Short-Cut MSE approach for robustness tests of harvest control rules in sex-structured models

Henning Winker (GFCM)

19 February, 2025

Contents

Load packages

```
# Load
library(ggplot2)
library(FLCore)
library(ggplotFL)
library(mse)
library(FLRef)
library(ggpubr)
library(mseviz)
```

1 Short-Cut MSE for Harvest Control Rule (HCR)

In contrast to a full Management Strategy Evaluation (MSE) simulation design (Punt et al. 2017), the MSE 'shortcut' approach, omits the step of the annual updating of the estimation model (assessment) in the feedback control. Instead, it emulates an annual update of the benchmark assessment model by passing outcomes (SSB and F) from the 'true' age-structured dynamics from the operating model (OM) with assessment error to the harvest control rule (HCR) and catch implementation system.

The HCRs were implemented using a simulated feedback control loop between the implementation system and the operating model, where the implementation system translates the emulated assessment outcome via the HRC into the Total Allowable Catch (TAC) advice. The feedback control loop between the implementation system and the OM allows accounting for the lag between the last of year data used in the assessment and the implementation year of catch advice (C_{adv}).

For blackspot seabream, the implementation system of the harvest control rule is based on the assumption that advice is given for year $y + 1$ based on an assessment completed in year y, which is fitted to data up until last data year $y - 1$. Therefore implementation of the derived C_{adv} through HCR requires projection of the stock dynamics by way of a short-term forecast. To do this, numbers-at-age were projected through the year of assessment. Status quo recruitment, M_a, w_a and mat_a were set as the mean of the last 3 years. A projection based on a fixed fishing mortality-at-age to the last year ($y-1$) in the assessment is then made through to the implementation year ($y + 1$).

The limitations of the MSE short-cut approach are that it cannot fully account for uncertainties resulting from imperfect sampling of the full age-structure (e.g. poorly sampled recruits), observation error, misspecified model assumptions and selectivity. On the other hand, the short-cut MSE approach is straight-forward to implement (`FLR`) and reduced complexity and computation time when the focus is predominantly optimizing HCRs for setting quotas on the premises that a benchmark assessment form the basis for the advice.

Here, the MSE short-cut approach is implemented using the tools available in the Fisheries Library for R (FLR; Kell et al., 2007; https://flr-project.org/)

1.1 Glossary

The following glossary summarizes key HCR parameters and associated target and limit reference points that are considered for tuning the candidate HCRs to optimise the trade-offs between maximising fishing opportunity and risk:

- F_{MSY}: target reference point for fishing mortality at Fmsy (or its proxy), (e.g. F_{B35})
- B_{MSY}: the average biomass around which the biomass fluctuated when fishing at F_{MSY} or its proxy (e.g. B_{35})
- B_{lim}: a deterministic biomass limit reference point below which a stock is considered to have reduced reproductive capacity. Here B_{lim} was set to $0.25B_{tgt}$
- B_{pa}: a precautionary biomass reference point set with high probability that biomass is above B_{lim}, which acts as a safety margin below which the risk of reduced reproductive capacity is increasing. When the biomass is estimated to be above Bpa, the stock is considered to be within safe biological limits in terms of its reproductive capacity.
- C_{adv}: advised catch as output of the management procedure
- $B_{trigger}$: biomass trigger point of the HCR, specified as change point of biomass below which fishing mortality reduced relative to Ftgt. Btrigger is typically specified as ratio to B_{MSY}.

2 Build FLStock

SS3 outputs are loaded with the `readFLSss3()` into an `FLStock` object. The folder that contains the model outputs has to be specified.

In the following, the area outside is evaluated first.

```r
run = "mon.27.8c9a"
stk = window(ss3om::readFLSss3(run, wtatage = TRUE))
   Warning: replacing previous import 'FLCore::profile' by 'r4ss::profile' when
   loading 'ss3om'
# Fill NAs
stk@m.spwn[] = 0
stk@harvest.spwn[] = 0
sr = ss3om::readFLSRss3(run)
stk@name = "mon.27.8c9a"
stk@desc = "2024, ICES, SS3"
out = ss3om::readOutputss3(run)

dir.create("rdata", showWarnings = F)
```

Check that the fbar range is consistent with `ss.starter` input.

```r
range(stk)
        min       max plusgroup   minyear   maxyear   minfbar   maxfbar
          0        30        30      1980      2023         2        15
```

```r
plot(stk, metrics = list(SSB = function(x) unitSums(ssb(x)[,
    , , 1]), F = function(x) fbar(x), Catch = function(x) catch(x),
    Rec = function(x) unitSums(rec(x)))) + theme_bw() +
    ylab("F") + xlab("Year") + facet_wrap(~qname, scales = "free_y")
```

Convert to combined sex trajectories to test `stockMedians()` adjustment

```r
stkm = stockMedians(stk)
plot(FLStocks(Median = stkm)) + theme_bw() + ylab("F") +
    xlab("Year")
```

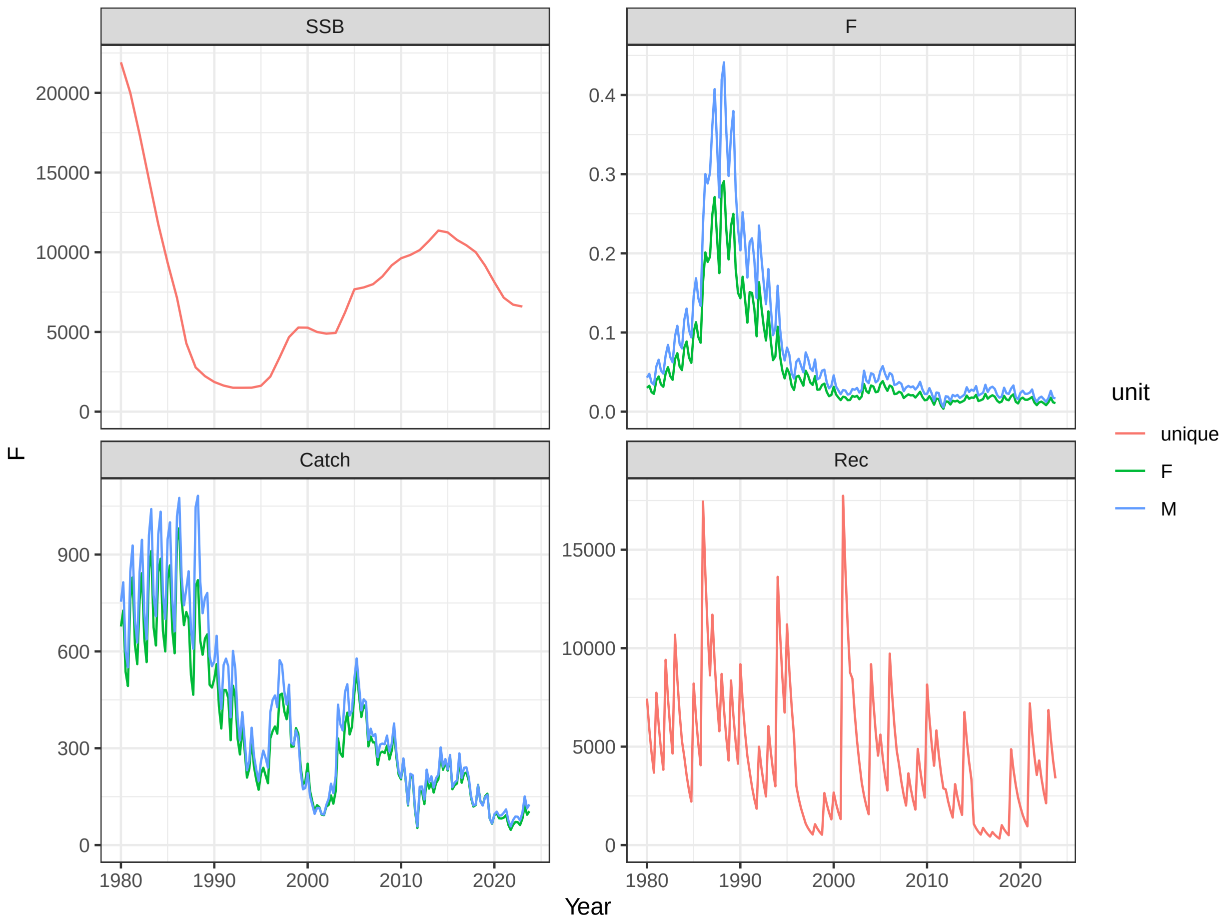

Figure 1: Seasonal stock trajectories

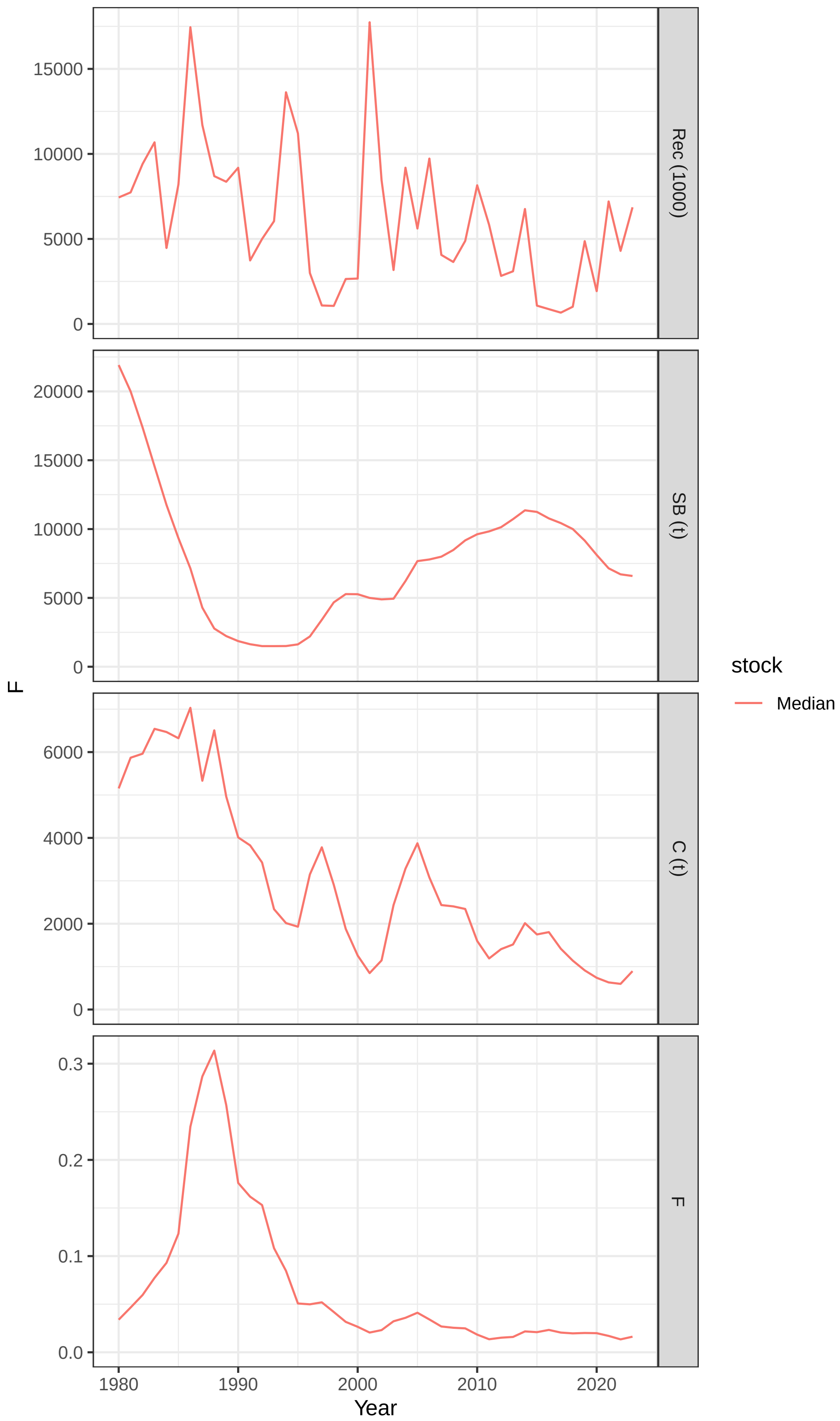

Figure 2: Seasonal stock trajectories

```r
ssbQ = 1
recQ = 1

flqs = FLQuants(SSB = unitSums(ssb(stk)[, , , ssbQ]), Vuln.bio = unitSums(vb(stk)),
    Tot.bio = unitSums(stock(stk)), N = unitSums(quantSums(stock.n(stk))))
plot(flqs) + theme_bw() + xlab("Year") + facet_wrap(~qname,
    scales = "free_y")
```

