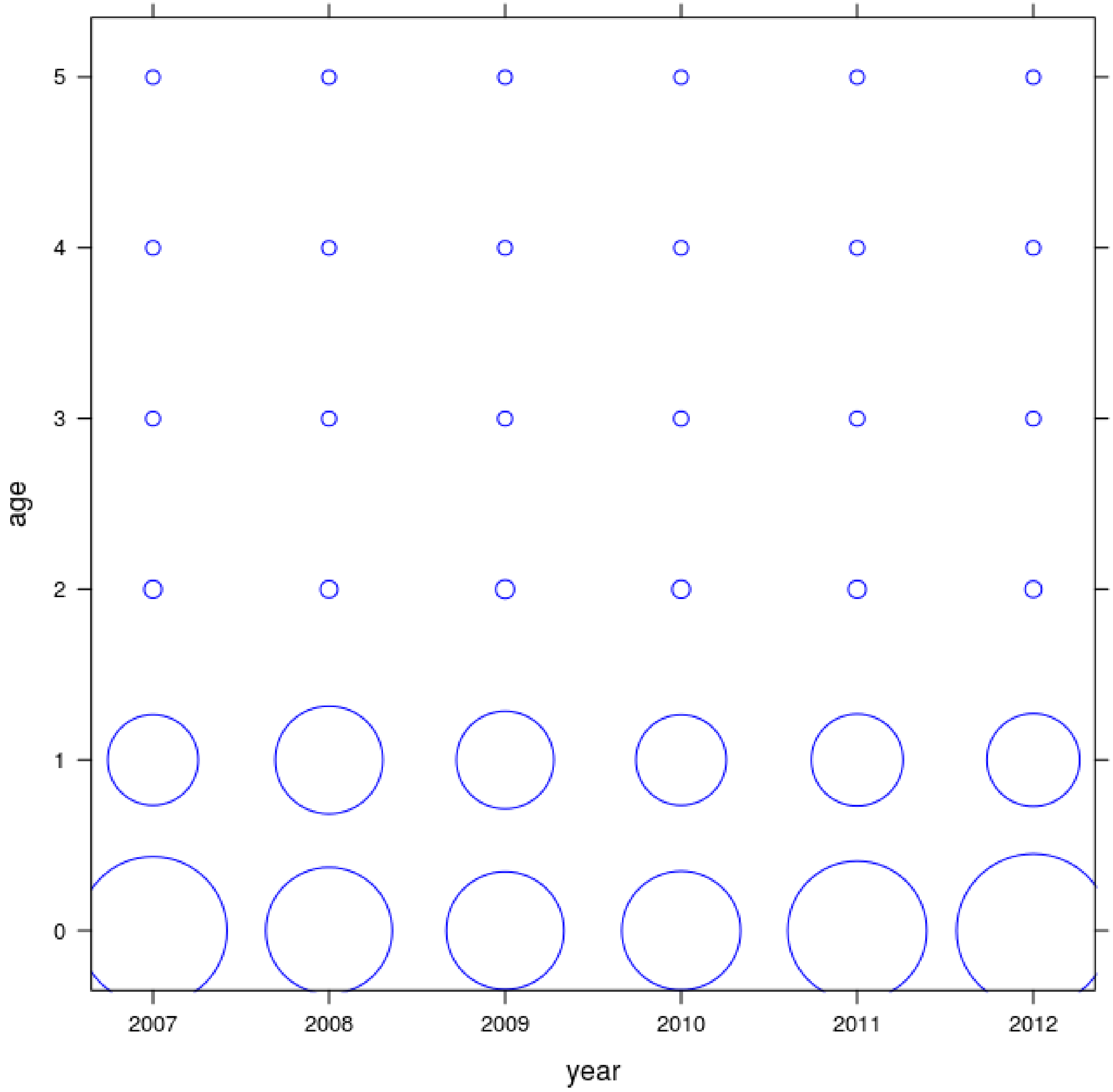


Figure 6.54: The catch matrix

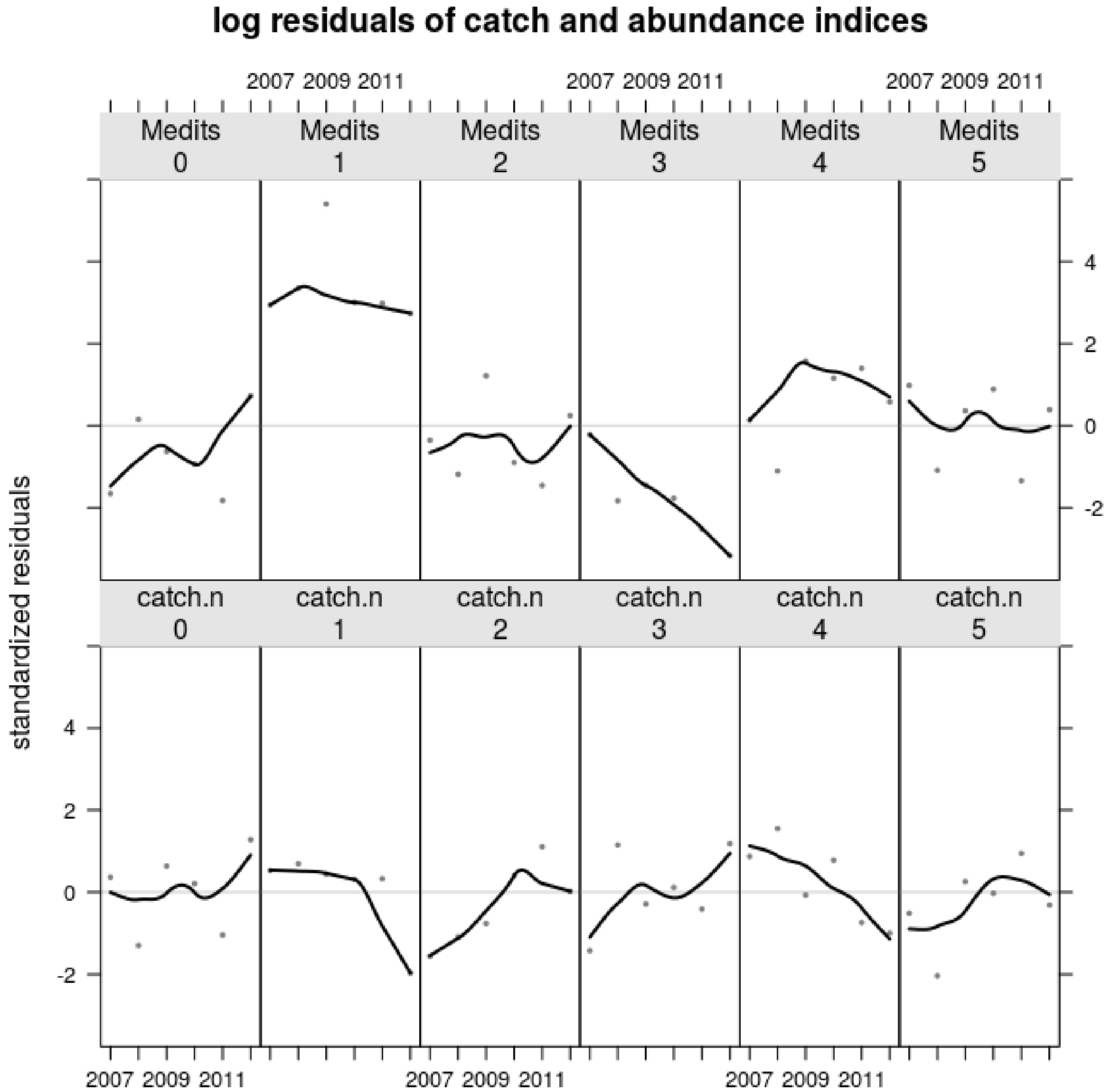


Figure 6.55: Residuals

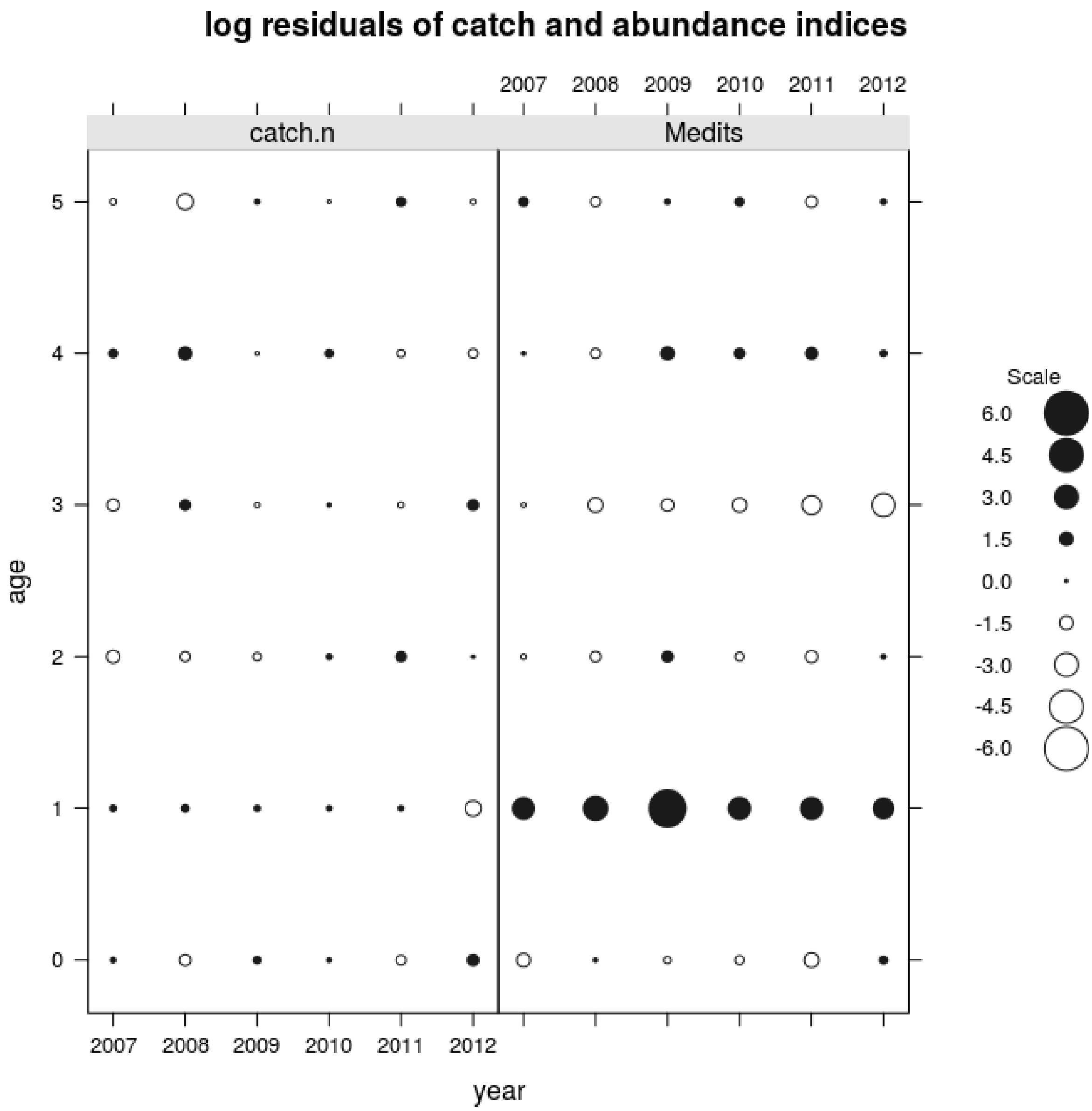


Figure 6.56: Residuals by age and year

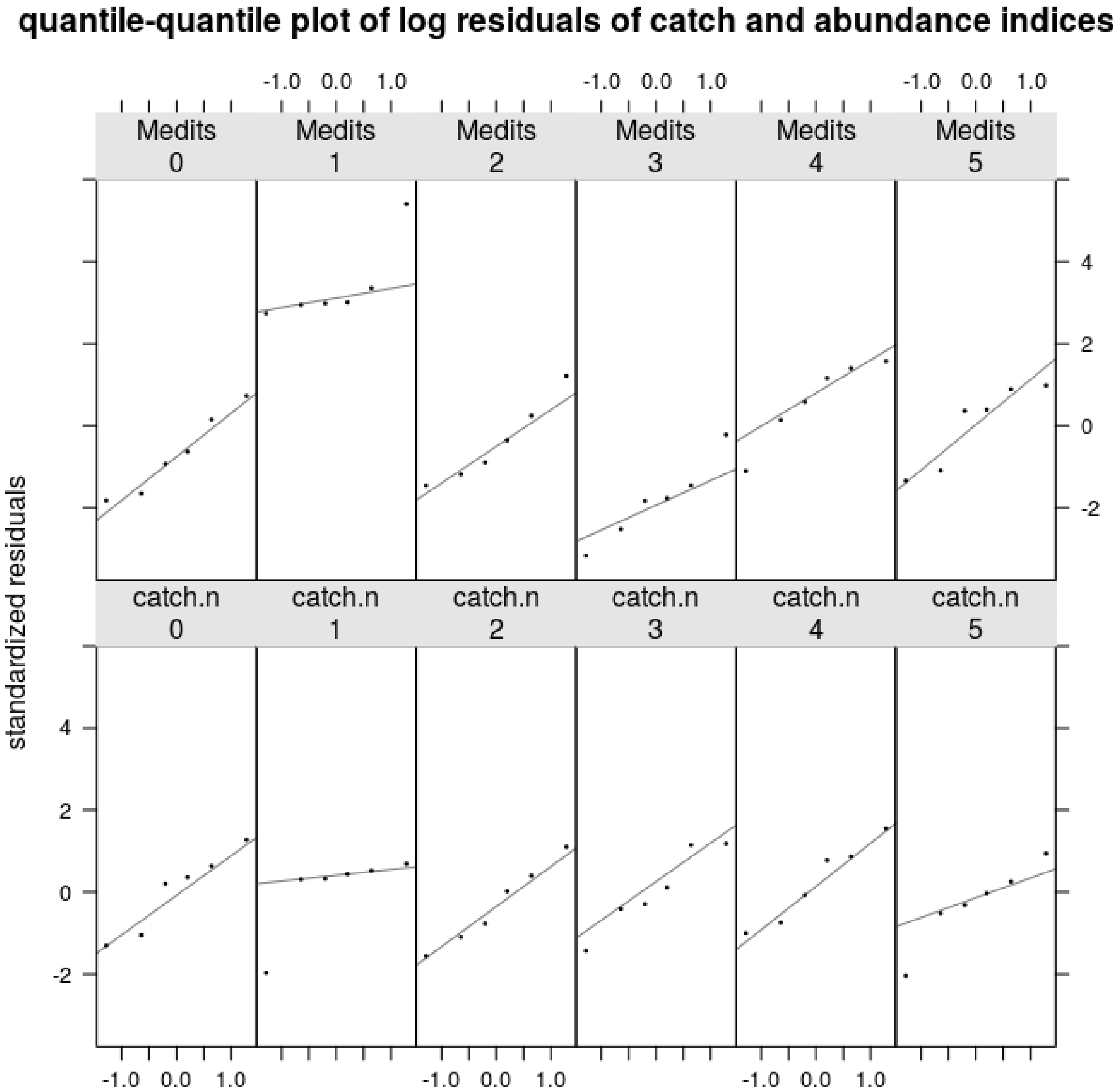


Figure 6.57: Quantile-quantile plot of residuals

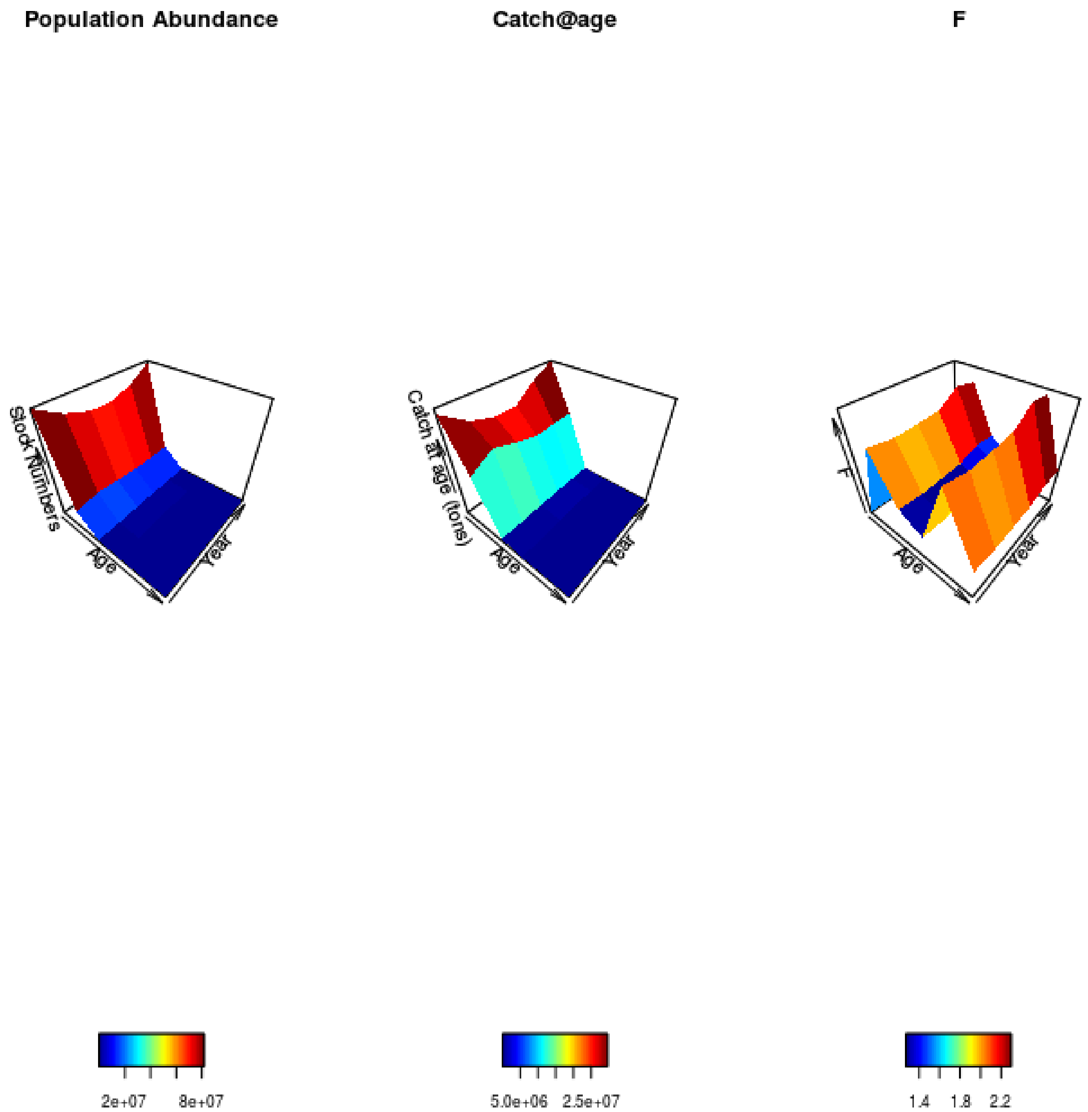


Figure 6.58: Assessment summary

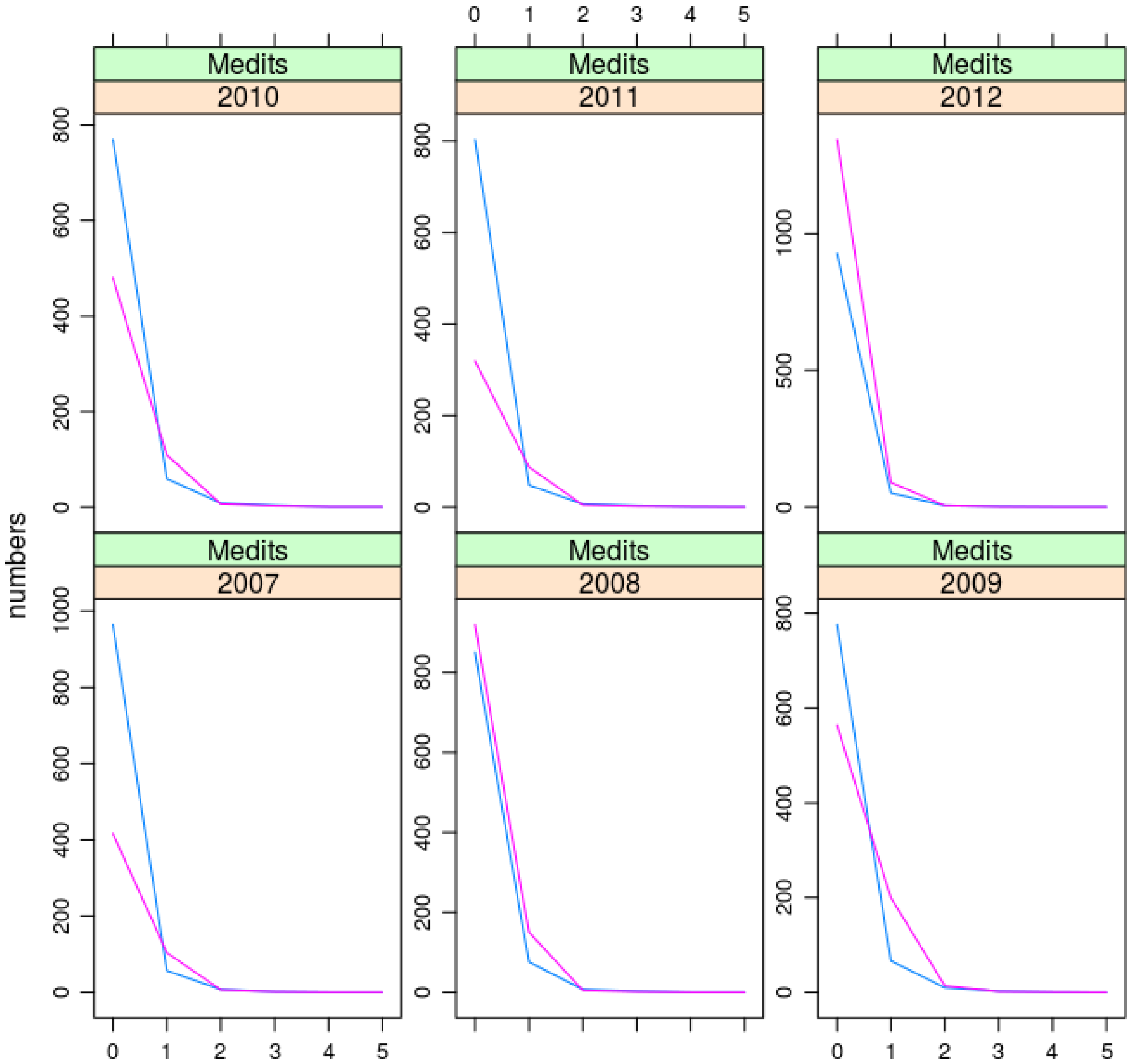


Figure 6.59: Plot fitted against observed for survey

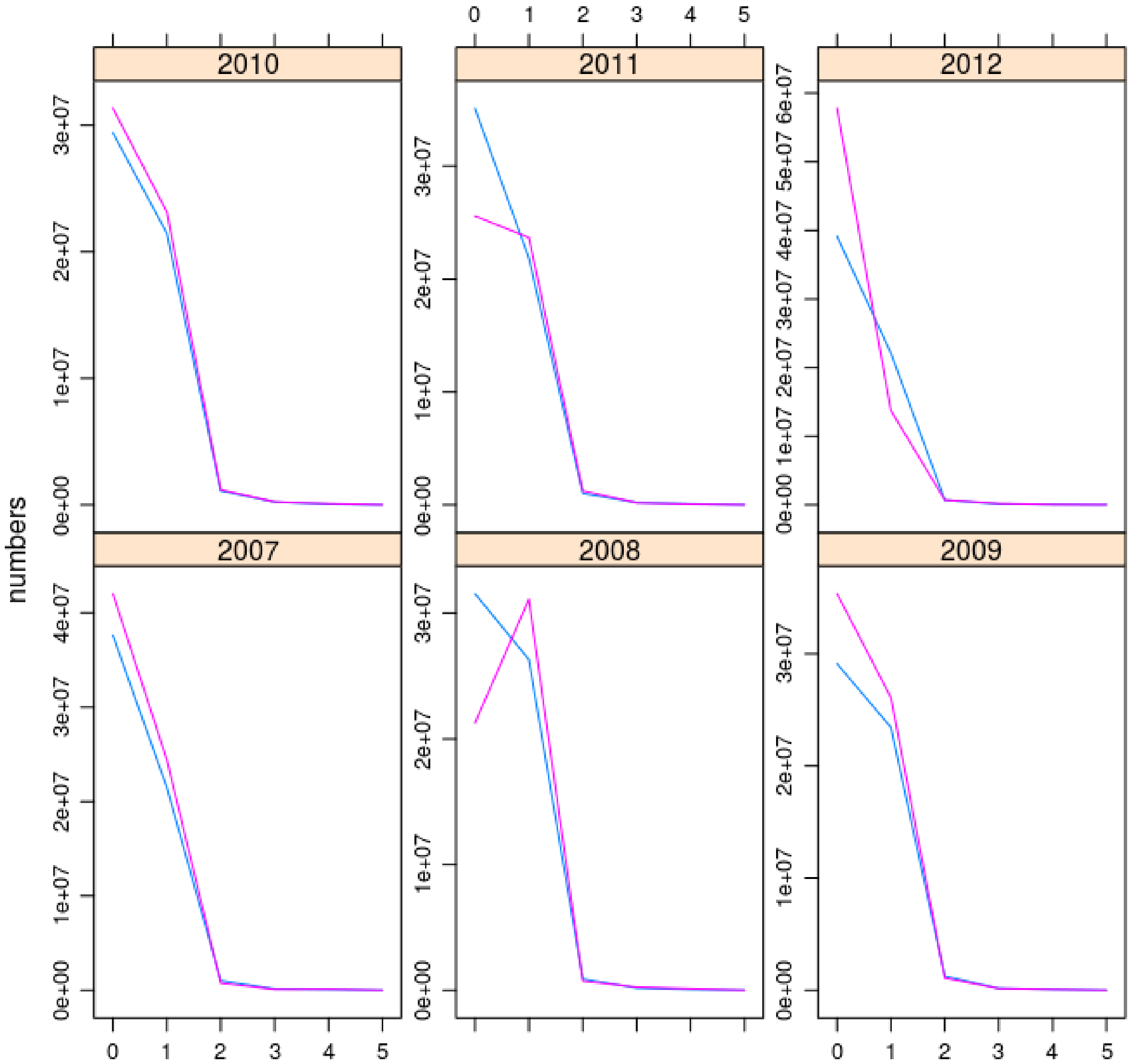


Figure 6.60: Plot fitted agains observed for catch at age

Model 3

Can't get more complex than this... te() is not running even with few knots, so use additive effects with max k's. AIC-BIC loooooooves complexity...

```
Hake18.stk <- hketemp
qmod3 <- list(~s(age, k = 6))
fmod3 <- ~s(age, k = 6) + s(year, k = 6)
srmod3 <- ~s(year, k = 6)

fit3 <- sca(stock = Hake18.stk, indices = Hake18.S.Ind.new, fmodel = fmod3,
    qmodel = qmod3, srmodel = srmod3)
```

```
Hake18.stk.a4a.3 <- Hake18.stk + fit3
landings(Hake18.stk.a4a.3) <- computeLandings(Hake18.stk.a4a.3)
```

