SS3: User Guide for generating GFCM assessment advice output

Henning Winker (GFCM - WGAs)

Contents

1 Background

In response to the requests of the SAC in 2021, 2022 and 2023, a new Summary Sheet Template is proposed towards improving and harmonizing the reporting on the stock status advice, with particular focus on quantitative advice for key stocks under management plan. This proposal was discussed and presented to all SRCs, the SAC and the WGBS in 2024 to gather their views. The SAC endorsed the canvas for a more detailed advice reporting and agreed to test and implement a new summary sheet in the WGSAs 2024.

This proposed Summary Sheet is designed to provide standardized tables and figures to summarize the following key information:

1. Headline advice in the form of a Table Advice, with text elements provided a summary box, including a new summary section of the assessment with respect to stock status and, where applicable, future fishing opportunity scenarios based on forecasts. The Advice Table shall include a the % $F_{reduction} = (1 - (1/F_{ratio})) * 100$, $F < F_{target}$.

2. Time series plots of stock status indicators including uncertainty. For age-structured models these include time-series of Recruitment, SSB, F and total, together with reference points for F_{target} and biomass reference points. For model ensembles (e.g. SOL17) time series of biomass and fishing mortality should be shown a ratios to B_{target}, F/F_{target}. Similarly, this also applies stocks for which advise is based on Surplus production Models (SPiCT, JABBA), but there is no need to provide information for recruitment.

3. Tables and a Figure summarizing fishing opportunity scenarios based on short- and medium-term forecasts. These may include fishing mortality (effort) and/or catch based forecasts. Where applicable. This should be aligned to the specifications in the in the multi annual management plan (MAP), for example, if catch limits are specified. In general it is suggested to implement:

    - F-based or Catch-based short-term forecasts for all stocks with quantitative advice (intermediate years + 1 year).
    - F-based medium-term forecasts (typically 5 years or until the end of the long-term management plan)
    - F-based long-term forecasts (typically 2 generation times), where the rebuild horizon is to levels around B_{target} is beyond the medium-term forecast (e.g. SBR1-3 or HKE17-18)

4. A technical summary of the assessment based on the text section existing the previous Summary Sheet template:

    - Brief description of the fishery
    - Assessment method
    - Input data
    - Assessment Quality

5. Comparative plot of the updated assessment with the previous assessment

6. Plot showing the partial fishing mortality by gear or fishing fleet. This is only applicable to integrated assessment models that can separate fishing fleets or gear (e.g. Stock Synthesis). This is particularly relevant for evaluating partial impacts on European hake.

This document seeks to provide step-by-step user guide to produce standardised outputs from Stock Synthesis assessments in the form input data for the STAR template, and the proposed Tables and Figures for the new Summary Sheet template.
For this purpose, several R functions were developed and implemented in the `FLR` package `FLRef` to facilitate the process of generating:

- `.csv` files of time series (SSB,F, Recruitment, Catch) and reference points that are compatible for input to the to STAR template, as well as, forecast results
- Figures showing the the stock indicators with uncertainty, fishing opportunity scenarios (forecasts), comparison of assessment updates.

The procedure is currently implement for the most commonly used assessment platforms for providing quantitative advice, including a4a, FLSAM, xsa, Stock Synthesis, SPiCT and JABBA.

1.1 Overview of the approach and general data organisation

Stock Synthesis is typically executed in a model folder(s). It is recommended the the model folder with the final model run is conventionally named using the following elements, separated by "_":

- The 3-letter alpha code for the species (e.g. "HKE")
- GSAs (e.g. "_17_18")
- Model name (e.g. "ss3")
- Reference Year for the most recent input data (e.g. "_2023")

So that the final model run for European hake in the Adriatic Sea is named "HKE_12_13_15_16_16_ss3_2023". This file should always be uploaded for future reference as a zip file `HKE_17_18_ss3_2023.zip` to the GFCM SharePoint. At a minimum, the zip should contain the key files to reproduce model:

- `starter.ss`
- `forecast.ss`
- `Dat` file
- `Control` file

In addition the assessment model output should be read as `ss3rep = SSoutput()` with the package `r4ss` and saved as corresponding `HKE_17_18_ss3_2023.rdata` file on the SharePoint.

Ideally, all `.rdata` and `.rds` files that are produced by the following step-by-step procedure be provided in the respective assessment folder on the SharePoint.

Where applicable the results from the forecasts should also be uploaded to SharePoint as `.rdata` file (See Supplement).

The `FLR` package `FLRef` has implemented an option for loading Stock Synthesis assessment output and then converting the outputs into a simplified `FLStockR` object with associated reference points, which holds all required data for input to the STAR Template. To do this, the original `ss3diags` function `SSdeltaMLVN()` (see ss3diags handbook to do this has been adopted and slightly modified in the form the function `ssmvln()` for `FLRef`.

This simplified `FLStockR` object should be saved as `HKE_17_18_stk2.rds`, file and uploaded to the GFCM WGAs subfolder `OutputFiles_stk2.rds` under `Input data and Scripts`. Therefore, the file name should comprise the following elements, separated by "_":

- The 3-letter alpha code for the species (e.g. "HKE")
- GSAs (e.g. "_17_18")
- stk2 (i.e. "_stk2")

where "stk2" is intended to denote that the stock object `FLStockR` represents the model outcomes in constrast of an `FLStock` input ("stk1") for, e.g. `xsa` or `a4a` models

The generic R commands to save and load `.rds` files are:

```
# save
saveRDS(stk2, file = "HKE_17_18_stk2.rds")
# load
stk2 = readRDS(file = "HKE_17_18_stk2.rds")
```

2 Installation guidelines

2.1 Install FLR packages

2.1.1 Basic packages FLCore and ggplotFL

All FLR based packages are now available on r-universe (similar R cran), which means that for Windows users the packages are readily pre-compiled.

It is recommended to first install and test the two basic FLR packages FLCore and ggplot before proceeding with "heavier machinery"

Before installing the packages clear your R environment

```
rm(list = ls())
```

and re-start R.

```
install.packages(c("FLCore", "ggplotFL"), repo = "https://flr.r-universe.dev")
```

Load packages

```
library(FLCore)
library(ggplotFL)
```

2.2 Install FLBRP, FLFishery FLasher

- FLBRP solving for reference points in FLR
- FLasher forecasting (requires FLFishery)

both packages use C++ in the background.

Please ensure that FLash is not loaded before installing or using FLasher. There are many conflicts. Perhaps best to remove FLash.

```
install.packages(c("FLBRP", "FLFishery", "FLasher"), repo = "https://flr.r-universe.dev")
```

Load packages

```
library(FLBRP)
library(FLasher)
   Loading required package: FLFishery
   FLasher: No sanctuary
```

2.3 FLSRTMB, mse and FLRef

- FLSRTMB for stock-recruitment fitting in TMB
- mse many additional utilities and prerequisite for FLRef
- FLReffor advanced reference point estimation and producing advice plots

First, this needs some additional packages, first and foremost `devtools` and `TMB`

Note that it is recommended to install `TMB` from `type = source`, which will also be required for running `spict`. The installation of `TMB` required that `Rtools` is correctly installed (see above)!

```
install.packages("devtools")
install.packages("TMB", type = "source")
```

Furthermore, best to install the `ggplot2` packages and reshape2

```
install.packages("ggplot2")
install.packages("reshape2")
```

Load packages

```
library(TMB)
library(ggplot2)
library(reshape2)
```

Now install `FLSRTMB`

```
install.packages(c("FLSRTMB"), repo = "https://flr.r-universe.dev")
```

Next install `mse`

```
install.packages(c("mse"), repo = "https://flr.r-universe.dev")
```

Load packages

```
library(FLSRTMB)
library(mse)
```

Install `FLRef`

```
install.packages(c("FLRef"), repo = "https://flr.r-universe.dev")
```

Load package

```
library(FLRef)
```

2.4 Install packages for Stock Synthesis

First in install `r4ss`, which is designed to load and evaluate ss3 models in R. It is recommended to install the latest version of `r4ss` directly from github.

```
devtools::install_github("r4ss/r4ss")
```

In addition, it is suggest to install the `FLR` package `ss3om`, which is needed to produce the partial F plot by gear/fleet

```
install.packages(c("ss3om"), repo = "https://flr.r-universe.dev")
```

Load `r4ss`, `FLRef` and `ss3om`

```
library(r4ss)
library(FLRef)
library(ss3om)
```

3 SS3 assessment summary for European hake in GSAs 17-18 (HKE 17-18)

Authors: Angelini S., Bitetto I., Carbonara P., Casciaro L., Chiarini M., Casini M., Colella S., Ianelli J., Ikika Z., Isajlovic I., Kule M., Manfredi C., Marceta B., Palermino A., Panfili M., Santojanni A., Scarcella G., Spedicato M.T., Vrgoc N., Winker H.

The R code and data to reproduce this worked example is available on **WGSAD2024** SharePoint under `Ìnput data and Scripts/SummarySheet2024` **here**.

3.1 Step 1: Set up file paths and folders structure for loading and saving the SS3 model

Set up the file path to the folder where the SS3 model folder with run is located.