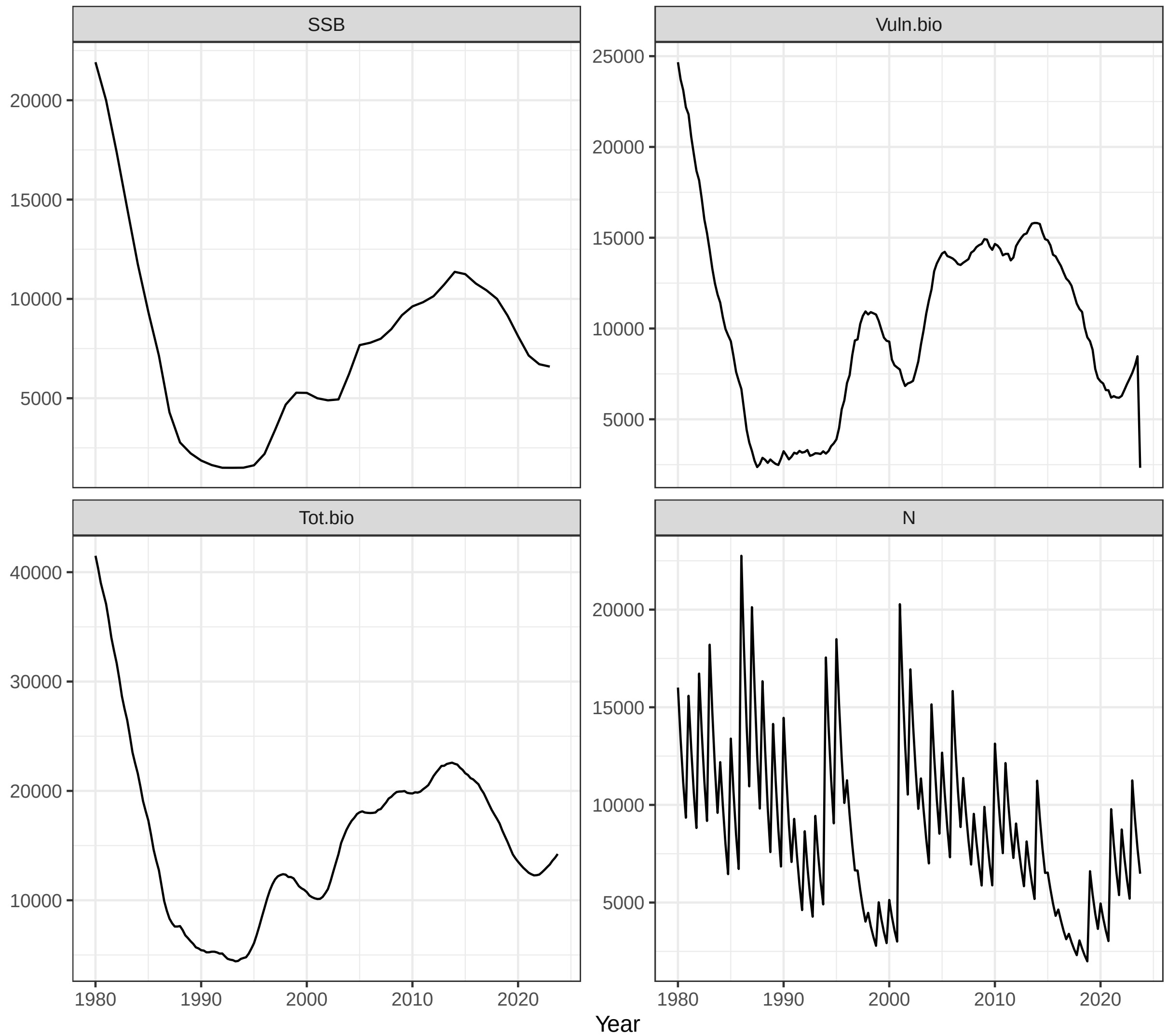

Figure 3: Combined Stock trajectories for Hake 17-18

```r
flqs = FLQuants(SSB = (ssb(stk[, , 1, ssbQ])), Vuln.bio = (vb(stk)),
    Tot.bio = (stock(stk)), N = (quantSums(stock.n(stk))))
plot(flqs) + theme_bw() + xlab("Year") + facet_wrap(~qname,
    scales = "free_y")
```

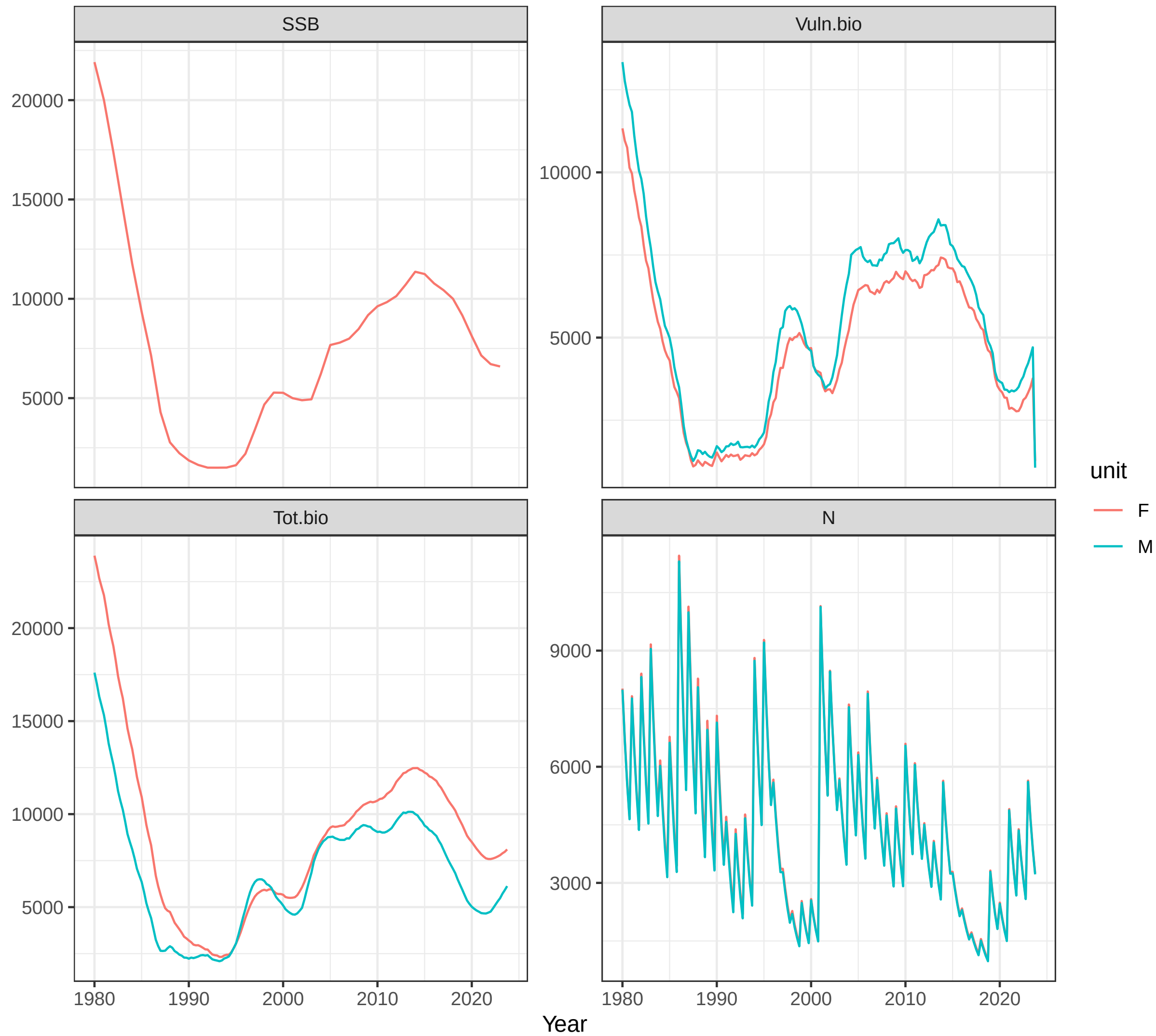

Figure 4: Combined Stock trajectories by sex for Hake 17-18

2.1 Plot SS3 Stock Dynamics

```
plotdyn(stk)
```

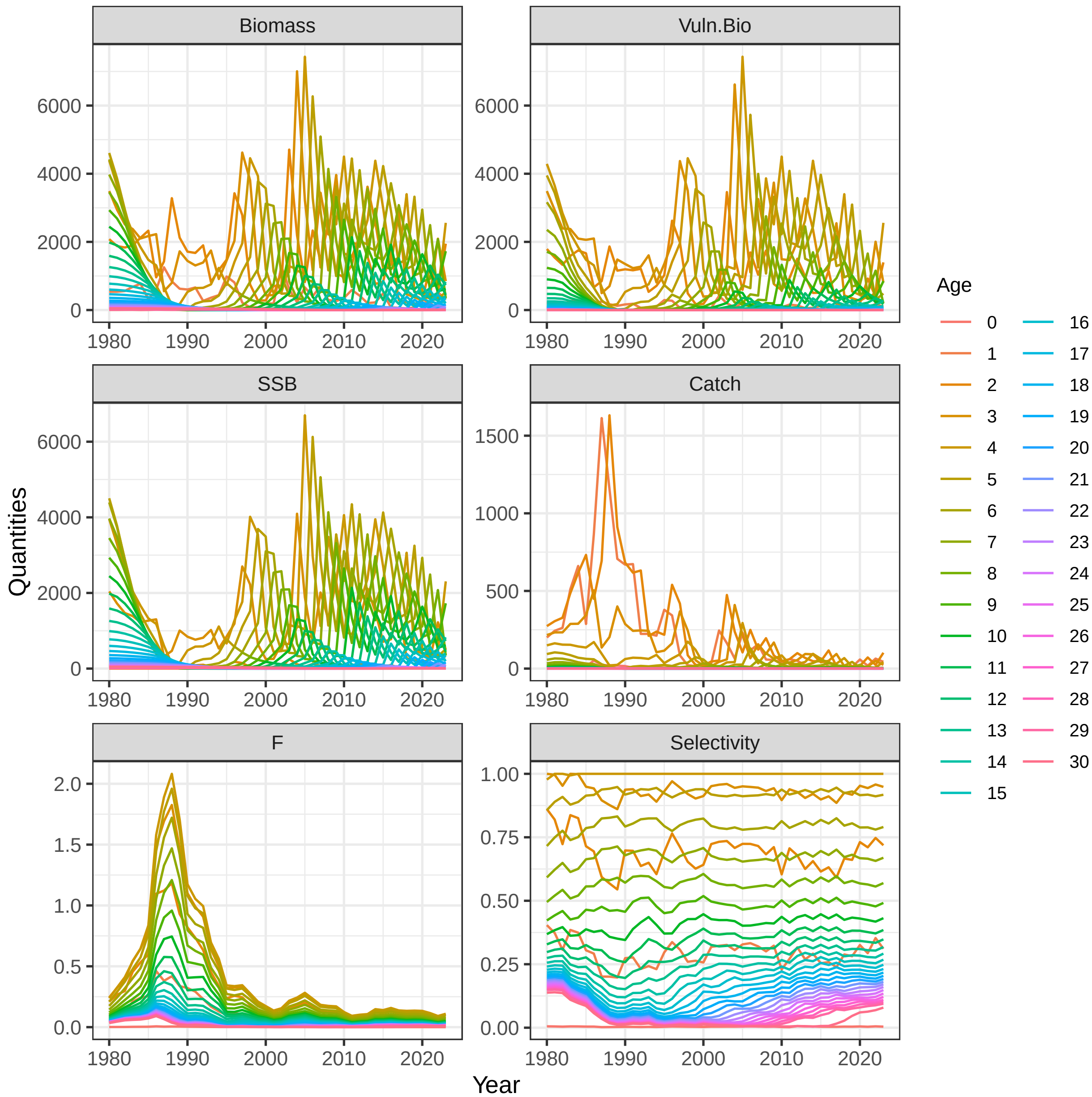

Figure 5: Stock assessment trajectories at age

```
plotbioyr(stk) + ggtitle(paste0(stk@name))
```