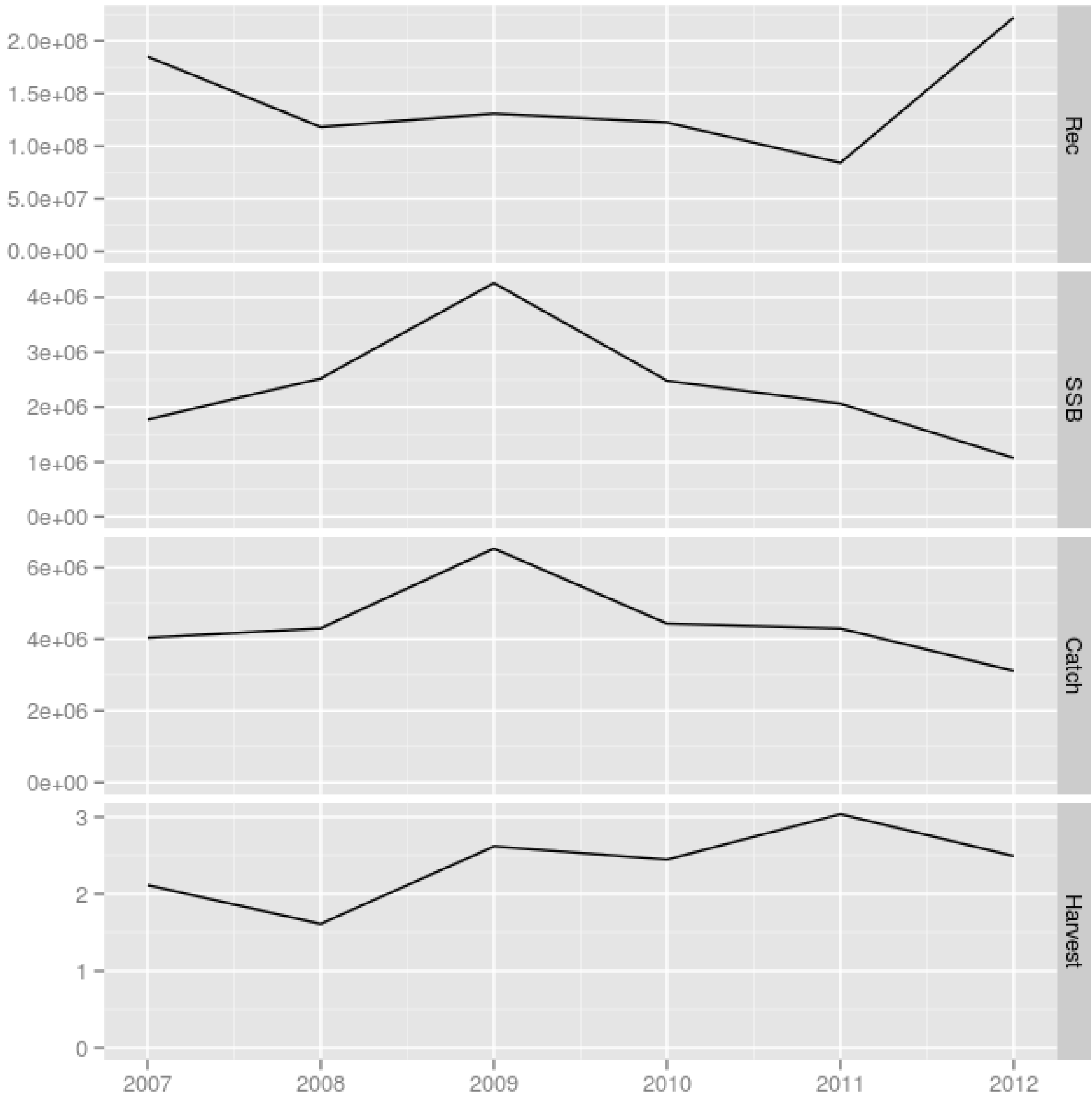


Figure 6.61: Stock summary

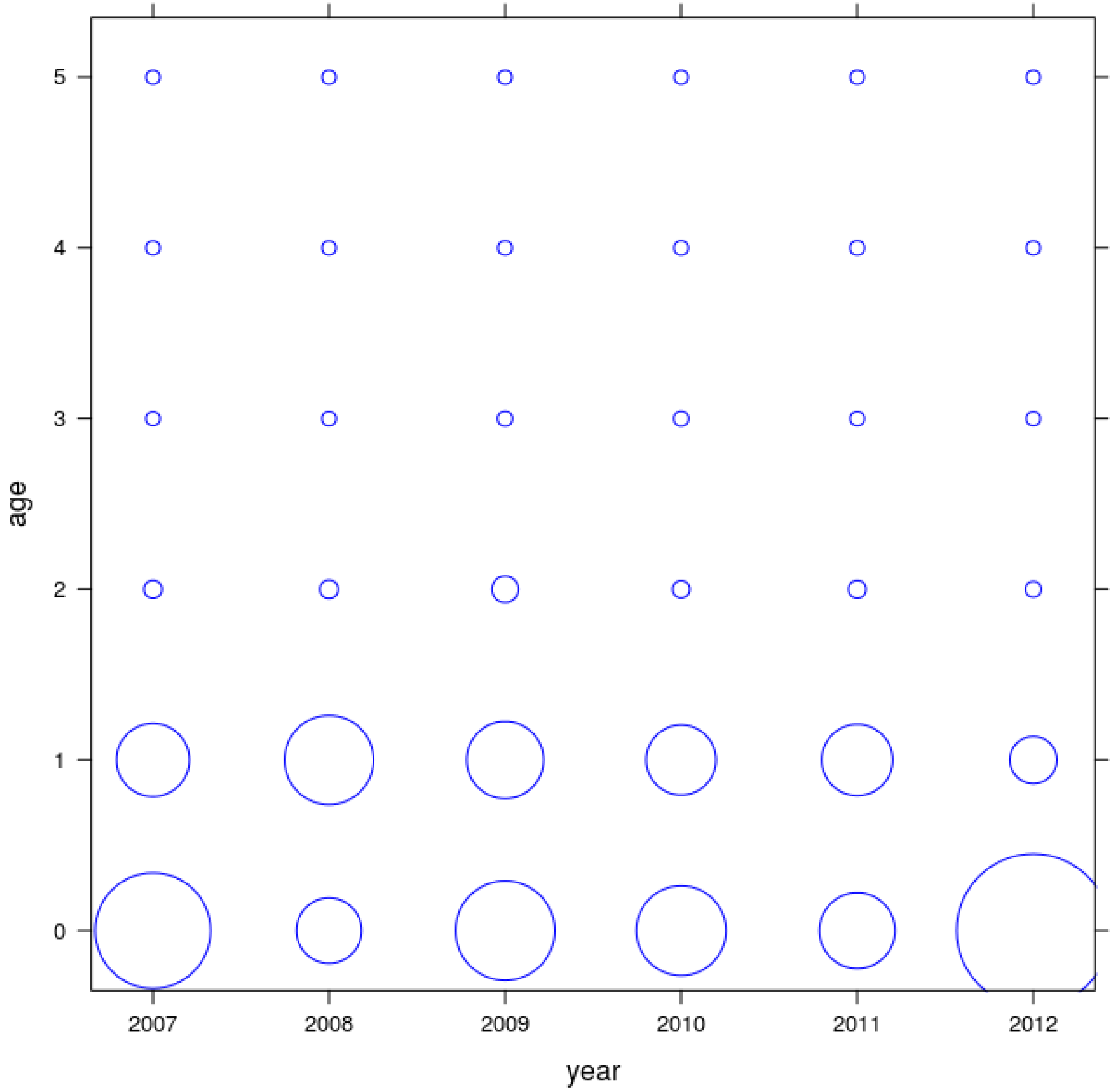


Figure 6.62: The catch matrix

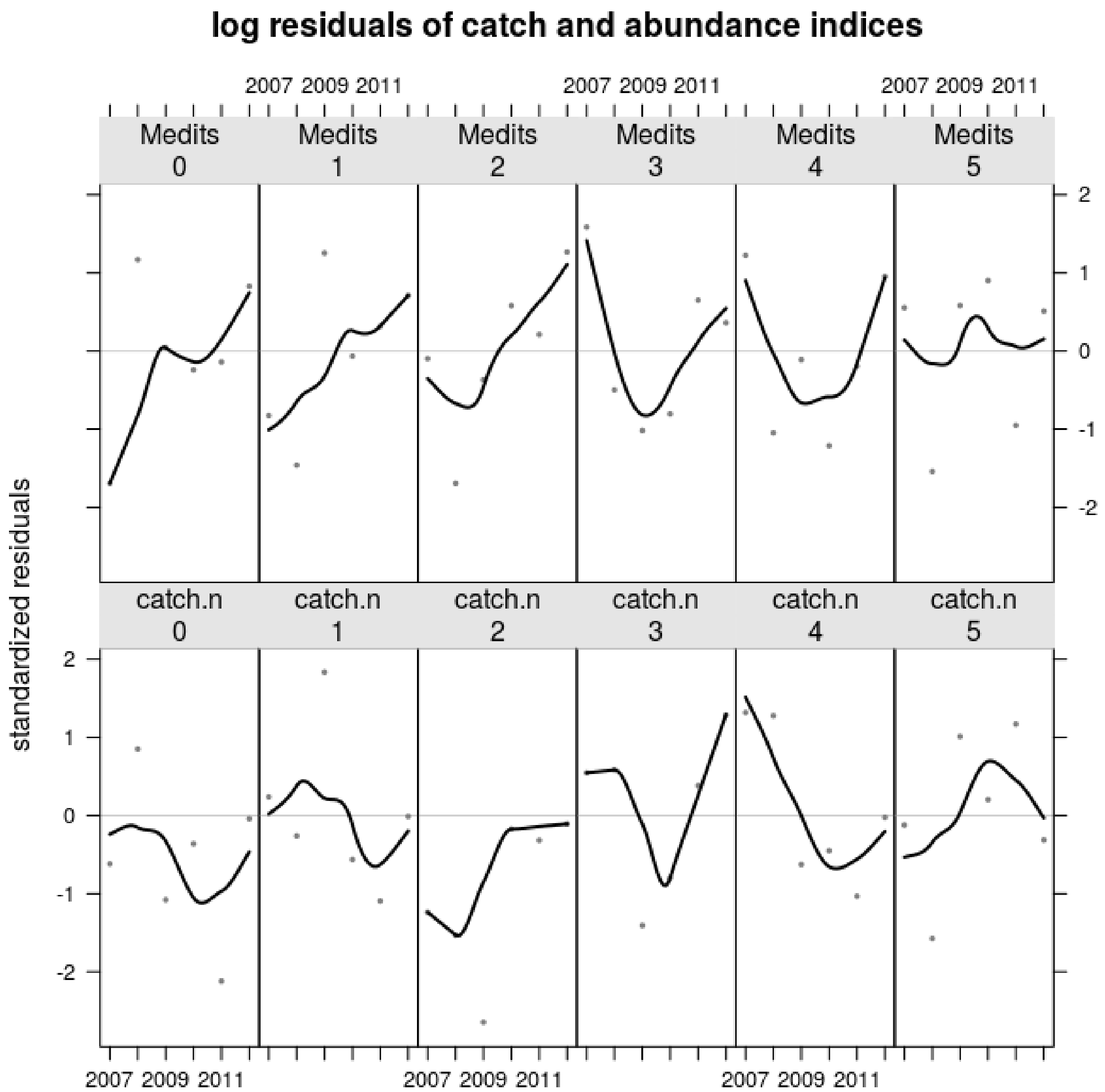


Figure 6.63: Residuals

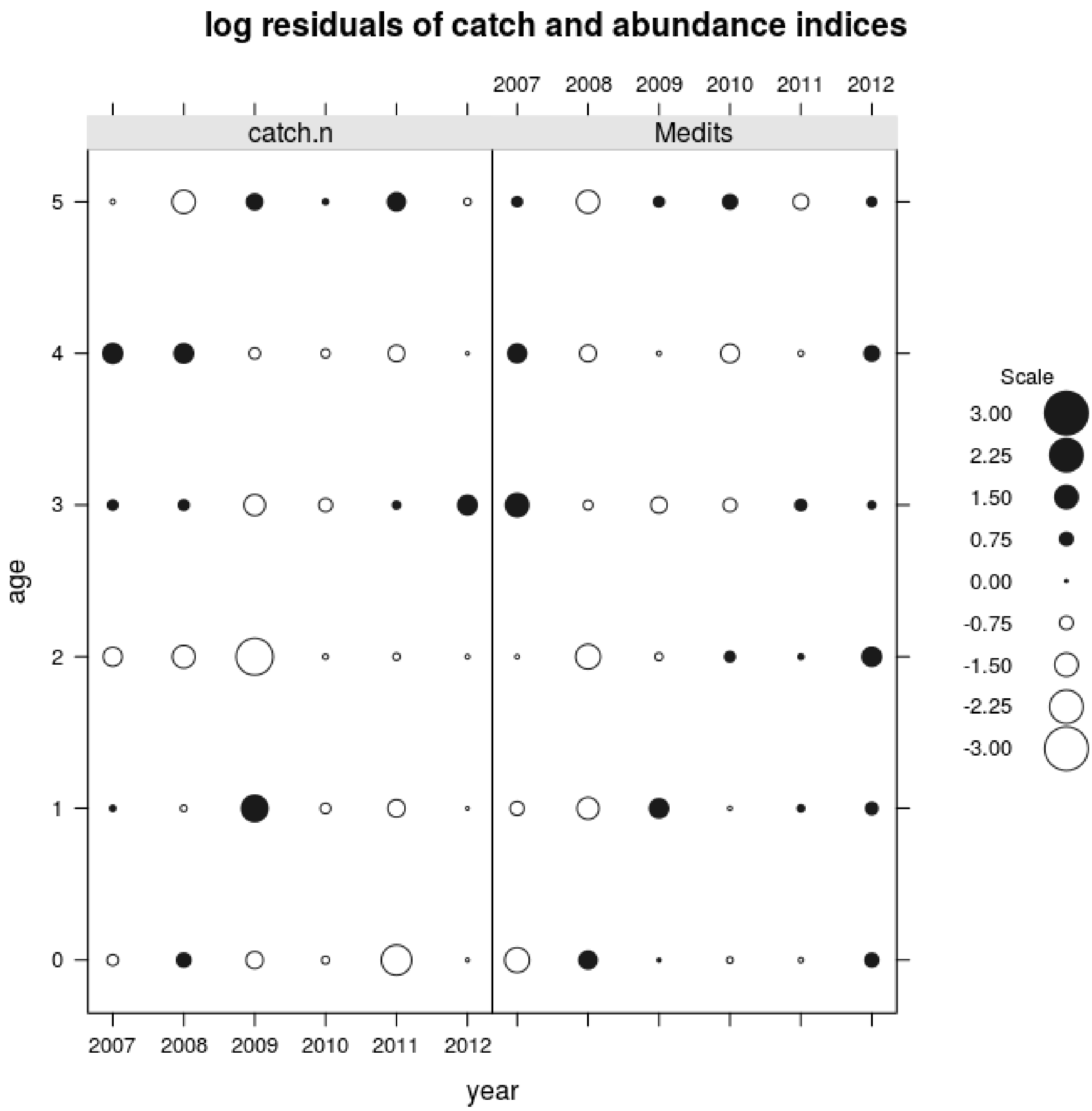


Figure 6.64: Residuals by age and year

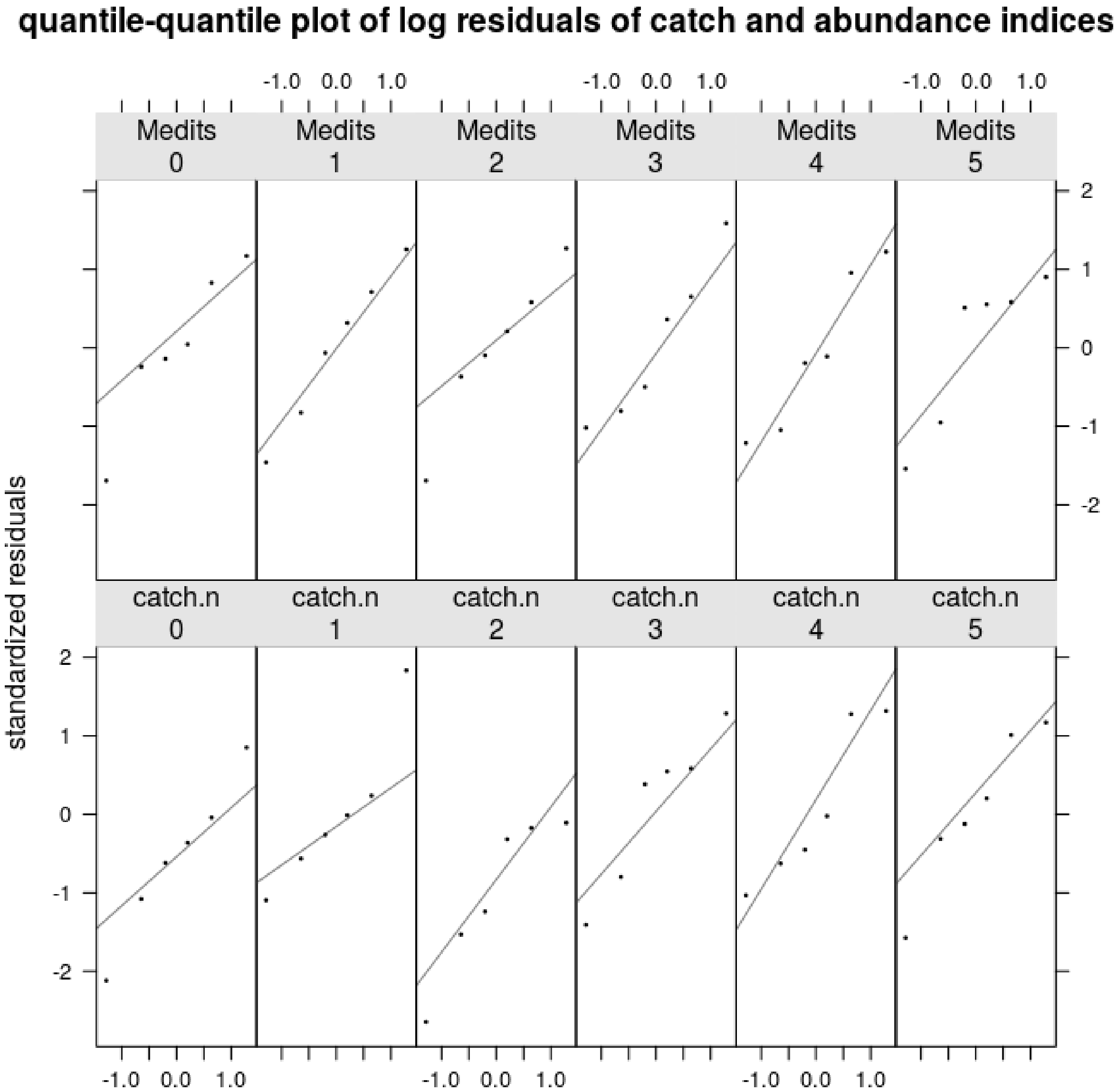


Figure 6.65: Quantile-quantile plot of residuals

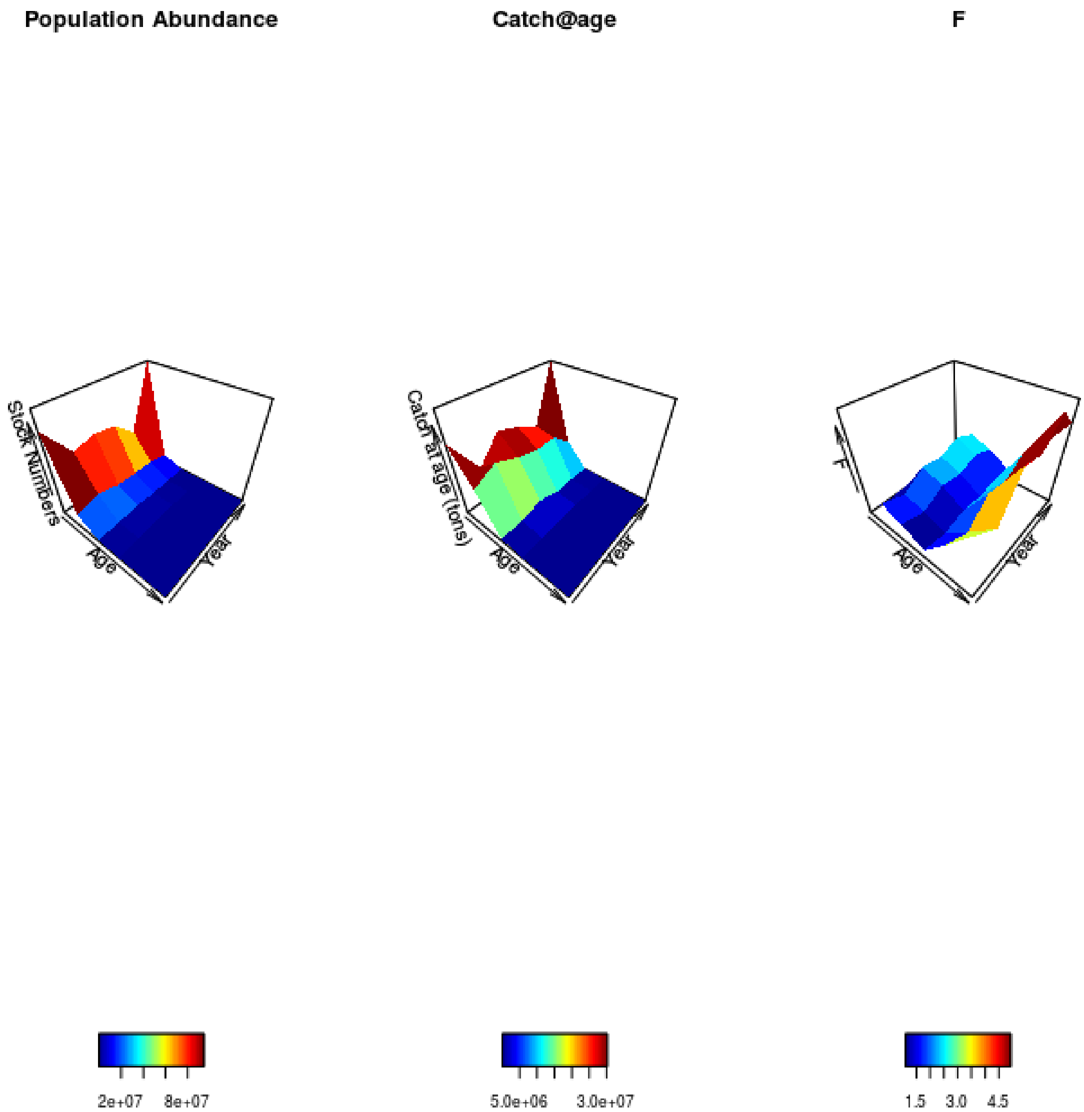


Figure 6.66: Assessment summary

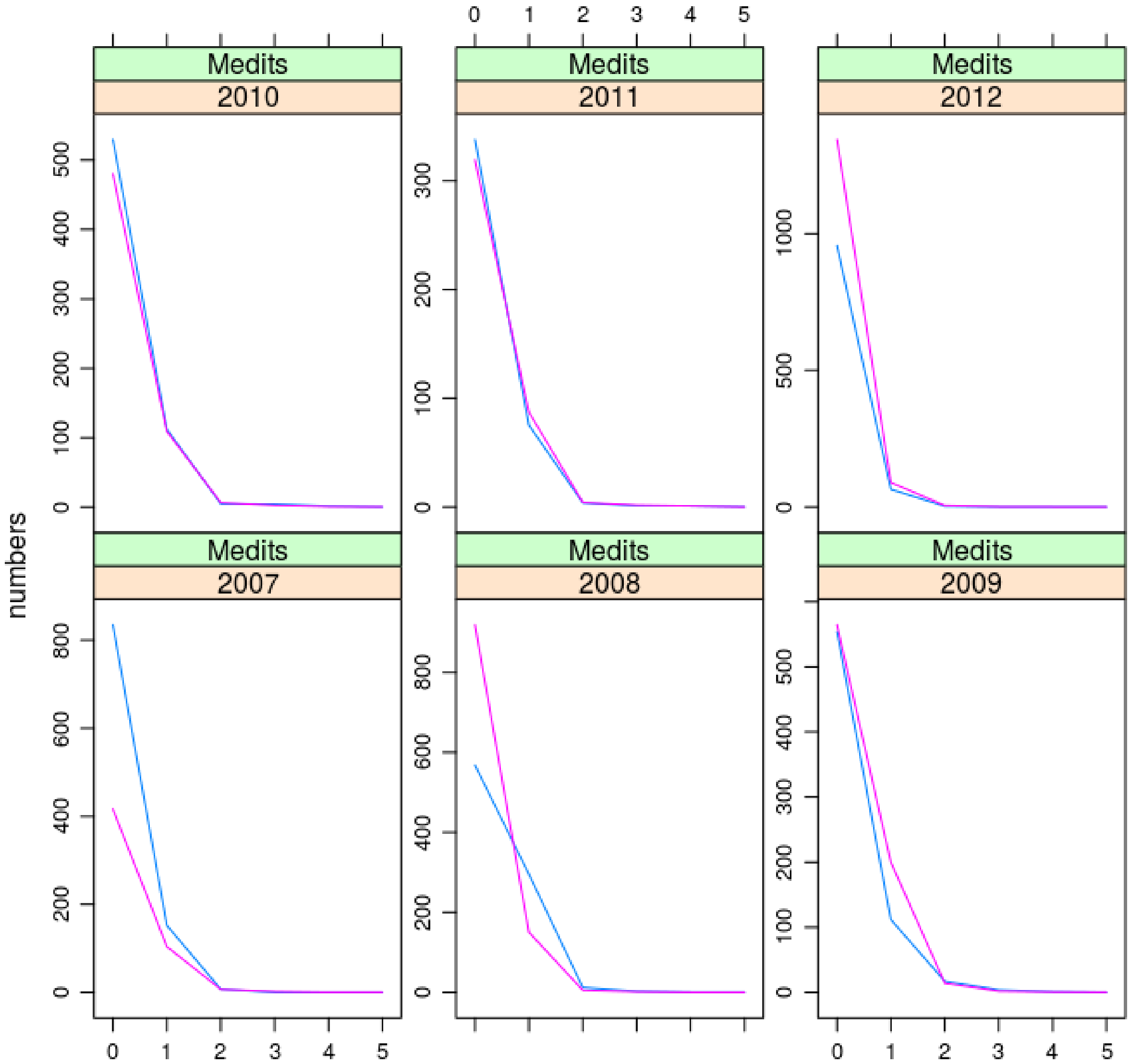


Figure 6.67: Plot fitted against observed for survey

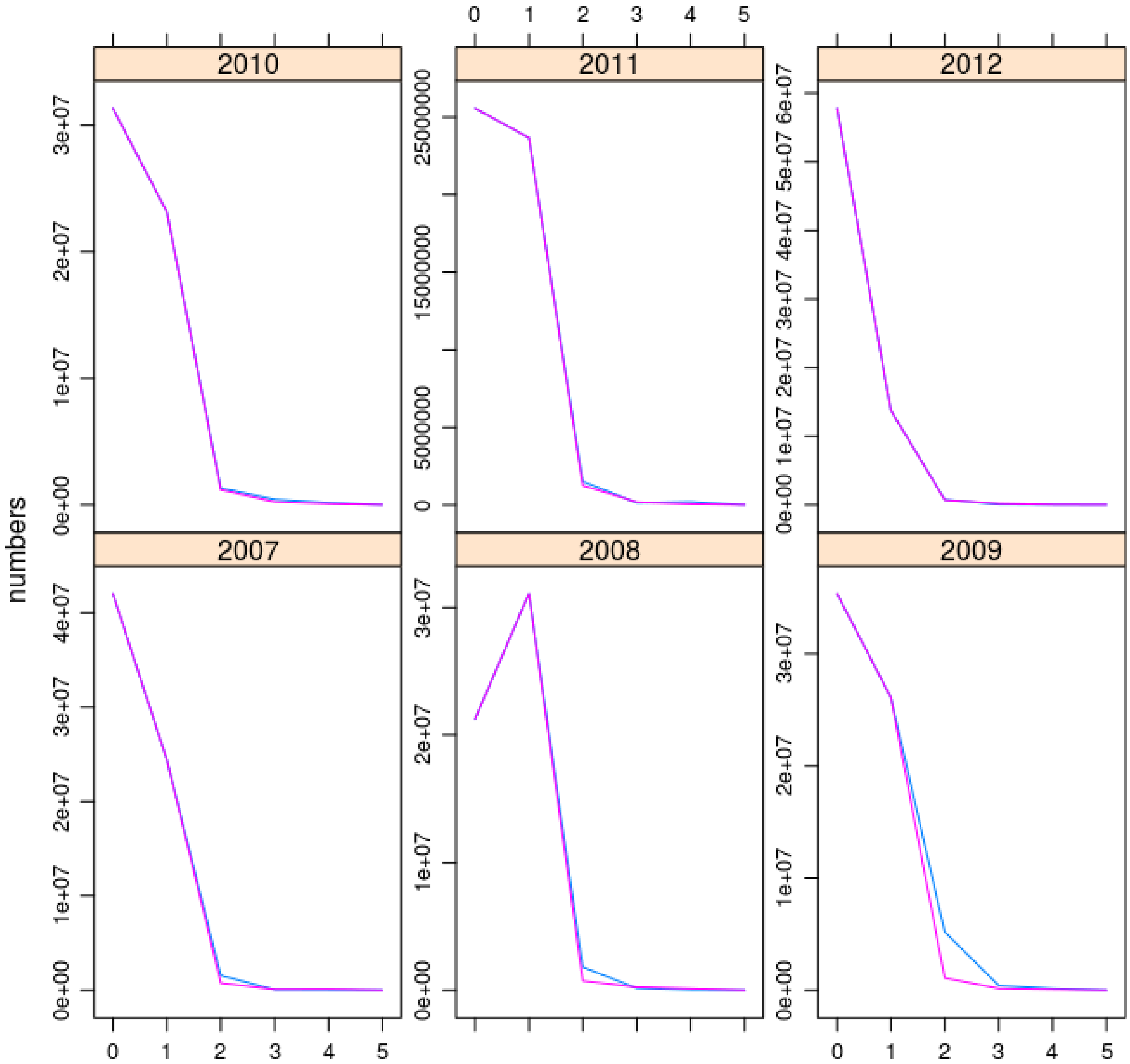


Figure 6.68: Plot fitted agains observed for catch at age

Model 4

Age as factor, year with max k. Best model (AIC-BIC)...

```
Hake18.stk <- hketemp
qmod4 <- list(~factor(age))
fmod4 <- ~factor(age) + s(year, k = 6)
srmod4 <- ~s(year, k = 6)

fit4 <- sca(stock = Hake18.stk, indices = Hake18.S.Ind.new, fmodel = fmod4,
    qmodel = qmod4, srmodel = srmod4)

Hake18.stk.a4a.4 <- Hake18.stk + fit4
```

```
landings(Hake18.stk.a4a.4) <- computeLandings(Hake18.stk.a4a.4)
```