In this case, this R script is located is located in `SummarySheet2024/rmd` and SS3 model runs are located in the folder `SummarySheet2024/data/hke1718`. To jump out of the `/rmd` subfolder and into the `/data` subfolder the command `../data` is used.

```
mod.dir = file.path("../data/hke1718")
```

Define name of reference model folder with the SS3 model outputs

```
model = "HKE_17_18_2023_ss3"
```

Create `.rds` stock file name

```
stock.file = paste0(model, ".rds")
```

Load reference model

```r
ss3rep = SS_output(file.path(mod.dir, model))
```

Create an `rdata` folder in the assessment model subdirectory.

```r
rdata = file.path(mod.dir, "rdata")
dir.create(rdata, showWarnings = FALSE)
```

Save the model as `rdata` file

```r
saveRDS(ss3rep, file = file.path(rdata, stock.file))
```

…or load directly as `.rdata` if these had been saved already

```r
ss3rep = readRDS(file.path(rdata, stock.file))
```

3.2 Step 2: Convert SS3 to `FLStockR`

First, the `ssmvln()` from `FLRef` is used to generate the stock trajectories with uncertainty using a Monte-Carlo to generate a large number of iterations from multivariate log-normal approximation of the variance-covariance estimates.

Here, a new standard for presenting the of SSB_{cur} is introduced, by setting to assign SSB_{y+1} to SSB_y. The reasoning is that SS3 typically computes (but see MUT17-18) SSB at the start of the year for the 1st of January. Therefore, there is only a nominal difference by 1 day for making the assumption that SSB_{y+1} on first January is the same as SSB at the end of the previous year,i.e. 31th Dec. In practice, this SSB_{y+1} is directly informed updated data, whereas SSB_y is only indirectly informed by the new data in terms of the fit.

To implement this re-assigment of SSB_{y+1} to SSB_y to account for the so caused lag in SSB response, `FLRef` now provides the function `FLRef::blag()`

First, it is important to extend the assessment horizon to the reference year+1, i.e. 2024.

```r
years = 1970:2024
mvn = FLRef::ssmvln(ss3rep, addprj = F, Fref = "Btgt", verbose = F,
    years = years)
```

Apply `blag()`

```r
mvn = blag(mvn)
```

Next the `mvn` object can easily converted into the `FLStockR` object, but subsetting the assessment horizon back to the 2023 end year with `window()` needs to be applied

```r
stk = window(ss2FLStockR(mvn), end = 2023)
```

Although the `ssmvln()` function is designed to find the reference year (here 2023), it is always good practice to specify the end of the assessment horizon, while it does not matter if the first year is smaller than the star year. The option `Fref=Btgt`, and not `Fref=MSY` is chosen because the reference points were based on $B_{tgt} = B_{35}$, with a corresponding $F_{tgt} = F_{B35}$.

By default, the reference points for F_{tgt} and B_{tgt} are extracted together with `MSY`, `B0` and `R0`.

```r
stk@refpts
   An object of class "FLPar"
   params
       Ftgt     Btgt      MSY       B0       R0
   2.36e-01 9.04e+04 7.84e+03 2.58e+05 8.86e+05
   units:  NA
```

However, for the final advice plot only the agreed reference points should shown, which currently excludes `MSY`, `B0` and `R0`.

For a **new** assessment the reference points can be computed directly from the model outputs

```r
# only select Ftgt and Btgt
rps = stk@refpts[1:2]
# Set Bpa and Blim as agreed ratios to the Btgt
rps = rbind(rps[1:2], FLPar(Bpa = rps["Btgt"] * 0.5, Blim = rps["Btgt"] *
    0.25))

stk@refpts = rps
```

However, for **benchmark** updates the reference points need to be inputted manually as agreed as part of the benchmark process.

```r
Fb35 = 0.231
B35 = 104752.1
Bpa = 52376.1
Blim = 26188
```

```r
# only select Ftgt and Btgt
rps = stk@refpts[1:2]
rps["Ftgt"] = Fb35
rps["Btgt"] = B35
# Set Bpa and Blim as agreed ratios to the Btgt
rps = rbind(rps, FLPar(Bpa = Bpa, Blim = Blim))
# Attach to the stock
stk@refpts = rps
```

For evaluating the effect of applying `blag()`, a reference `FLStockR` is generated.

```r
mvn0 = FLRef::ssmvln(ss3rep, addprj = F, Fref = "Btgt", verbose = F,
    years = years)
stk0 = window(ss2FLStockR(mvn0), end = 2023)
stk0@refpts = rps
```

```r
plotAdvice(FLStocks(SSBy = stk0, `SSBy+1` = stk))
```

Next make a `FLStockR` with iterations generated MVLN Monte-Carlo (default nsim = 1000) to depict uncertainty

```r
# with uncertainty
stki = ss2FLStockR(mvn, output = "iters", thin = 1)
# assign benchmark reference points
stki@refpts = rps
```

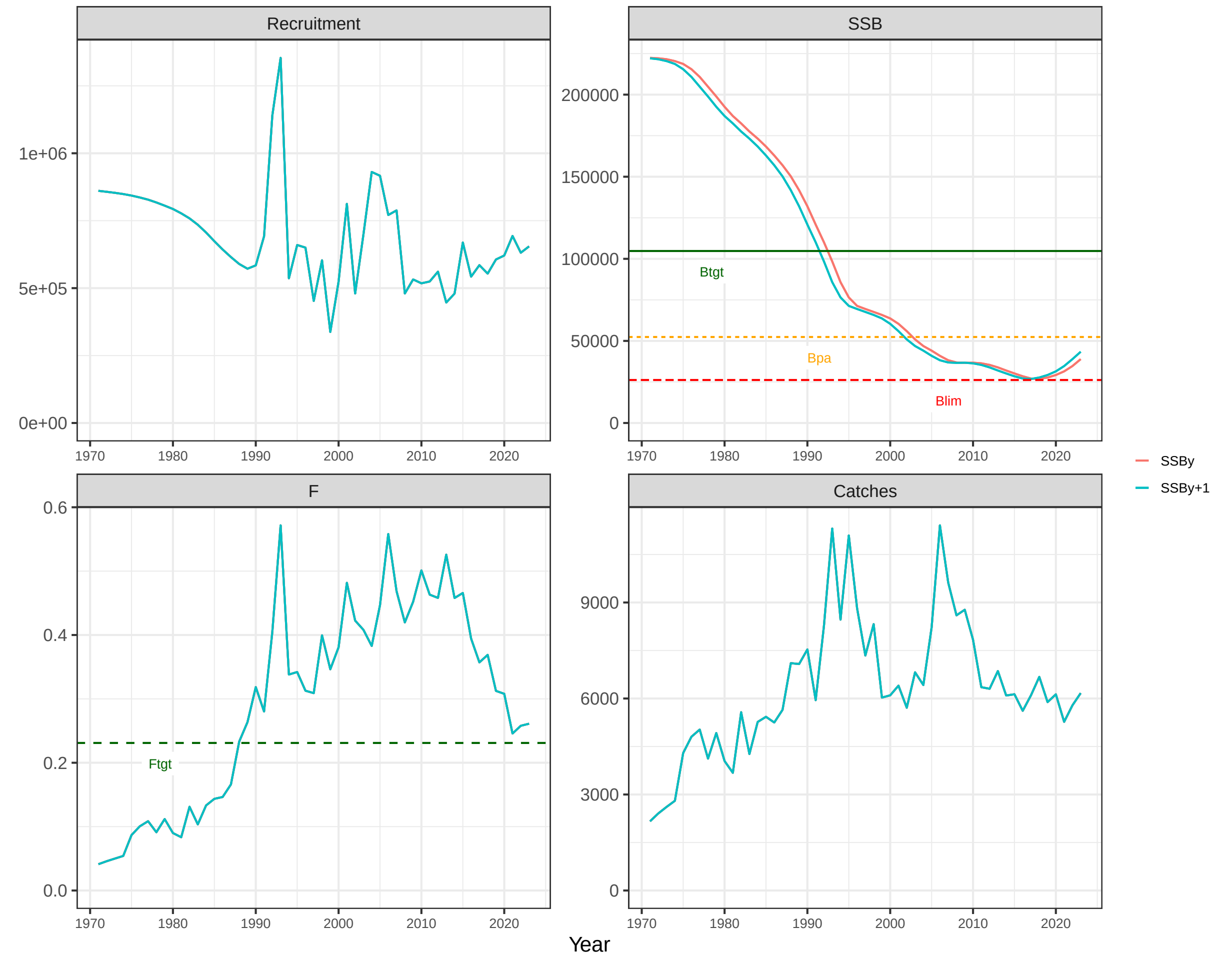

Figure 1: Comparison of estimated stock status trajectories with and without re-assigning SSB to the end of the year for the 2024 benchmark update of European hake in GSAs 17-18 through 2023