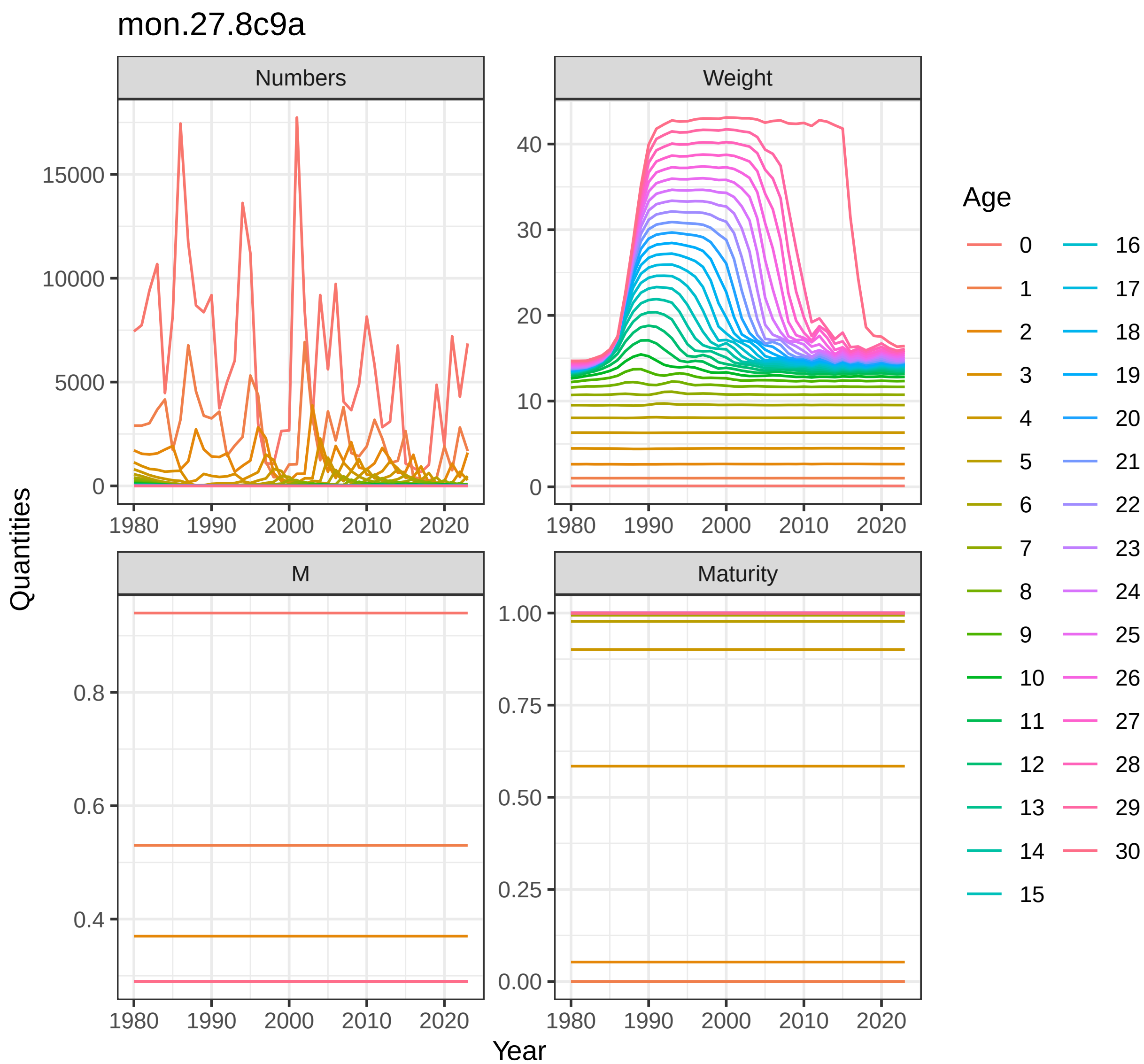

Figure 6: Stock biology trajectories at age

```r
plotbioage(stk) + theme(legend.position = "none")
```

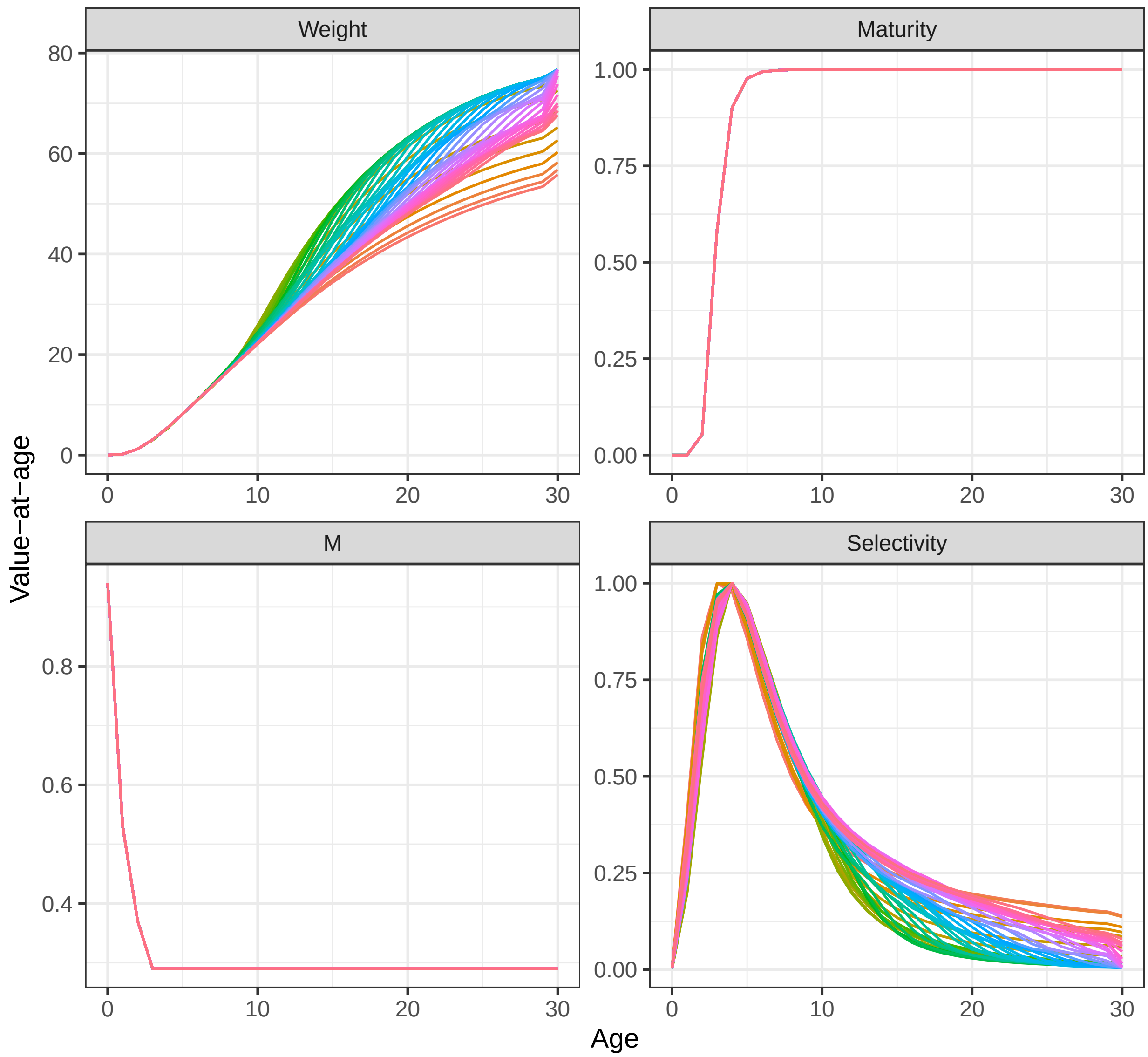

Figure 7: Annual stock quanties at age

2.2 Consistency checks using backtesting

Set seed

```
set.seed(1507)
```

Get bias adjusted recruitment deviations from ss3 model

Simplify to annual sex-structured model

```
if (dims(stk)$season > 1) {
    stka = simplify(stk, "season", weighted = TRUE, harvest = TRUE)

    discards.wt(stka) = stock.wt(stka)
    stka@discards = computeDiscards(stka)
    # Make annual sra
    sra = sr
    params(sra) = FLPar(an(sr@params[, 1]), params = rownames(sr@params))
} else {
    sra = sr
    stka = stk

}
```

```
yrs = an(dimnames(stk)$year)
recruit = out$recruit[out$recruit$Yr %in% yrs, ]
dms <- list(year = yrs)
sigR = mean(an(out$sigma_R_info[1:2, "SD_of_devs_over_sigma_R"]))  # Realised sigR
residuals <- FLQuant(exp(recruit$dev - 0.5 * recruit$biasadjuster *
    sigR^2), dimnames = c(age = 0, dms), units = "")
recs = FLQuant(recruit$pred_recr, dimnames = c(age = 0,
    dms), units = "")


if (dims(stk)$unit == 2) recs <- expand(recs, unit = c("F",
    "M"))
```

```
testC = fwd(stka, sr = recs[, ac(1981:2023)], control = fwdControl(year = 1981:2023,
    value = (unitSums(catch(stka)[, ac(1981:2023)])), quant = "catch"))

testF = fwd(stka, sr = recs[, ac(1981:2023)], control = fwdControl(year = 1981:2023,
    value = unitMeans(fbar(stka)[, ac(1981:2023)]), quant = "fbar"))
```

```
plot(window(FLStocks(ss3om = stka, backtestC = testC, backtestF = testF))) +
    theme_bw() + facet_wrap(~qname, scale = "free_y")
```

Note that minor deviations are likely due to difficulties in precisely adjusting the rec devs with bias correction.