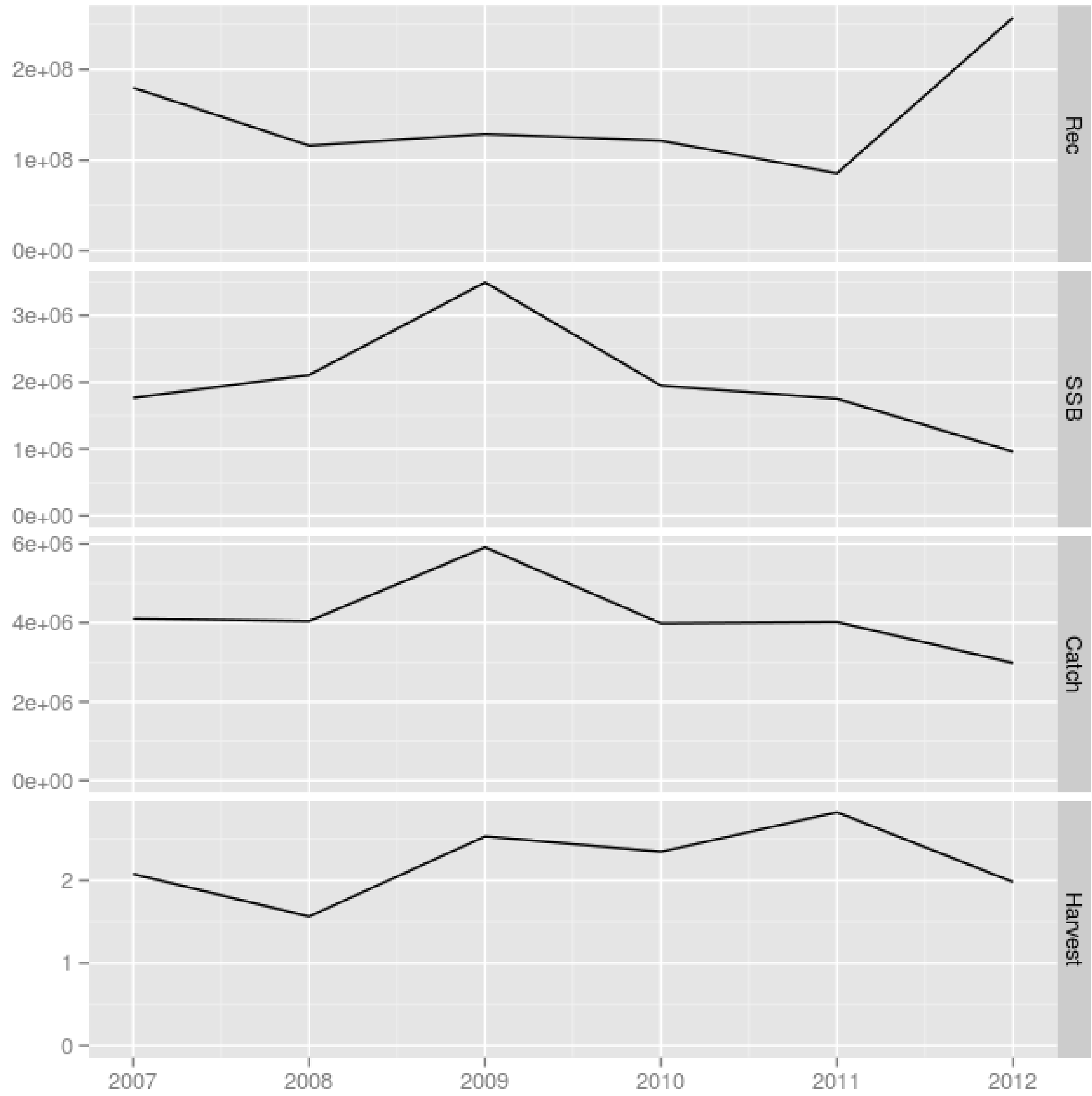


Figure 6.69: Stock summary

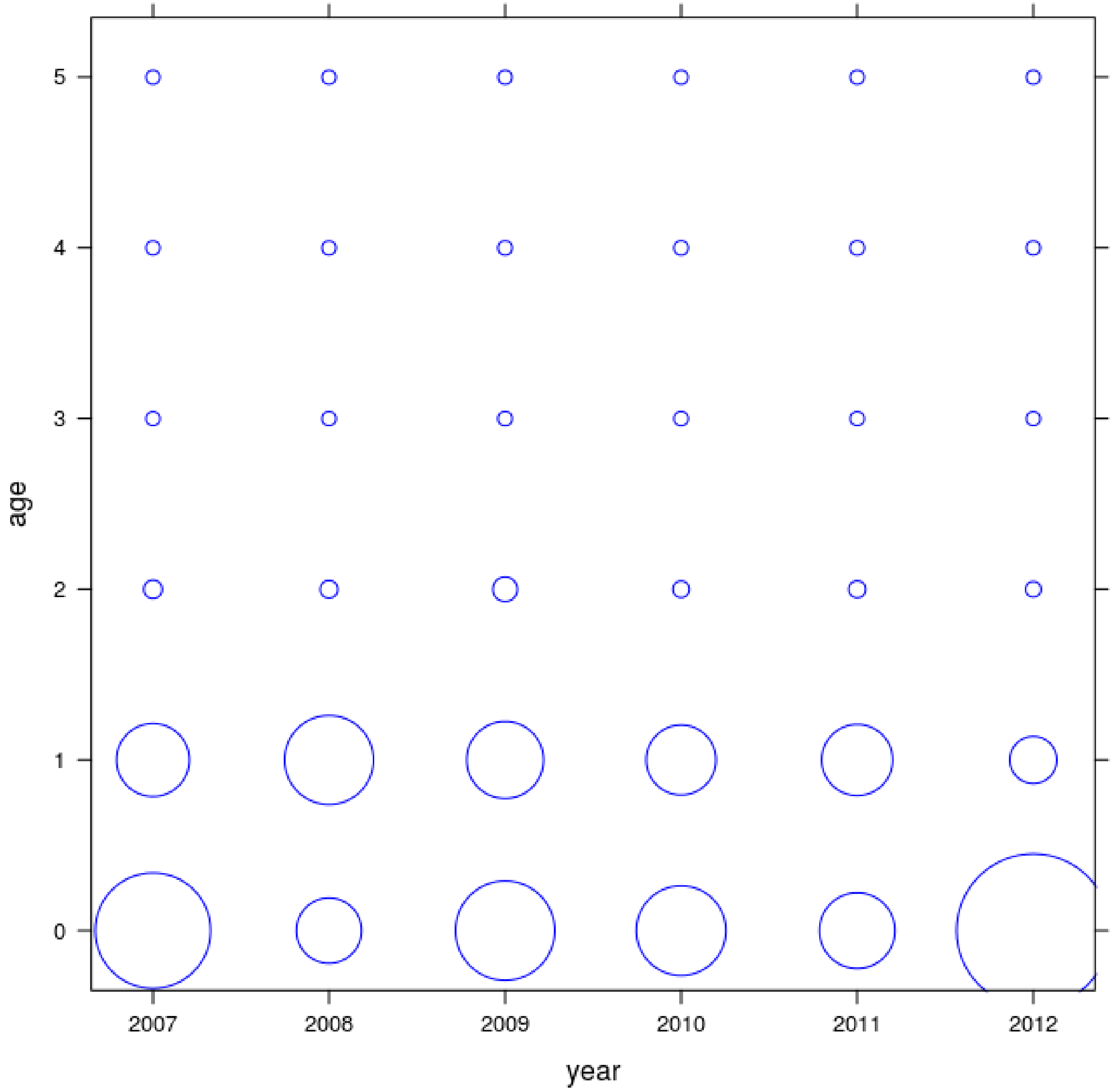


Figure 6.70: The catch matrix

Much better diagnostics with this model.

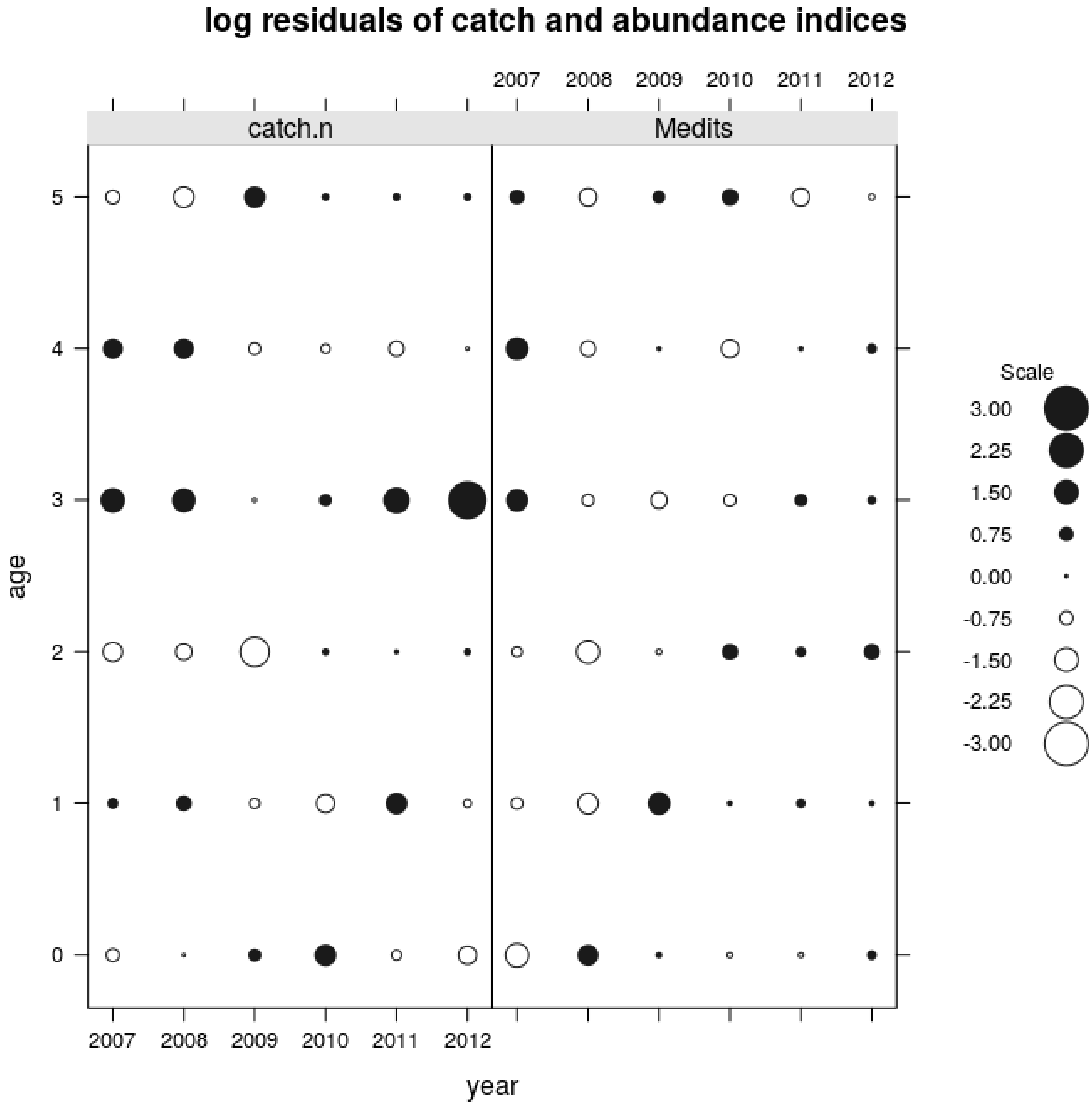


Figure 6.71: Residuals by age and year

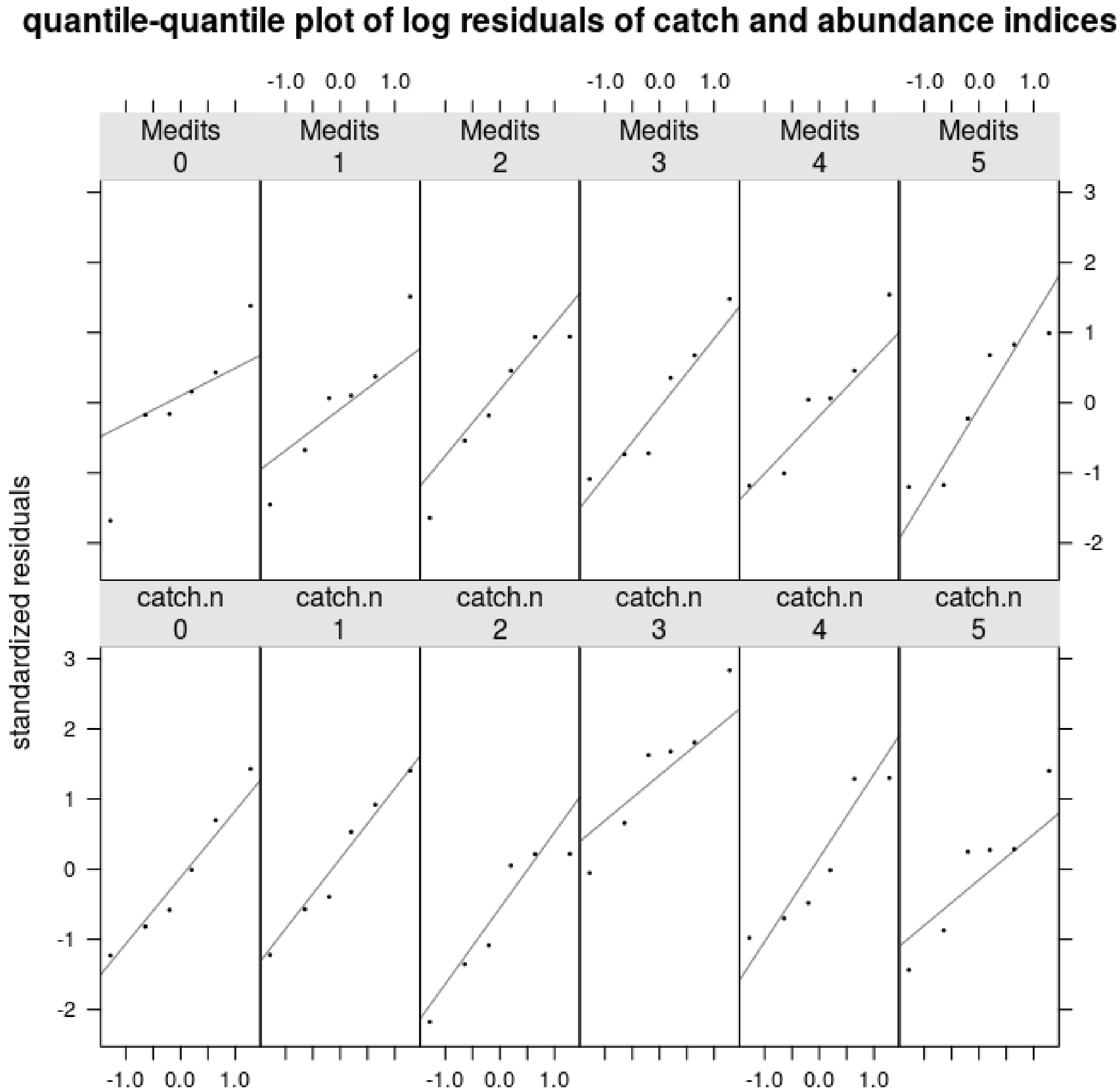


Figure 6.72: Quantile-quantile plot of residuals

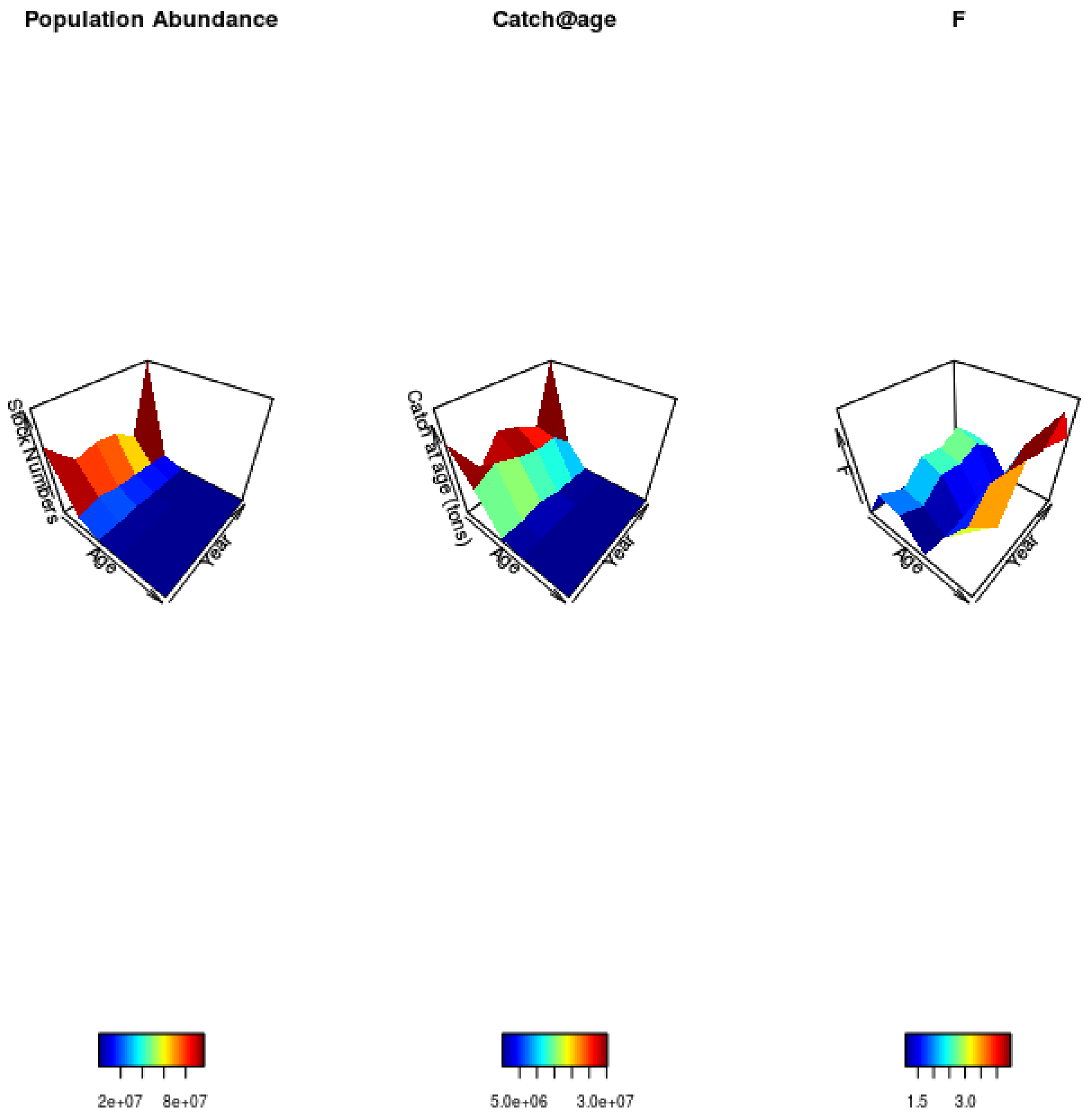


Figure 6.73: Assessment summary

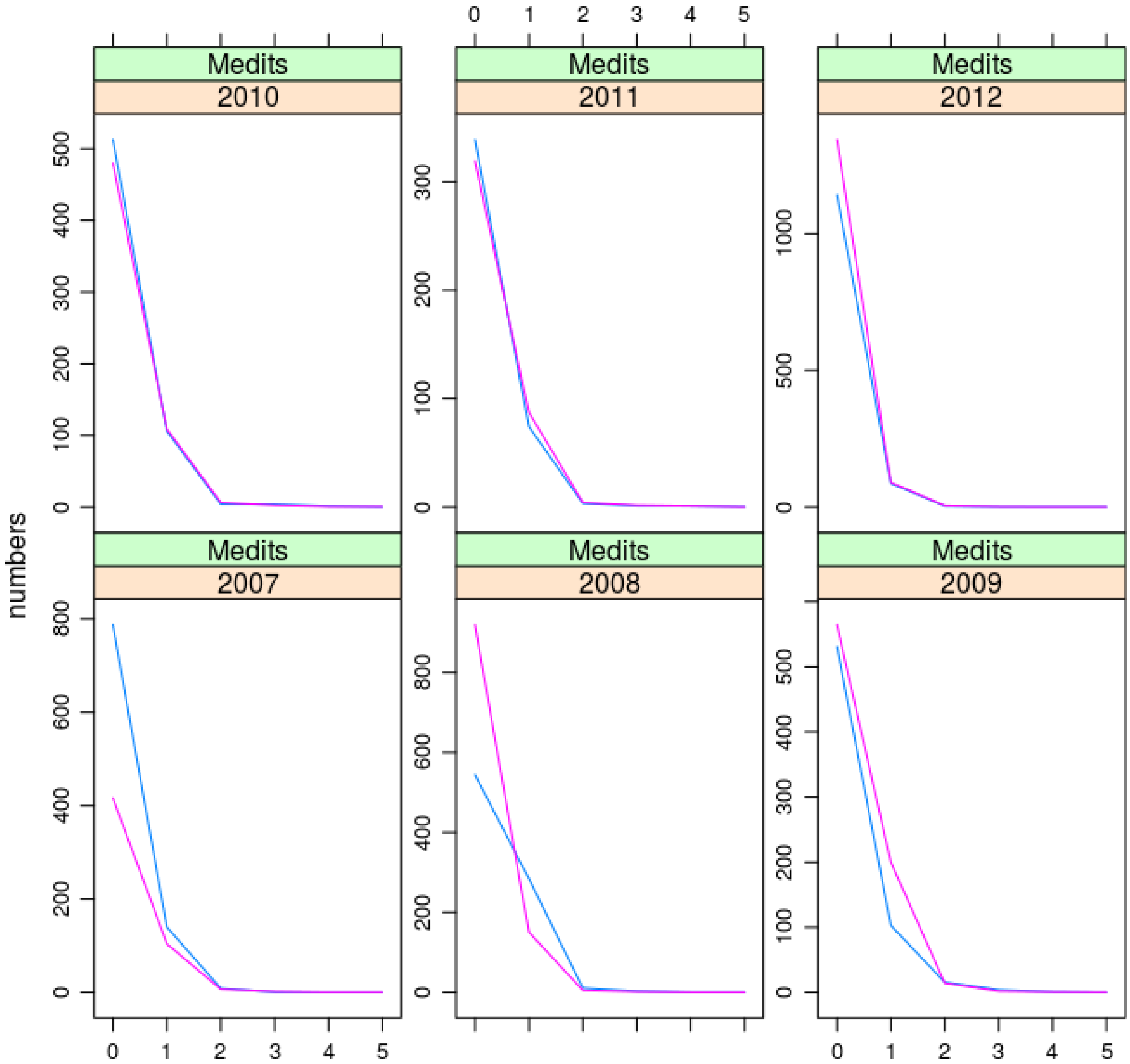


Figure 6.74: Plot fitted against observed for survey

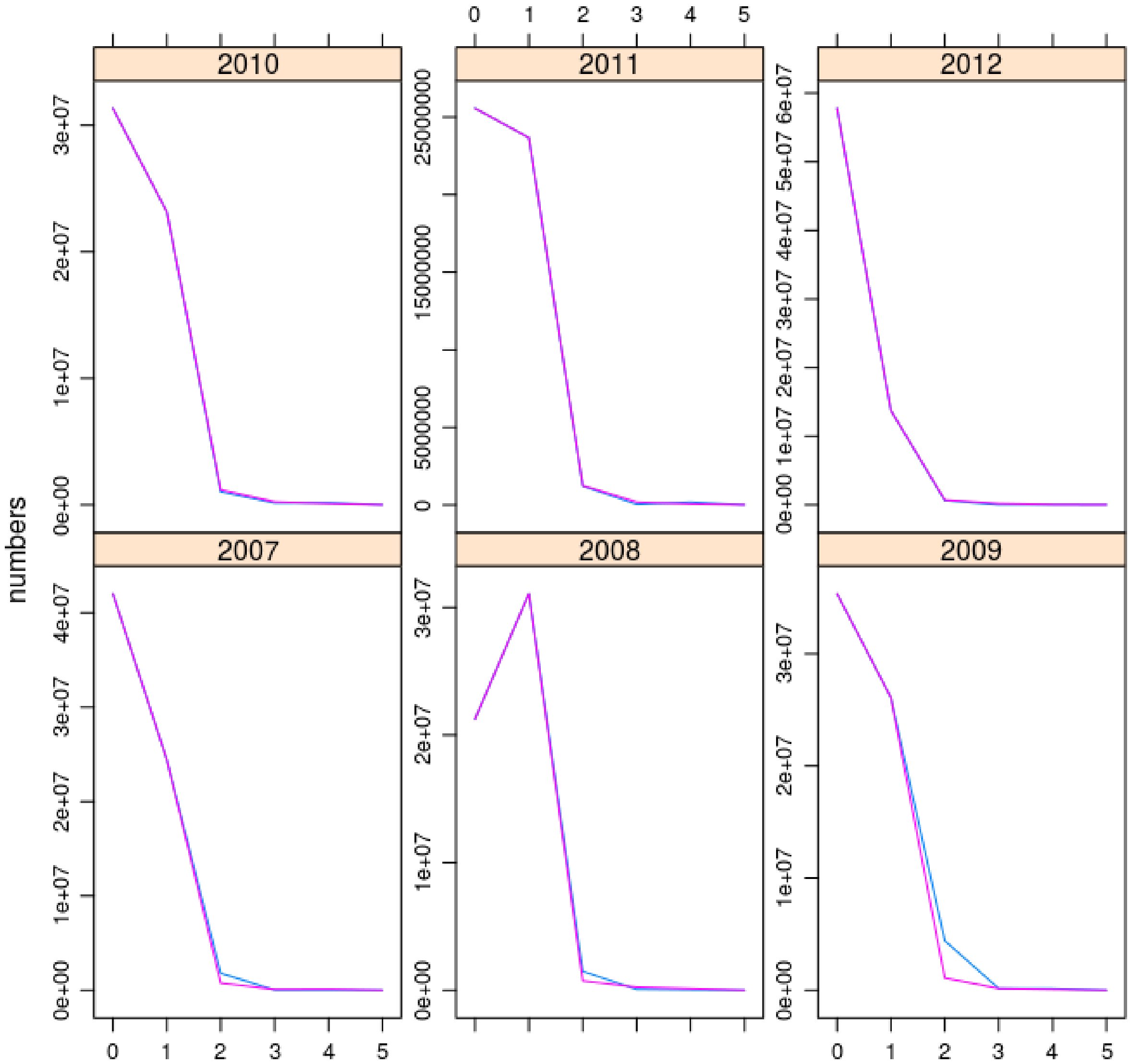


Figure 6.75: Plot fitted against observed for catch at age

Still the survey is not fitted well...