Check

```
plotAdvice(stki)
```

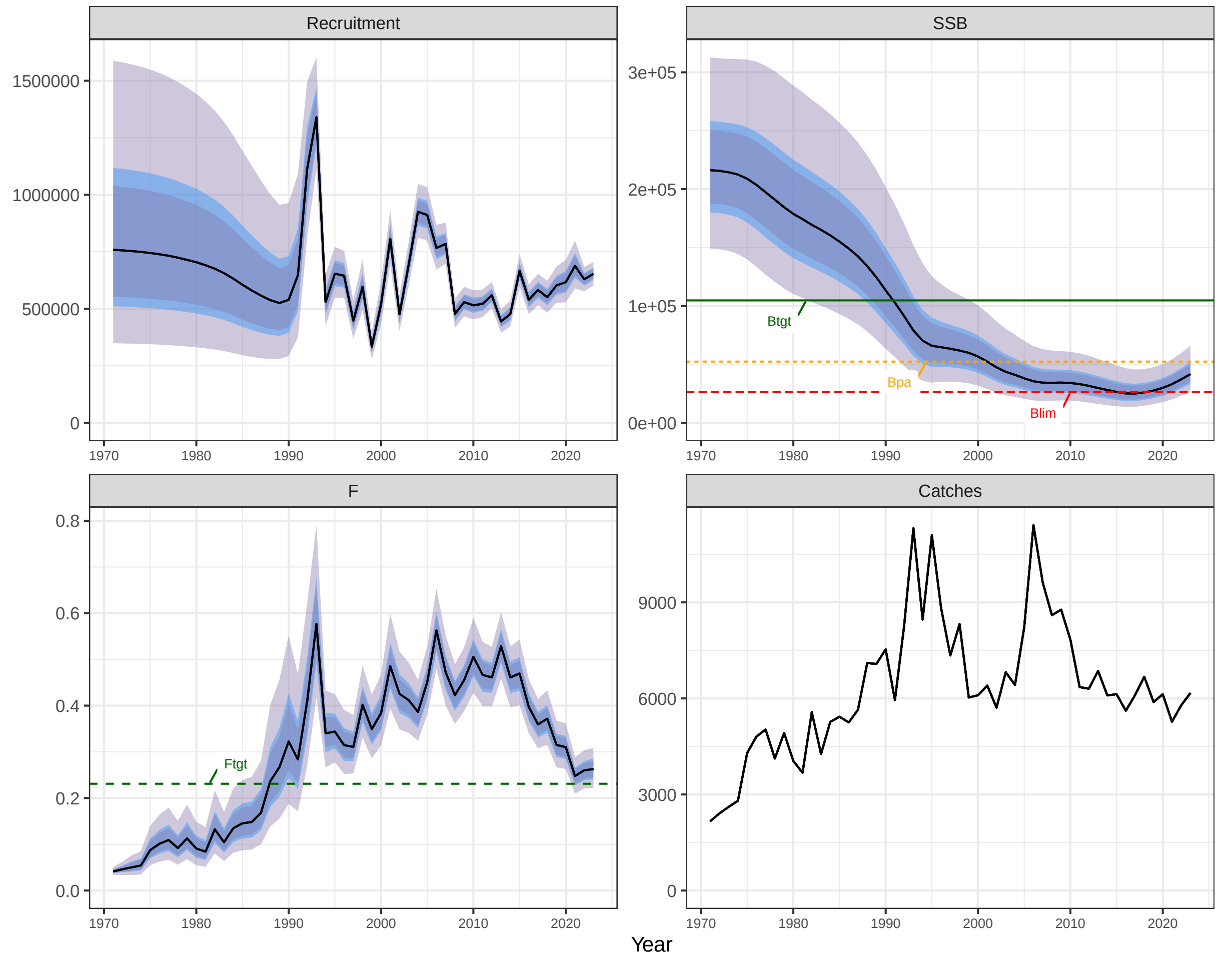

Figure 2: Uncertainty of estimated stock status trajectories with associated reference points for the 2024 benchmark update of European hake in GSAs 17-18 through 2023, with solid line depicting the median

3.3 Step 3: Make Advice plot of stock status indicators with uncertainty

The final advice plot seeks to provide a standard format for presenting stock status indicators that shows the exact maximum likelihood estimates from the model (`stk`) and depicts the uncertainty around those from the Monte-Carlo approach (stki).

The plotting code allows to specify the years shown along the x-axis by adjusting the option `break=seq(1970,2023,2)` depending on the length of the time series (here every 2 years)

```
# Name plot padv
padv = plotAdvice(FLStocks(CIs = stki, Fit = stk)) + scale_fill_manual(values = c("dodgerblue",
    "black")) + scale_color_manual(values = c(0, "black")) +
    theme(legend.position = "none") + scale_x_continuous(breaks = c(seq(1971,
    2023, 2))) + theme(axis.text.x = element_text(size = 8, angle = 90,
```

```
    vjust = 0.5))
padv
```

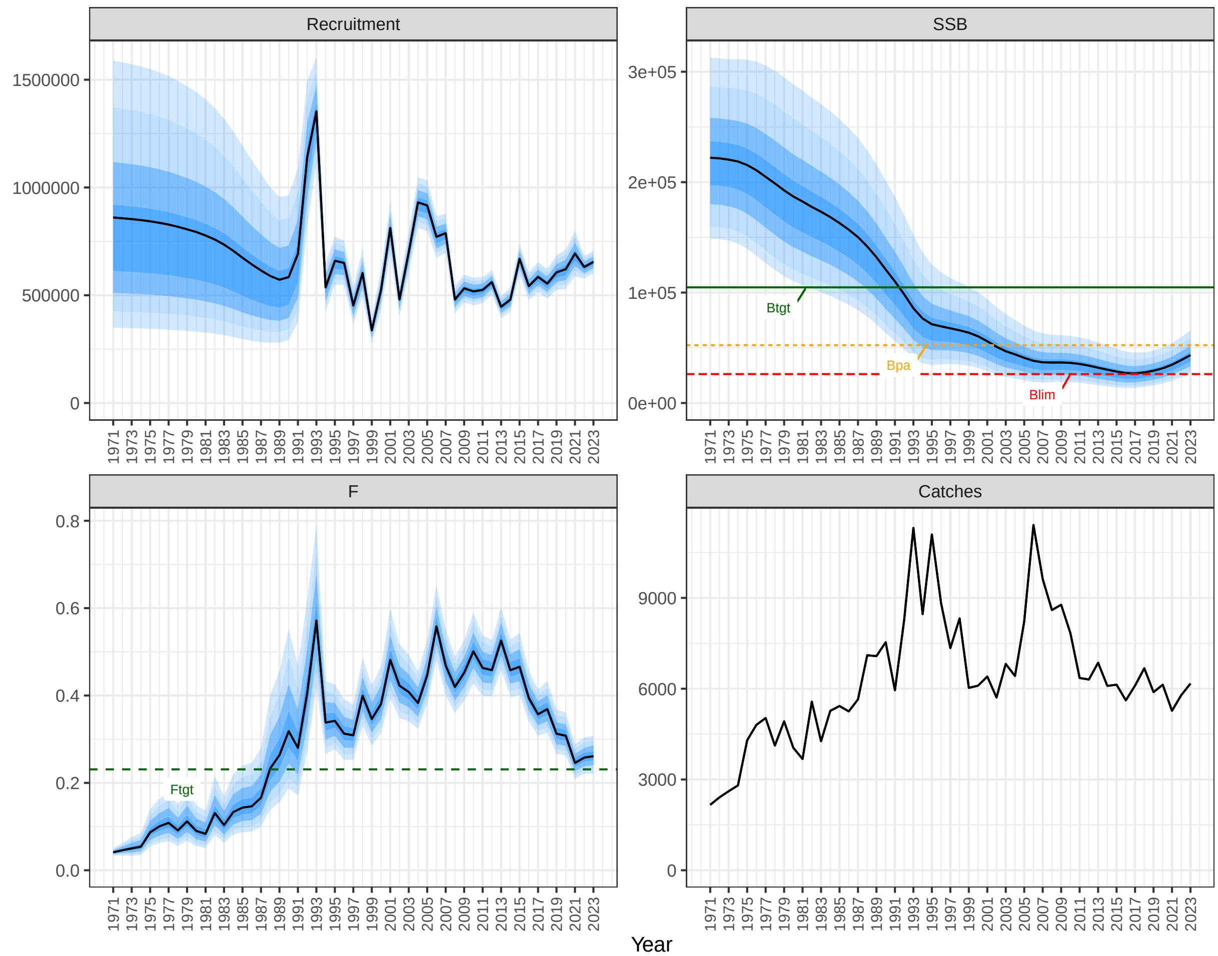

Figure 3: Estimated stock status trajectories with associated reference points for the 2024 benchmark update of European hake in GSAs 17-18

3.3.1 Save `FLStockR` objects in `.rds` format to `rdata`

It is adviced to specify additional information in the `FLStockR` object before saving it.

Label the `FLStockR` object properly

```
stk2 = stk
stk2@name = "HKE_17_18"
stk2@desc = "2023, SS3, GFCM"
```

Note that `stk@name` will be used through this script to label file names!

```
saveRDS(stk2, file = file.path(rdata, paste0(stk2@name, "_stk2.rds")))
saveRDS(stki, file = file.path(rdata, paste0(stk2@name, "_stki.rds")))
```

3.4 Step 5: Generate input data for the STAR template

First a folder **star** is created to save the outputs (Tables, Figures) for input to the STAR template and the new Summary Sheet.

```
star.dir = file.path(mod.dir, "star")
dir.create(star.dir, showWarnings = FALSE)
```

The output from `ssmvln` can now be directly converted in the STAR compatible format from SS3 model by

```
star = ss2stars(mvn)
```

Note that as long as the `blag()` adjustment is applied to the `mvn` object, it will also reflect in the `STAR` time series output (‘*timeseries*).