2.3 Estimate candidate reference points

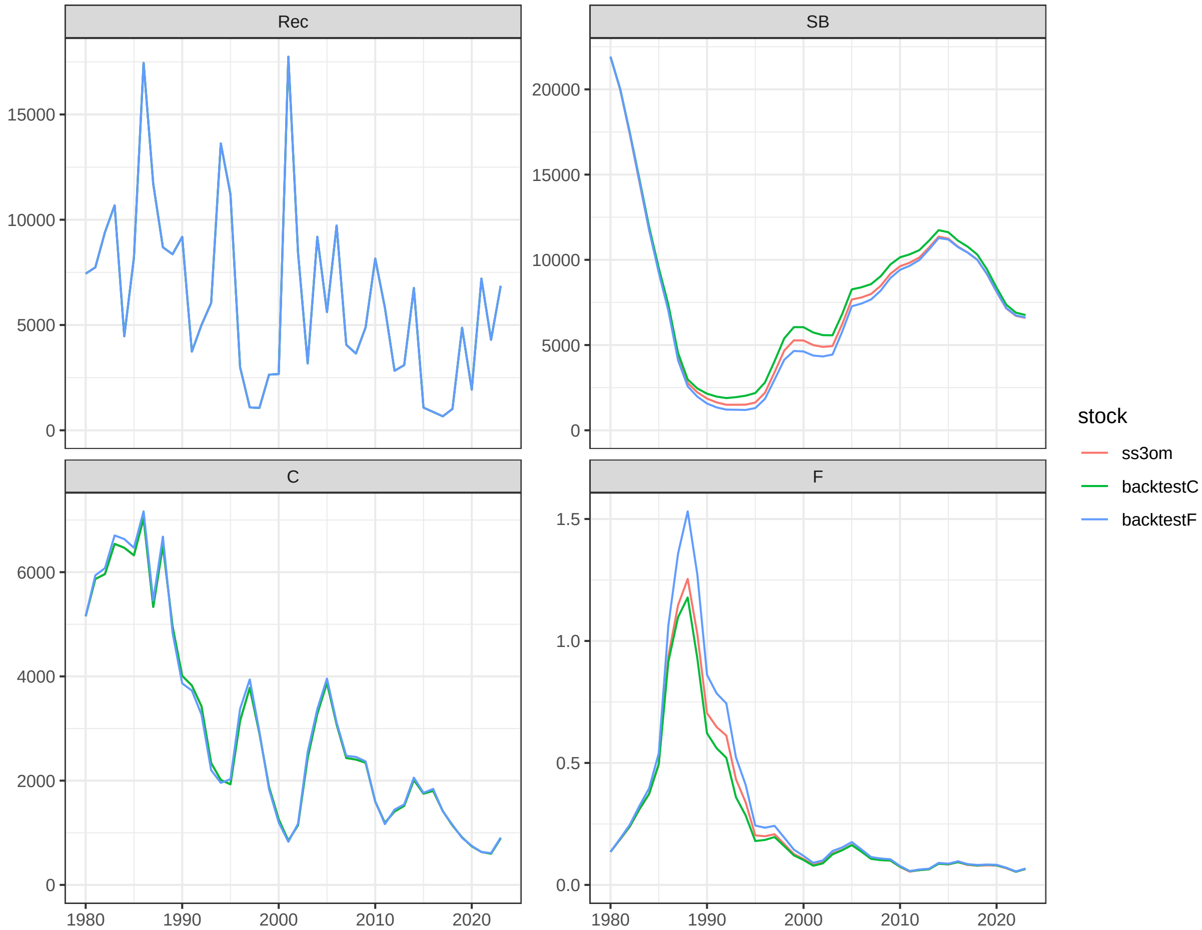

Figure 8: Comparison of stock trajectories from ss3om and a backtest under the same Fbar

```
# Extract pars
s = params(sra)[[1]]
R0 = params(sra)[[2]]
B0 = params(sra)[[3]]
# Main recdevs
recyrs = recruit$Yr[recruit$era == "Main"]
maindevs = unitSums(residuals[, ac(recyrs)])
rho = cor(maindevs[, -1], maindevs[, -length(maindevs)])
sigmaR = mean(an(out$sigma_R_info[1, "SD_of_devs"]))  # Realised sigR

rho
   [1] 0.4441913
sigmaR
   [1] 0.8046849

# MSY refpts
Bmsy <- out$derived_quants$Value[out$derived_quants$Label ==
    "SSB_MSY"]
Fmsy <- out$derived_quants$Value[out$derived_quants$Label ==
    "annF_MSY"]
MSY <- out$derived_quants$Value[out$derived_quants$Label ==
    "Dead_Catch_MSY"]

# Short cut devs
ay = out$endyr  # assessment year
SSBcv <- out$derived_quants$StdDev[out$derived_quants$Label ==
    paste0("SSB_", ay)]/out$derived_quants$Value[out$derived_quants$Label ==
    paste0("SSB_", ay)]

Fcv <- out$derived_quants$StdDev[out$derived_quants$Label ==
    paste0("F_", ay)]/out$derived_quants$Value[out$derived_quants$Label ==
    paste0("F_", ay)]
```

Set B_{MSY} proxies as ratio of B_0

```
Bfx = seq(25, 40, 5)
Bx = FLPar(Bfx/100 * B0, params = paste0("B", Bfx))
```

Find corresponding F_{Bx} with bisection algorithm

Test search for Fb40

```
Fb35 = fwdF4B(stka, sra, btgt = an(Bx["B35"]), nfy = 100,
    niy = 0, ftune = c(0.1 * Fmsy, 1.5 * Fmsy), tol = 0.001,
    verbose = TRUE)
```

Check

```


Fb35 = FLStockR(Fb35)
fb35 = tail(unitMeans(fbar(Fb35), 1))
```

```r
cb35 = tail(unitSums(catch(Fb35), 1))
an(fb35)
   [1] 0.09819662
Fb35@refpts = FLPar(Fb35 = fb35, B35 = Bx[3], Yeq35 = cb35)


plotAdvice(Fb35)
```

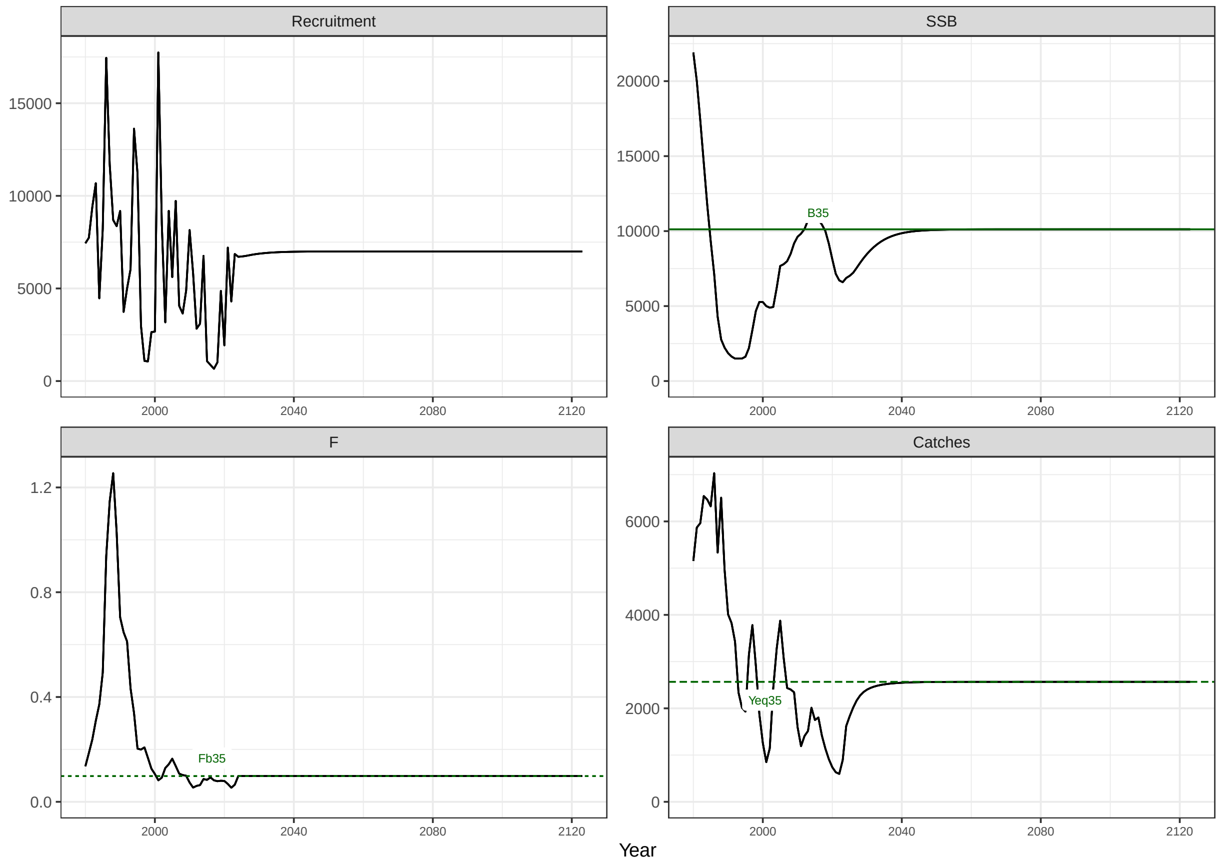

Figure 9: Stock trajectories relative to SBB_{35} and F_{SB35}

```r
# Check scale B35/B0
round(an(tail(unitSums(ssb(Fb35)))/B0), 2)
   [1] 0.35
```

Run

```r
Fx = do.call(c, lapply(an(Bx), function(x) {
    fs = fwdF4B(stka, sra, btgt = x, nfy = 100, niy = 0,
        ftune = c(0.3 * Fmsy, 1.5 * Fmsy), tol = 0.001,
        verbose = F)
    an(tail(unitMeans(fbar(fs)), 1))
}))
```

```r
Fx = FLPar(Fx, params = paste0("F", Bfx))
```

```r
refpts = rbind(FLPar(Fmsy = Fmsy, Bmsy = Bmsy, Blim = 0.1 *
    B0, B0 = B0, R0 = R0), Fx, Bx)

refpts
   An object of class "FLPar"
   params
       Fmsy      Bmsy      Blim        B0        R0       F25       F30       F35
   1.82e-01 5.72e+03 2.89e+03 2.89e+04 7.44e+03 1.31e-01 1.13e-01 9.81e-02
        F40       B25       B30       B35       B40
   8.53e-02 7.23e+03 8.67e+03 1.01e+04 1.16e+04
   units:  NA
```

Create FLStockR with @refpts

```r
stkr = FLStockR(stka)

stkr@refpts = refpts
```

Summarize short-cut params

```r
spars = FLPar(s = s, sigmaR = sigmaR, rho = rho, Fcv = Fcv,
    SSBcv = SSBcv)
```

```r
plotAdvice(stkr)
```

```r
save(stk, stka, stkr, refpts, sr, sra, spars, file = "rdata/om.ss3ref.rdata")
```

3 Set up short-cut MSE

Load OM conditioned to Stock Synthesis

```r
load("rdata/om.ss3ref.rdata", verbose = T)
   Loading objects:
     stk
     stka
     stkr
     refpts
     sr
     sra
     spars
```

Next set up the MSE horizon

```r
# data year
dy <- dims(stka)$maxyear
# FINAL year
```