Model 5

Reasonable F comes at a cost... "Fixing" the excess F of all other models, drives biomass of 5+ group to explode.

```
Hake18.stk <- hketemp

qmod5 <- list(~s(age, k = 4))
fmod5 <- ~s(age, k = 3) + s(year, k = 5)
srmod5 <- ~s(year, k = 6)
fit5 <- sca(stock = Hake18.stk, indices = Hake18.S.Ind.new, fmodel = fmod5,
```

```
    qmodel = qmod5, srmodel = srmod5)

Hake18.stk.a4a.5 <- Hake18.stk + fit5
landings(Hake18.stk.a4a.5) <- computeLandings(Hake18.stk.a4a.5)
```

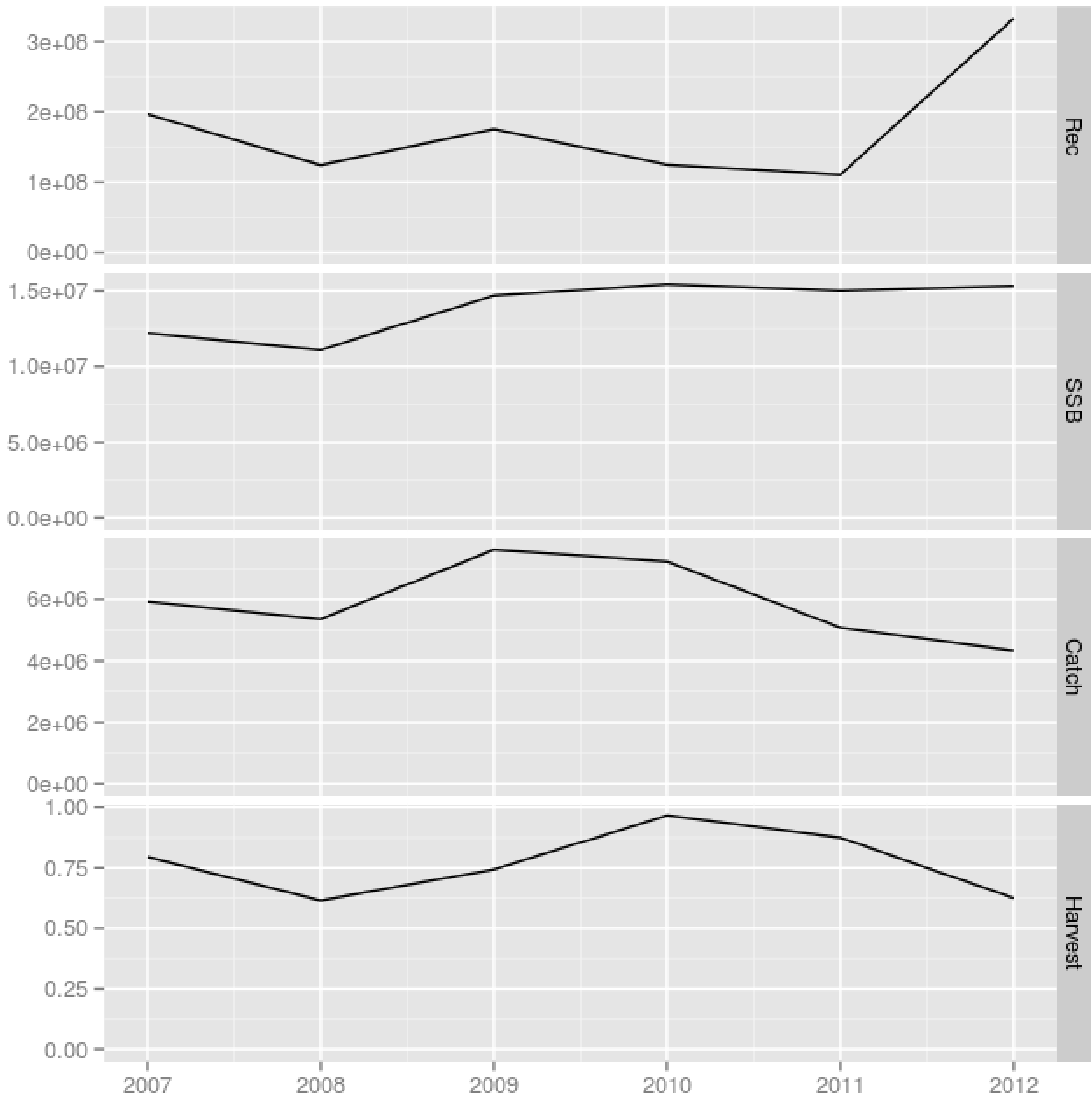


Figure 6.76: Stock summary

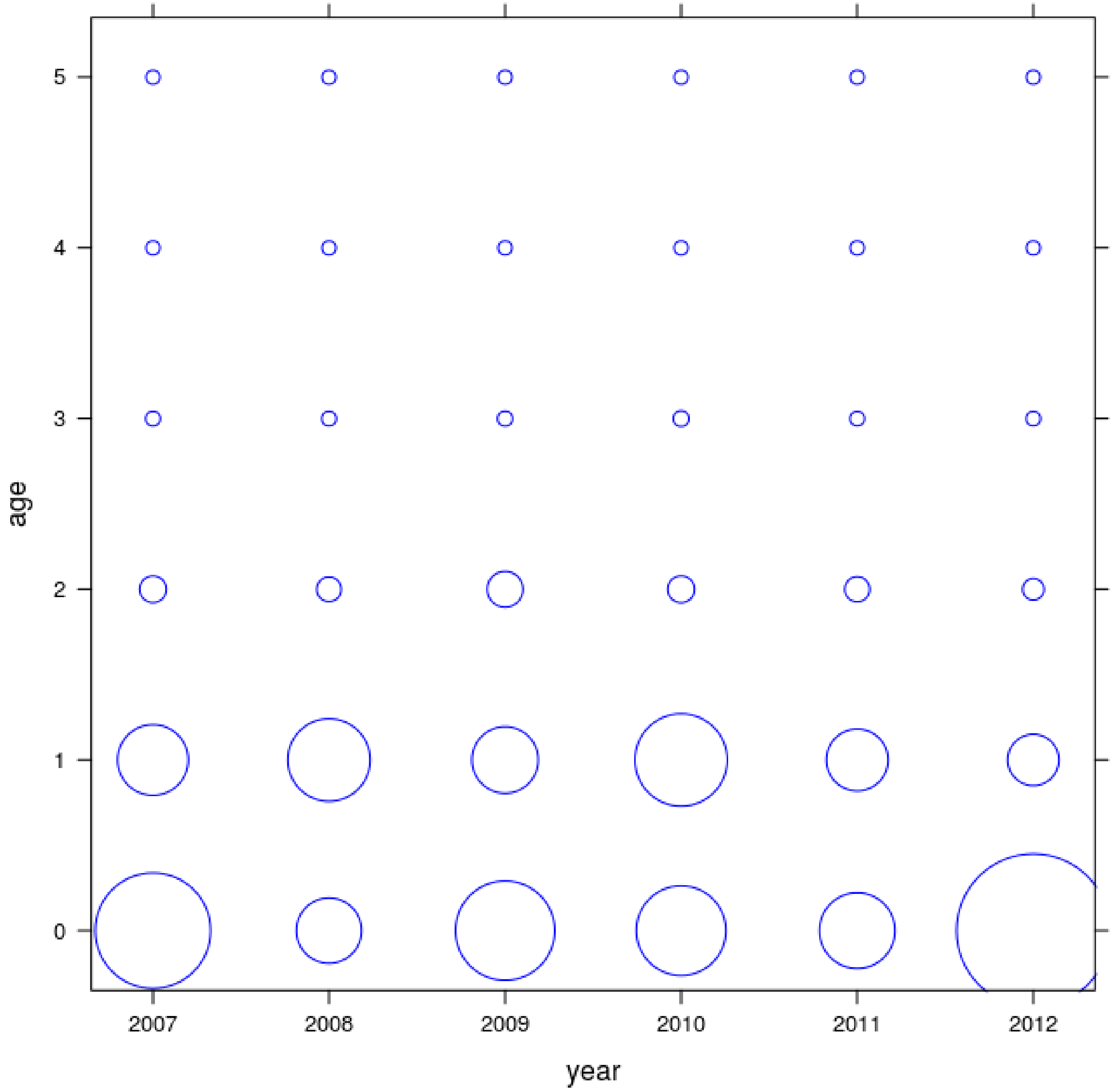


Figure 6.77: The catch matrix

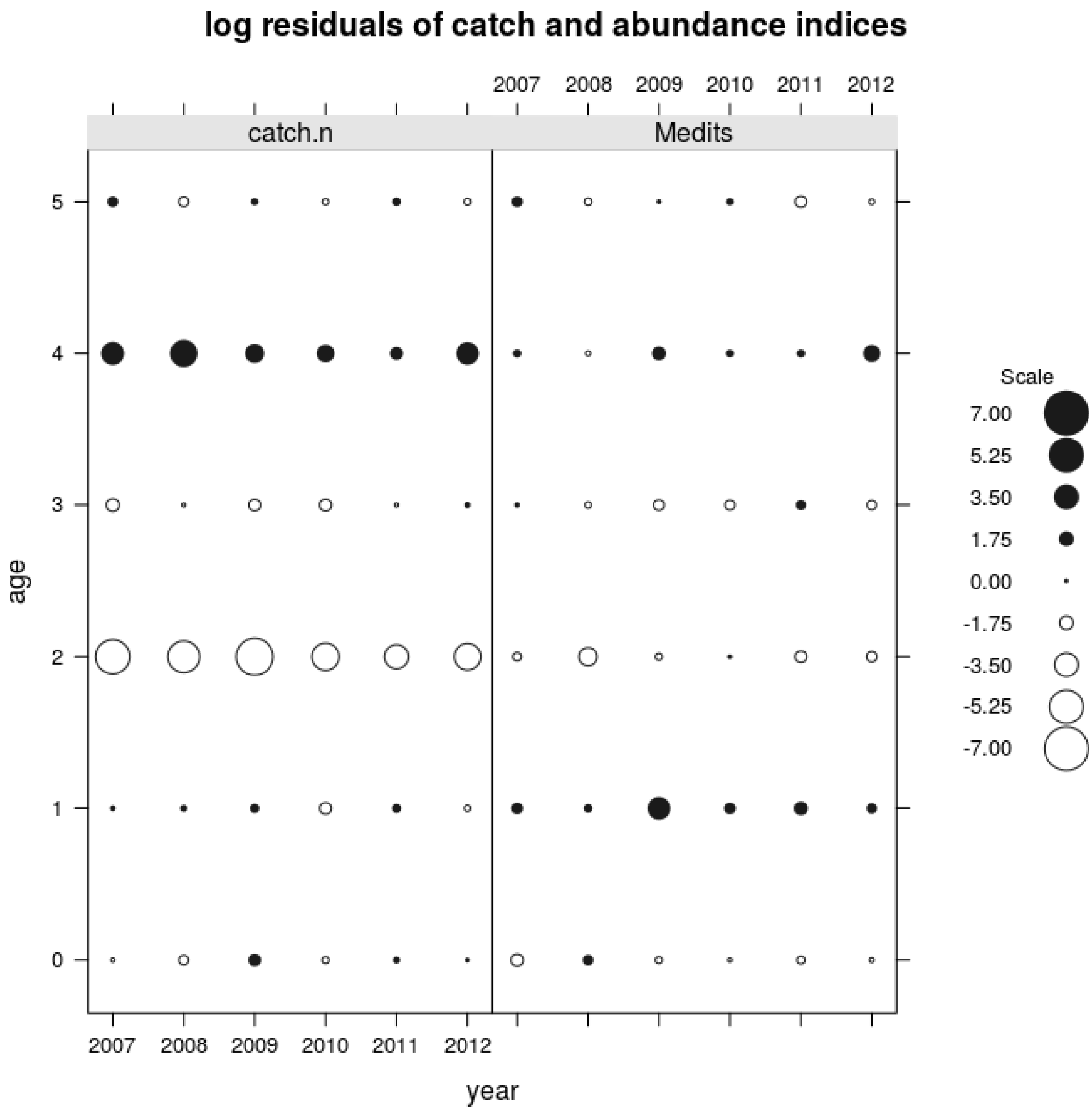


Figure 6.78: Residuals by age and year

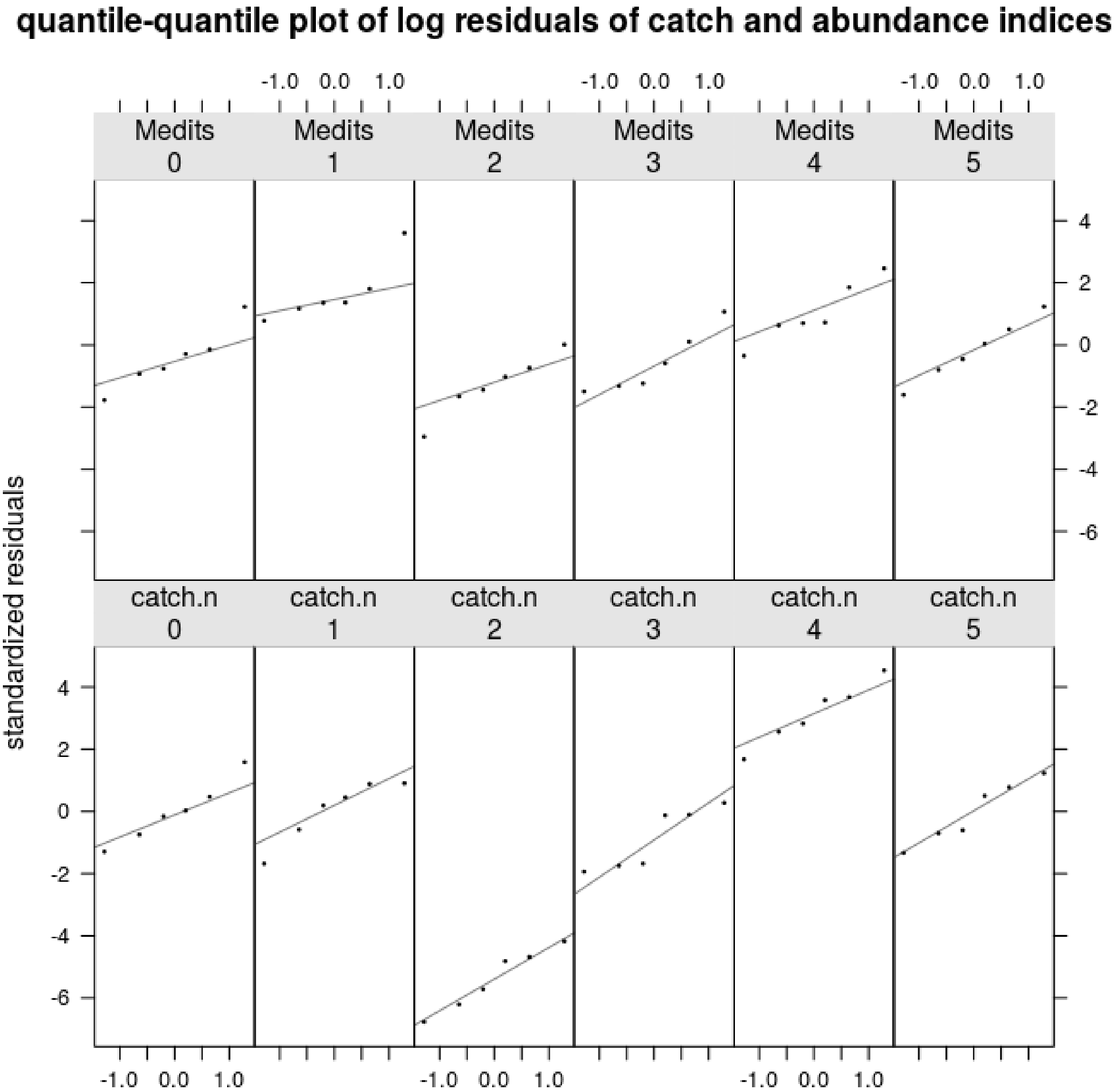


Figure 6.79: Quantile-quantile plot of residuals

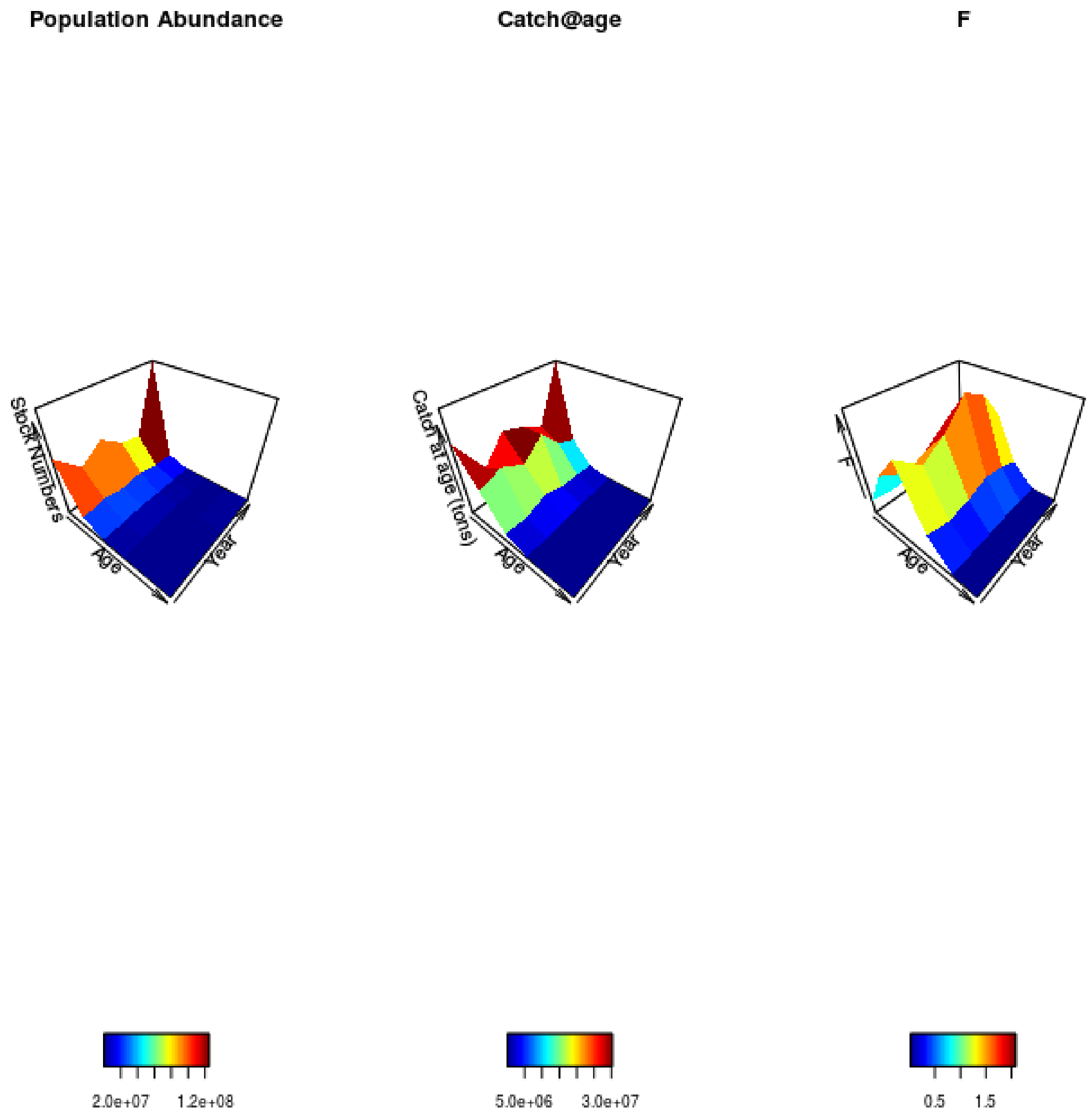


Figure 6.80: Assessment summary

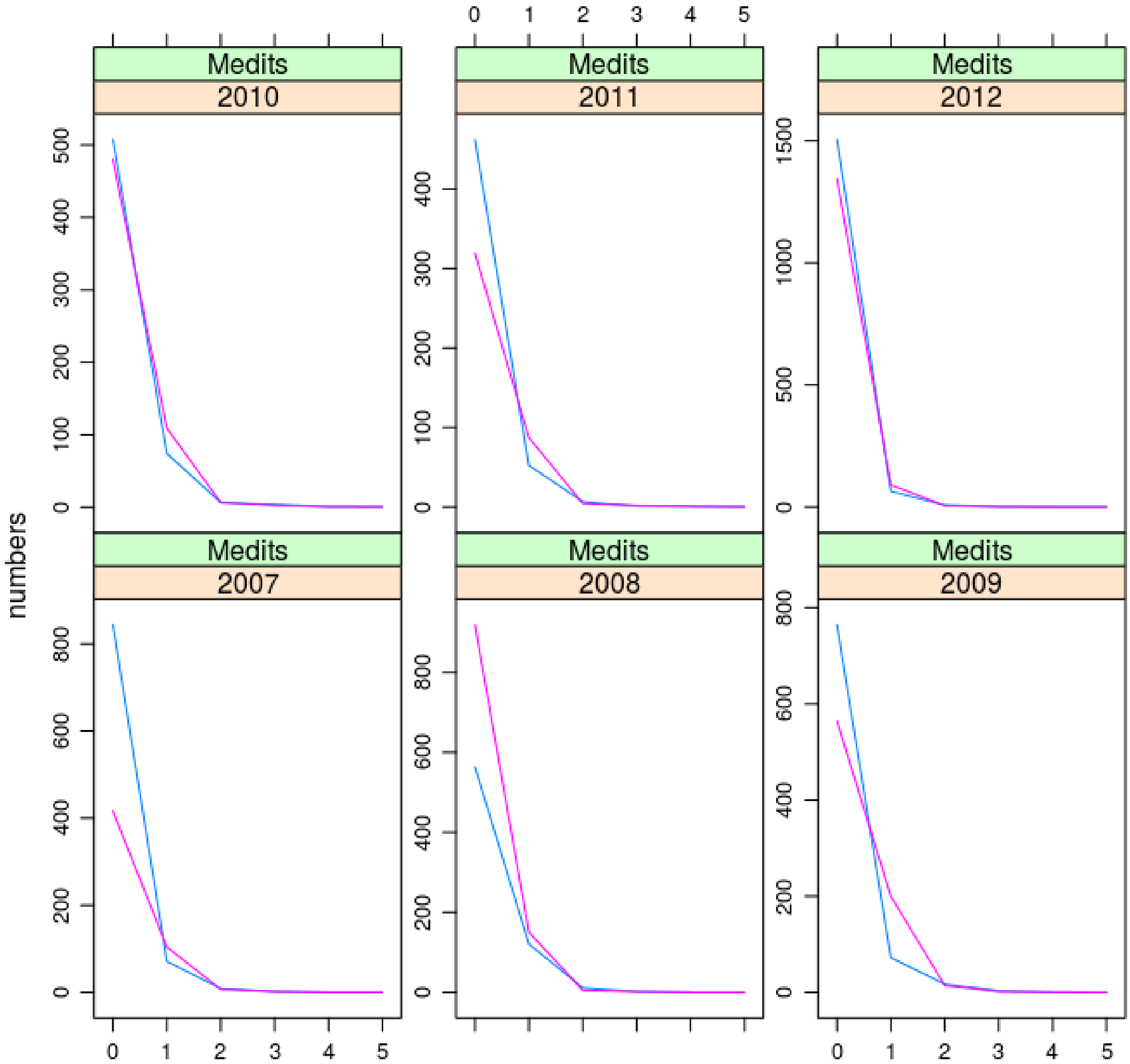


Figure 6.81: Plot fitted against observed for survey

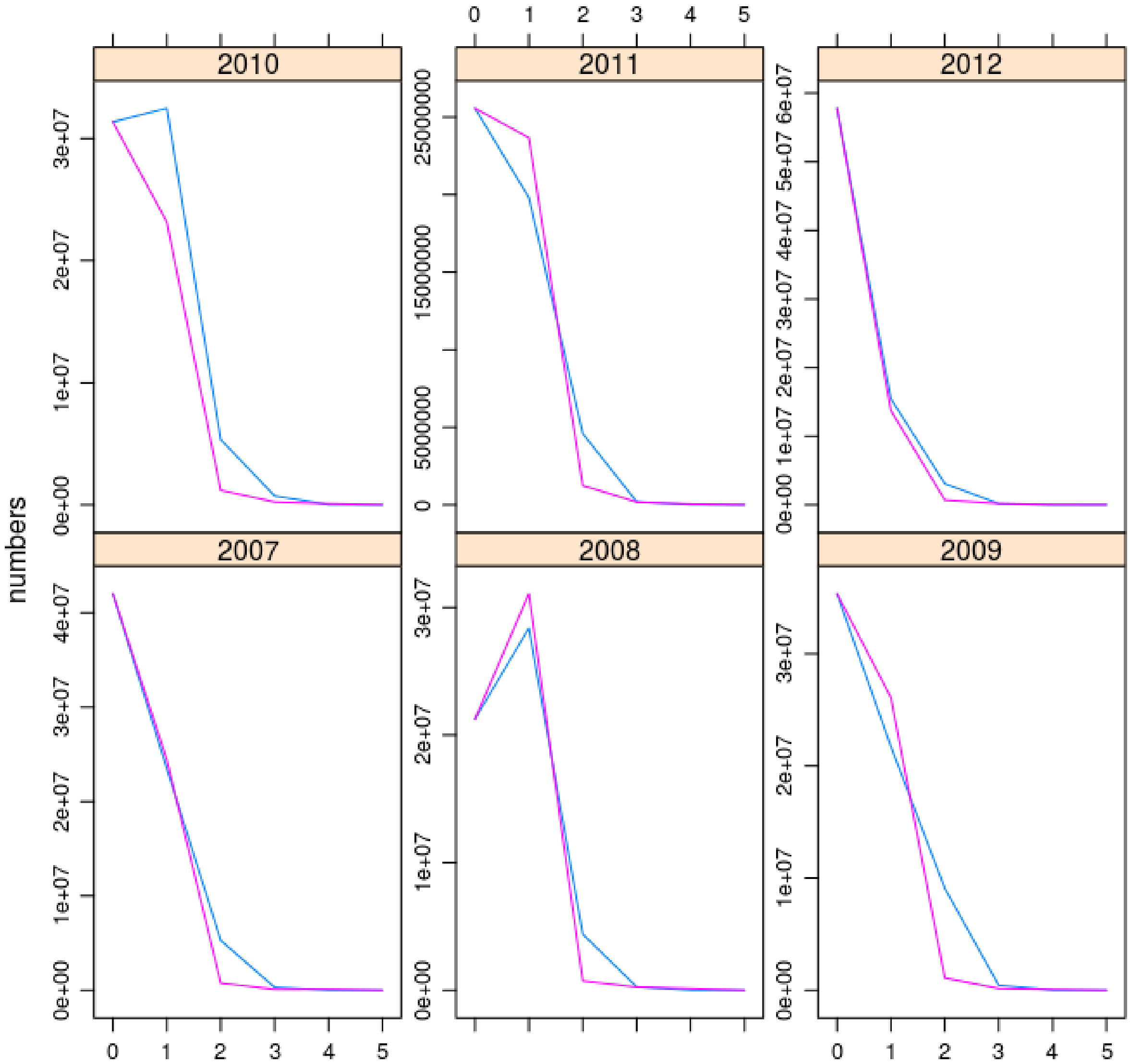


Figure 6.82: Plot fitted against observed for catch at age

```
# Which model has the best fit
AIC.BIC <- rbind(t(AIC(fit0, fit1, fit2, fit3, fit4, fit5)[2]),
    t(BIC(fit0, fit1, fit2, fit3, fit4, fit5)[2]))
AIC.BIC

##       fit0   fit1  fit2   fit3   fit4  fit5
## AIC 173.9  65.39 122.0 -407.5 -441.1 123.2
## BIC 201.2 138.24 181.2 -334.7 -368.2 182.3
```