In Stock Synthesis, the reference points are estimated based on the most recent assessment updates and `ssmvln` only extracted F_{tgt} and B_{tgt} from the updated fit.

```
star$refpts
    RefPoint      Value
  1     Ftgt      0.236
  2     Btgt  90374.597
  3     Bthr         NA
  4     Blim         NA
  5     Fcur      0.261
  6     Bcur  43393.900
  7    B0.33  41310.448
  8    B0.66 135112.800
```

The function `updstar()` is therefore provided to update the STAR output with agreed set of reference points for assessment and benchmark updates, but also add limit reference points that are already adjusted in `stk@refpts`.

```
star = updstars(star, newrefpts = stk@refpts)
star$refpts
    RefPoint      Value
  1     Ftgt      0.231
  2     Btgt 104752.100
  3     Bthr  52376.100
  4     Blim  26188.000
  5     Fcur      0.261
  6     Bcur  43393.900
  7    B0.33  41310.448
  8    B0.66 135112.800
```

Note that the ratio B/B_{tgt} and F/F_{tgt} are also recalculated automatically.

```
head(star$timeseries[, c(8:10, 14:16)])
    Bratio_lower    Bratio Bratio_upper F_lower     F F_upper
  1    0.1450736 0.1787879    0.2206753   0.034 0.041   0.050
  2    0.1460952 0.1981991    0.2717576   0.034 0.046   0.062
  3    0.1430303 0.2165887    0.3310130   0.033 0.050   0.076
  4    0.1491602 0.2339567    0.3708571   0.035 0.054   0.085
  5    0.2349784 0.3759654    0.6027706   0.055 0.087   0.140
  6    0.2676710 0.4352208    0.7080000   0.062 0.100   0.164
```

The `$timeseries` and `refpts` can be saved as `.csv` files and to then copy this key information into the STAR template. The STAR template will also automatically produce Table of Advice.

```
write.csv(star$timeseries, file = file.path(star.dir, paste0(stk2@name,
    "_ts.star.csv")), row.names = F)
write.csv(star$refpts, file = file.path(star.dir, paste0(stk2@name,
    "_refpts.star.csv")), row.names = F)
```

3.5 Step 4: Fishing Opportunity scenarios based on forecasts

Here, the outcomes (`.rdata`) of the SS3 forecasts (see Supplement) are loaded

```
load(file = file.path(rdata, paste0(stk2@name, "_Ffwd.rdata")),
    verbose = T)
   Loading objects:
     Ffwd
```

The forecast scenarios should be defined with appropriate naming.

Here, fractions relative to F_{cur} were used and a addition a forecast with F_{tgt} was conducted

```
Ffrac = c(0.01, seq(0.5, 1.1, 0.1))

scenarios = c(paste0(100 * Ffrac, "% Fcur"), "Ftgt")
scenarios
   [1] "1% Fcur"   "50% Fcur"  "60% Fcur"  "70% Fcur"  "80% Fcur"  "90% Fcur"
   [7] "100% Fcur" "110% Fcur" "Ftgt"
```

The next step is looping through forecast scenarios to convert to a `FLStockR` format, compiled within a list of `FLStocks`

Note, here `blag()` needs to be applied again!

```
fstks = FLStocks(Map(function(x, y) {
    out = (FLRef::ssmvln(x, Fref = "Btgt", verbose = F, run = y,
        addprj = T))
    out = blag(out)  # adjust SSBy to SSBy+1
    out = window(ss2FLStockR(out, thin = 1), start = 2022)

    out@refpts = stk@refpts
    return(out)
}, x = Ffwd, y = as.list(scenarios)))
names(fstks) = scenarios
```

```r
pstks = FLStocks(c(FLStocks(Assessment = window(stk)), fstks))

# Name plot pffwd (pffwd for F-based and pcfwd for
# catch-based)
pffwd = plotAdvice(window(pstks, start = 1971)) + geom_vline(xintercept = c(2023,
    2026, 2030), linetype = 2) + scale_color_manual(values = c("black",
    ss3col(length(fstks)))) + scale_x_continuous(breaks = seq(1970,
    2035, 5)) + theme(axis.text.x = element_text(size = 8, angle = 90,
    vjust = 0.5))

pffwd
```

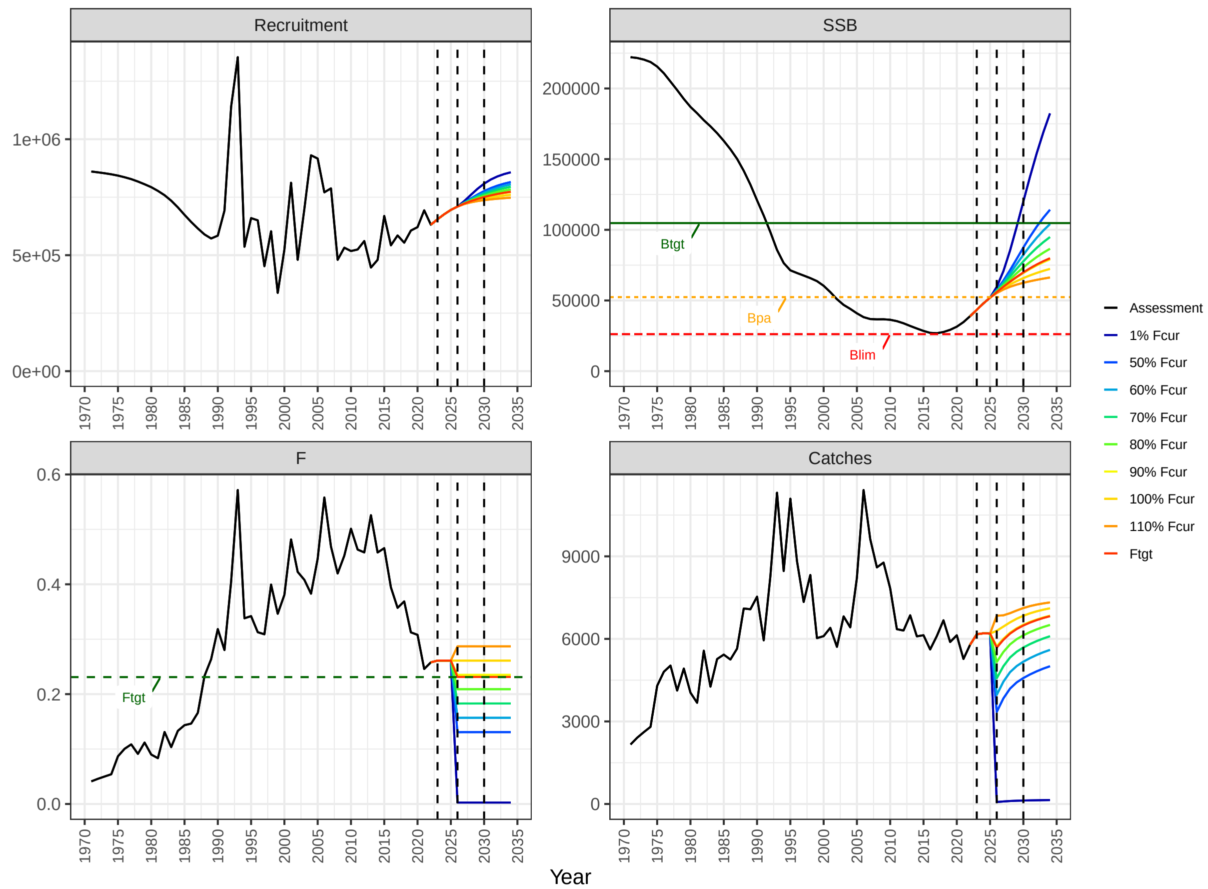

Figure 4: Trajectories for the basecase run and forecast scenarios relative F_{sq} and for F_{tgt}. The vertical dashed lines denote from left to right: reference year, implementation year for short-term forecasts and the end of the long-term management plan for the medium-term forecasts

Now the output summary from the short (2026) and medium (2030) can be prepared, which is suitable for input in STAR Template `Forecast` tab.

```
out = cbind(FLRef::fwd2stars(fstks, eval.yrs = 2026, rel = T),
    FLRef::fwd2stars(fstks, eval.yrs = 2030, rel = T)[, -1])

out[, -1] = round(out[, -1], 2)
out[, c(2, 7)] = round(out[, c(2, 7)], 1)
```

Make table for short-term forecasts through 2026

```
stf.tab = FLRef::fwd2stars(fstks, eval.yrs = 2026, rel = T)
```

Save `stf.tab` for input in the new Summary Sheet

```
write.csv(stf.tab, file = file.path(star.dir, paste0(stk2@name,
    "_stf.tab.csv")), row.names = F)
```

```
knitr::kable(stf.tab, "pipe", align = "lccccc", caption = "Summary of short-term forecast scenarios for
```

Table 1: Summary of short-term forecast scenarios for 2026

scenario	C2026	F2026/Ftgt	B2026/Btgt	B2026/	B2026/Bpa	B2026/Blim
1% Fcur	71.2	0.011	0.569	0.569	1.138	2.276
50% Fcur	3346.3	0.566	0.549	0.549	1.098	2.195
60% Fcur	3965.9	0.679	0.545	0.545	1.090	2.180
70% Fcur	4570.1	0.792	0.541	0.541	1.083	2.165
80% Fcur	5159.4	0.905	0.538	0.538	1.075	2.151
90% Fcur	5734.2	1.017	0.534	0.534	1.068	2.136
100% Fcur	6295.0	1.130	0.531	0.531	1.061	2.122
110% Fcur	6842.1	1.243	0.527	0.527	1.054	2.109
Ftgt	5683.3	1.001	0.534	0.534	1.069	2.137

Make table for medium-term forecasts through 2030

```
mtf.tab = FLRef::fwd2stars(fstks, eval.yrs = 2030, rel = T)
```

Save `stf.tab` for input in the new Summary Sheet

```
write.csv(mtf.tab, file = file.path(star.dir, paste0(stk2@name,
    "_mtf.tab.csv")), row.names = F)
```

```
knitr::kable(mtf.tab, "pipe", align = "lccccc", caption = "Summary of medium-term forecast scenarios fo
```

Table 2: Summary of medium-term forecast scenarios for 2030

scenario	C2030	F2030/Ftgt	B2030/Btgt	B2030/	B2030/Bpa	B2030/Blim
1% Fcur	124.8	0.011	1.146	1.146	2.292	4.584
50% Fcur	4577.8	0.566	0.835	0.835	1.670	3.340
60% Fcur	5171.9	0.679	0.786	0.786	1.573	3.145
70% Fcur	5686.6	0.792	0.742	0.742	1.483	2.966
80% Fcur	6131.1	0.905	0.701	0.701	1.401	2.803
90% Fcur	6513.6	1.017	0.663	0.663	1.326	2.652
100% Fcur	6841.2	1.130	0.628	0.628	1.257	2.514
110% Fcur	7120.6	1.242	0.597	0.597	1.193	2.387
Ftgt	6488.7	1.001	0.667	0.667	1.334	2.667

Save STAR forecast data table

```
write.csv(out, file = file.path(star.dir, paste0(stk2@name, "_fwd2star.csv")),
    row.names = F)
```

3.6 Step 6: Comparitive plot with previous assessment

Load the benchmark model 2024 for the previous reference year 2022.