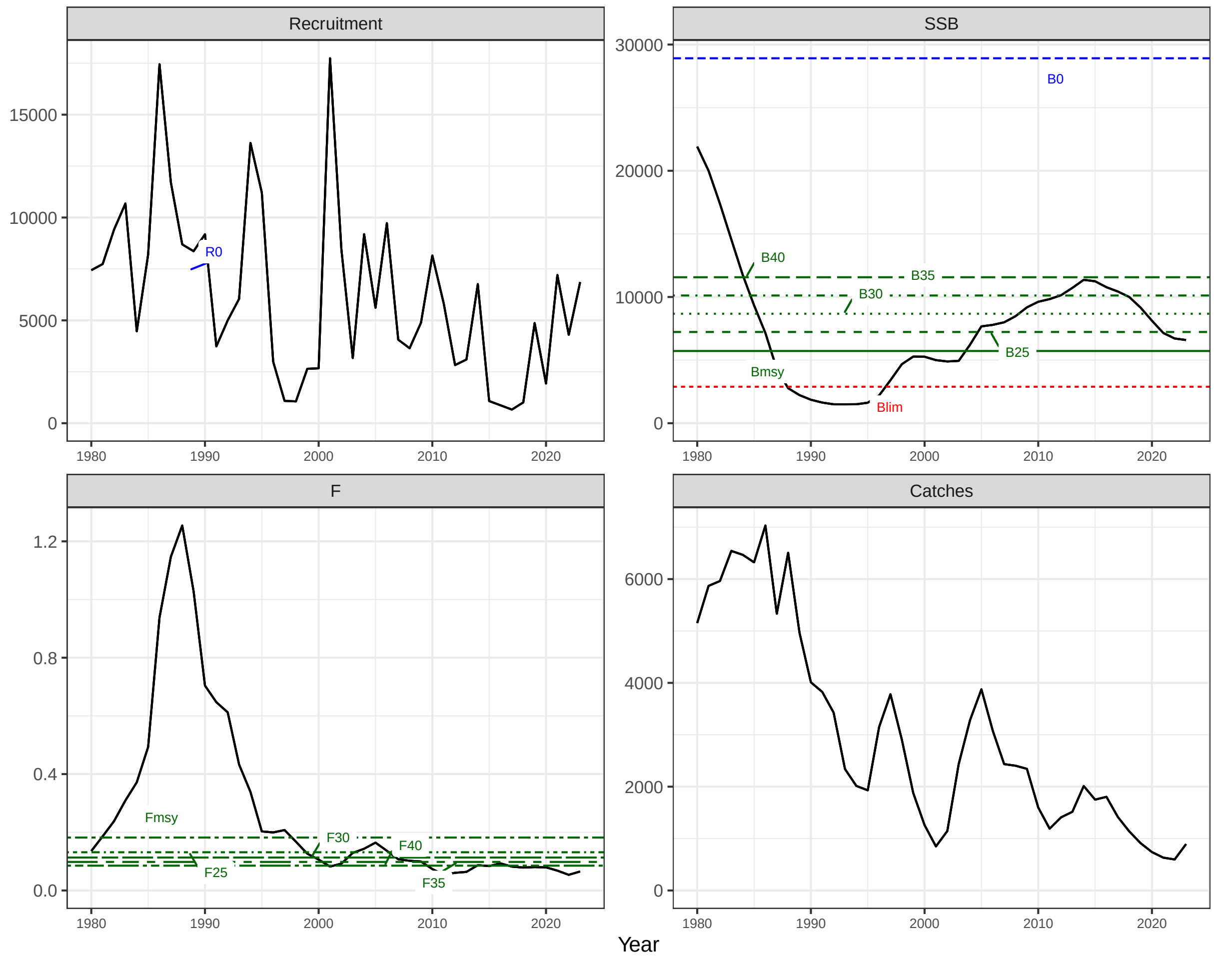

Figure 10: Status Advice plot showing stock trajectories of Recruitment, SSB, F, recruitment and $Yield$

```
fy <- dy + 50
# assessment year
ay = dy + 1
# intermediate years
iy = ay
```

For illustration the number of iterations are reduced to 100.

```
# NUMBER iterations
it <- 200
```

Subset the 1000 simulated stock iterations to the first 100

```
stki = propagate(stka, it)
```

Generate recruitment

```
srdevs <- rlnormar1(n = it, sdlog = spars["sigmaR"], rho = spars["rho"],
    years = seq(dy, fy))
# Sex-structured
if (dims(stki)$unit == 2) srdevs <- expand(srdevs, unit = c("F",
    "M"))
```

Now construct the `FLom` object from the `mse` package by passing on `FLStock`,`refpts`, `sr` and the method used for foward projections.

```
om <- FLom(stock = stki, refpts = refpts, sr = sra, projection = mseCtrl(method = fwd.om),
    deviances = srdevs)

class(om)
   [1] "FLom"
   attr(,"package")
   [1] "mse"
```

Next add the structure for the future years: average of last 3 years

```
om <- fwdWindow(om, end = fy)
```

Next, a so called observation error is constructed. In the case of the short-cut MSE, it simply holds the "perfect" stock information. For a full MSE with inbuilt estimation model it would also generate the observations with errors, such a catch-at-age and survey numbers at age for SAM or a4a, or biomass surveys indices and catches for SPicT or JABBA.

```
oem <- FLoem(observations = list(stk = stock(om)), method = perfect.oem)
```

However, there is increasing realisation that the assessment estimates are imperfect. Therefore, ICES has implemented procedures to add uncertainty about the key quantities F and SSB, where the error on F is specified by a the random error term Fcv and a first order autocorrelation parameter $Fphi$ and the precision of SSB can specified by $SSBcv$

Short-cut deviations

```r
sdevs <- shortcut_devs(om, Fcv = spars["Fcv"], Fphi = 0.432,
    SSBcv = spars["SSBcv"])
```

Finally, the implementation error module `iem` is setup. In this case, with a random catch implementation error of 10%.

```r
iem <- FLiem(method = noise.iem, args = list(noise = rlnorm(it,
    rec(om) %=% 0, 0.1)))
```

```r
save(om, oem, sdevs, iem, file = "rdata/flom.rda", compress = "xz")
```

3.1 Setting up harvest control rules

This can be effectively implemented ICES advice rule hockey-stick by setting the $B_{trigger}$ to zero, using the `icesControl` function. This function can also take the `SSBdevs` and `Fdevs` that implement the deviations from the SSB and F with aim to account for assessment errors.

Adjusted mse controls for sex-structured stocks

```r
shortcut.sa2 <- function(stk, idx, SSBdevs=unitSums(ssb(stk)) %=% 1, args, tracking, ...) {
  # DIMS
  y0 <- args$y0
  dy <- args$dy
  ay <- args$ay
  it <- args$it

  # SUBSET oem stock
  stk <- window(stk, end=dy)

  ind <- FLQuants(
    # SSB + devs
    #><> add unitSums
    ssb=unitSums(ssb(stk)) * window(SSBdevs, start=y0, end=dy))

  track(tracking, "conv.est", ac(ay)) <- 1

  list(stk=stk, ind=ind, tracking=tracking)
}



tac.is2 <- function(stk, ctrl, args, output="catch", recyrs=-2,
  Fdevs=unitMeans(fbar(fut)) %=% 1, dtaclow=NA, dtacupp=NA, fmin=0, reuse=TRUE,
  initac=metrics(stk, output)[, ac(iy - 1)], tracking) {
  # EXTRACT args
  spread(args)


  # SET control years
  cys <- seq(ay + management_lag, ay + management_lag + frq - 1)
```

```r
# PREPARE stk for cys, biology as in last nsqy years
fut <- fwdWindow(stk, end=cys[length(cys)], nsq=nsqy)

# PARSE recyrs if numeric
id <- dimnames(stk)$year

# COERCE to list
if(!is.list(recyrs)) {
  recyrs <- list(recyrs)
}

# PARSE list
for(i in recyrs) {
  if(is(i, 'character')) {
    id <- id[!id %in% i]
  } else if(all(i < 0)) {
    if(length(i) == 1)
      id <- rev(rev(id)[-seq(abs(i))])
    else
      id <- rev(rev(id)[i])
  } else if(all(i > 0)) {
    id <- rev(rev(id)[seq(abs(i))])
  }
}

# SET years to use
recyrs <- id

# CHECK recyrs
if(!all(recyrs %in% dimnames(stk)$year)) {
  stop("'recyrs' cannot be found in input stk")
}

# TODO: OTHER rec options


# SET GM recruitment from past
#><> add unitSums()
gmnrec <- exp(yearMeans(log(unitSums(rec(stk))[, recyrs])))

# SETUP SRR
srr <- predictModel(model=rec~a, params=FLPar(a=gmnrec))

# STORE geomeanrec value
track(tracking, "gmrec.isys", ay + management_lag) <- gmnrec

# ADD F deviances for 1 year

# reuse = TRUE
if(isTRUE(reuse) | toupper(reuse) == 'F') {
  ftar <- rep(c(ctrl[1,]$value * Fdevs[, ac(cys[1])]), length(cys))
# reuse = FALSE
} else {
```

```r
  ftar <- c(ctrl$value * Fdevs[, ac(cys)])
}

# TRACK Ftarget
track(tracking, "fbar.isys", cys) <- ftar

# FORECAST for iyrs and my IF mlag > 0,
if(management_lag > 0) {

  # SET F for intermediate year #><> added unitMeans
  #><> add unitMeans()

  fsq <- unitMeans(fbar(stk))[, ac(dy)]

  # TODO: ADD TAC option

  # CONSTRUCT fwd control
  fctrl <- fwdControl(
    # ay as intermediate with Fsq TODO: Other options
    list(year=seq(ay - data_lag + 1, length=management_lag),
      quant="fbar", value=rep(c(fsq), management_lag)),
    # target
    list(year=cys, quant="fbar", value=c(ftar))
  )

# else only for my
} else {
  fctrl <- fwdControl(
    list(year=ay + management_lag, quant="fbar", value=ftar))
}

# RUN STF ffwd
fut <- ffwd(fut, sr=srr, control=fctrl)

# ID iters where hcr set met trigger and F > fmin
id <- c(tracking[[1]]["decision.hcr", ac(ay)] > 2) &
  c(unitMeans(fbar(fut))[, ac(ay + management_lag)] > fmin)

# EXTRACT catches
if(isTRUE(reuse) | toupper(reuse) == "C") {
  TAC <- expand(unitSums(catch(fut))[, ac(cys)[1]], year=seq(length(cys)))
} else {
  TAC <- unitSums(catch(fut))[, ac(cys)]
}

# GET TAC dy / ay - 1
if(ay == iy)
  prev_tac <- rep(c(initac), length=args$it)
else
  prev_tac <- c(tracking[[1]]["isys", ac(ay)])

# APPLY upper and lower TAC limit, if not NA and only for id iters
if(!is.na(dtacupp)) {
```

```r
    iter(TAC, id) <- pmin(c(iter(TAC, id)), prev_tac[id] * dtacupp)
  }
  if(!is.na(dtaclow)) {
    iter(TAC, id) <- pmax(c(iter(TAC, id)), prev_tac[id] * dtaclow)
  }

  # CONSTRUCT fwdControl
  # TODO: USE frq here
  ctrl <- fwdControl(lapply(seq(length(cys)), function(x)
    list(year=cys[x], quant=output, value=TAC[,x])))

  return(list(ctrl=ctrl, tracking=tracking))
}

# Add fixed Fmsy
```

```r
arule <- mpCtrl(list(
  # (est)imation method: shortcut.sa + SSB deviances
  est = mseCtrl(method=shortcut.sa2,
    args=list(SSBdevs=sdevs$SSB)),

  # hcr: hockeystick (fbar ~ ssb | lim, trigger, target, min)
  hcr = mseCtrl(method=hockeystick.hcr,
    args=list(lim=0, trigger=0.01, target=Fmsy,
    min=0, metric="ssb", output="fbar")),

  # (i)mplementation (sys)tem: tac.is (C ~ F) + F deviances
  isys = mseCtrl(method=tac.is2,
    args=list(recyrs=-2, fmin=0, Fdevs=sdevs$F))
  ))
```

This rule can now be run passing on the `om`, `oem` and `arule` and an additional argument to set the implementation year to 2024.