Compare fits

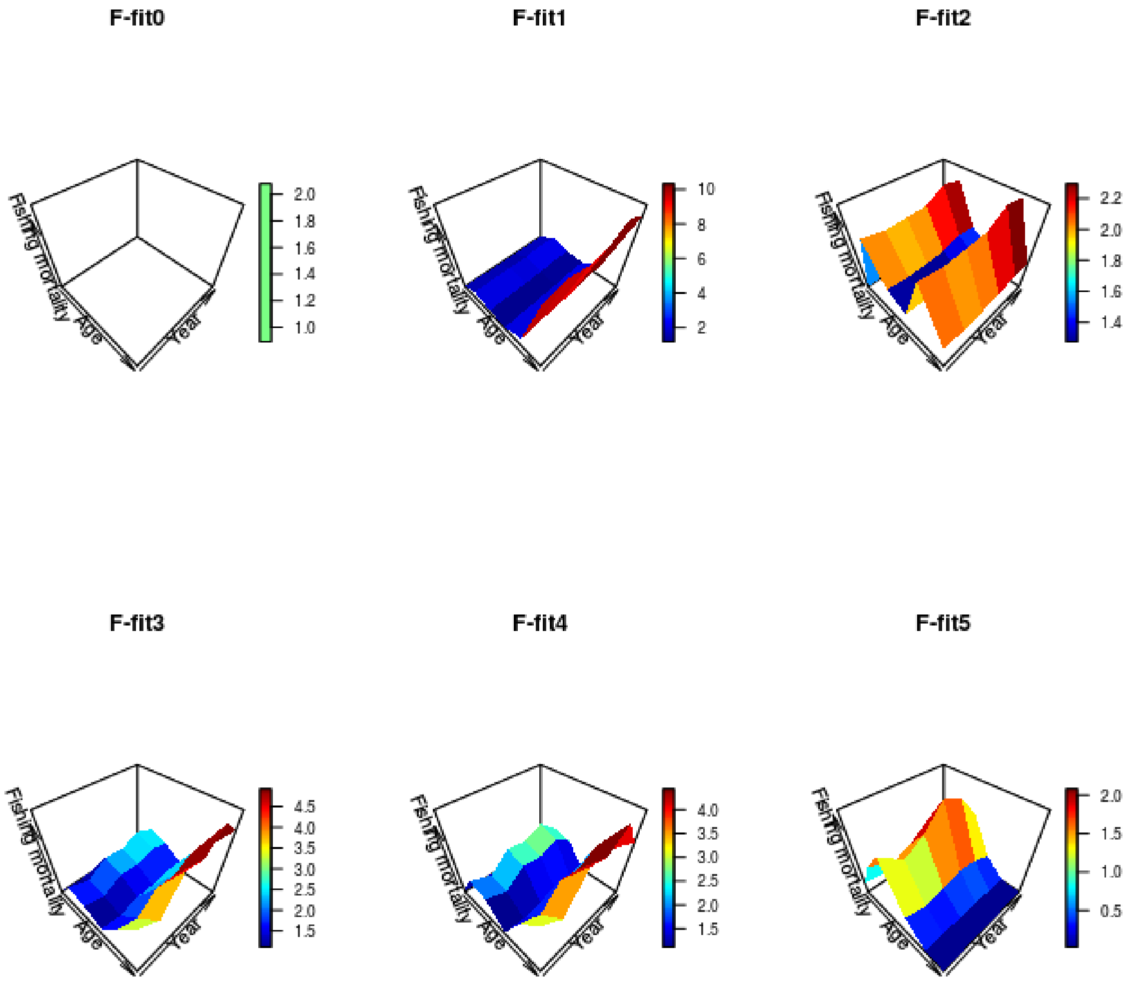


Figure 6.83: F-at-age estimates by each model

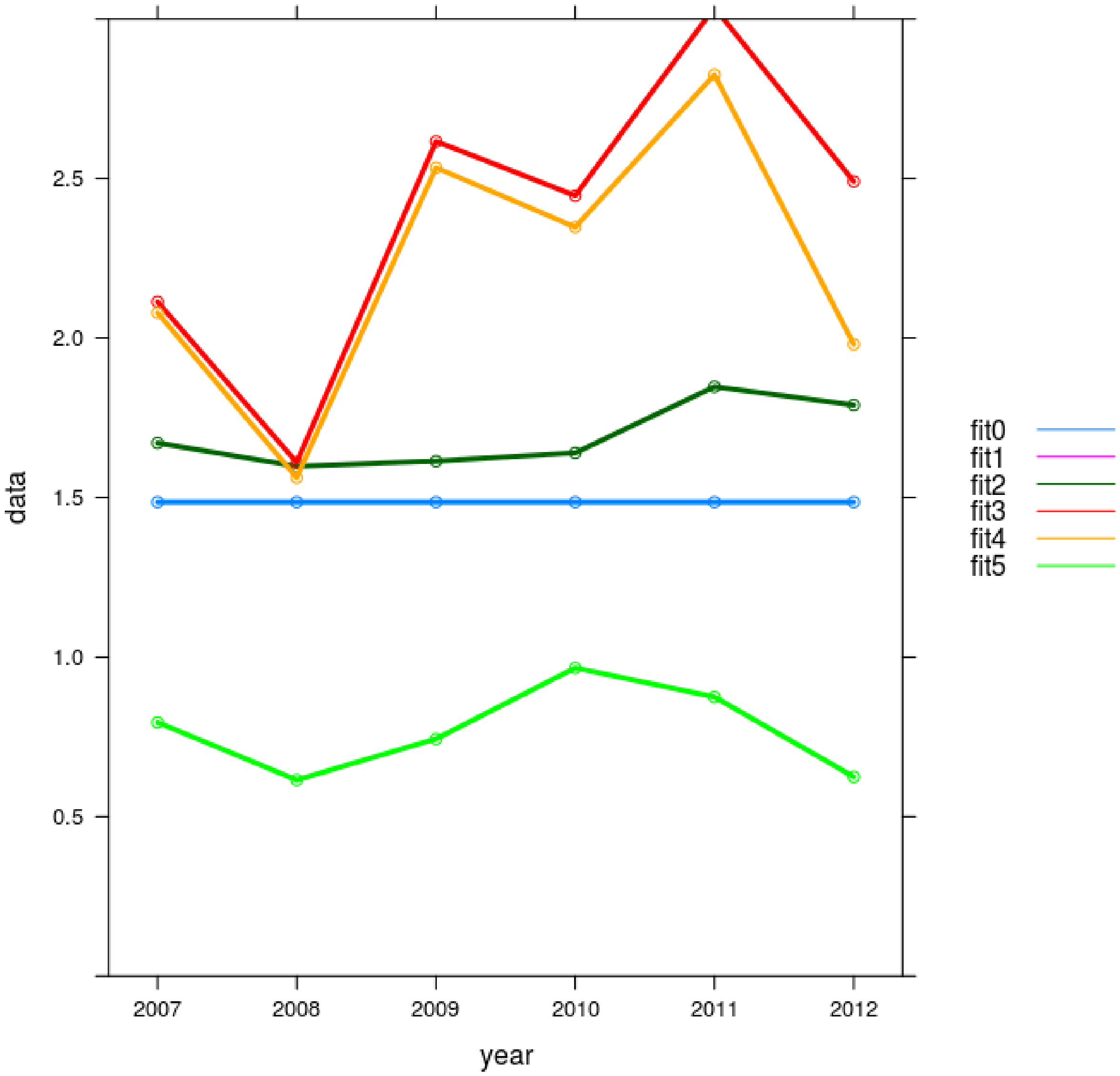


Figure 6.84: F estimates by each model

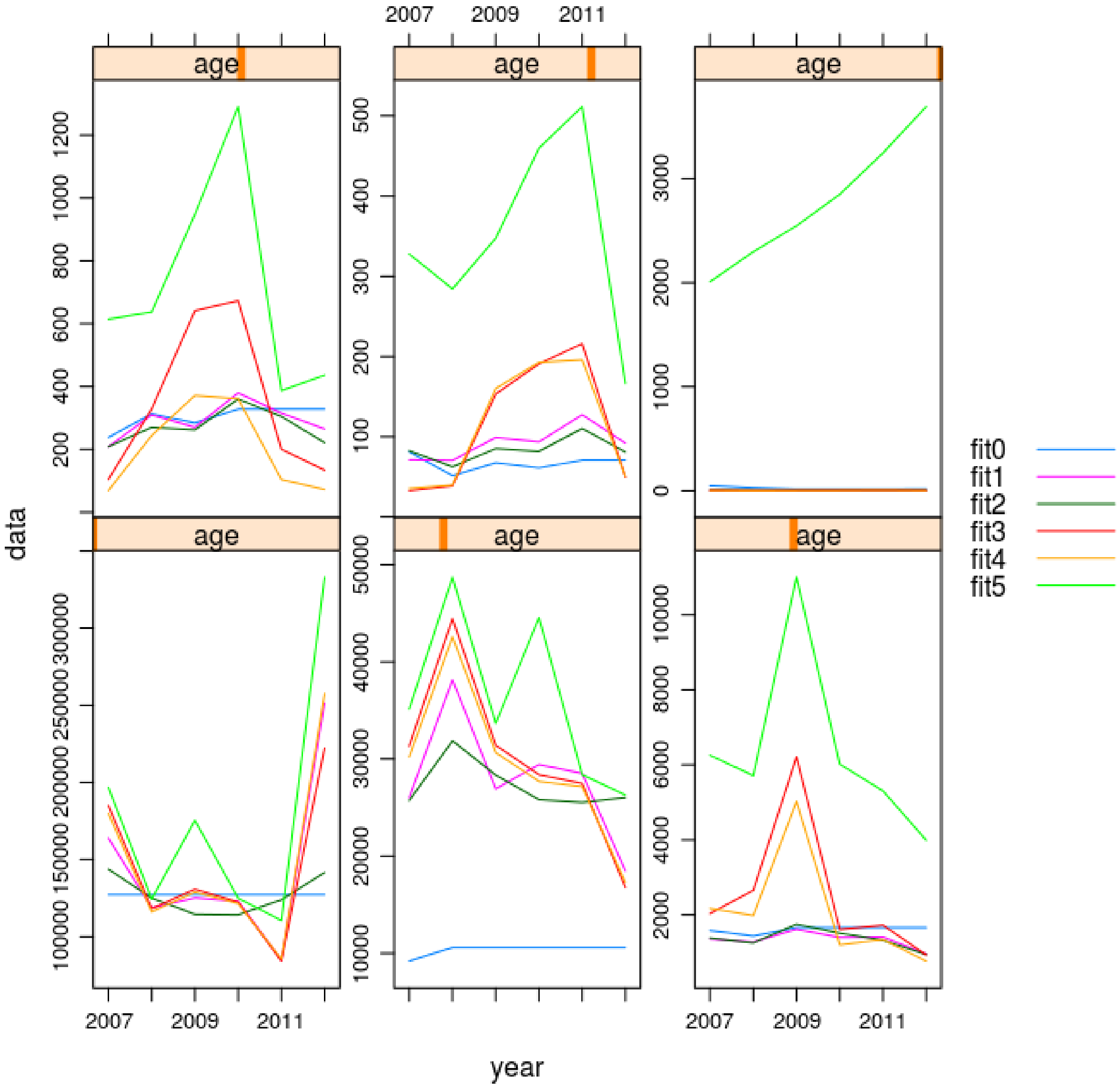


Figure 6.85: N-at-age estimates by each model

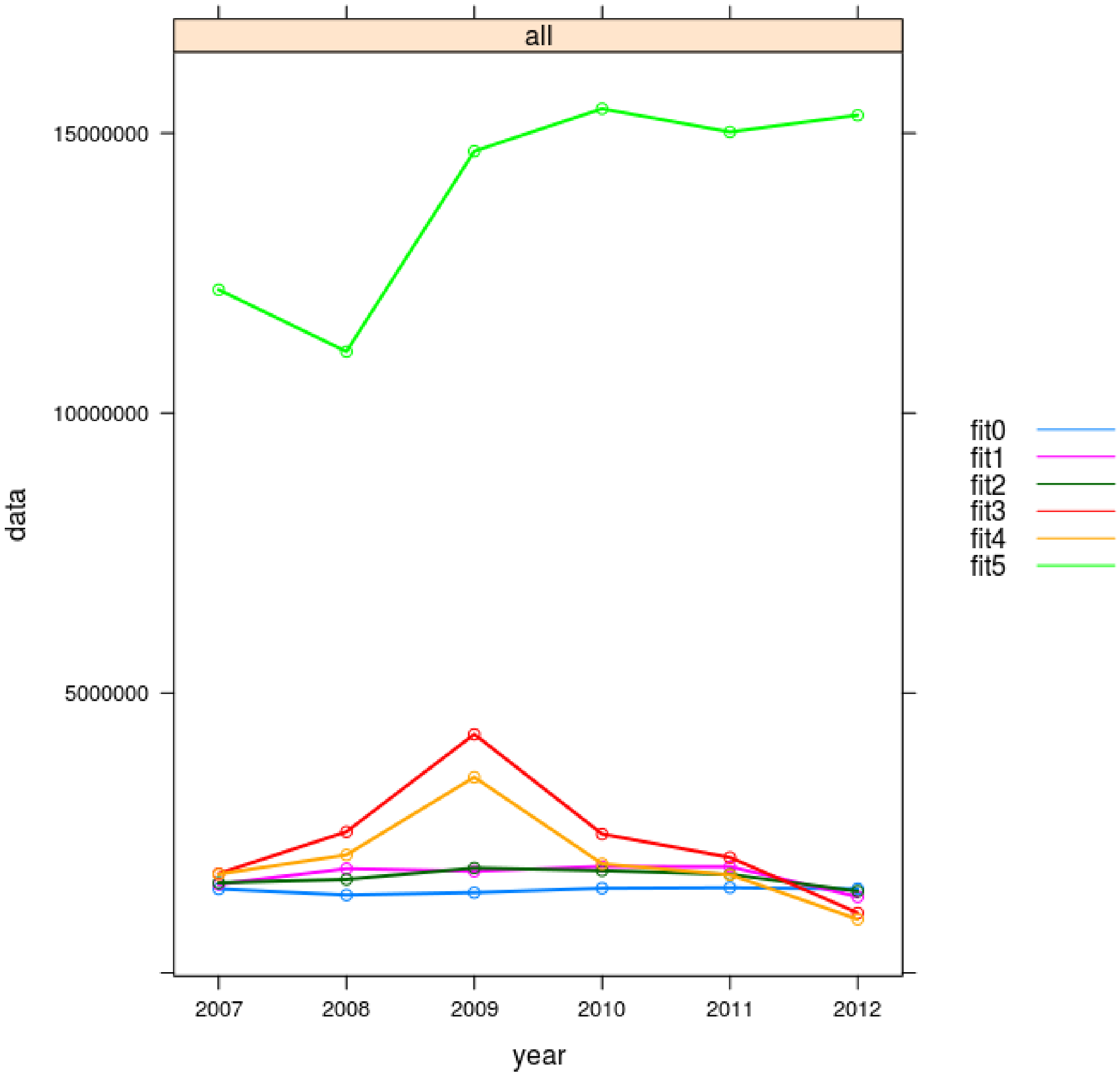


Figure 6.86: SSB estimates by each model

Final comments

4 out of the 5 investigated models gave very high values of F (>1.0). Model 5 succeeded to give reasonable F values, but stock biomass exploded... Model 4 trends, of all stock features, are similar to the recent STECF EWG 13-09 outcomes however F's are 50%-75% higher. This model was also suggested as the optimal from an information theoretic point of view (AIC, BIC criteria).

7 Sole in the North Adriatic Sea

Giuseppe Scarcella & Finlay Scott

7.1 Replicating accepted assessments

7.1.1 Reading in the data

We read the stock data.

```
sole_stk_xsa <- readFLStock("data/SOLE17IND.DAT", no.discards = TRUE)
summary(sole_stk_xsa)

## An object of class "FLStock"
##
## Name: NORTHERN ADRIATIC SEA (GSA 17) COMMON SOLE 2006-2012
## Description: Imported from a VPA file. ( data/SOLE17IND.DAT ).  Wed Sep 24 17:41:28 2014
## Range:  min max pgroup minyear maxyear minfbar maxfbar
##  0 5 NA 2006 2012 0 5
## Quant: age
##
## catch         : [ 1 7 1 1 1 1 ], units =  NA NA
## catch.n       : [ 6 7 1 1 1 1 ], units =  NA
## catch.wt      : [ 6 7 1 1 1 1 ], units =  NA
## discards      : [ 1 7 1 1 1 1 ], units =  NA
## discards.n    : [ 6 7 1 1 1 1 ], units =  NA
## discards.wt   : [ 6 7 1 1 1 1 ], units =  NA
## landings      : [ 1 7 1 1 1 1 ], units =  NA
## landings.n    : [ 6 7 1 1 1 1 ], units =  NA
## landings.wt   : [ 6 7 1 1 1 1 ], units =  NA
## stock         : [ 1 7 1 1 1 1 ], units =  NA * NA
## stock.n       : [ 6 7 1 1 1 1 ], units =  NA
## stock.wt      : [ 6 7 1 1 1 1 ], units =  NA
## m             : [ 6 7 1 1 1 1 ], units =  NA
## mat           : [ 6 7 1 1 1 1 ], units =  NA
## harvest       : [ 6 7 1 1 1 1 ], units =  f
## harvest.spwn  : [ 6 7 1 1 1 1 ], units =  NA
## m.spwn        : [ 6 7 1 1 1 1 ], units =  NA
```

There are 7 years of data (2006 to 2012) and 6 ages (0 to 5).

Set units, fbar range and plusgroup.

```
units(harvest(sole_stk_xsa)) <- "f"
range(sole_stk_xsa)["minfbar"] <- 0
range(sole_stk_xsa)["maxfbar"] <- 4
sole_stk_xsa <- setPlusGroup(sole_stk_xsa, 5)
```

Then we read the index data. There is just a single tuning index.

```
sole_idxs_xsa <- readFLIndices("data/TUNEFF.DAT")
summary(sole_idxs_xsa[[1]])

## An object of class "FLIndex"
##
## Name: SoleMon Radipo-trawl survey
## Description: NORTHERN ADRIATIC SEA (GSA 17) COMMON SOLE 2006-2012 TUNING DATA . Imported fron
## Range:  min max pgroup minyear maxyear startf endf
##  1 5 5 2006 2012 0.75 1
## Type :  numbers
## Distribution :
## Quant: age
##
## index         : [ 5 7 1 1 1 1 ], units =  NA
## index.var     : [ 5 7 1 1 1 1 ], units =  NA
## catch.n       : [ 5 7 1 1 1 1 ], units =  NA
## catch.wt      : [ 5 7 1 1 1 1 ], units =  NA
## effort        : [ 1 7 1 1 1 1 ], units =  NA
## sel.pattern   : [ 5 7 1 1 1 1 ], units =  NA
## index.q       : [ 5 7 1 1 1 1 ], units =  NA
```

7.1.2 Running the XSA

Here we run the XSA using the same settings as used in SGMED.

```
FLXSA.control.sole1 <- FLXSA.control(x = NULL, tol = 1e-09, maxit = 30,
    min.nse = 0.3, fse = 1, rage = 0, qage = 4, shk.n = TRUE,
    shk.f = TRUE, shk.yrs = 5, shk.ages = 5, window = 100, tsrange = 20,
    tspower = 3, vpa = FALSE)
sole_xsa1 <- FLXSA(sole_stk_xsa, sole_idxs_xsa, FLXSA.control.sole1)
sole_stk_xsa_res <- sole_stk_xsa + sole_xsa1
```

The results of the XSA analysis can be seen in Figure 7.1.

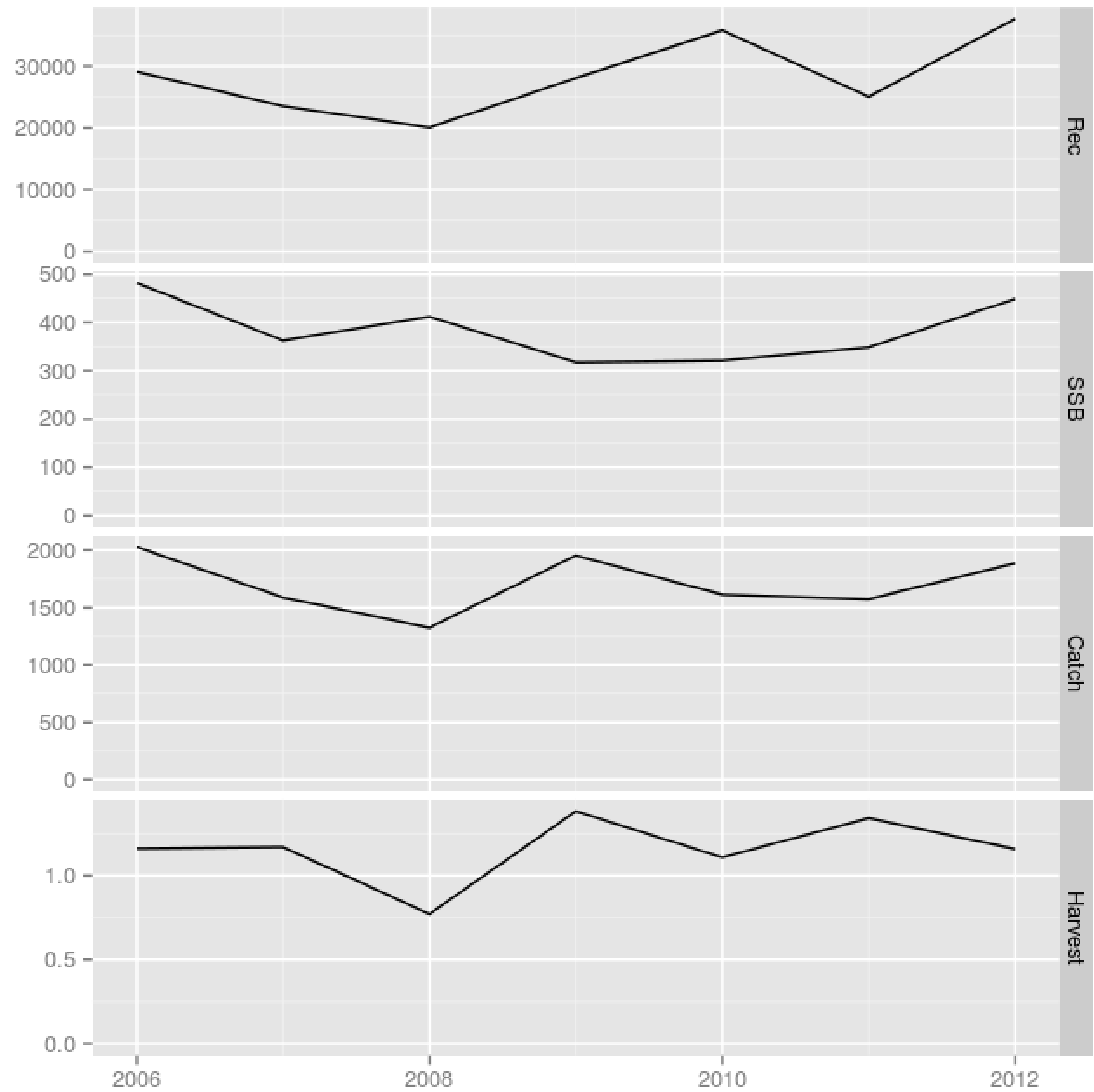


Figure 7.1: Results from running XSA

7.2 Assessments with the statistical catch-at-age method

Here we set up a wide range of a4a model options and put them into a data.frame. The `fmodel` options are a basic mix of factors and smoothers. The `qmodel` options only have an age factor and are constant in time. The `srmodel` options are either a factor by year or a Beverton-Holt shape with a low CV. Every combination of model options is used.

```
# fmodels age and year seperable
fmodel1 <- ~factor(age) + factor(year)
# age and year seperable but smooth
fmodel2 <- ~s(age, k = 3) + s(year, k = 3)
# age and year smooth interaction
```

```r
fmodel3 <- ~te(age, year, k = c(3, 3))
# age and year seperable, with age > 4 being the same as age
# 4
fmodel4 <- ~factor(replace(age, age > 4, 4)) + factor(year)
# qmodels
qmodel1 <- list(~factor(age))
qmodel2 <- list(~s(age, k = 3))
# Mimics the XSA settings of flat catchability after age 4
# (qage)
qmodel3 <- list(~factor(replace(age, age > 4, 4)))
# srmodels
rmodel1 <- ~factor(year)
rmodel2 <- ~bevholt(CV = 0.1)
# Build the data.frame
fmodels <- c("fmodel1", "fmodel2", "fmodel3", "fmodel4")
qmodels <- c("qmodel1", "qmodel2", "qmodel3")
rmodels <- c("rmodel1", "rmodel2")
model_data <- expand.grid(fmodel = fmodels, qmodel = qmodels,
    rmodel = rmodels)
model_data <- cbind(model_id = 1:nrow(model_data), model_data)
```

For reference, the model combinations are:

```r
model_data
```

```
##    model_id  fmodel  qmodel  rmodel
## 1         1 fmodel1 qmodel1 rmodel1
## 2         2 fmodel2 qmodel1 rmodel1
## 3         3 fmodel3 qmodel1 rmodel1
## 4         4 fmodel4 qmodel1 rmodel1
## 5         5 fmodel1 qmodel2 rmodel1
## 6         6 fmodel2 qmodel2 rmodel1
## 7         7 fmodel3 qmodel2 rmodel1
## 8         8 fmodel4 qmodel2 rmodel1
## 9         9 fmodel1 qmodel3 rmodel1
## 10       10 fmodel2 qmodel3 rmodel1
## 11       11 fmodel3 qmodel3 rmodel1
## 12       12 fmodel4 qmodel3 rmodel1
## 13       13 fmodel1 qmodel1 rmodel2
## 14       14 fmodel2 qmodel1 rmodel2
## 15       15 fmodel3 qmodel1 rmodel2
## 16       16 fmodel4 qmodel1 rmodel2
## 17       17 fmodel1 qmodel2 rmodel2
## 18       18 fmodel2 qmodel2 rmodel2
## 19       19 fmodel3 qmodel2 rmodel2
## 20       20 fmodel4 qmodel2 rmodel2
## 21       21 fmodel1 qmodel3 rmodel2
## 22       22 fmodel2 qmodel3 rmodel2
## 23       23 fmodel3 qmodel3 rmodel2
## 24       24 fmodel4 qmodel3 rmodel2
```

7.2.1 Fitting the a4a models

We fit the models using a `for` loop and store the results (the new *FLStock* and the fitted object) in *lists*. We also store the AIC and BIC as attributes.

```
# Stores
sole_stks_fit <- list()
fits <- list()

for (model_count in 1:nrow(model_data)){
    stk_fit_name <- paste("model",model_count,sep="")
    cat("model_count: ", model_count, "\n")
    fit <- try(a4aSCA(stock=sole_stk_xsa,
        indices = sole_idxs_xsa,
        fmodel = eval(parse(text=as.character(model_data[model_count,"fmodel"]))),
        qmodel = eval(parse(text=as.character(model_data[model_count,"qmodel"]))),
        # variance of catches constant - better mimics XSA?
        vmodel = list(~1,~1),
        srmodel = eval(parse(text=as.character(model_data[model_count,"rmodel"])))))
    if (!is(fit, "try-error")) {
        fits[[stk_fit_name]] <- fit
        sole_stks_fit[[stk_fit_name]] <- sole_stk_xsa + fit
        attr(sole_stks_fit[[stk_fit_name]],"aic") <- AIC(fit)
        attr(sole_stks_fit[[stk_fit_name]],"bic") <- BIC(fit)
        attr(sole_stks_fit[[stk_fit_name]],"fitSumm") <- fit@fitSumm
    }
}

## model_count:  1
## model_count:  2
## model_count:  3
## model_count:  4
## model_count:  5
## model_count:  6
## model_count:  7
## model_count:  8
## model_count:  9
## model_count:  10
## model_count:  11
## model_count:  12
## model_count:  13
## model_count:  14
## model_count:  15
## model_count:  16
## model_count:  17
## model_count:  18
## model_count:  19
## model_count:  20
## model_count:  21
## model_count:  22
## model_count:  23
## model_count:  24
```

7.2.2 Exploring the a4a results

The model summary results, along with the XSA results, can be seen in Figure 7.2.

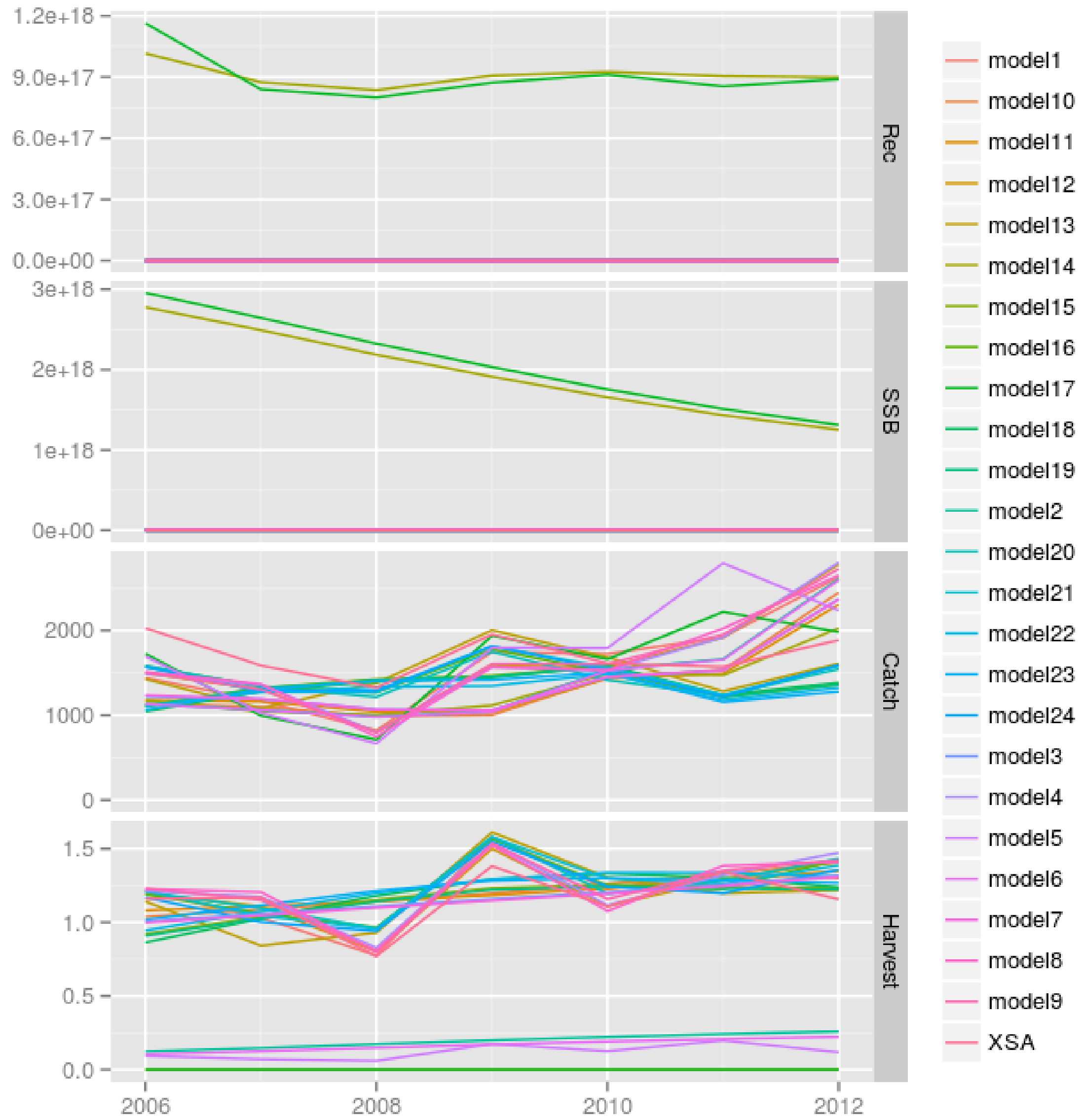


Figure 7.2: Comparing the results from all of the a4a assessments to the XSA assessment. Some models are clearly unbelievable.

Some models have very high SSBs (Figure 7.3). We need to filter these out.

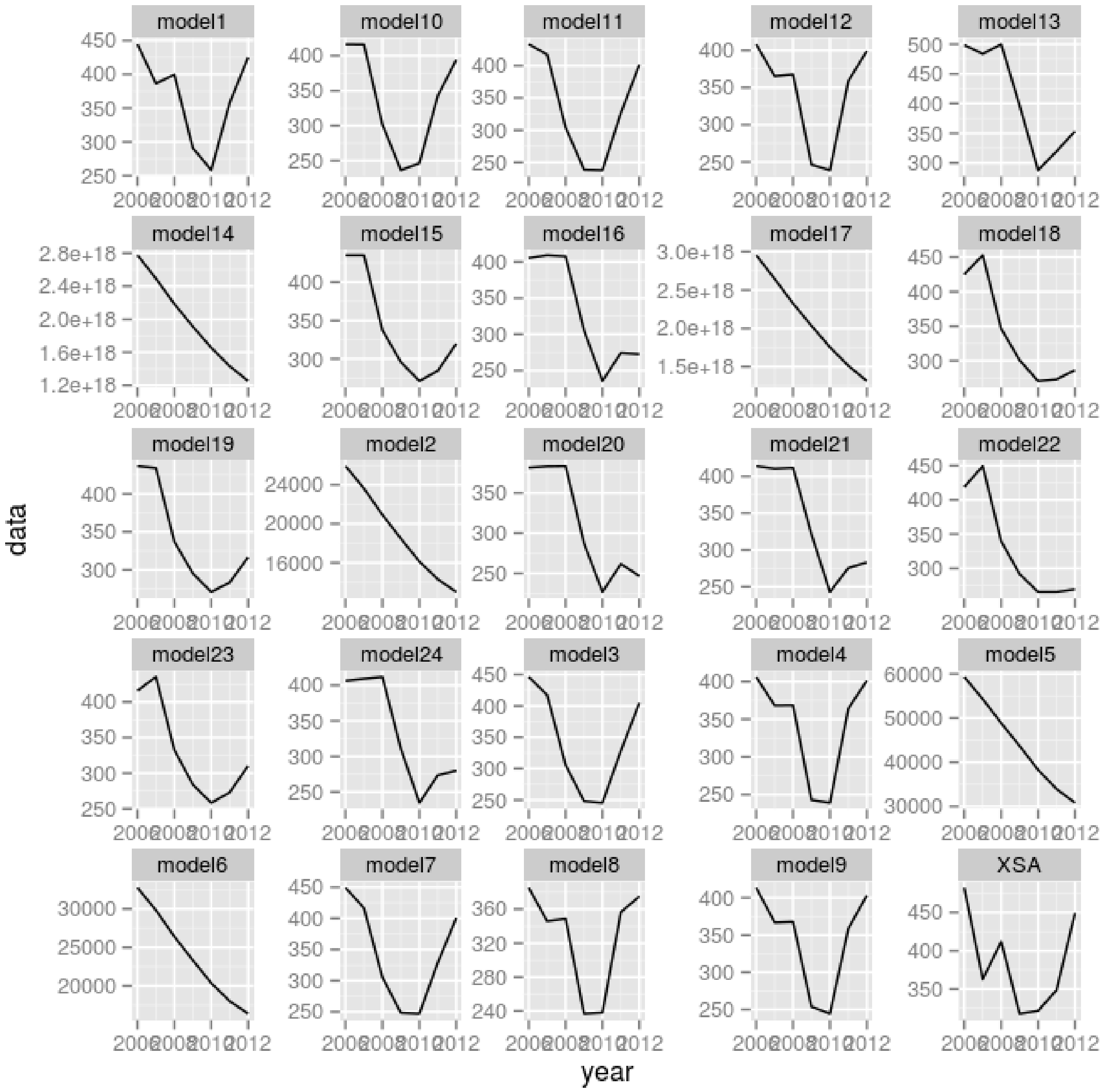


Figure 7.3: SSB estimates of all the model fits and the XSA results to identify bad models.