Define name of reference model folder with the SS3 model outputs

```
mod.pre = "HKE_17_18_2022_ss3"
```

Create stock file name

```
stock.file.pre = paste0(mod.pre, ".rds")
```

Load reference model

```
rep.pre = SS_output(file.path(mod.dir, mod.pre))
```

Save to `rdata`

```
saveRDS(rep.pre, file = file.path(rdata, stock.file.pre))
```

Check if the model can be loaded

```
rep.pre = readRDS(file = file.path(rdata, stock.file.pre))
```

Convert to `FLStockR`

If `blag()` is applied to the update, apply it here, too.

```
mvn.pre = blag(FLRef::ssmvln(rep.pre, Fref = "Btgt", verbose = F))
stk.pre = ss2FLStockR(mvn.pre)
```

Assign benchmarks

```
stk.pre@refpts = stk@refpts
```

```
# Name plot pcomp
compstks = FLStocks(Ref2022 = stk.pre, Upd2023 = stk)
pcomp = plotAdvice(compstks) + facet_wrap(~qname, scales = "free_y") +
    theme_bw() + scale_x_continuous(breaks = seq(1971, 2023,
    2)) + theme(axis.text.x = element_text(size = 8, angle = 90,
    vjust = 0.5)) + scale_color_manual(values = c(ss3col(length(compstks))))

pcomp
```

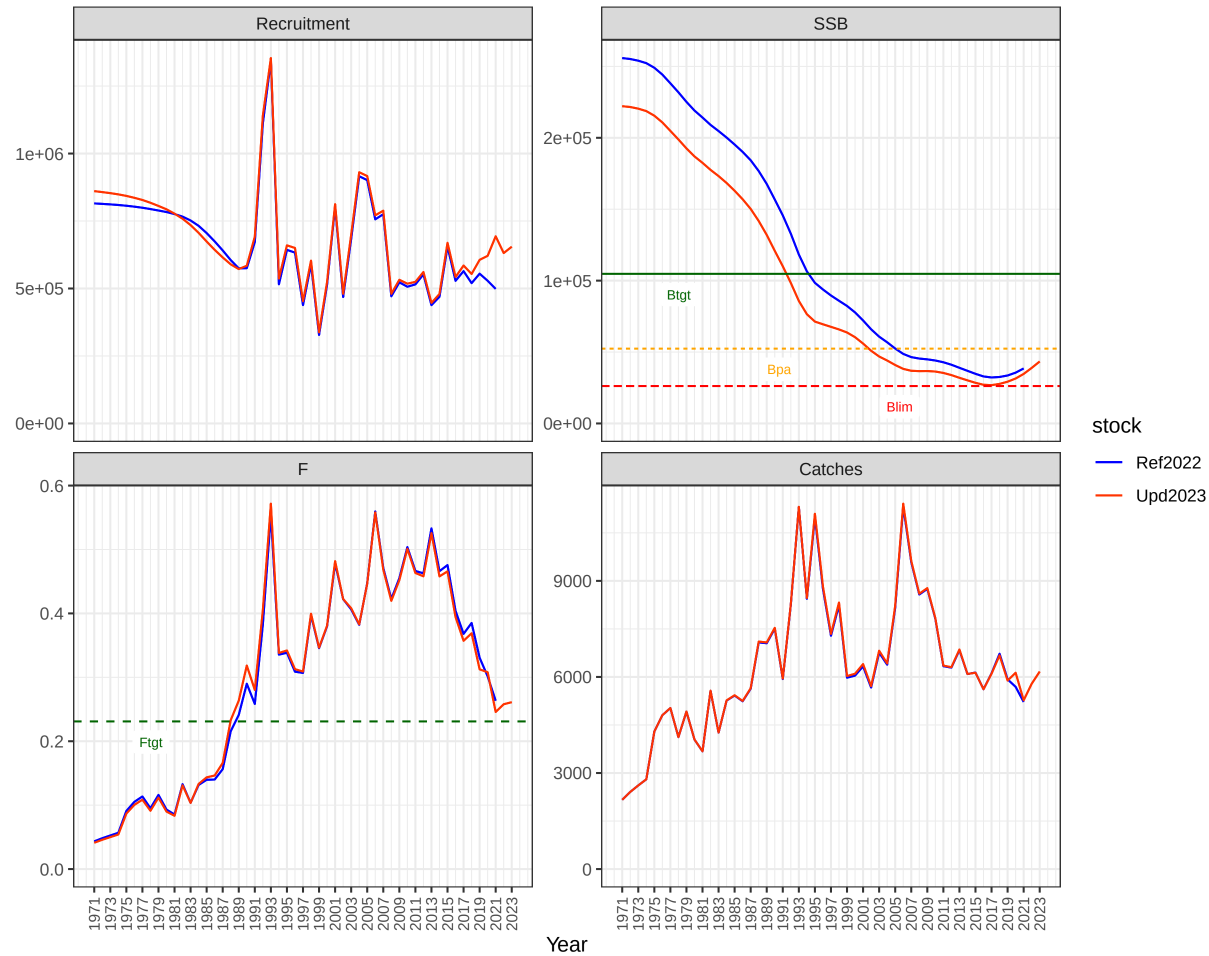

Figure 5: Comparison between the estimated stock status trajectories from the updated benchmark for the reference year 2023 and the previous benchmark model for the reference model 2022

3.7 Step 7: Make partial F plot

Load `ss3om`

```
library(ss3om)
```

Read SS3 run into `ss3om`

```
out = ss3om::readOutputss3(file.path(mod.dir, model))
```

To extract the F_a for all fleets in the model the following function is provided.

```
## Get partial Fs {{{
#' getFa()
#'
#' Helper function to extract F_a by fleet from ss3rep files
#' @param out FLStock from ss3om
#' @author Iago Mosqueira
#' @return *FLQuants*
#' @export
getFa <- function(out) {
    fatage <- data.table(out$fatage)[Era %in% c("TIME")]
    # DIMS
    fis <- unique(fatage$Fleet)

    res <- FLQuants(lapply(fis, function(x) {

        dat <- data.table::melt(fatage[Fleet == x, ], measure.vars = names(fatage)[-seq(7)],
            variable.name = "age", value.name = "data")[, .(age,
            Yr, Sex, Seas, Area, data)]

        setnames(dat, c("age", "year", "unit", "season", "area",
            "data"))

        as.FLQuant(dat)

    }))

    names(res) <- out$FleetNames[out$fleet_type == 1]

    return(res)
}
```

Next we can extract the F_a by fleet

```
fa = getFa(out)
```

In this case, these comprise all Gear and Country combinations, which are treated as seperate fleets in the model.

```
names(fa)
   [1] "ITA_OTB_17" "HRV_OTB_17" "HRV_LLS_17" "HRV_GNS_17" "ITA_OTB_18"
   [6] "ITA_LLS_18" "MNE_OTB_18" "ALB_OTB_18"
```

Here, these are combined by Gear type.

Re-organize

```r
fa = rev(fa[c(6, 3, 4, 8, 7, 2, 5, 1)])
names(fa)
   [1] "ITA_OTB_17" "ITA_OTB_18" "HRV_OTB_17" "MNE_OTB_18" "ALB_OTB_18"
   [6] "HRV_GNS_17" "HRV_LLS_17" "ITA_LLS_18"
```

Combine

```r
# OTB
fa[[1]] = iterSums(combine(fa[1:5]))
# GNS
fa[[2]] = fa$HRV_GNS_17
# LL
fa[[3]] = iterSums(combine(fa[7:8]))
fa = fa[c(1:3)]
names(fa) = c("OTB", "GNS", "LSS")
```

Now the Partial F plot by gear can be produced. Note that the partial impact by Gear is expressed as F_{apic}, which represent the F_a with maximum impact on any age class. By contrast, F_{bar} is not suitable for comparing different selectivity impact, as it is sensitive to choice of the age range. For, example, an F_{bar} for ages 1-6 would underestimate the partial impact of long-line, which impact more on age-6+ fish that are excluded from the average.

```r
dff = unitSums(fa)  # combine sex
fapic = FLQuants(lapply(fa, function(x) {
    apply(x, 2, max)
}))
# Name plot
pPF = ggplot(fapic) + geom_line(aes(year, data, color = qname),
    linewidth = 0.9) + theme_bw() + scale_x_continuous(breaks = seq(1971,
    2023, 2)) + theme(axis.text.x = element_text(size = 8, angle = 90,
    vjust = 0.5)) + scale_color_manual(values = c(ss3col(length(fapic)))) +
    ylab(expression(F[apic])) + xlab("Year") + theme(legend.title = element_blank())

pPF
```

3.8 Step 8: Save figures for the new Summary Sheet

In this final step all relevant figures are saved as `.jpg` for new Summary Sheet.