Note that the default setting assumes 1 year lag between data (reference) year and assessment (reporting) year and that the TAC is implemented the next year. In case the assessment is conducted in 2023, based on data from 2022 and the TAC is implemented for 2024.

```
Loading objects:
  run1
Loading objects:
  run2
```

```r
mseargs <- list(iy = dy, fy = fy, data_lag = 1, management_lag = 1,
    frq = 1)

system.time(run <- mp(om, oem = oem, ctrl = arule, args = mseargs,
    verbose = T))
```

```r
# make FLStocks from om until 2024 (implementation)
# and the run
stks = FLStocks(stock = window(om@stock, end = 2023), fixedFmsy = run@om@stock)
```

```r
plot(stks) + facet_wrap(~qname, scales = "free") + theme_bw() +
    geom_vline(xintercept = c(dy), linetype = 2, col = 1) +
    geom_vline(xintercept = c(fy), linetype = 2, col = 4)
```

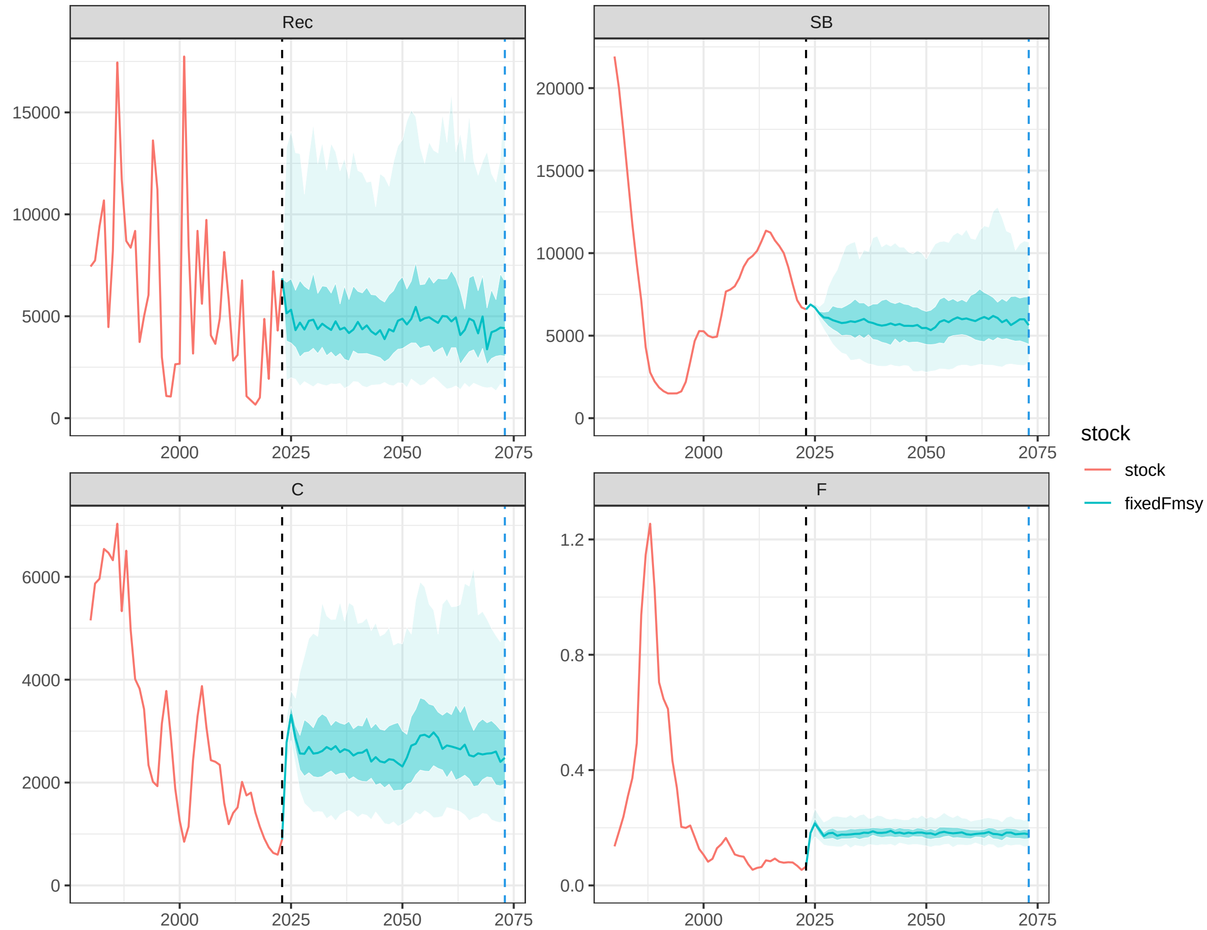

Figure 11: Initial OM and the MSE forecast horizon under a fixed F_{tgt} rule

The `run` can also be appended to the `om` to make a single `FLStockR` object with reference points

```r
runR = FLStockR(append(window(om@stock, end = dy), run@om@stock))
runR@refpts = refpts[1:5]
```

This allows to quickly evaluate the stock status under a fixed F_{MSY} rule.

It can be seen that despite "perfect" knowledge of the "true" F_{MSY}, and fishing pressure is on average F_{MSY}, the stock fails to attain biomass levels at B_{MSY} with a relative high risk to fall below B_{lim}. This is a well known fact as a result of the lags between data and management and asymetric risks in that exceeding F_{MSY} is more consequential on both SSB and long term yield, then fishing below F_{MSY}, In the case of the latter, more biomass is left in the water, which provides increased future reproduction potential and catch opportunity.

```r
plotAdvice(runR) + geom_vline(xintercept = c(dy), linetype = 2,
    col = 1) + geom_vline(xintercept = c(fy), linetype = 2,
    col = 4) + scale_x_continuous(breaks = seq(1970, 3000,
    5)) + theme(axis.text.x = element_text(size = 8, angle = 90,
    vjust = 0.5))
```

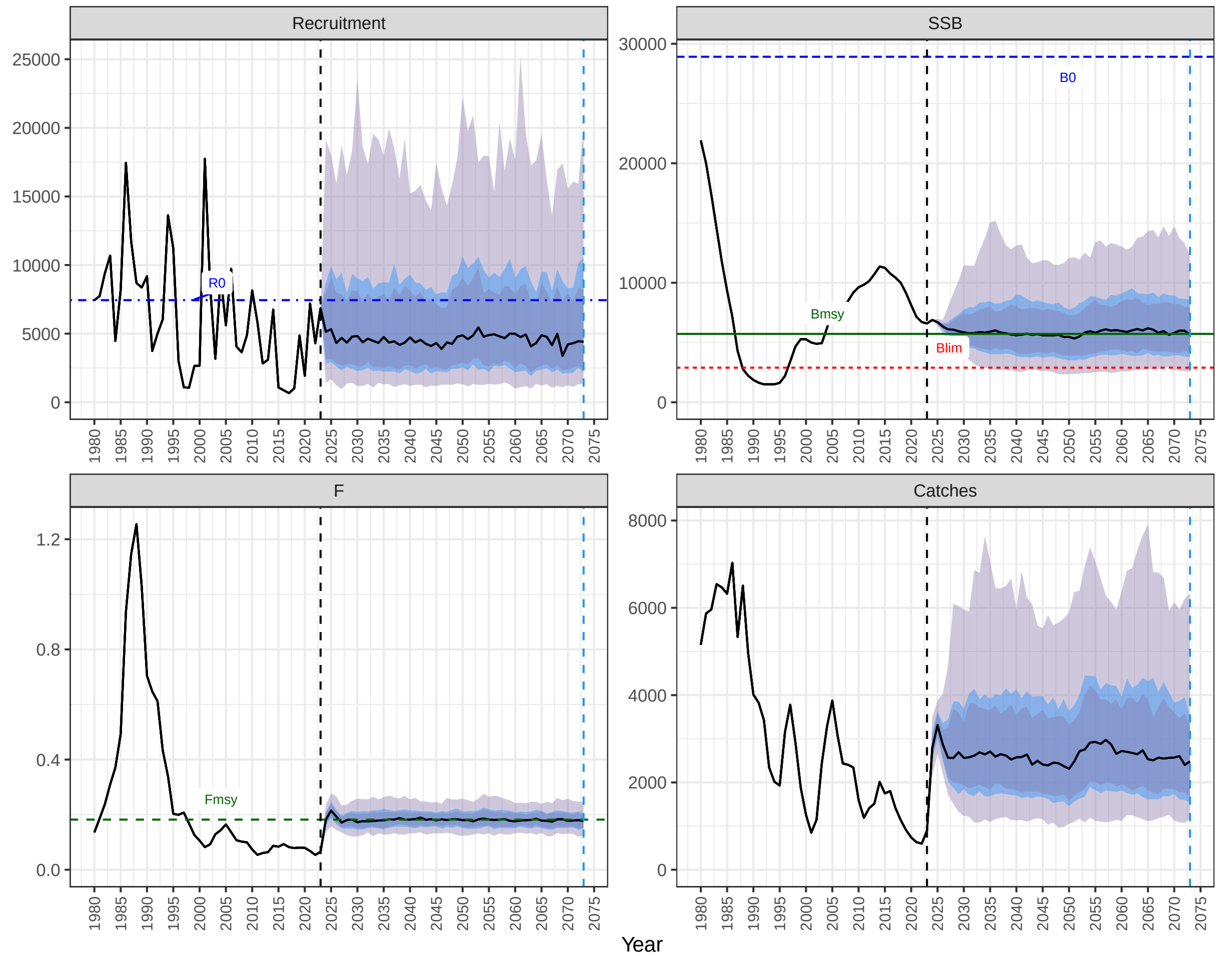

Figure 12: Stock Status under a fixed F_{tgt} short-cut MSE rule

Even relatively simplified short-cut MSE frameworks provide a powerful to explore alternative HCRs to achieve better trade-off between risks and yield.

Here, the convential hockey-stick control rule is explored with different ratios of F_{adv}/F_{tgt} and $B_{trigger}/B_{tgt}$ settings, where the $B_{trigger}$ promts a linear reduction in F_{adv} if SSB falls below it.

```r
Fadv = c(Fmsy, rep(Fmsy, 3), rep(an(Fx), each = 3))
Btgt = c(Bmsy, rep(Bmsy, 3), rep(an(Bx), each = 3))
Btri = c(Btgt * c(0, rep(c(0.5, 0.6, 0.7), 5)))

hcrs = Map(function(x, y, z) {
    p = plotGFCM(fadv = x, btrigger = y, ftgt = x, btgt = z,
        blim = an(refpts["Blim"]), fmin = 0, bclose = 0,
```

```
        kobe = FALSE, text = F, rel = F)
    p = p
    return(p)
}, x = Fadv, y = Btri, z = Btgt)
# plot ggplot list
ggarrange(plotlist = hcrs[-1], ncol = 3, nrow = 5)
```

The same settings can be specified for the new `mps` function in `mse`, which allow to explore variations of the HCR parameters.

The function `combinations` enables to vary more than one parameter at the time.

```
hcrs = combinations(target = Fadv, trigger = Bx[[2]])
hcrs$trigger = Btri  # Overwrite
hcrs
   $target
    [1] 0.18199700 0.18199700 0.18199700 0.18199700 0.13137908 0.13137908
    [7] 0.13137908 0.11325048 0.11325048 0.11325048 0.09810776 0.09810776
   [13] 0.09810776 0.08531109 0.08531109 0.08531109

   $trigger
    [1]    0.000 2860.250 3432.300 4004.350 3614.100 4336.920 5059.740 4336.920
    [9] 5204.304 6071.688 5059.740 6071.688 7083.636 5782.560 6939.072 8095.584

save(om, oem, arule, hcrs, mseargs, file = "rdata/inpMP.rdata")
```

These changes in parameters can simply be passed on to the existing `arule` to run the variations with `mps`.

```
runs <- mps(om, oem = oem, ctrl = arule, args = mseargs,
    hcr = hcrs)
```

Now combine with the Fixed F_{MSY} run and see if we can do better.

```
scenarios = c("FixedFmsy", paste0(rep(c("Fmsy", rownames(Fx)),
    each = 3), ".Btri.", rep(c(0.5, 0.6, 0.7), 5)))

names(runs) = scenarios
```

Save

```
save(runs, mseargs, file = "rdata/shortcut_runs.rda", compress = "xz")
```

An easy way of basic plotting is to extract the `FLStocks` from the list of MSE runs with `lapply`

```
stks = FLStocks(lapply(runs, function(x) {
    out = (x@om@stock)
    out = FLStockR(out)
    out@refpts = refpts[c(1:5)]
    out
}))
```