The unwanted models are 2, 5, 6, 14, 17 and 18. Model 14 is particularly wrong. We only want to include models that have believable SSB estimates.

```
good_models <- c(1, 3, 4, 7:13, 15, 16, 19:24)
```

Figure 7.4 shows the results from the selection of model fits that have reasonable SSB estimates.

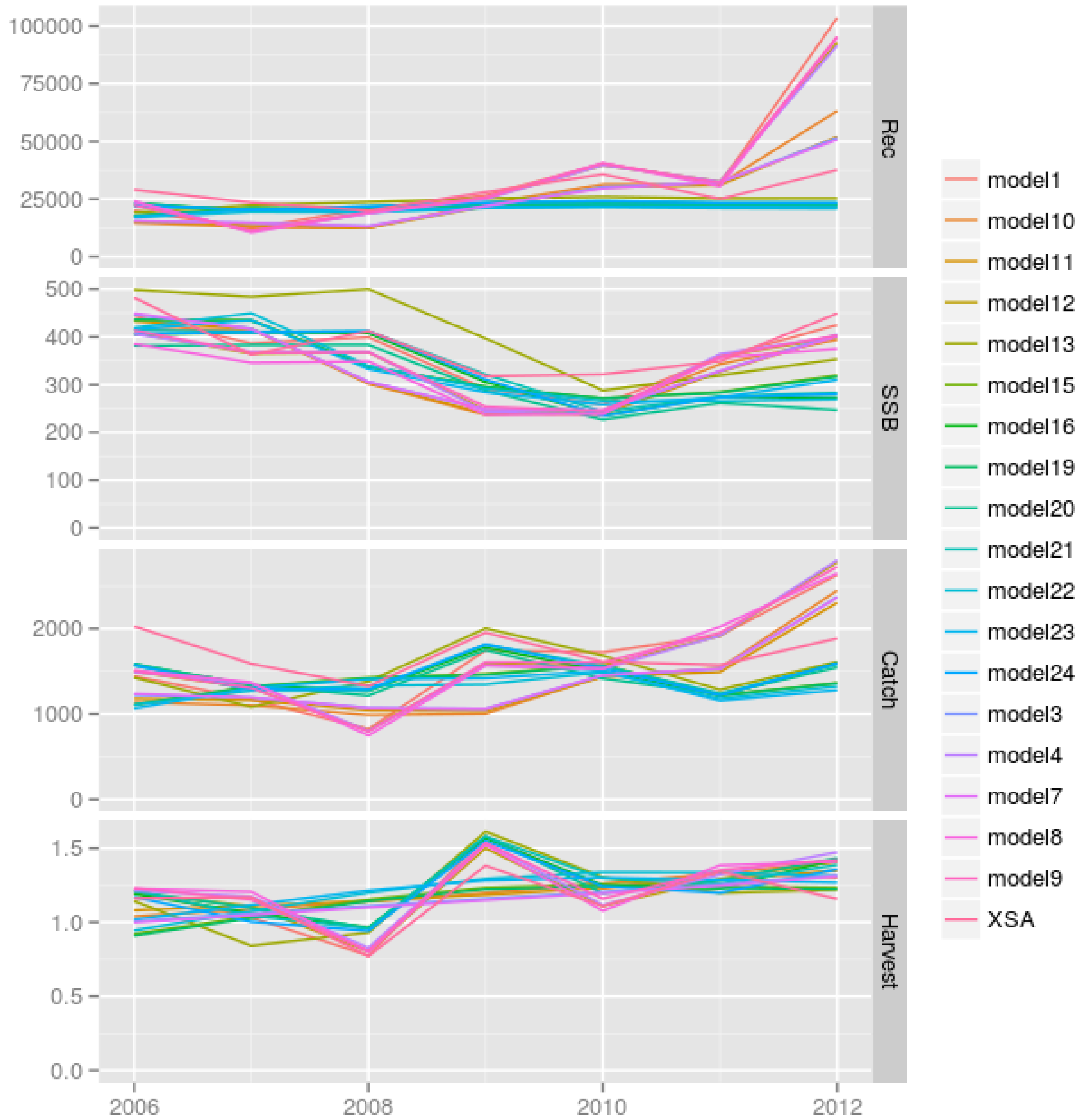


Figure 7.4: Comparing the results from the a4a assessments that have reasonable SSB estimates to the XSA assessment.

Some of the Fbar estimates are very linear looking (Figure 7.5). We need to identify and remove them.

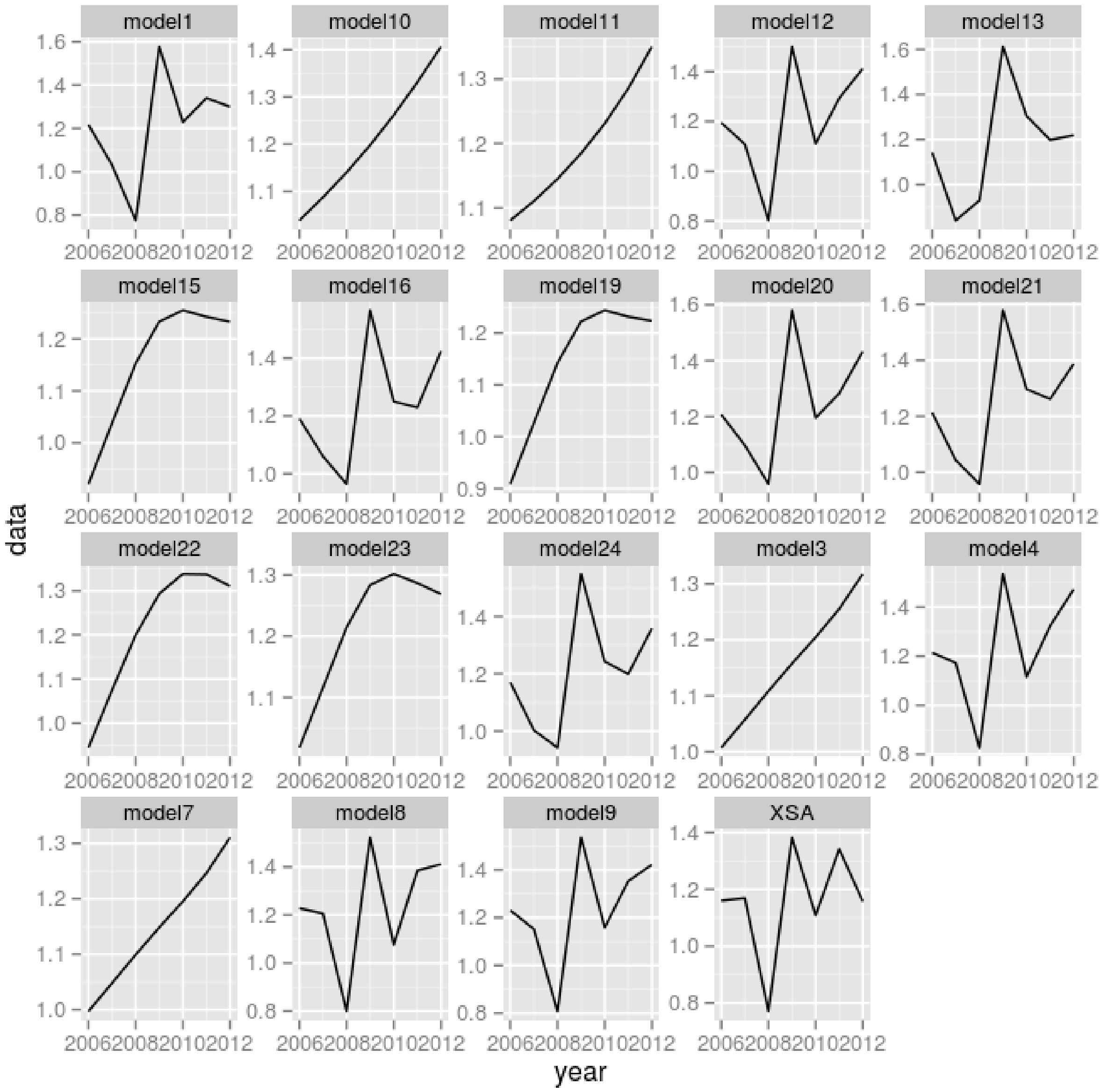


Figure 7.5: Fbar estimates of the model fits to identify the linear looking ones.

We want to remove models 3, 7, 11, 15, 19, 22 and 23 from the list of good models. These all use `fmodel3`, the smooth interaction between age and year.

```
fmodel3 <- ~te(age, year, k = c(3, 3))  # age and year smooth interaction.
```

This may be because we don't have many years of data and ideally we would like more degrees of freedom in the smoother.

This leaves 10 models that look reasonable (Figure 7.6).

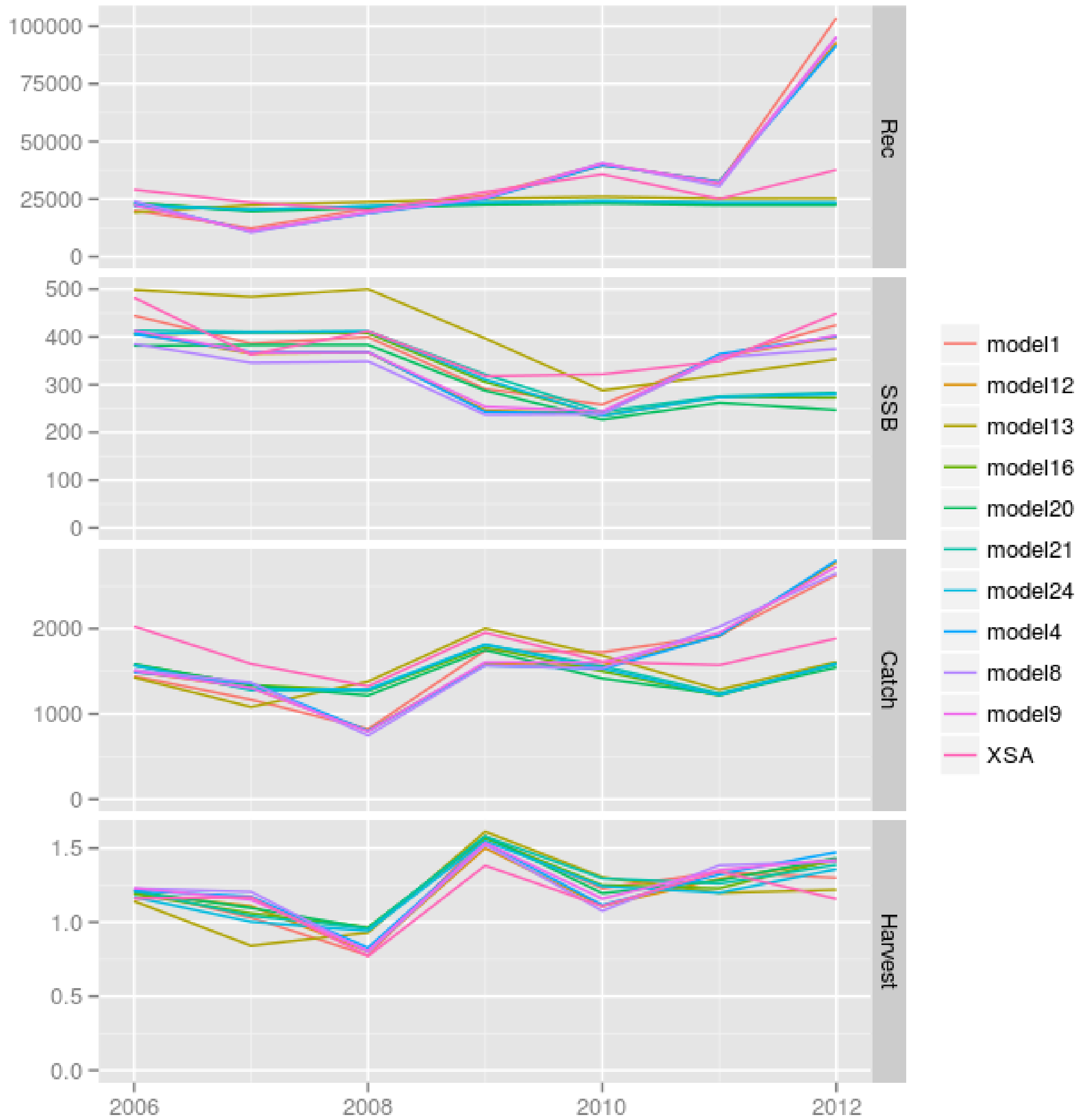


Figure 7.6: Comparing the good model fits to the XSA results. The difference in recruitment in the final year is driven by the two different srmodels

The main difference between the models is the recruitment in the final year. The recruitment in the models which have an `srmodel` of a factor on year (models 1, 4, 8, 9 and 12) are less constrained than the models which have an `srmodel` of a Beverton-Holt (models 13, 16, 20, 21 and 24).

We can look at the AIC and BIC of the good models to see if they are good guides as to which model is the best.

```
good_model_data <- model_data[good_models, ]
good_model_data$aic <- lapply(sole_stks_fit[paste("model", good_models,
    sep = "")], function(x) x@aic)
good_model_data$bic <- lapply(sole_stks_fit[paste("model", good_models,
    sep = "")], function(x) x@bic)
good_model_data
```

```
##    model_id  fmodel  qmodel  rmodel   aic   bic
## 1         1 fmodel1 qmodel1 rmodel1   176 248.7
## 4         4 fmodel4 qmodel1 rmodel1 176.3 246.6
## 8         8 fmodel4 qmodel2 rmodel1 180.6 246.3
## 9         9 fmodel1 qmodel3 rmodel1 175.7   246
## 12       12 fmodel4 qmodel3 rmodel1 176.1 244.1
## 13       13 fmodel1 qmodel1 rmodel2 179.3 256.7
## 16       16 fmodel4 qmodel1 rmodel2 183.3 258.3
## 20       20 fmodel4 qmodel2 rmodel2 187.6 257.9
## 21       21 fmodel1 qmodel3 rmodel2 182.1 257.1
## 24       24 fmodel4 qmodel3 rmodel2   182 254.6
```

The models which have recruitment modelled as a factor on year have the lowest AIC and BIC (probably because there are fewer parameters in the recruitment model). The residuals from model 4 are show in Figure 7.7.

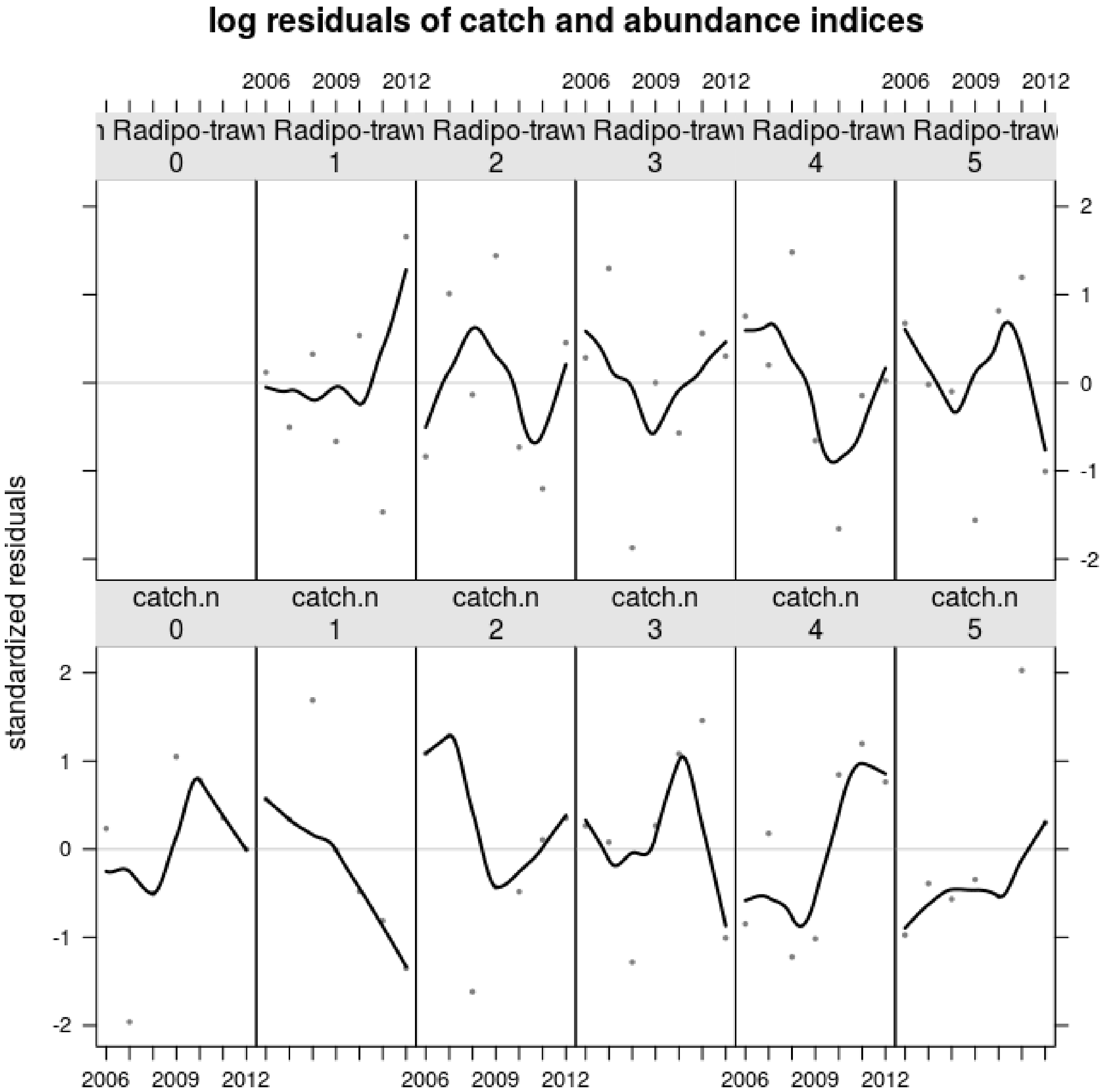


Figure 7.7: Residuals plot of model 4

As the AICs and BICs are close to model average (if we want to).

```
names(sole_stks_fit)


##  [1] "model1"  "model2"  "model3"  "model4"  "model5"  "model6"  "model7"
##  [8] "model8"  "model9"  "model10" "model11" "model12" "model13" "model14"
## [15] "model15" "model16" "model17" "model18" "model19" "model20" "model21"
## [22] "model22" "model23" "model24"


# Check AIC and BICs of these guys Shrink model_data
fits[paste("model", good_models, sep = "")]


## $model1
```

```
## a4a model fit for: NORTHERN ADRIATIC SEA (GSA 17) COMMON SOLE 2006-2012
##
## Call:
## .local(stock = stock, indices = indices, fmodel = ..1, qmodel = ..2,
##     srmodel = ..4, vmodel = ..3)
##
## Time used:
##  Pre-processing     Running a4a Post-processing           Total
##         0.31154         0.09765         0.07571         0.48491
##
## Submodels:
##   fmodel: ~factor(age) + factor(year)
##  srmodel: ~factor(year)
##  n1model: ~factor(age)
##   qmodel:
##     SoleMon Radipo-trawl survey: ~factor(age)
##   vmodel:
##     catch:                       ~1
##     SoleMon Radipo-trawl survey: ~1
##
## $model4
## a4a model fit for: NORTHERN ADRIATIC SEA (GSA 17) COMMON SOLE 2006-2012
##
## Call:
## .local(stock = stock, indices = indices, fmodel = ..1, qmodel = ..2,
##     srmodel = ..4, vmodel = ..3)
##
## Time used:
##  Pre-processing     Running a4a Post-processing           Total
##         0.25776         0.09510         0.05362         0.40647
##
## Submodels:
##   fmodel: ~factor(replace(age, age > 4, 4)) + factor(year)
##  srmodel: ~factor(year)
##  n1model: ~factor(age)
##   qmodel:
##     SoleMon Radipo-trawl survey: ~factor(age)
##   vmodel:
##     catch:                       ~1
##     SoleMon Radipo-trawl survey: ~1
##
## $model8
## a4a model fit for: NORTHERN ADRIATIC SEA (GSA 17) COMMON SOLE 2006-2012
##
## Call:
## .local(stock = stock, indices = indices, fmodel = ..1, qmodel = ..2,
##     srmodel = ..4, vmodel = ..3)
##
## Time used:
##  Pre-processing     Running a4a Post-processing           Total
##         0.27679         0.09268         0.05392         0.42339
```

```
##
## Submodels:
##   fmodel: ~factor(replace(age, age > 4, 4)) + factor(year)
##  srmodel: ~factor(year)
##  n1model: ~factor(age)
##   qmodel:
##     SoleMon Radipo-trawl survey: ~s(age, k = 3)
##   vmodel:
##     catch:                       ~1
##     SoleMon Radipo-trawl survey: ~1
##
## $model9
## a4a model fit for: NORTHERN ADRIATIC SEA (GSA 17) COMMON SOLE 2006-2012
##
## Call:
## .local(stock = stock, indices = indices, fmodel = ..1, qmodel = ..2,
##     srmodel = ..4, vmodel = ..3)
##
## Time used:
##  Pre-processing     Running a4a Post-processing           Total
##         0.27057         0.09515         0.05109         0.41680
##
## Submodels:
##   fmodel: ~factor(age) + factor(year)
##  srmodel: ~factor(year)
##  n1model: ~factor(age)
##   qmodel:
##     SoleMon Radipo-trawl survey: ~factor(replace(age, age > 4, 4))
##   vmodel:
##     catch:                       ~1
##     SoleMon Radipo-trawl survey: ~1
##
## $model12
## a4a model fit for: NORTHERN ADRIATIC SEA (GSA 17) COMMON SOLE 2006-2012
##
## Call:
## .local(stock = stock, indices = indices, fmodel = ..1, qmodel = ..2,
##     srmodel = ..4, vmodel = ..3)
##
## Time used:
##  Pre-processing     Running a4a Post-processing           Total
##         0.26976         0.08821         0.05295         0.41092
##
## Submodels:
##   fmodel: ~factor(replace(age, age > 4, 4)) + factor(year)
##  srmodel: ~factor(year)
##  n1model: ~factor(age)
##   qmodel:
##     SoleMon Radipo-trawl survey: ~factor(replace(age, age > 4, 4))
##   vmodel:
##     catch:                       ~1
```

```
##     SoleMon Radipo-trawl survey: ~1
##
## $model13
## a4a model fit for: NORTHERN ADRIATIC SEA (GSA 17) COMMON SOLE 2006-2012
##
## Call:
## .local(stock = stock, indices = indices, fmodel = ..1, qmodel = ..2,
##     srmodel = ..4, vmodel = ..3)
##
## Time used:
##  Pre-processing     Running a4a Post-processing           Total
##         0.27201         0.10715         0.05176         0.43092
##
## Submodels:
##   fmodel: ~factor(age) + factor(year)
##  srmodel: ~bevholt(CV = 0.1)
##  n1model: ~factor(age)
##   qmodel:
##     SoleMon Radipo-trawl survey: ~factor(age)
##   vmodel:
##     catch:                       ~1
##     SoleMon Radipo-trawl survey: ~1
##
## $model16
## a4a model fit for: NORTHERN ADRIATIC SEA (GSA 17) COMMON SOLE 2006-2012
##
## Call:
## .local(stock = stock, indices = indices, fmodel = ..1, qmodel = ..2,
##     srmodel = ..4, vmodel = ..3)
##
## Time used:
##  Pre-processing     Running a4a Post-processing           Total
##         0.27044         0.10929         0.05316         0.43290
##
## Submodels:
##   fmodel: ~factor(replace(age, age > 4, 4)) + factor(year)
##  srmodel: ~bevholt(CV = 0.1)
##  n1model: ~factor(age)
##   qmodel:
##     SoleMon Radipo-trawl survey: ~factor(age)
##   vmodel:
##     catch:                       ~1
##     SoleMon Radipo-trawl survey: ~1
##
## $model20
## a4a model fit for: NORTHERN ADRIATIC SEA (GSA 17) COMMON SOLE 2006-2012
##
## Call:
## .local(stock = stock, indices = indices, fmodel = ..1, qmodel = ..2,
##     srmodel = ..4, vmodel = ..3)
##
```

```
## Time used:
##  Pre-processing     Running a4a Post-processing           Total
##         0.27918         0.10762         0.05374         0.44054
##
## Submodels:
##   fmodel: ~factor(replace(age, age > 4, 4)) + factor(year)
##  srmodel: ~bevholt(CV = 0.1)
##  n1model: ~factor(age)
##   qmodel:
##     SoleMon Radipo-trawl survey: ~s(age, k = 3)
##   vmodel:
##     catch:                       ~1
##     SoleMon Radipo-trawl survey: ~1
##
## $model21
## a4a model fit for: NORTHERN ADRIATIC SEA (GSA 17) COMMON SOLE 2006-2012
##
## Call:
## .local(stock = stock, indices = indices, fmodel = ..1, qmodel = ..2,
##     srmodel = ..4, vmodel = ..3)
##
## Time used:
##  Pre-processing     Running a4a Post-processing           Total
##         0.27208         0.11029         0.05159         0.43396
##
## Submodels:
##   fmodel: ~factor(age) + factor(year)
##  srmodel: ~bevholt(CV = 0.1)
##  n1model: ~factor(age)
##   qmodel:
##     SoleMon Radipo-trawl survey: ~factor(replace(age, age > 4, 4))
##   vmodel:
##     catch:                       ~1
##     SoleMon Radipo-trawl survey: ~1
##
## $model24
## a4a model fit for: NORTHERN ADRIATIC SEA (GSA 17) COMMON SOLE 2006-2012
##
## Call:
## .local(stock = stock, indices = indices, fmodel = ..1, qmodel = ..2,
##     srmodel = ..4, vmodel = ..3)
##
## Time used:
##  Pre-processing     Running a4a Post-processing           Total
##         0.26059         0.11783         0.05441         0.43283
##
## Submodels:
##   fmodel: ~factor(replace(age, age > 4, 4)) + factor(year)
##  srmodel: ~bevholt(CV = 0.1)
##  n1model: ~factor(age)
##   qmodel:
```

```
##     SoleMon Radipo-trawl survey: ~factor(replace(age, age > 4, 4))
##   vmodel:
##     catch:                       ~1
##     SoleMon Radipo-trawl survey: ~1

model_average <- ma(a4aFitSAs(fits[paste("model", good_models,
    sep = "")]), sole_stk_xsa, AIC, nsim = 1000)
```

8 Multi-fleet projections for Sole in the Adriatic Sea

Finlay Scott & Giuseppe Scarcella

Here we demonstrate how the single fleet projection tools in FLR can be used to approximate multiple fleet projections using historical partial catches. Three fleets catch Sole in GSA 17: trammel nets, set nets and a trawl. We are particularly interested in exploring what happens if the trawl fleet stops operating. Therefore, in the following examples we compare scenarios where all three fleets are in operation and when only the trammel and set nets are in operation. The operation of multiple fleets is approximated by changing the selectivity pattern in the projection model. The selection pattern of each fleet is calculated through the partial catches and the total estimated fishing mortality.