- Save Figure: Stock Status Indicators

```r
png(file.path(star.dir, paste0(stk2@name, "_advicePlot.jpg")),
    width = 9, height = 7, res = 300, units = "in")
padv
dev.off()
   pdf
     2
```

- Save Figure: Fishing Opportunities

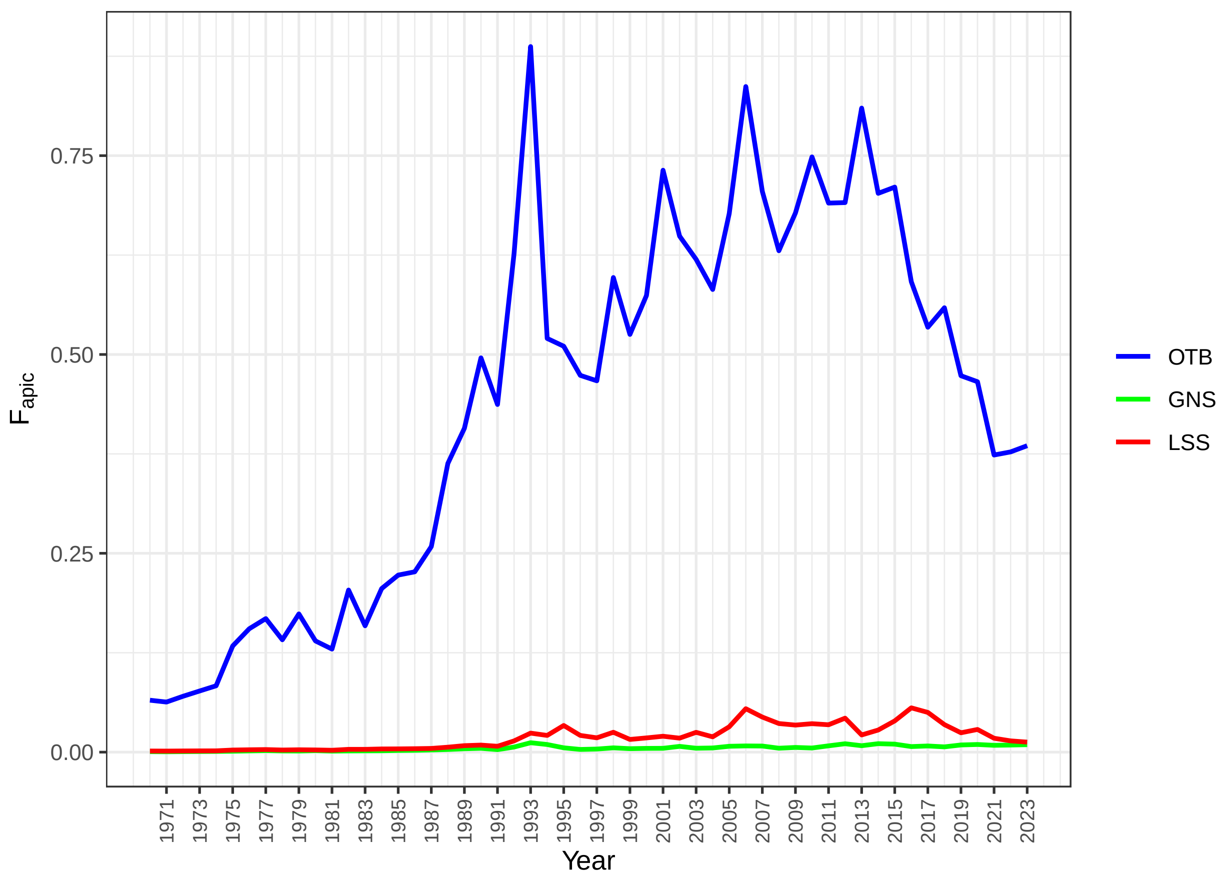

Figure 6: Evolution of fishing mortality (apical F) by gear type

```r
png(file.path(star.dir, paste0(stk2@name, "_FforecastPlot.jpg")),
    width = 9, height = 7, res = 300, units = "in")
pffwd
dev.off()
   pdf
     2
```

- Save Figure: Comparison with previous year assessment

```r
png(file.path(star.dir, paste0(stk2@name, "_compPlot.jpg")),
    width = 9, height = 7, res = 300, units = "in")
pcomp
dev.off()
   pdf
     2
```

- Save Figure: Partial F by fleet

```r
png(file.path(star.dir, paste0(stk2@name, "_FpartialPlot.jpg")),
    width = 9, height = 5.5, res = 300, units = "in")
pPF
dev.off()
   pdf
     2
```

4 Supplement

This supplement introduces some new features for the functions `SSdeltaMVLN` and `SSplotEnsemble` and then uses their utilities for illustrating a worked example for F-based forecasts for the multi-fleet models of European hake in the Adriatic Sea (GSAs 17-18).

Specific emphasis is given to F-based forecasting with SS3, which is based on the so-called apic F values (F_{apic}), whereas the choice of the reference F-basis and the associated reference points may differ from the F_{apic} scale. For instance, GFCM and ICES, F_{bar} (option 5) is the default option. It is therefore necessary to rescale F-basis to F_{apic} for generating forecasts that are consistent with, e.g., F_{tgt} or F_{cur}.

4.1 F-based forecasts for HKE 17-18

Forecasting with SS3 is based on so apic F values (F_{apic}), whereas the choice of the reference F-basis and the associated reference points may differ from the F_{apic} scale. For instance, GFCM and ICES, F_{bar} (option 5) is the default option. It is therefore necessary to rescale F-basis to F_{apic} for generating forecasts that are consistent with, e.g., F_{tgt} or F_{cur}.

4.1.1 Install and load packages

To quickly visualize the forecasts and make use of several helper functions, first install the latest versions of `ss3diags` (JABBAmodel branch)

```
devtools::install_github("JABBAmodel/ss3diags")
```

and load both `r4ss` and `ss3diags`

```
library(FLRef)
library(r4ss)
library(ss3diags)
```

4.1.2 Approach to F-based forecasting

F_{apic} is used for good reason in forecasts in order to account for multi- fleet selectivity. Comparing the partial impacts selectivity pattern requires setting the instantaneous rate of fishing mortaly F at comparable constant levels. For this purpose, it is important to consider that the definition of selectivity differs across regions (e.g. $Fbar$ or exploitation rate). With regards to temporal compatibility of partial fleet selectivity effects, F_{bar} has the undesirable property that its scale depends on the pre-specified age range across which F_a is averaged. For example, if F_{bar} is set to ages 2-4 to represent the dominant age classes under the current selectivity regime, but the goal is to evaluate the effect of selecting fish only at age-5, a common F_{bar} would result in disproportionately high F_a on ages 5+. This is because F_{bar} is computed for age ranges that are hardly selected for the definition $S_a = F_a/max(F_a)$. For this reason, it is more straight forward to use F_{apical} as the standardized quantity F quantify to account for partial impacts of fleet selectivity.

In the following, step-by-step guidelines are provided to setup an F_{apic}, so that it correctly corresponds to the F_{bar} baseline for F_{tgt} across multiple fleets and seasons.

4.1.3 Step 1: Basic setup

In this a case, a folder with the reference model run is created and the model outputs are loaded with `r4ss::SS_output`

next set up the folder where the SS3 model folder with run is located

```
mod.dir = file.path("../data/hke1718")
```

Define name of reference model folder with the SS3 model outputs

```
model = "hke_17_18_2023_ss3"
```

Create stock file name

```
stock.file = paste0(model, ".rdata")
```

Create an `rdata` folder in the assessment model subdirectory.

```
rdata = file.path(mod.dir, "rdata")

dir.create(rdata, showWarnings = FALSE)
```

Load and save assessment model

```
ss3rep = SS_output(file.path(mod.dir, model))
# save
save(ss3rep, file = file.path(rdata, stock.file))
```

..load directly as `.rdata` if it already exists (faster)

```
load(file.path(rdata, stock.file), verbose = T)
   Loading objects:
     ss3rep
```

To organise the forecast outputs, first create a subfolder `forecast`

```
forecast.dir = file.path(mod.dir, model, "forecast")

dir.create(forecast.dir, showWarnings = F)
```

A new helper function `SSnewrun` was added to `ss3diags` to easily create subfolders for the forecast scenarios. First a `Ftgt` reference folder is created for initial cross-checks

To this specify a new subfolder path, where to run the forecast

```
ftgtdir = file.path(forecast.dir, "Ftgt")
```

Create new F forecast model folder. Note that the data and control file and ss.exe names need to be specified if these diverge from the defaults `data.ss`, `control.ss` and `ss3.exe`

```
dat = "hke1718_dat.ss"
ctl = "hke1718_ctl.ss"
ss.exe = "ss3.exe"

SSnewrun(model = file.path(mod.dir, model), dat = dat, ctl = ctl,
    newdir = ftgtdir, ss.exe = "ss3.exe")
```

Now the forecast file can be read be read with `r4ss`

```
fc <- SS_readforecast(file.path(ftgtdir, "forecast.ss"), verbose = F)
```

4.1.4 Step 2: Initial F exploitation calculations for Fapic forecast

Extract the `$exploitation` output from the report file

```
Fexp = ss3rep$exploitation
```

Importantly, the `annual_F` are scaled to the F-basis (here F_{bar}), whereas fleet specific F values are always given as F_{apic}

Next compute the combined F_{apic} generically across fleets

```
Fexp$Fapic = apply(as.matrix(ss3rep$exploitation[, -c(1:6)]),
    1, sum, na.rm = T)
```

and aggregate across seasons, by taking the `mean` and not the `sum`.

```
Fapic = aggregate(Fapic ~ Yr, Fexp, mean)
```

Next compute the corresponding annual F_{bar} values from the `annual_F`

```
Fbar = aggregate(annual_F ~ Yr, Fexp, mean)
```

To work out exact ratio between F_{apic} and F_{bar} so that it is consistent with the benchmark calculations with ss3, it is necessary to extract the reference years for selectivity from the `forecast.ss` file.