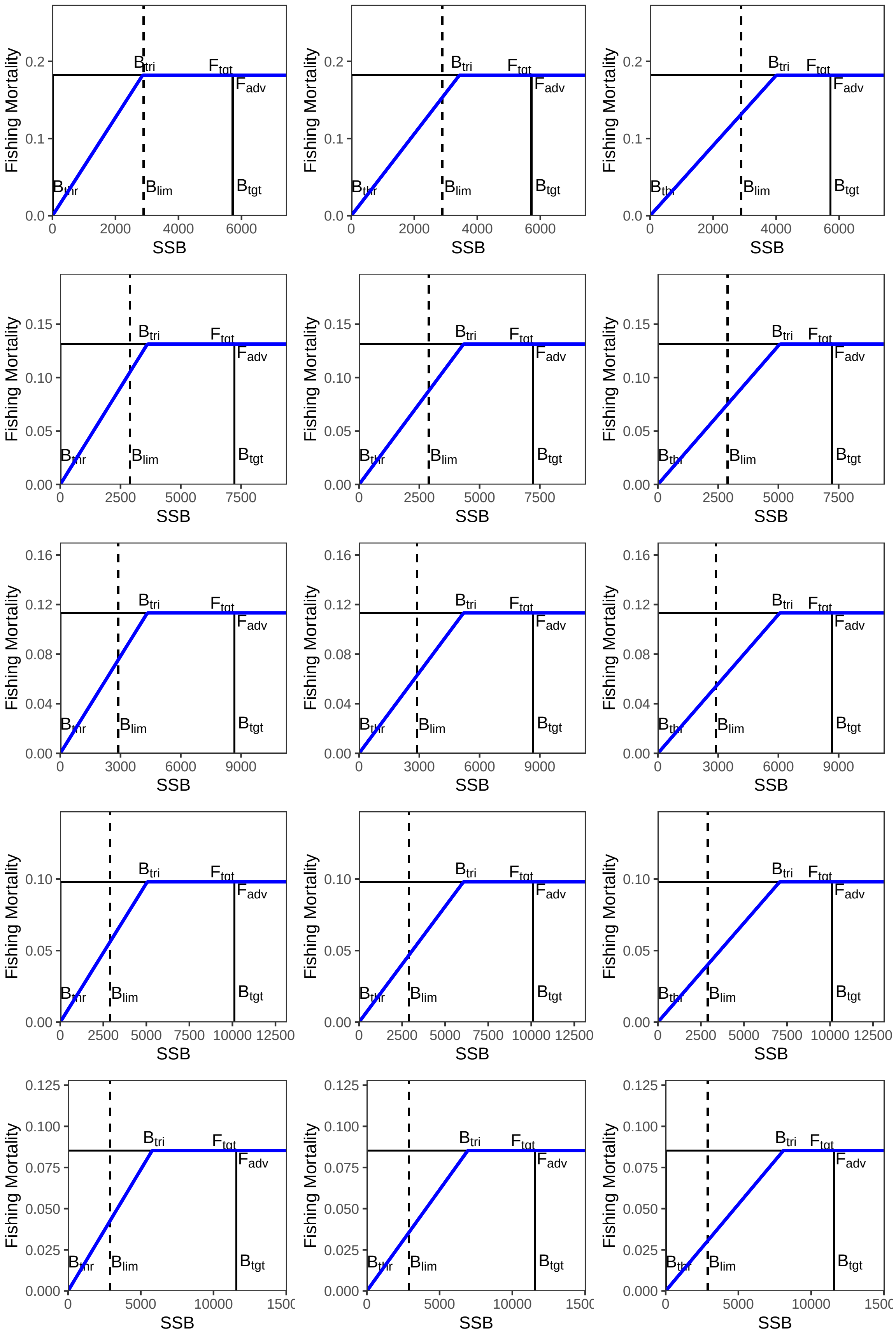

Figure 13: Alternative HCR formulations, with $F_{min} = 0$ (top) and $F_{min} = 0.5F_{tgt}$ (bottom) under different F_{adv} options set at $F_{adv} = F_{adv}$, $F_{adv} = 0.8F_{adv}$ and at $F_{adv} = 0.7F_{adv}$ from left to right

```r
ref = FLStockR(window(stock(om), end = dy))
ref@refpts = refpts
pstks = FLStocks(c(FLStocks(stock = window(ref, end = dy)),
    stks))

plotAdvice(pstks) + facet_wrap(~qname, scales = "free") +
    theme_bw() + scale_color_manual(values = c("black",
    ss3col(length(pstks))[-1])) + scale_fill_manual(values = c("darkgrey",
    ss3col(length(pstks))[-1])) + geom_vline(xintercept = c(dy),
    linetype = 2, col = 1) + geom_vline(xintercept = c(fy),
    linetype = 2, col = 4) + scale_x_continuous(breaks = seq(1900,
    3000, 5)) + theme(axis.text.x = element_text(size = 8,
    angle = 90, vjust = 0.5))
```

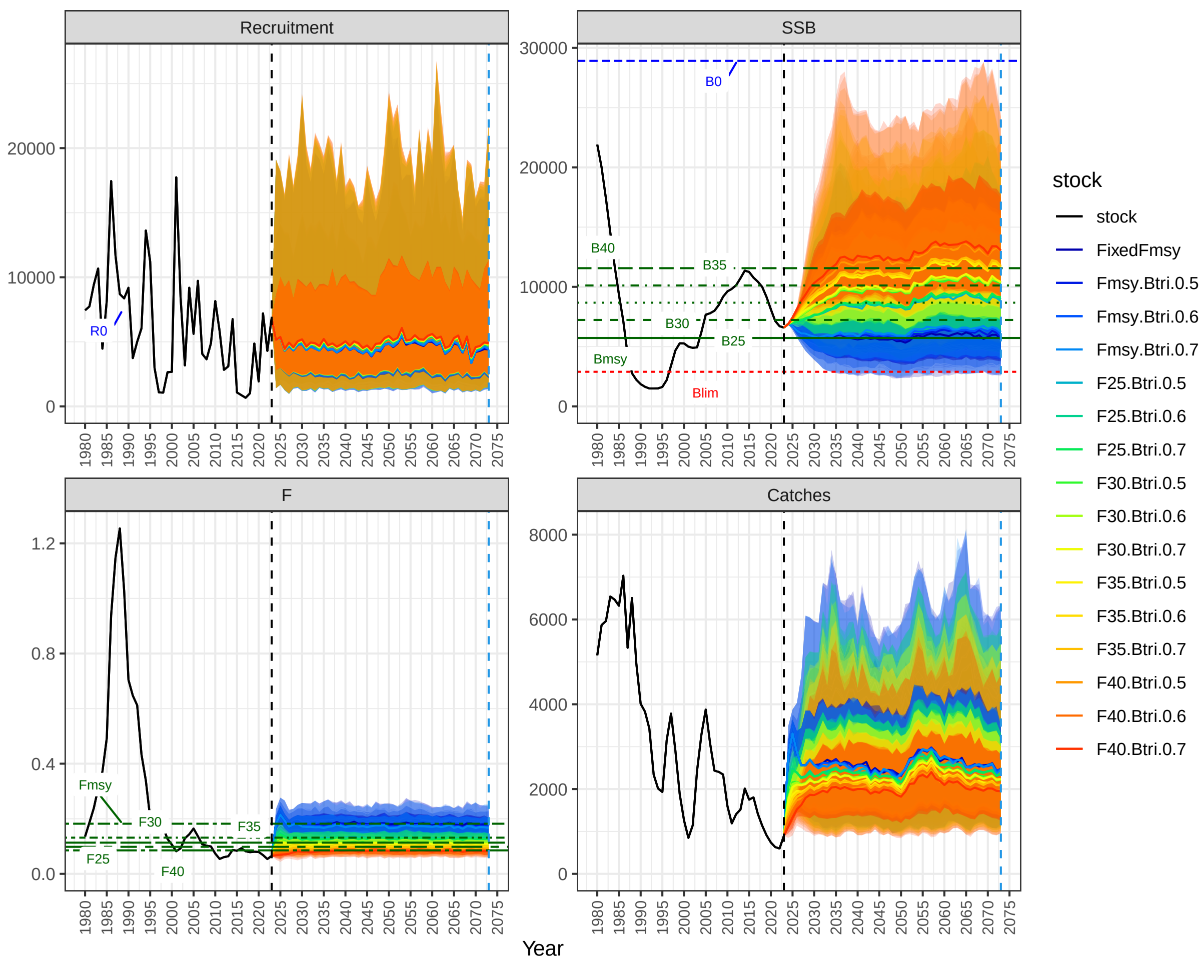

Figure 14: Initial OM and the MSE forecast horizon under a fixed F_{tgt} rule and 6 HCR hockey-stick scenarios

```r
medstks = FLStocks(lapply(runs, function(x) {
    stockMedians(x@om@stock)
}))
medref = stockMedians(ref)
```

```r
pmstks = FLStocks(c(FLStocks(stock = window(medref, end = dy)),
    medstks))

plotAdvice(pmstks) + facet_wrap(~qname, scales = "free") +
    scale_color_manual(values = c("black", ss3col(length(pmstks))[-1])) +
    theme_bw() + xlab("Year") + geom_vline(xintercept = c(dy),
    linetype = 2, col = 1) + geom_vline(xintercept = c(fy),
    linetype = 2, col = 4) + scale_x_continuous(breaks = seq(1900,
    3000, 5)) + theme(axis.text.x = element_text(size = 8,
    angle = 90, vjust = 0.5))
```

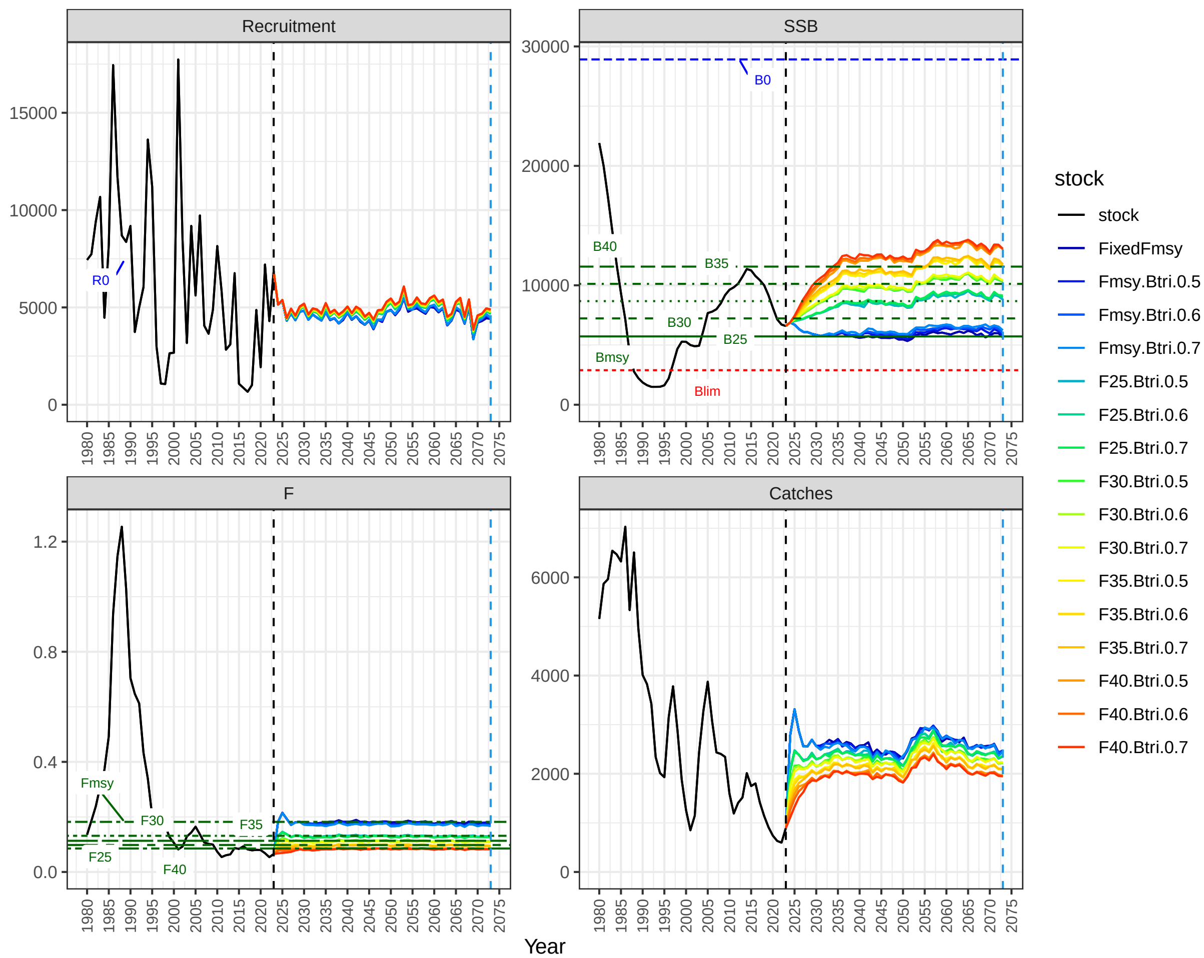

Figure 15: Median MSE forecast horizon under a fixed F_{tgt} rule and 6 HCR scenarios