8.1 Running the assessment

Read in the tuning indices and the stock object.

```
# FLIndices
idxs <- readFLIndices("data/TUNEFF.DAT")
# FLStock
load("data/stk.Rdata")
```

Run a stock assessment using `FLa4a` using the final settings from the sole assessment.

```
# Run chosen assessment - model 4 from previous section age
# and year seperable, with age > 4 being the same as age 4
fmodel <- ~factor(replace(age, age > 4, 4)) + factor(year)
qmodel <- list(~factor(age))
rmodel <- ~factor(year)
fit <- a4aSCA(stock = sole, indices = idxs, fmodel = fmodel,
    qmodel = qmodel, srmodel = rmodel)
sole_det <- sole + fit
```

We simulate from the fitted object to generate a stock object with multiple iterations that represent the uncertainty in the stock assessment (Figure 8.1).

```
niters <- 1000
sole_sim <- sole + simulate(fit, niters, seed = 0)
```

```r
plot(sole_sim)
```

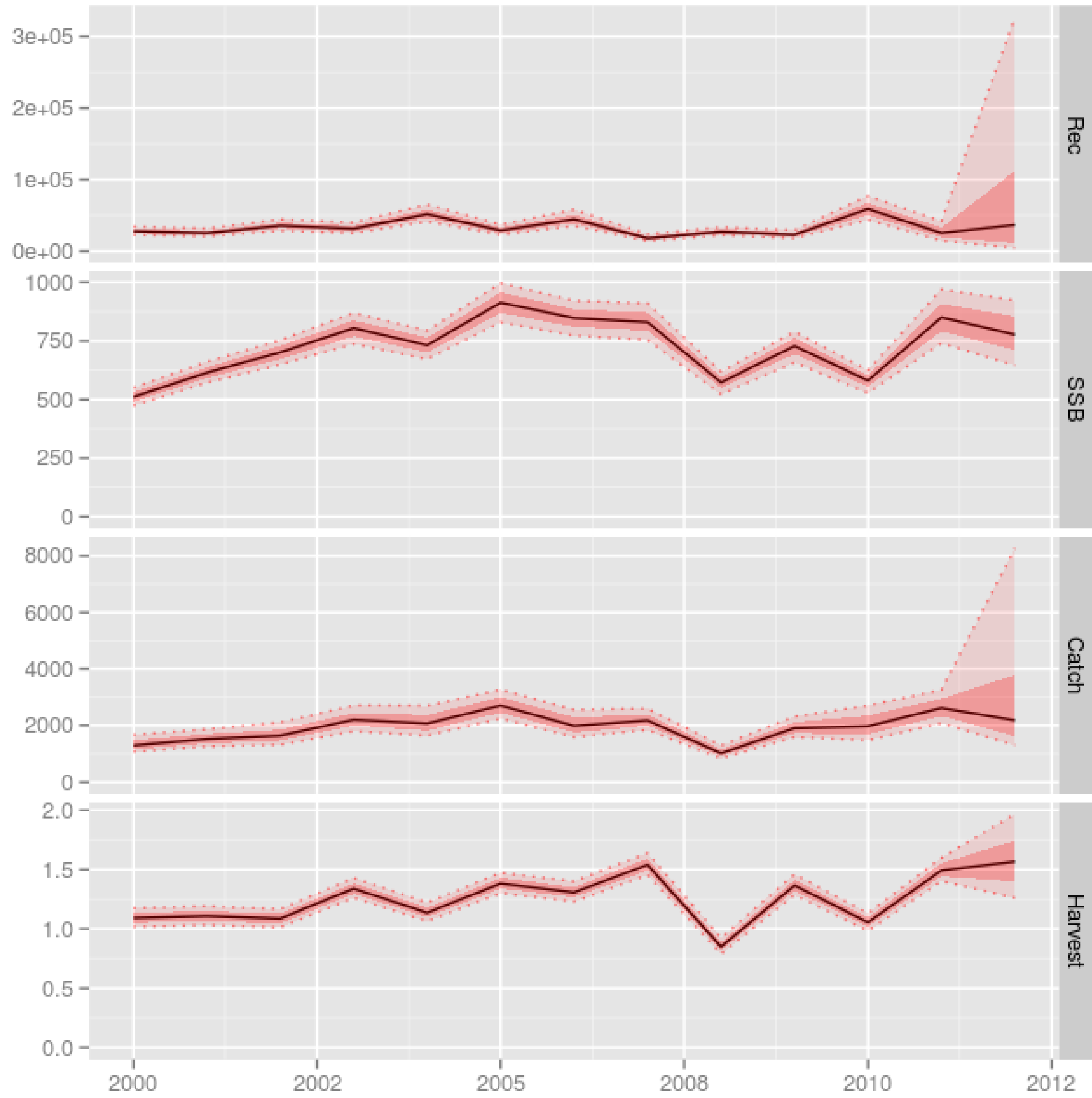


Figure 8.1: Results of the a4a assessment

8.2 Calculate the partial catches

The general approach to approximating multiple fleet projections is to estimate historical partial fishing mortality at age using the the historical partial catches. There are three main fleets fishing on Sole in GSA 17: set nets, trawlers and trammel nets.

First we read in the catch histories of the three fleets and calculate the proportion they contribute to the total catches (Figure 8.2). The trawl and set net fleet catches younger fish and the trammel net catches older fish.

```
# Read in catch numbers from the fleets
set_net <- read.csv("data/ITA_SET_NET.csv", header = TRUE, sep = ";")
set_net <- FLQuant(t(as.matrix(set_net)[, 2:8]), dimnames = list(age = 0:6,
    year = 2000:2012))
trawl <- read.csv("data/ITA_TRAWL.csv", header = TRUE, sep = ";")
trawl <- FLQuant(t(as.matrix(trawl)[, 2:8]), dimnames = list(age = 0:6,
    year = 2000:2012))
tram <- read.csv("data/SLO_CRO_TRAMMEL.csv", header = TRUE, sep = ";")
tram <- FLQuant(t(as.matrix(tram)[, 2:8]), dimnames = list(age = 0:6,
    year = 2000:2012))
total_catch <- set_net + trawl + tram
prop_catch_set_net <- set_net/total_catch
prop_catch_trawl <- trawl/total_catch
prop_catch_tram <- tram/total_catch
prop_catches <- FLQuants(set_net = prop_catch_set_net, trawl = prop_catch_trawl,
    tram = prop_catch_tram)
```

```
ggplot(as.data.frame(prop_catches), aes(x = year, y = data)) +
    geom_line(aes(colour = qname)) + facet_wrap(~age)
```

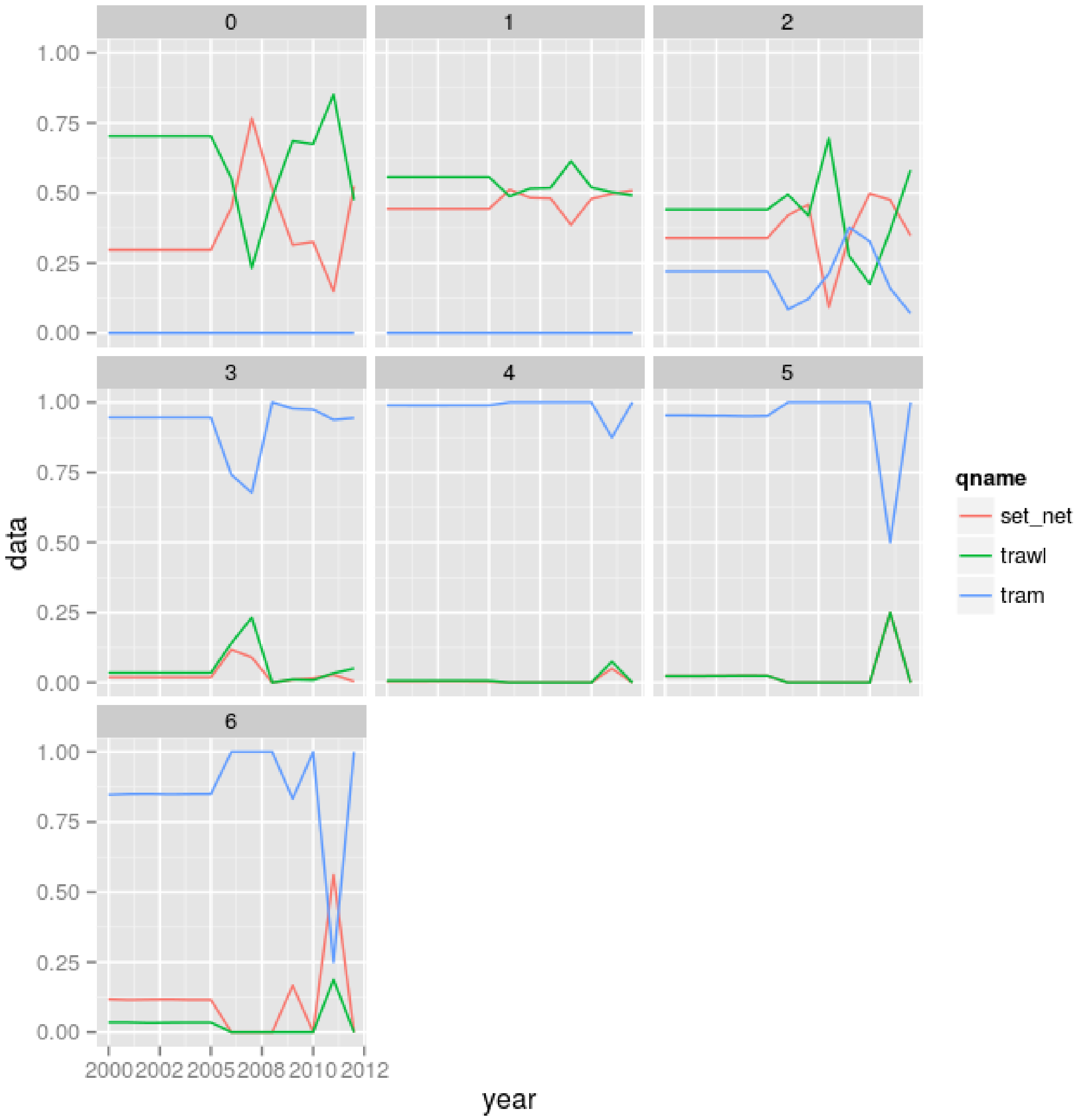


Figure 8.2: Catch proportions of the fleets by age over the years 2000 to 2012

We now calculate the partial fishing mortalities by multiplying the estimated fishing mortality from the assessment by the catch proportions (Figure 8.3). The selection patterns used in the projections are based on the mean of the partial fishing mortalities over the years 2006 to 2012 (Figure 8.4).

```
pfs <- lapply(prop_catches, function(x) sweep(harvest(sole_sim),
    1:5, x, "*"))
pfs_mean <- lapply(pfs, function(x) apply(x[, as.character(2006:2012)],
    c(1, 3:6), mean))
```

```
pfs_df <- as.data.frame(pfs)
ggplot(pfs_df[pfs_df$year > 2003, ], aes(factor(year), data)) +
    geom_boxplot(aes(colour = qname)) + facet_wrap(~age)
```

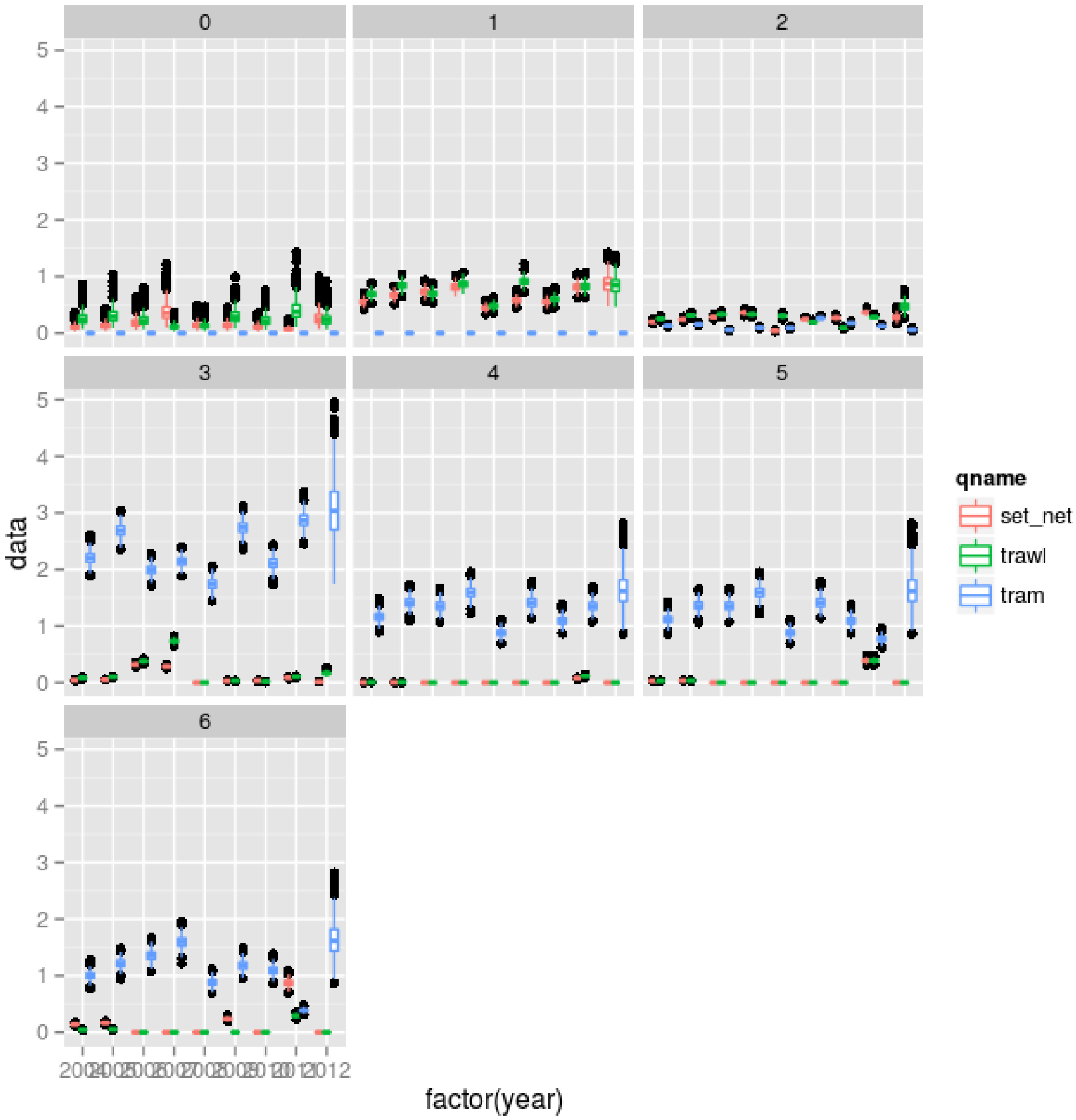


Figure 8.3: Partial fishing mortalities of the three fleets by age from 2003 to 2012

```
ggplot(as.data.frame(pfs_mean), aes(factor(age), data)) + geom_boxplot(aes(colour = qname))
```

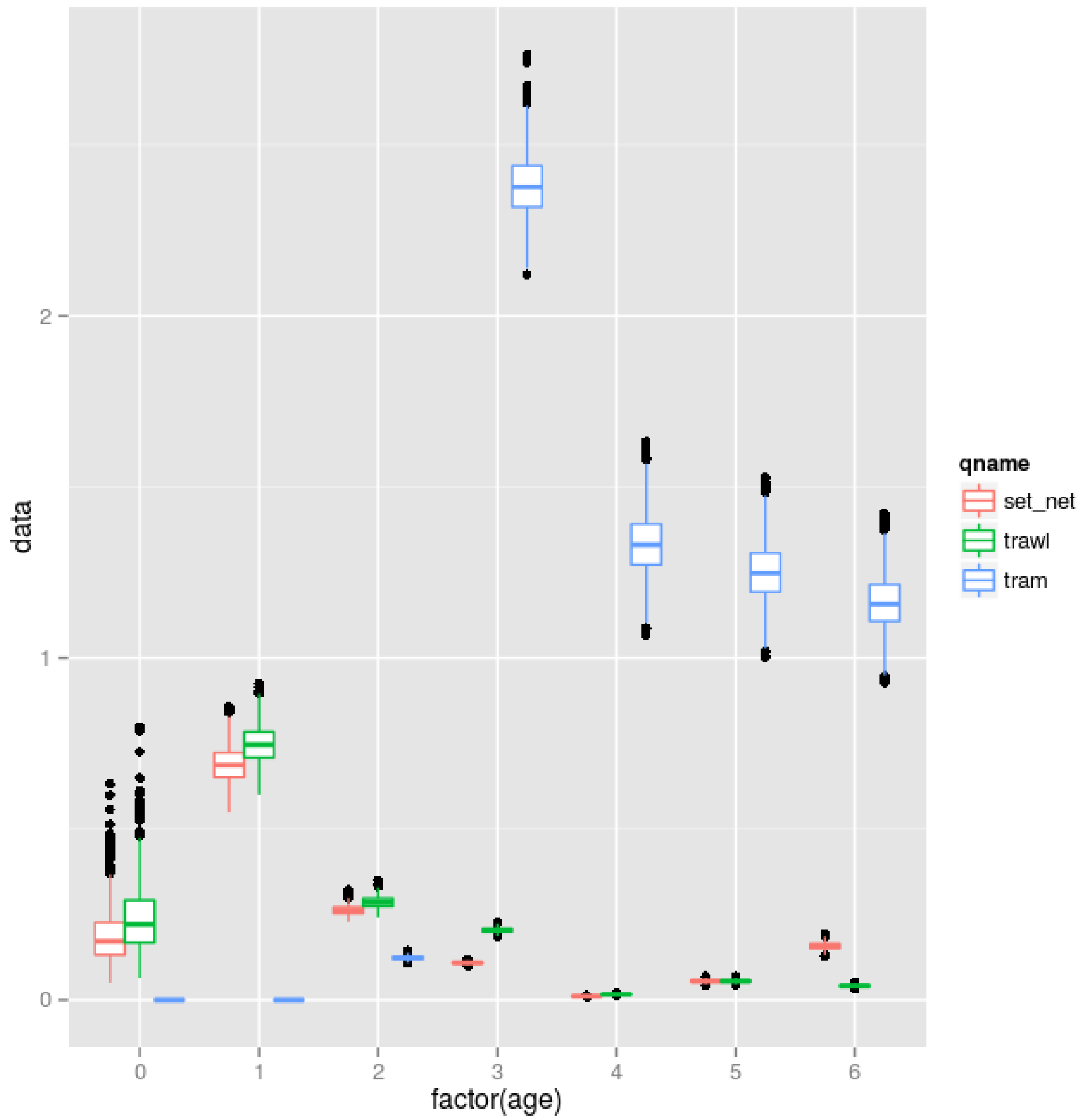


Figure 8.4: Mean partial fishing mortality over the years 2006 to 2012. This is used for the selection patterns in the projections.

8.3 Stock recruitment

For the projection we need some kind of stock recruitment relationship. First we attempt to fit a Beverton-Holt relationship between SSB and recruitment using the stock assessment results without uncertainty. However, the shortage of the data set (13 years) and the poorness of the fit (Figure 8.5) means that we cannot justifiably use this relationship for the projections.

```
sole_srr <- fmle(as.FLSR(sole_det, model = "bevholt"), control = list(trace = 0))
```

```
plot(sole_srr)
```

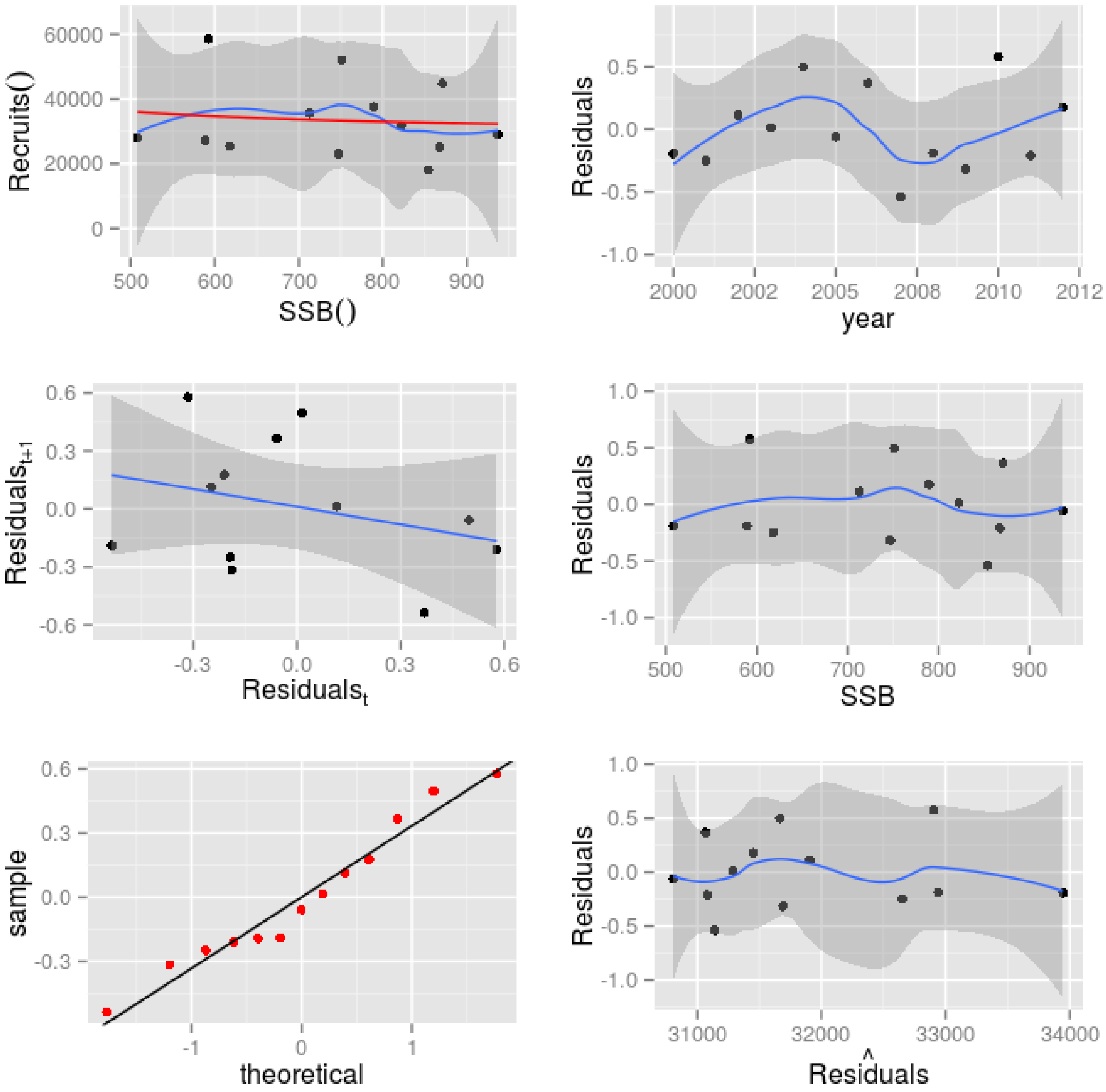


Figure 8.5: The fitted stock recruitment relationship using a Beverton-Holt function. This relationship was not used in the projections.

Instead, we used the geometric mean recruitment of the last three years of the simulated stock. The residuals are simply the historic estimated recruitment minus the mean recruitment.

```
obj <- rec(sole_sim)[, as.character(2010:2012)]
mean_rec <- apply(obj, c(1, 3:6), function(x) exp(mean(log(x))))
mult_residuals_rec <- sweep(rec(sole_sim), c(1, 3:6), mean_rec,
    "/")
```

8.4 The projections

We need to set up the future extended stock that will be used for the projections. There are several assumptions made about the future stock. For example, the future mean weights, natural mortality and maturity at age are calculated as the mean of the last three years. The future selection pattern is based on the mean partial fishing mortalities from the three fleets of the years 2006 to 2012 (see Figure 8.4).

```
nyears <- 20
# Multiplicative residuals for the SRR
residual_array <- aperm(apply(mult_residuals_rec, c(1, 3:6),
    sample, size = nyears, replace = TRUE), c(2, 1, 3, 4, 5,
    6))
dimnames(residual_array) <- list(age = 0, year = 2013:(2013 +
    nyears - 1), unit = "unique", season = "all", area = "unique",
    iter = 1:niters)
sr_residuals <- FLQuant(residual_array)
```

8.4.1 Status quo with all fleets

The status quo scenario is that the total fishing mortality will be the same as the mean of the last three years (Figure 8.6). The future selection pattern is based on the mean partial fishing mortality from the years 2006 to 2012 (see Figure 8.7). The projections are performed with multiplicative residuals on the mean recruitment

```
sole_stf <- stf(sole_sim, nyears = nyears)
# Set the future selection pattern using the partial fishing
# mortalities
obj <- pfs_mean[["tram"]] + pfs_mean[["set_net"]] + pfs_mean[["trawl"]]
harvest(sole_stf)[, as.character(2013:(2013 + nyears - 1))] <- obj

# get F status quo
fsq <- apply(fbar(sole_sim)[, as.character(c(2010, 2012))], c(1,
    3:6), mean)
ctrl_target <- fwdControl(data.frame(year = 2013:(2013 + nyears -
    1), quantity = "f", val = rep(c(iter(fsq, 1)), nyears)))
# Fix trgtArray
trgtArray <- array(NA, dim = c(nyears, 3, niters), dimnames = list(1:nyears,
    c("min", "val", "max"), iter = 1:niters))
trgtArray[, "val", ] <- rep(c(fsq), each = nyears)
ctrl_target@trgtArray <- trgtArray
stf_sq <- fwd(sole_stf, ctrl_target, sr = list(model = "mean",
    params = FLPar(a = c(mean_rec), iter = niters)), sr.residuals = sr_residuals,
    sr.residuals.mult = TRUE)
```

```
plot(stf_sq)
```

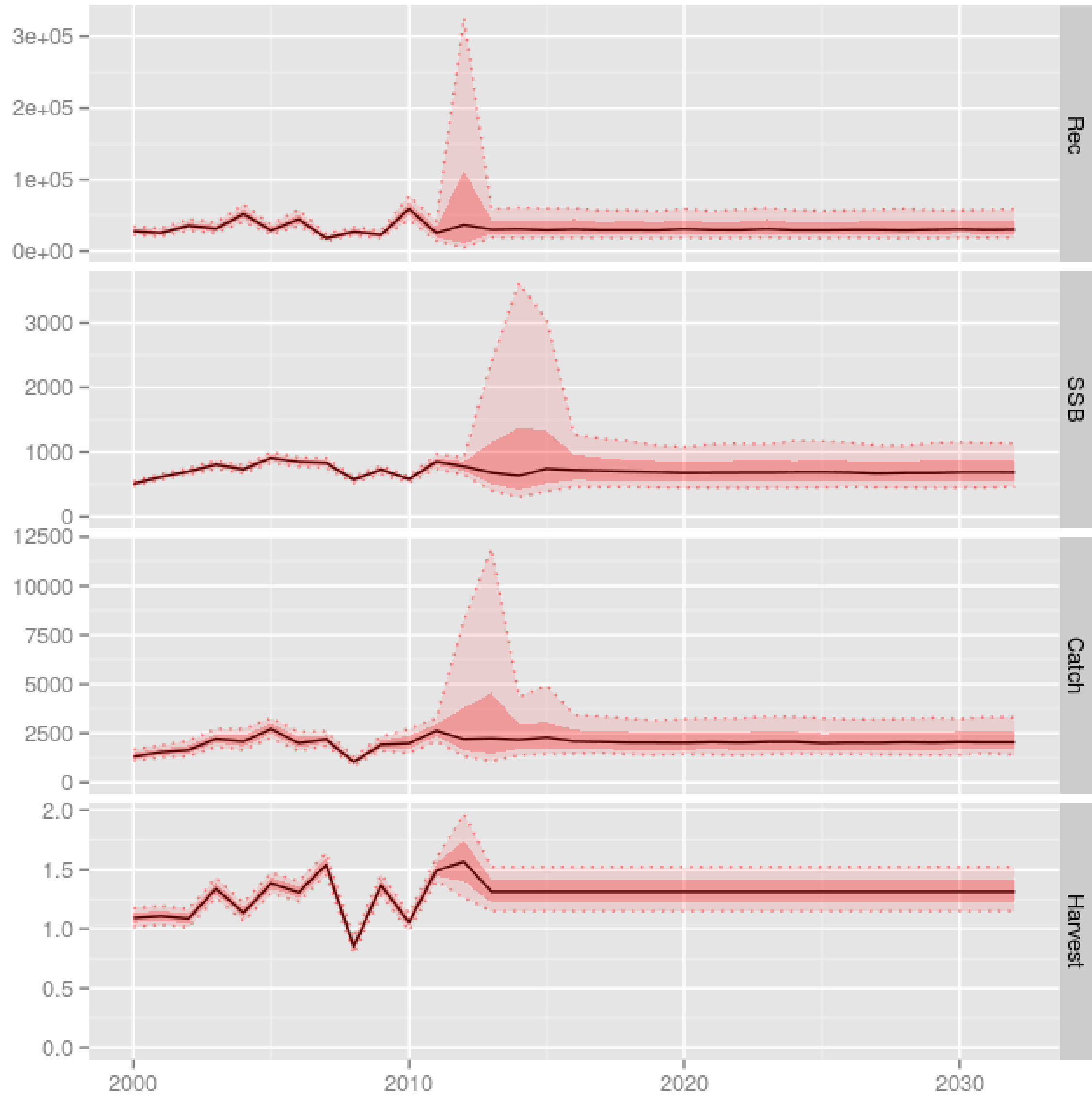


Figure 8.6: The results of the status quo projection.

8.4.2 Status quo projection without the trawl fleet

Here we project forward without the trawling fleet. The fishing mortalities of the other two fleets (set net and trammel) are kept at their status quo levels. As we are only able to perform a 'single fleet' projection, the selectivity in the projection will be the sum of the selectivities of the two fleets. We use the mean partial fishing mortalities we calculated earlier as the future selection pattern (this will be scaled accordingly in the projection). The status quo fishing mortality is the mean of the last three years. As before, projections are performed with multiplicative residuals on the mean recruitment (Figure 8.7).