The information required for the average selectivity conditions can be found in the `forecast.ss` file under `$Bmark_years`. The third and fourth position define the time horizon for the average selectivity across fleet, a value of -999 (here) indicates that the whole time series is use, but more commonly averages are taken, e.g. over the last 3 years, which can be specified as -2 0 or 2019 2021. The following code attempts to compute this generically.

```
endyr = ss3rep$endyr
if (fc$Bmark_years[3] < -90) {
    nfc = length(min(ss3rep$exploitation$Yr + 1):endyr)  # excluded init year
} else {
    # if specified (e.g. -2, 0)
    nfc = fc$Bmark_years[4] - fc$Bmark_years[3] + 1
}
# Benchmark reference years
bmyrs = (endyr - nfc + 1):endyr
```

```
Fratio = mean(Fapic$Fapic[Fapic$Yr %in% max(bmyrs)]/Fbar$annual_F[Fbar$Yr %in%
    max(bmyrs)])
Fratio
   [1] 1.809787
```

`Fratio` defines the ratio of F_{apic} to F_{bar} for the reference period

For a **new** asssessment, the F_{tgt} reference point, here defined as $F_{B_{35}}$ can be directed extracted from the model. For $F_{B_{35}}$ the `annF_Btgt` is extracted.

```
Fref = c("annF_Btgt", "annF_MSY", "annF_SPR")[1]
Ftgt = ss3rep$derived_quants$Value[ss3rep$derived_quants$Label ==
    Fref]
```

However, for benchmark updates (excluding model ensemble) the benchmarked reference point should be used.

```
Fb35 = 0.231
B35 = 104752.098
Bpa = 52376.049
Blim = 26188.025
```

Set F_{tgt} to benchmark

```
Ftgt = Fb35
```

This value is given as F_{bar} and therefore needs to be transformed to F_{apic}

```
Ftgt.apic = Ftgt * Fratio
Ftgt  # Fbar
   [1] 0.231
Ftgt.apic
   [1] 0.4180607
```

4.1.5 Step 3: Setting up the manual F forecast input structure

First, do some basic house keeping for the model structure

```
nseas = length(unique(ss3rep$exploitation$Seas))  # number of seasons
fleets = unique(ss3rep$fatage$Fleet)  # fleets
nfleets = length(fleets)  # number of fleet
```

Next, the mean Fapic by fleet and season is calculated

```
# subset to benchmark years for selectivity
fexp = ss3rep$exploitation[ss3rep$exploitation$Yr %in% bmyrs,
    ]
fexp = cbind(fexp[, 1:2], fexp[, -c(1:5)])[, -3]  #><>  single fleet trick
# flip
fexp = reshape2::melt(fexp, id.vars = c("Yr", "Seas"), variable.name = "Fleet",
    value.name = "Fapic")
head(fexp)
       Yr Seas      Fleet     Fapic
   1 2021    1 ITA_OTB_17 0.1574590
   2 2022    1 ITA_OTB_17 0.1836090
   3 2023    1 ITA_OTB_17 0.1896770
   4 2021    1 HRV_OTB_17 0.0933417
   5 2022    1 HRV_OTB_17 0.1006600
   6 2023    1 HRV_OTB_17 0.0869246
```

The forecast file requires Fleet IDs not names. In the next step these are extracted and fleet names are converted in to Fleet IDs

```
fleet = data.frame(Fleet = ss3rep$FleetNames, ID = ss3rep$fleet_ID)
fexp$Fleet = fleet[match(fexp$Fleet, fleet$Fleet), 2]
```

Then, the relative proportions of F_{apic} by fleet and season can be computed

```
Fap = aggregate(Fapic ~ Seas + Fleet, fexp, mean)
Fap$prop = Fap$Fapic/sum(Fap$Fapic) * nseas
Fap
     Seas Fleet       Fapic        prop
   1    1     1 0.176915000 0.378317573
   2    1     2 0.093642100 0.200245610
   3    1     3 0.010491673 0.022435545
   4    1     4 0.013842067 0.029600074
   5    1     5 0.106649067 0.228059894
   6    1     6 0.006688280 0.014302314
   7    1     7 0.002379633 0.005088642
   8    1     8 0.057028400 0.121950348
```

In the next step, status quo F_{sq} for forecasting over the intermediate year(s) is defined. This can be relatively easily changed to intermediate catch years. Here, the F_{sq} is taken as F_{2022} to account for the systematically decreasing trend, and the intermediate years are set to 2, account for 1 data lag year and additional management lag year. Note that cases where F flactutes the average F over the most recent three maybe considered as F_{cur} (`status quo`) as more representative for the intermediate years.

```
# F status q
nfsq = 1
nint = 2
```

Compute the F_{sq} as F_{apic} vector by season and fleet

```
fsq = ss3rep$exploitation[ss3rep$exploitation$Yr %in% ((endyr -
    nfsq + 1):endyr), ]
fsq = cbind(fsq[, 1:2], fsq[, -c(1:5)])[, -3]  #><>  single fleet trick
fsq = reshape2::melt(fsq, id.vars = c("Yr", "Seas"), variable.name = "Fleet",
    value.name = "Fapic")
Fsq = aggregate(Fapic ~ Seas + Fleet, fsq, mean)
```

Now, the forecast horizon can be defined in the loaded `starter.ss` object `fc`. Given the low current stock level, a slightly longer forecast horizon through 2035 is chosen (2 intermediate years + 10 implementation years) to also provide a first idea of the predicted recovery time to B_{tgt}. Summary statistics for short-term forecasts (3 years) and medium-term forecast until 2030 can then me extracted from these longer term forecasts,

```
fc$Nforecastyrs = 12
nfyrs = fc$Nforecastyrs
fyrs = endyr + c(1:nfyrs)
```

The F-vector that is passed on the forecast file comprises the season/fleet structure replicates for `ninit` for F_{sq} and the forecast years under F_{tgt} that is scaled to F_{apic} by the `Fratio` and portioned by fleets.

```
fvec = c(rep(Fsq$Fapic, nint), rep(Ftgt * Fratio * Fap$prop,
    nfyrs - nint))
```

Given the fleet, season, intermediate year and forecast years structures, the forecast table for the `forecast.ss` file can finally be constructed.

```
fc$ForeCatch = data.frame(Year = rep(fyrs, each = nseas * nfleets),
    Seas = 1:nseas, Fleet = rep(fleets, each = nseas), `Catch or F` = fvec,
    Basis = 99)
head(fc$ForeCatch, 9)
     Year Seas Fleet Catch.or.F Basis
   1 2024    1     1 0.18967700    99
   2 2024    1     2 0.08692460    99
   3 2024    1     3 0.00879872    99
   4 2024    1     4 0.01449440    99
   5 2024    1     5 0.10776300    99
   6 2024    1     6 0.00587590    99
   7 2024    1     7 0.00232619    99
   8 2024    1     8 0.05670080    99
   9 2025    1     1 0.18967700    99
```

Note that the `Basis` 99 specifies that Fs are inputted (2 would be Catch). Finally, the forecast options need to be adjusted for manual input

```
fc$eof = TRUE
fc$InputBasis = -1
```

and then the modified `starter.ss` file can be saved

```
SS_writeforecast(fc, file = file.path(ftgtdir, "forecast.ss"),
    overwrite = T, verbose = F)
```

4.1.6 Step 4: Running Ftgt forecasts with checks

In principle, the `Ftgt` can serve as a reference and the model does not have to be run if the goal is set up a number forecasts relative to F_{tgt}.

However, for illustration, the `Ftgt` forecast is run to check that the F_{apic} will produce F_{bar} estimates that are consistent with F_{tgt}.

To run

```
r4ss::run(ftgtdir, skipfinished = F, show_in_console = T, exe = "ss3.exe")
```

After the run is finished (here under 2.5 min) the output can be loaded again.

```
ftgtrep = SS_output(ftgtdir)
# safe as rdata
save(ftgtrep, file = file.path(rdata, "HKE_17_18_fwd.ftgt.rdata"))
```

For a quick check the `plotAdvice()` from `FLRef` can be used, but first the forecast needs to be converted into a "simplified" `FLStock` object, using the function `ssmvln`.

```r
mvn = FLRef::ssmvln(ftgtrep, Fref = "Btgt", addprj = T, verbose = F)
stkf = ss2FLStockR(mvn)
```

Again, for **new** assessments the reference points can be computed directly from the model outputs

```r
# only select Ftgt and Btgt
rps = stkf@refpts[1:2]
# Set Bpa and Blim as agreed ratios to the Btgt
rps = rbind(rps[1:2], FLPar(Bpa = rps["Btgt"] * 0.5, Blim = rps["Btgt"] *
    0.25))

stkf@refpts = rps
```

But reference points need to be inputted manually for benchmark assessments

```r
# only select Ftgt and Btgt
rps = stkf@refpts[1:2]
rps["Ftgt"] = Fb35
rps["Btgt"] = B35
# Set Bpa and Blim as agreed ratios to the Btgt
rps = rbind(rps, FLPar(Bpa = Bpa, Blim = Blim))
stkf@refpts = rps
```

It can be readily seen that the F_{apic} based F_{tgt} forecast corresponds indeed to the F_{tgt} estimate on F_{bar} scale.

```r
plotAdvice(stkf) + geom_vline(xintercept = 2022.5, linetype = 2)
```