4 Performance Evaluation with adjustment for sex-tructured stocks

4.1 Define Metrics for performance evaluation of sex-structured

```
metrics <- list(SB = function(x) unitSums(ssb(x)), F = function(x) unitMeans(fbar(x)),
    C = function(x) unitSums(catch(x)), Rec = function(x) unitSums(rec(x)))
```

Define performance statistics - ordered alphabetically

```
stats <- list(a.medianFmsy = list(~yearMedians(F/Fmsy),
    name = "F/Fmsy", desc = "Median annual F/Fmsy"), b.medianBmsy = list(~yearMedians(SB/Bmsy),
    name = "B/Bmsy", desc = "Median annual B/Bmsy"), c.medianCmsy = list(~yearMedians(C/MSY),
    name = "Catch/MSY", desc = "Median Catch/MSY over years"),
    d.aavC = list(~yearMedians(iav(C)), name = "AAV", desc = "Median annual variation in catches"),
    e.riskBlim = list(~apply(iterMeans((SB/Blim) < 1), 1,
        max), name = "P3(B<Blim)", desc = "Probability that SSB < Blim"),
    f.risk10SB0 = list(~apply(iterMeans((SB/(B0 * 0.1)) <
        1), 1, mean), name = "P(B<SB0.10)", desc = "Probability that SSB < 10% SB0"),
    g.P80BMSY = list(~apply(iterMeans((SB/(Bmsy * 0.8)) >
        1), 1, max), name = "B>80Bmsy", desc = "Probability that SSB > 80% x Bmsy"),
    h.medianSBMSY = list(~yearMedians(SB/Bmsy), name = "SSB/SSB[MSY]",
        desc = "Median annual SSB/SSBmsy"), e.medianFMSY = list(~yearMedians(F/Fmsy),
        name = "F/F[MSY]", desc = "Median annual F/FMSY"))
```

4.2 Long-term last 25 years (50 years)

Replace deterministic MSY with MSE MSY from a run with the true, fixed F_{MSY} in the reference points

```
refpts.perf = refpts[1:5]
refpts.perf = rbind(refpts.perf, FLPar(MSY = median(unitSums(catch(stks$FixedFmsy[,
    ac((fy - 24):fy)])))))
```

Compute performance metrics

```
perf <- performance(stks, refpts = refpts.perf, metrics = metrics,
    statistics = stats, years = list((fy - 20):fy))
```

4.3 PLOT performance

```
ncol = length(unique(perf$mp))
pbp = plotBPs(perf,
statistics=c("a.medianFmsy","b.medianBmsy","c.medianCmsy", "d.aavC", "e.riskBlim", "g.P80BMSY"),
size=3, target = c(a.medianFmsy=1,b.medianBmsy=1, c.medianCmsy=1),
limit= c(e.riskBlim=0.05,c.medianCmsy=0.95),
yminmax = c(0.05, 0.95))+theme_bw()+
facet_wrap(~name,scales = "free_y",ncol=2)+
ggtitle(paste0("Performance: Reference Points"))+
ylab("Performance statistics")+
scale_fill_manual(values=ss3col(ncol))+ # USE FLRef::ss3col
theme(axis.text.x=element_blank())+xlab("Candidates")
pbp
```

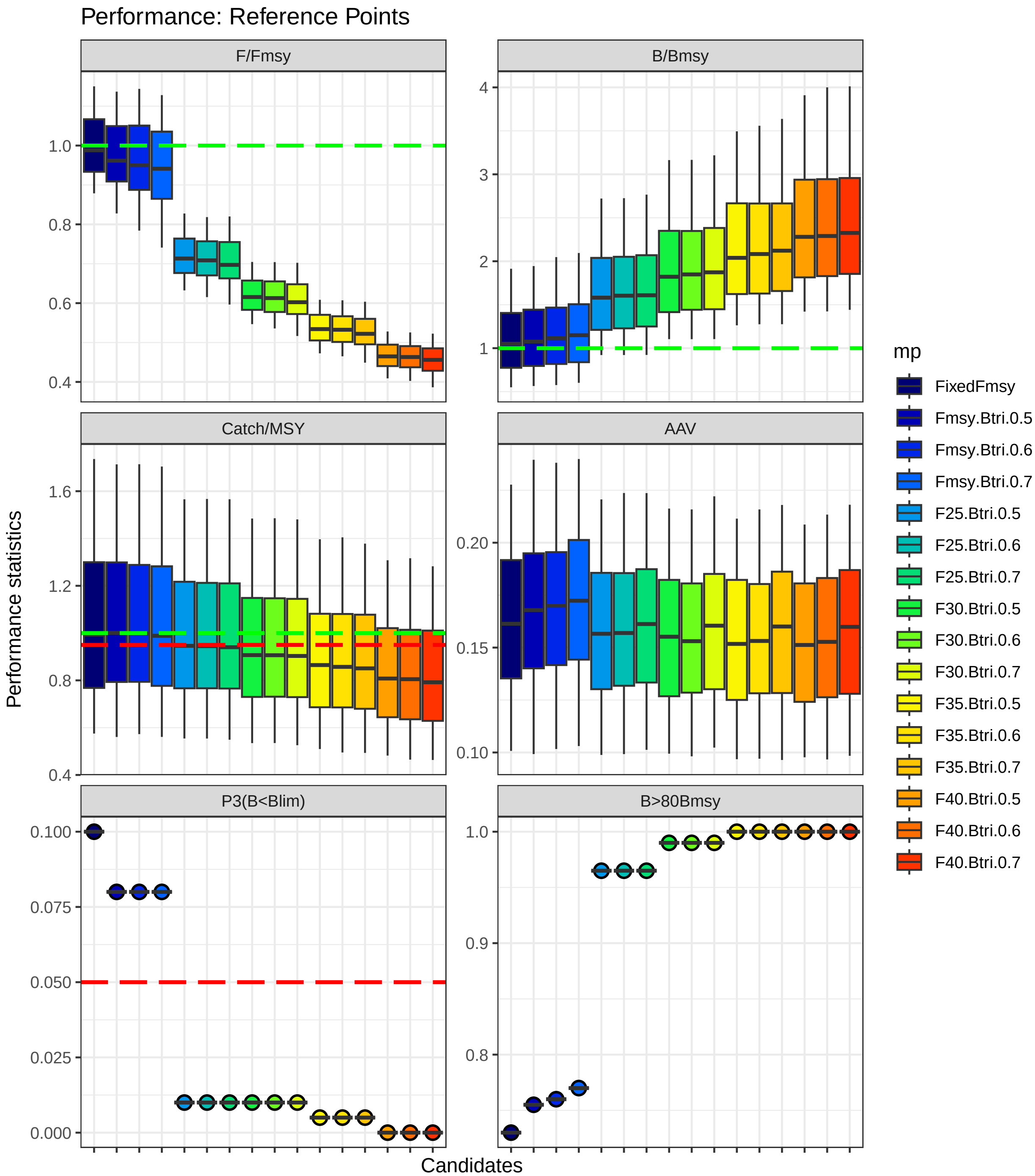

Figure 16: Performance: Long-term through 2045

4.3.1 MSE kobe plot

```
kbcex =function(){theme(plot.title = element_text(size=10),
                        legend.key.size = unit(0.3, 'cm'), #change legend key size
                        legend.key.height = unit(0.4, 'cm'), #change legend key height
                        legend.key.width = unit(0.4, 'cm'), #change legend key width
                        legend.text = element_text(size=10)) #change legend text font size
}
kobeMPs(perf,y="a.medianFmsy", x="b.medianBmsy", SBlim=NULL, Ftarget = 1)+
  ylab(expression(F/F[MSY]))+xlab(expression(B/B[MSY]))+ylim(0,2.5)+kbcex()+theme()
   Scale for y is already present.
   Adding another scale for y, which will replace the existing scale.
```

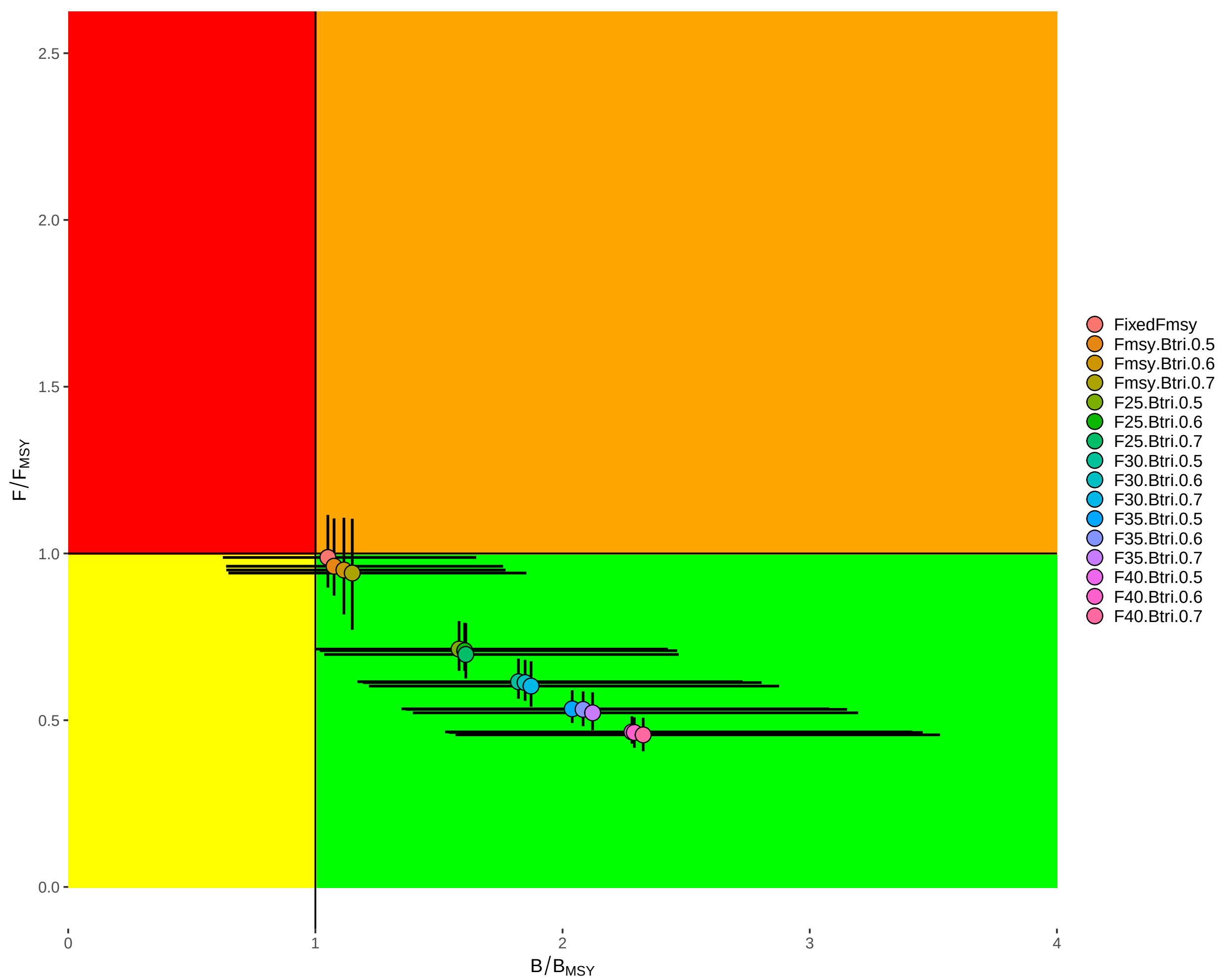

Figure 17: MSE kobe plot Advice rules