The difference between this scenario and the status quo scenario is that here the selection pattern is based only the trammel and set net fleets, and the future fishing mortality level is the mean of the last three years of the trammel and set net fleets only.

```
sole_notrawl <- stf(sole_sim, nyears = nyears)
# Overwrite harvest slot with sum of fleets set_net + trammel
# to set the future selectivity (2013 onwards)
obj <- pfs_mean[["tram"]] + pfs_mean[["set_net"]]
harvest(sole_notrawl)[, as.character(2013:(2013 + nyears - 1))] <- obj

# The status quo fishing mortality is the mean of the last
# three years (of the two fleets only)
fbar_range <- as.character(range(sole_sim)["minfbar"]:range(sole_sim)["maxfbar"])
fbar_set_net_and_tram <- apply((pfs[["set_net"]] + pfs[["tram"]])[fbar_range,
    ], 2:6, mean)
fsq_notrawl <- apply(fbar_set_net_and_tram[, as.character(c(2010,
    2012))], c(1, 3:6), mean)
# Set the control object
ctrl_target_notrawl <- fwdControl(data.frame(year = 2013:(2013 +
    nyears - 1), quantity = "f", val = rep(c(iter(fsq_notrawl,
    1)), nyears)))
# Fix trgtArray
trgtArray <- array(NA, dim = c(nyears, 3, niters), dimnames = list(1:nyears,
    c("min", "val", "max"), iter = 1:niters))
trgtArray[, "val", ] <- rep(c(fsq_notrawl), each = nyears)
ctrl_target_notrawl@trgtArray <- trgtArray
# With multiplicative residuals
stf_sq_notrawl <- fwd(sole_notrawl, ctrl_target_notrawl, sr = list(model = "mean",
    params = FLPar(a = c(mean_rec), iter = niters)), sr.residuals = sr_residuals,
    sr.residuals.mult = TRUE)
```

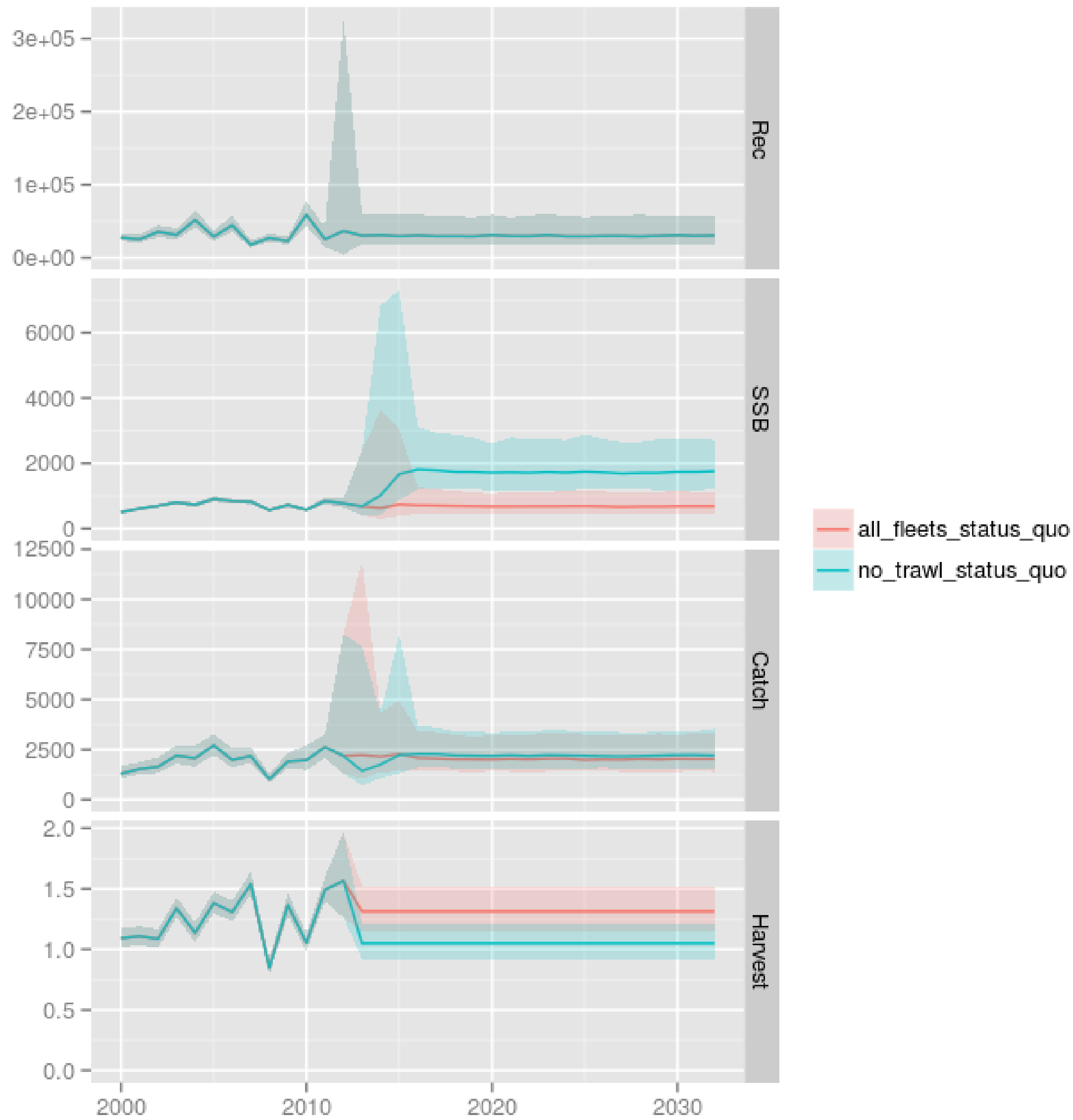


Figure 8.7: Comparing the results of the status quo scenario to the status quo without the trawl fleet scenario.

We can compare the total fishing mortality in the projection years with the status quo scenario in Section 8.4.1 (Figure 8.8).

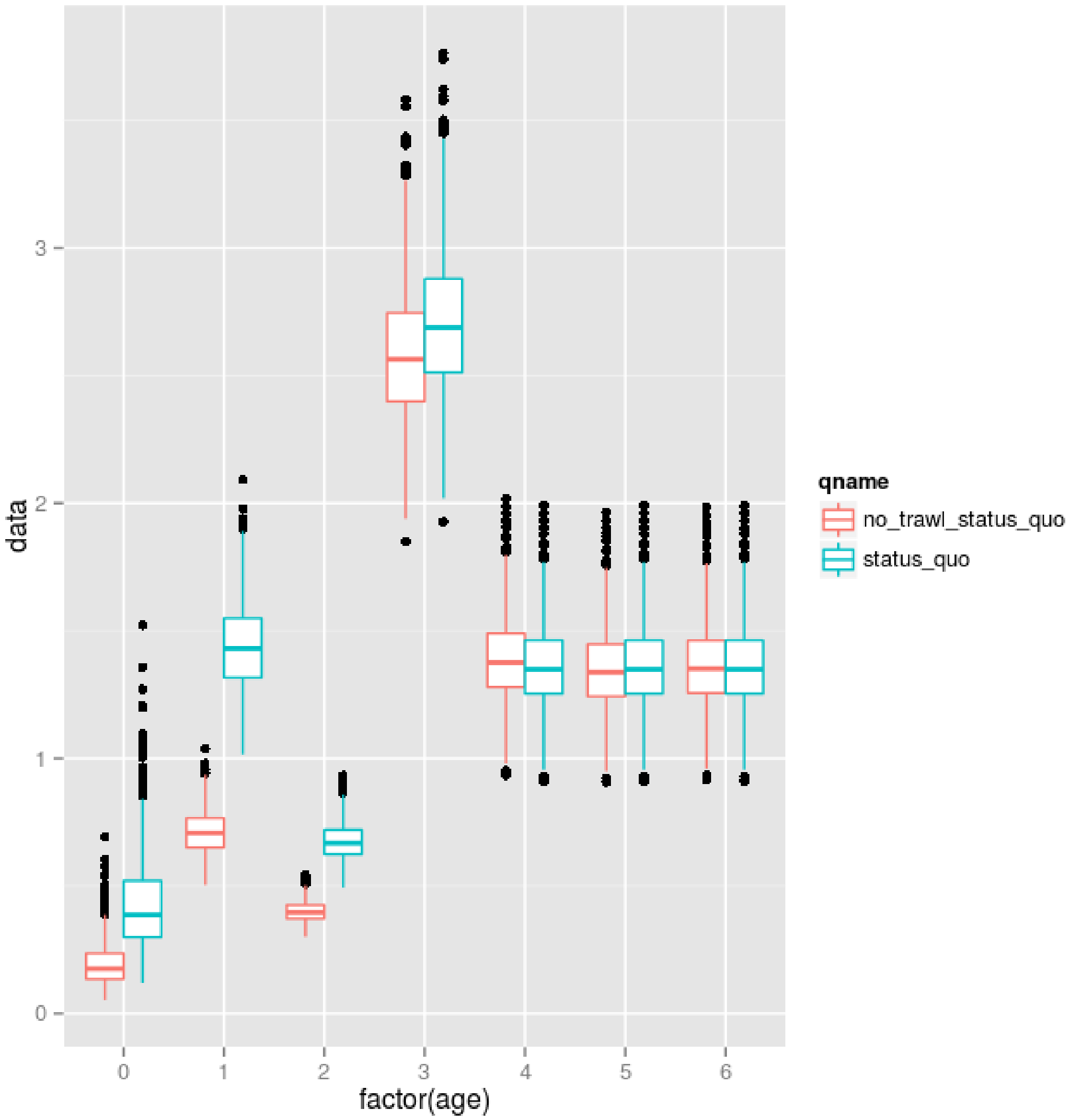


Figure 8.8: The fishing mortality by age in 2020 for the status quo and status quo without the trawl fleet scenarios.

8.4.3 Only set and trammel nets with catches at status quo level

In this scenario we take the catches from the status quo scenario with all of the fleets, and use these in the projection using only the set and trammel nets. This can be considered a 'compensation' scenario, i.e. if the trawl fleet is not operating the other two fleets will take the extra catch (so the total catch is the same as the status quo scenario when the trawl fleet is operating).

```
# Get the total catches from the status quo scenario
catch_trawl <- catch(stf_sq)[, ac(2013:(2013 + nyears - 1))]
ctrl_target_catch <- fwdControl(data.frame(year = 2013:(2013 +
    nyears - 1), quantity = "catch", val = c(iter(catch_trawl,
```

```
    1))))
# Fix trgtArray
trgtArray <- array(NA, dim = c(nyears, 3, niters), dimnames = list(1:nyears,
    c("min", "val", "max"), iter = 1:niters))
trgtArray[, "val", ] <- c(catch_trawl)
ctrl_target_catch@trgtArray <- trgtArray
# Use the notrawl fleet with the selection pattern based on
# set net and trammel net only
stf_trawlcatch_notrawl <- fwd(sole_notrawl, ctrl_target_catch,
    sr = list(model = "mean", params = FLPar(a = c(mean_rec),
        iter = niters)), sr.residuals = sr_residuals, sr.residuals.mult = TRUE)
```

Without the trawl fleet there is an initial increase in fishing mortality which slowly decreases below the status quo level. Additionally, without the trawl fleet the SSB level starts to increase (Figure 8.9). It must be remembered that there is no stock recruitment relationship in these projections, i.e. the recruitment is not affected by SSB. With a stock recruitment relationship it would be expected that the increase in SSB would occur earlier due to potential increases in recruitment.

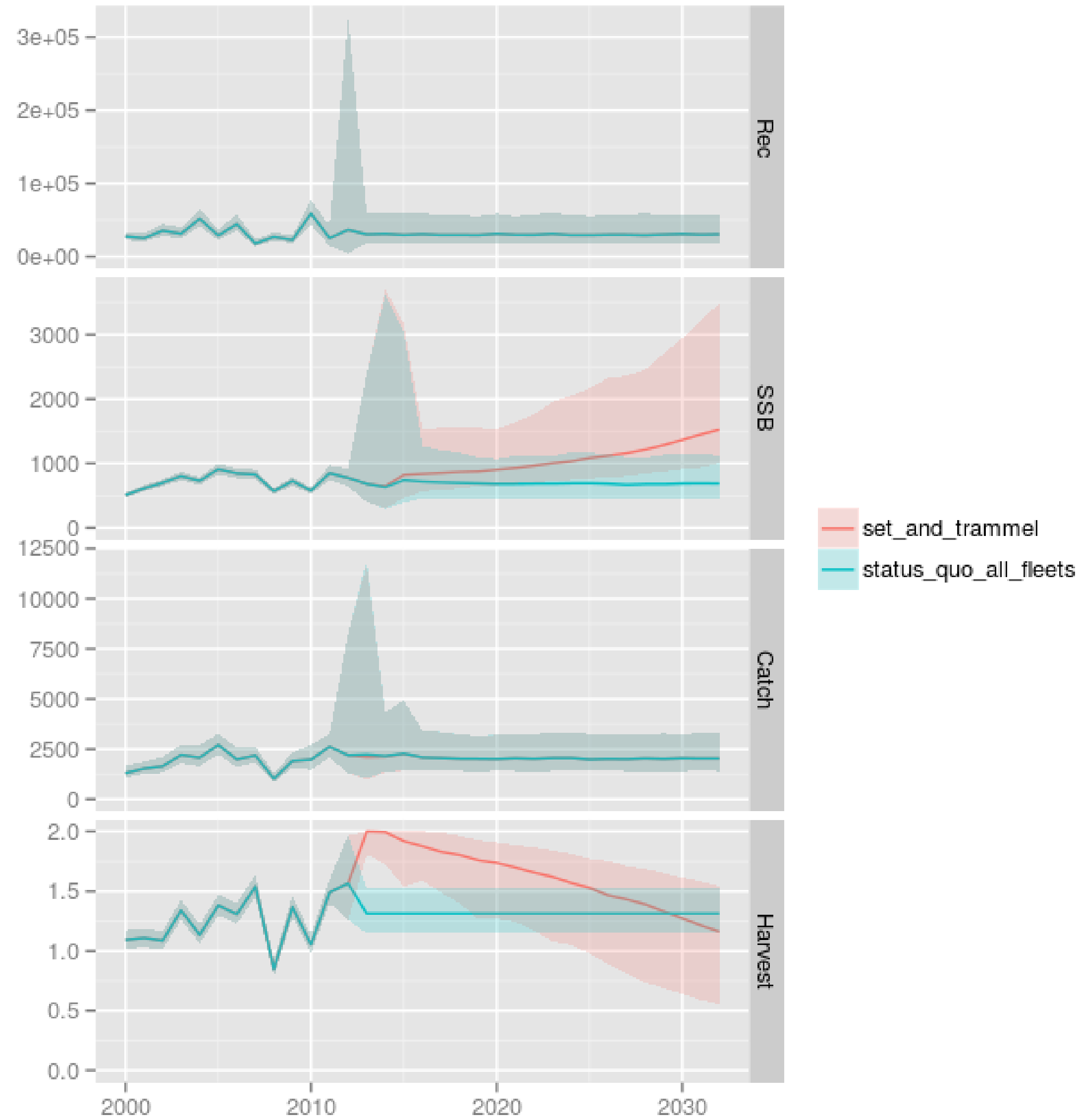


Figure 8.9: Comparing the status quo scenario with all three fleets to having only set and trammel nets in operation but the same catches.

8.4.4 Projecting at F01 with all fleets and with only the set net and trammel net fleets

In these scenarios we are interested in projecting at F0.1. The value of F0.1 will be affected by whether or not the trawl fleet is operating due to changes in the combined selectivity pattern. However, the values are similar (Figure 8.10).

```
# Now BRP these in batches of 200
f01_all <- c()
f01_all <- c(f01_all, c(refpts(brp(FLBRP(iter(stf_sq, 1:200))))["f0.1",
    "harvest"]))
```

```
f01_all <- c(f01_all, c(refpts(brp(FLBRP(iter(stf_sq, 201:400))))["f0.1",
    "harvest"]))
f01_all <- c(f01_all, c(refpts(brp(FLBRP(iter(stf_sq, 401:600))))["f0.1",
    "harvest"]))
f01_all <- c(f01_all, c(refpts(brp(FLBRP(iter(stf_sq, 601:800))))["f0.1",
    "harvest"]))
f01_all <- c(f01_all, c(refpts(brp(FLBRP(iter(stf_sq, 801:1000))))["f0.1",
    "harvest"]))
f01_notrawl <- c()
f01_notrawl <- c(f01_notrawl, c(refpts(brp(FLBRP(iter(sole_notrawl,
    1:200))))["f0.1", "harvest"]))
f01_notrawl <- c(f01_notrawl, c(refpts(brp(FLBRP(iter(sole_notrawl,
    201:400))))["f0.1", "harvest"]))
f01_notrawl <- c(f01_notrawl, c(refpts(brp(FLBRP(iter(sole_notrawl,
    401:600))))["f0.1", "harvest"]))
f01_notrawl <- c(f01_notrawl, c(refpts(brp(FLBRP(iter(sole_notrawl,
    601:800))))["f0.1", "harvest"]))
f01_notrawl <- c(f01_notrawl, c(refpts(brp(FLBRP(iter(sole_notrawl,
    801:1000))))["f0.1", "harvest"]))
```

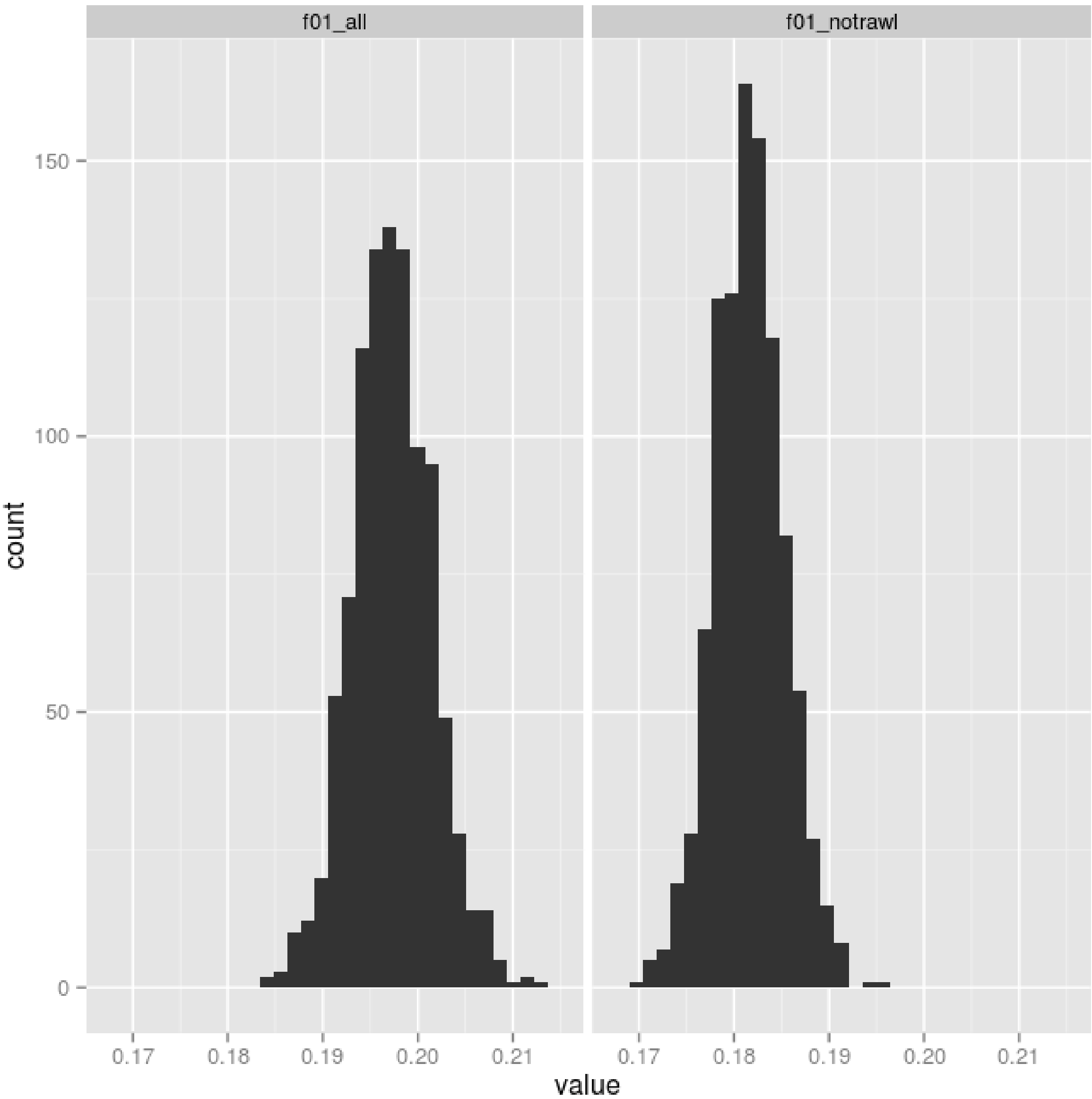


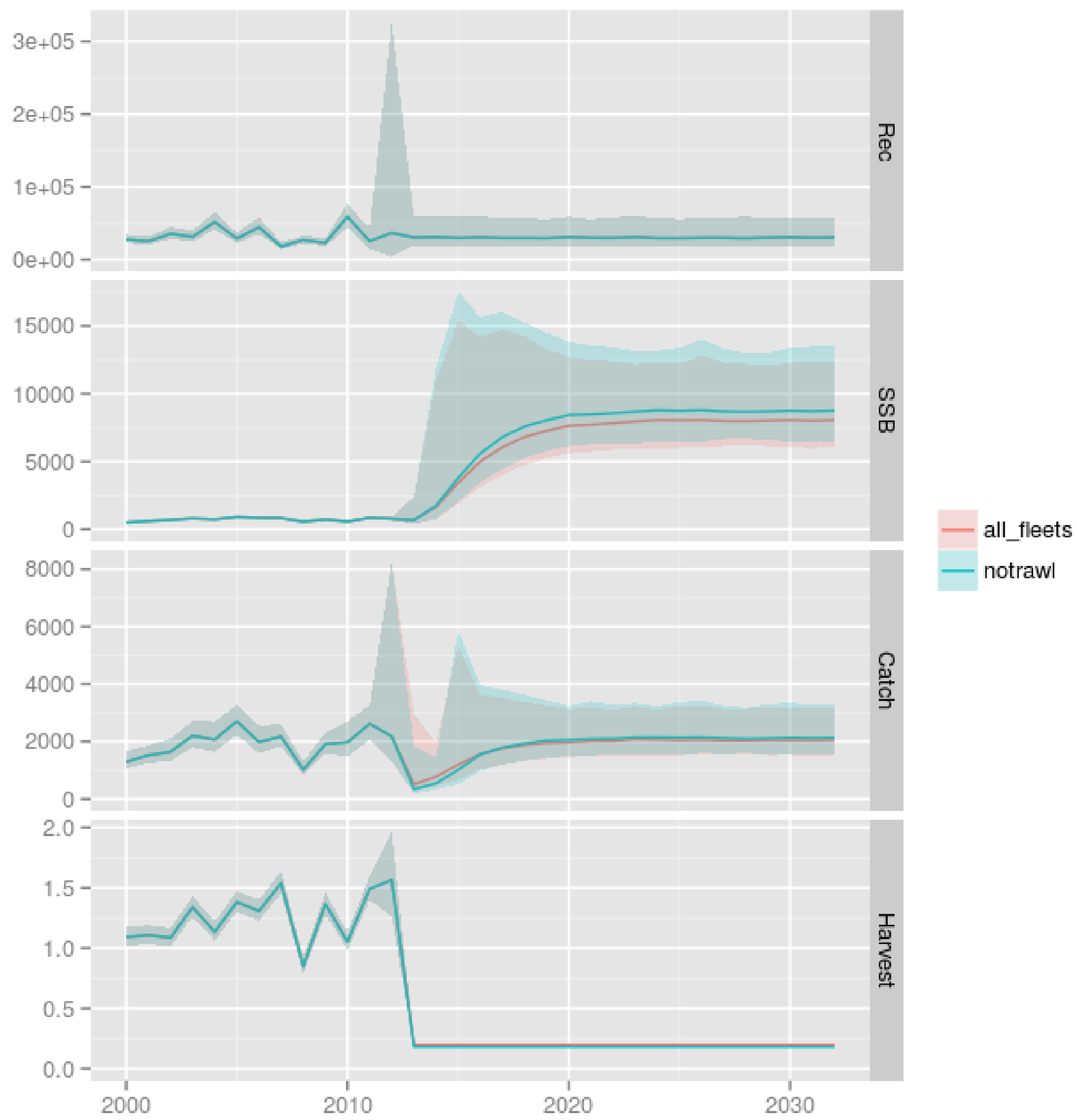
Figure 8.10: Histograms of the F0.1 values when all fleets and when only the set and trammel fleets are in operation

First we project with all fleets, but with the future fishing mortality being set at F0.1.

```
# Project all fleets at Fmsy
ctrl_target_f01_all <- fwdControl(data.frame(year = 2013:(2013 +
    nyears - 1), quantity = "f", val = rep(f01_all[1], nyears)))
# Fix trgtArray
trgtArray <- array(NA, dim = c(nyears, 3, niters), dimnames = list(1:nyears,
    c("min", "val", "max"), iter = 1:niters))
trgtArray[, "val", ] <- rep(f01_all, each = nyears)
ctrl_target_f01_all@trgtArray <- trgtArray
# Project
stf_f01_all <- fwd(sole_stf, ctrl_target_f01_all, sr = list(model = "mean",
    params = FLPar(a = c(mean_rec), iter = niters)), sr.residuals = sr_residuals,
    sr.residuals.mult = TRUE)
```

Then we project using only the set and trammel nets (by using the stock object with the adjusted selectivity pattern from above).

```
# Project notrawl fleets at Fmsy
ctrl_target_f01_notrawl <- fwdControl(data.frame(year = 2013:(2013 +
    nyears - 1), quantity = "f", val = rep(f01_notrawl[1], nyears)))
# Fix trgtArray
trgtArray <- array(NA, dim = c(nyears, 3, niters), dimnames = list(1:nyears,
    c("min", "val", "max"), iter = 1:niters))
trgtArray[, "val", ] <- rep(f01_notrawl, each = nyears)
ctrl_target_f01_notrawl@trgtArray <- trgtArray
# Project
stf_f01_notrawl <- fwd(sole_notrawl, ctrl_target_f01_notrawl,
    sr = list(model = "mean", params = FLPar(a = c(mean_rec),
        iter = niters)), sr.residuals = sr_residuals, sr.residuals.mult = TRUE)
```



9 Discussion and conclusions

Ernesto Jardim

The exercises carried out during this workshop were extremely valuable to test the a4a statistical catch-at-age model with real datasets. The stocks used as case studies, hake in the Gulf of Lions, hake in the Southern Adriatic, and anchovy, sardine and sole from the North Adriatic, encompass a diversity of life histories (small pelagics, demersals, round and flat fish) and fleets (purse seiners, trawlers, gillnetters) that cover a fair range of fisheries in Europe.

All stocks could fit in our definition of moderate-data, although in some cases the time series of the data are very short, which creates an additional problem to fit models. In most cases it was possible to replicate last year's assessments. For all cases it was possible to explore alternative models and get satisfactory fits.

The flexibility that the a4a stock assessment framework provides was the major factor fostering these results, but as expected flexibility has a downside. The most important one being the difficulty in selecting a model using the traditional statistical information criteria like AIC and BIC or visual analysis of diagnostics. The way forward proposed and applied to hake in the Gulf of Lions and sole in the Northern Adriatic, is to select a range of models that seem plausible for the problem and average across them. This is an area of research that deserves attention. We're all aware that modelling natural resources is hardly done by a single model, and most of the times a range of models is required to pick up all (or most) of the relevant processes.

In addition to stock assessment we used the two hake stocks to test the methods to model growth and natural mortality. Converting length to ages allows the user to use length based datasets and introduce uncertainty on the growth model, or test distinct models. With natural mortality the same principles apply, it allows the user to add uncertainty to the parameters of the model as well as uncertainty about this parameters.

One of the most interesting research questions we had was to test if it was possible to run projections in FLa4a/FLR using individual fleets. Such method allows testing management options that deal with fleet's effort or capacity, technical measures that impact the gear's characteristics, marine protect areas, etc. Any option that may change the overall fishing mortality deployed in each age group can be tested with this method. The case study used was sole in the Northern Adriatic. The methods presently implemented in FLa4a and FLR do not estimate or project individual fleets. The alternative is to use partial fishing mortality by fleet to set up distinct scenarios, which can be forecasted with FLR. The results were very promising and it was possible to test distinct scenarios of fleet management.

Finally, the a4a framework together with the existing tools in FLR, provided an efficient way of running stock assessments, including a fair amount of uncertainty sources, and forecasting. The methods implemented allow the user to run all the analysis in the same framework, which is a major improvement in terms of efficiency, allowing the analysts to address more time to the structural configuration of the processes they're modelling and spend less time dealing with technical details.

European Commission

EUR 26716 EN – Joint Research Centre – Institute for the Protection and Security of the Citizen

Title: a4a short research project - Stock assessment and multi-fleet forecast methods for some Mediterranean stocks.

Authors: Ernesto Jardim, Dimitrios Damalas, Iago Mosqueira, Chato Osio Tristan Rouyer, Giuseppe Scarcella & Finlay Scott

Luxembourg: Publications Office of the European Union

2014 – 250 pp. – 21 x 29.7 cm

EUR – Scientific and Technical Research series – ISSN 1831-9424 (online), ISSN 1018-5593 (print)

ISBN 978-92-79-38997-9

doi:10.2788/88416

Abstract

Under the scope of the a4a Initiative, the JRC is promoting cooperative activities between fisheries scientists with the aim to test, disseminate and promote a4a methods. These Small Research Projects (SRP) focus on comparing the results of assessments from other models to assessments obtained from the a4a statistical catch-at-age model, and explore research questions using case studies. The Workshop dedicated to the Mediterranean took place in Ispra, Italy, the 23rd to the 27th of June. The main objectives were to compare assessment models and develop multi-fleet forecasts methodologies. These can be applied in the context of ex-ante/ex-post evaluations of multi-annual plans, performed by STECF in order to provide scientific advice to the European Commission.

How to obtain EU publications

Our priced publications are available from EU Bookshop (http://bookshop.europa.eu), where you can place an order with the sales agent of your choice.

The Publications Office has a worldwide network of sales agents. You can obtain their contact details bysending a fax to (352) 29 29-42758.

LB-NA-26716-EN-N

As the Commission's in-house science service, the Joint Research Centre's mission is to provide EU policies with independent, evidence-based scientific and technical support throughout the whole policy cycle.

Working in close cooperation with policy Directorates-General, the JRC addresses key societal challenges while stimulating innovation through developing new standards, methods and tools, and sharing and transferring its know-how to the Member States and international community.

Key policy areas include: environment and climate change; energy and transport; agriculture and food security; health and consumer protection; information society and digital agenda; safety and security including nuclear; all supported through a cross-cutting and multi-disciplinary approach.

Publications Office

doi:10.2788/88416
ISBN 978-92-79-38997-9