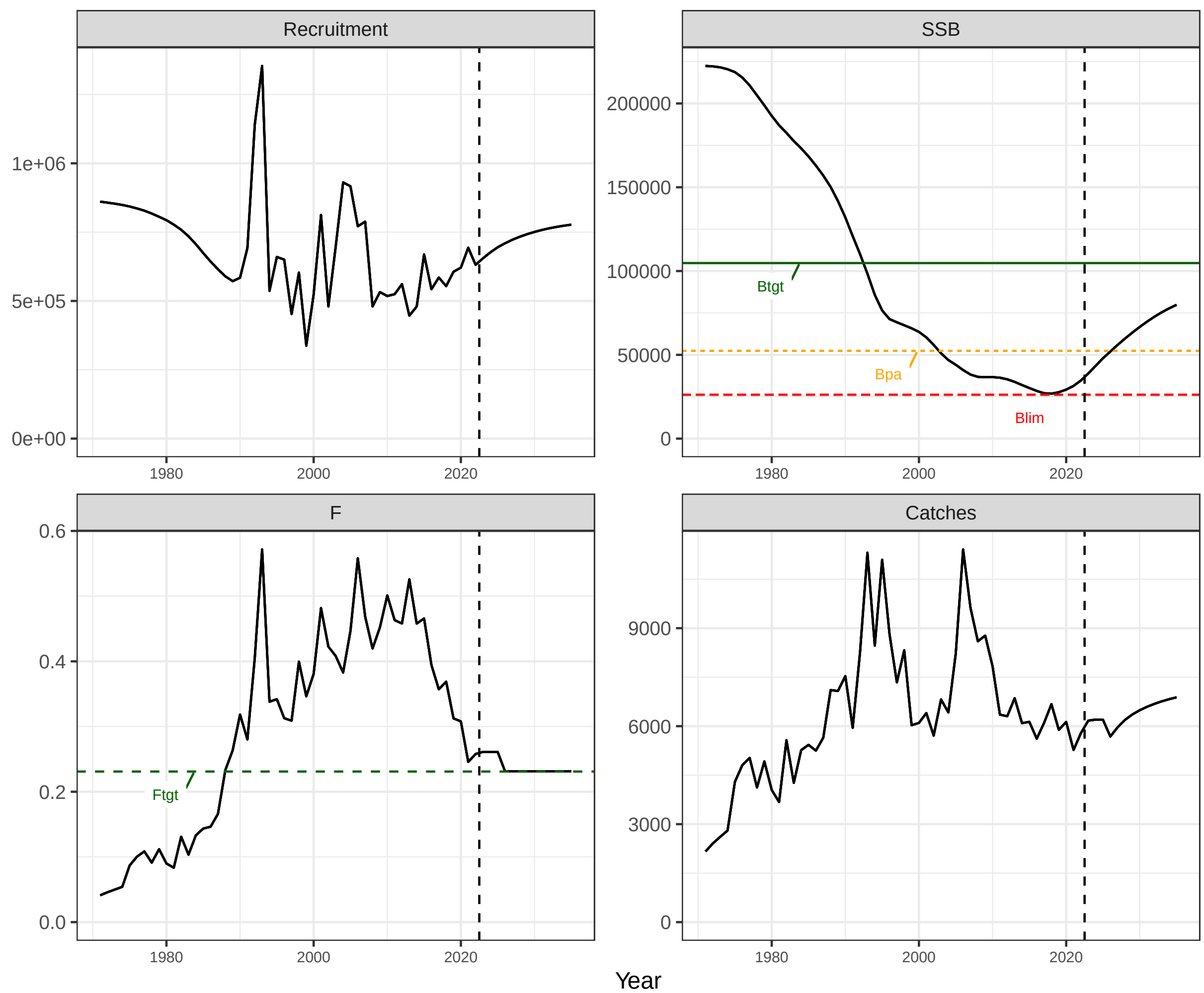

Figure 7: Stock trajectories for basecase run and a F_{tgt} forecast, relative to reference points

4.1.7 Step 5: Looping through forecast scenarios

Set up ratios relative to F_{sq} in this case

```
Ffrac = c(0.01, seq(0.5, 1.1, 0.1))
```

Specify forecast folders

```
fcdirs = file.path(forecast.dir, paste0("Fsq", Ffrac))
```

Loop through the process of modifying the `forecast.ss` file iteratively. The `Ffrac` is applied to apportioned F_{tgt} vector.

```
# TODO add optopm for parallel computing
for (i in 1:length(Ffrac)) {
    # create model folder
    SSnewrun(model = ftgtdir, dat = dat, ctl = ctl, newdir = fcdirs[i],
        ss.exe = "ss3.exe")
    # Read Forecast file
    fc <- SS_readforecast(file.path(fcdirs[i], "forecast.ss"))
    # Apply Ffrac Create F forecast vector (generic) Change
    # to Fsq
    fvec = c(rep(Fsq$Fapic, nint), rep(Fsq$Fapic * Ffrac[i],
        nfyrs - nint))
    # Creat F forecast table in forecast.ss
    fc$ForeCatch = data.frame(Year = rep(fyrs, each = nseas *
        nfleets), Seas = 1:nseas, Fleet = rep(fleets, each = nseas),
        `Catch or F` = fvec, Basis = 99)
    SS_writeforecast(fc, file = file.path(fcdirs[i], "forecast.ss"),
        overwrite = T)
    r4ss::run(fcdirs[i], skipfinished = F, show_in_console = TRUE,
        exe = ss.exe)
}
```

Load all runs in one go with `SSgetoutput`, inclduding the Ftgt run

```
Ffwd = SSgetoutput(dirvec = c(fcdirs, ftgtdir))
save(Ffwd, file = file.path(rdata, "HKE_17_18_Ffwd.rdata"))
```

Check that these can be loaded

```
load(file = file.path(rdata, "HKE_17_18_Ffwd.rdata"), verbose = T)
   Loading objects:
     Ffwd
```

Quick check and plot `FLRef`

```
fstks = FLStocks(Map(function(x, y) {
    out = FLRef::ssmvln(x, Fref = "Btgt", verbose = F, run = y,
        addprj = T)
    out = blag(out)  #apply blag
```

```r
    out = ss2FLStockR(out)
    out@refpts = rps  # replace with benchmarks
    return(out)
}, x = Ffwd, y = as.list(c(paste0("Fcur", Ffrac), "Ftgt"))))
names(fstks) = c(paste0("Fcur", Ffrac), "Ftgt")
```

```r
plotAdvice(window(fstks, start = 2015)) + geom_vline(xintercept = c(2023,
    2026), linetype = 28)
```

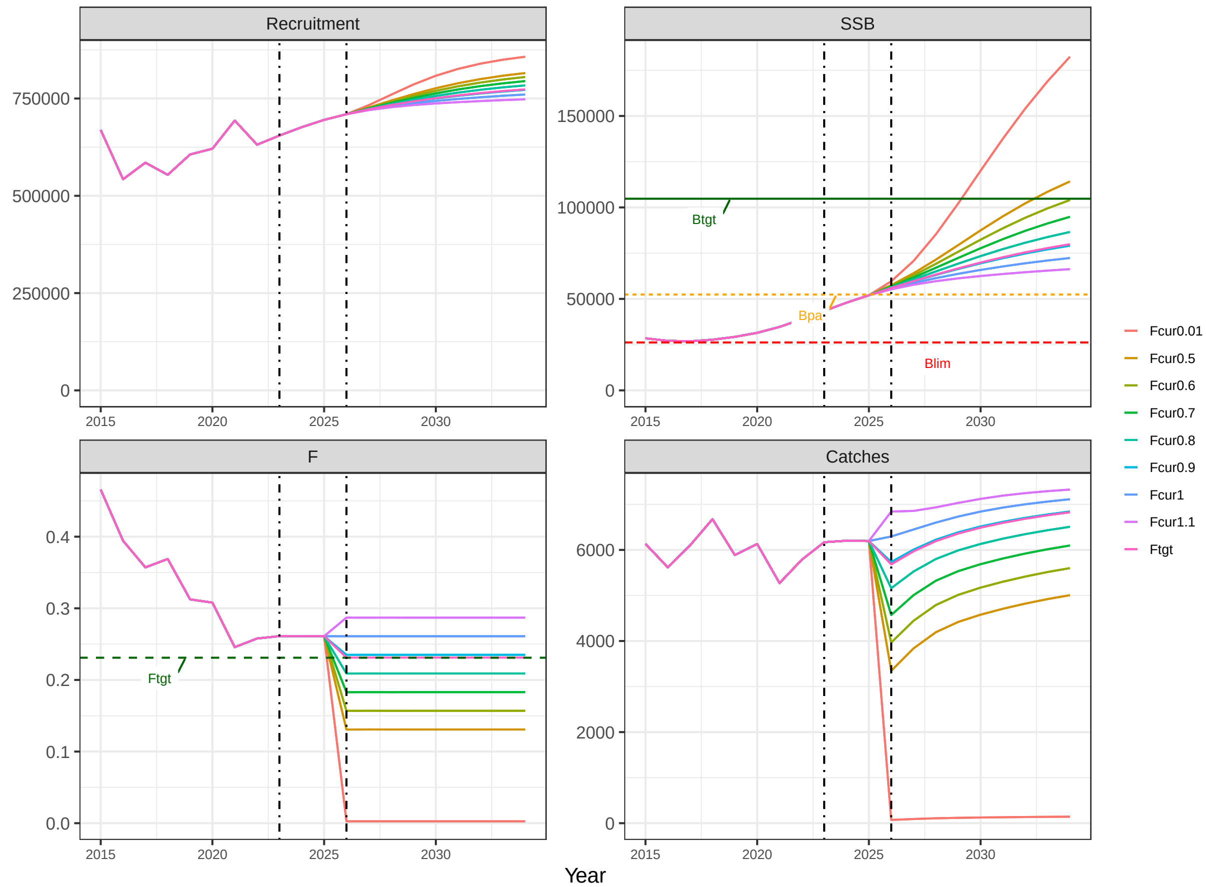

Figure 8: Trajectories for the basecase run and forecast scenarios relative F_{sq} and for F_{tgt}