FLRef Demo: Reference Point estimation and visualization in FLR

Hennig Winker (JRC-European Commission)

03 May, 2022

Contents

1 Getting started

This vignette introduces the `FLRef` R package available on https://github.com/Henning-Winker/FLRef, as a support tool for estimating and visualizing reference points in `FLR`. Specific emphasis is put to enable routine plotting of a wider a of biological reference points (BRPs), such, as F_{spr40}, F_{B35} or $F_{0.1}$.

1.1 Installation

`FLRef` requires very recent versions of `FLR` libraries `FLCore`, `FLBRP`, `FLasher`, `mse`, `FLSRTBM` and `ggplotFL`. This can be installed together with `FLRef` from gihtub using library(devtools):

```
installed.packages("devtools")

installed.packages("ggplot2")

installed.packages("ggpubr")

installed.packages("TMB")


devtools::install_github("flr/FLCore")

devtools::install_github("flr/FLBRP")
```

```r
devtools::install_github("flr/FLasher")

devtools::install_github("flr/mse")

devtools::install_github("flr/ggplotFL")

devtools::install_github("flr/FLSRTMB")

devtools::install_github("henning-winker/FLRef")

# only for demo
install.packages("ggpubr")
```

Due to common challenges to set up Windows systems correctly for compiling C++ code with Rtools, working versions for `FLBRP`, `FLasher`, `mse` and `FLSRTMB` are also provided as binary package zip files here.

```r
library(FLCore)
library(FLBRP)
library(FLasher)
library(FLSRTMB)
library(ggplotFL)
library(FLRef)
library(ggpubr)  # For this demo
   Warning: package 'ggpubr' was built under R version 4.1.3
```

1.2 Example stock

The North Sea Plaice `FLStock` object `ple4` from `FLCore` used here as an example.

```r
data(ple4)
stk = ple4
```

```r
plot(stk) + theme_bw() + facet_wrap(~qname, scales = "free", ncol = 2)
```

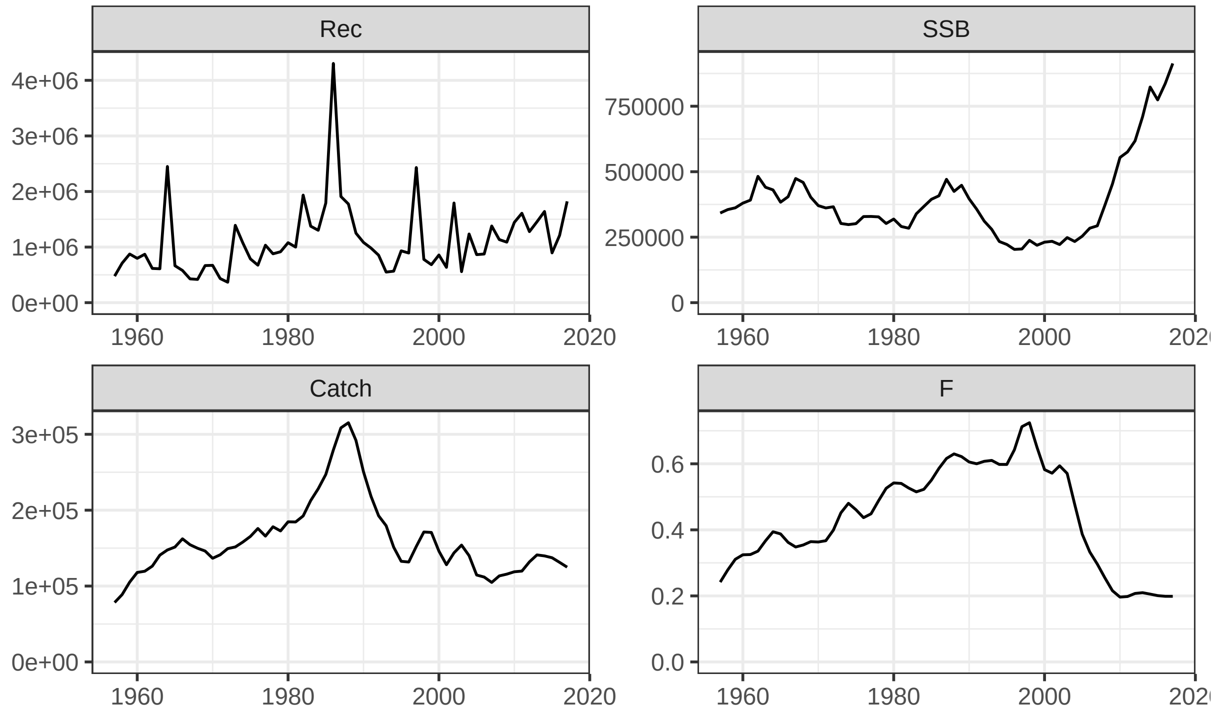

Figure 1: Estimated stock trajectories

2 Per-recruit reference points

Common proxies for F_{MSY} that do not neccessarly require a stock recruitment relationship are $F_{0.1}$ and $F_{SPR35-50}$, where SPR the spawning ratio potential expressed as spawning-biomass per-recruit relative to the unfished spawning biomass per-recruit at $F = 0$ (SPR_0). F_{SPR40} denotes a spawning-biomass per-recruit is reduced to 40 percent of SPR_0.

A range of these F_{BRP}'s can be computed quickly by:

```
fbrps = computeFbrps(stock = ple4, proxy = "sprx", f0.1 = TRUE, verbose = FALSE)
```

This range F_{BRP} values can easily visualised

```
ploteq(fbrps)
```

$Yield$ and SSB are in this case as yield- and spawning biomass per-recruit, respectively. $B0$ is the product of R_0 and SPR_0, where SPR_0 is a function of weight-at-age (w_a), maturity-at-age (mat_a) and natural mortality at age (M_a). Because R_0 is one (per-recruit), $B0$ equals SPR_0.
It is also possible to add some of the "default" reference points that are inbuilt in `FLBRP`.

```
ploteq(fbrps, refpts = "fmax")
```

A more targeted approach for exploring option of target an limit reference points is the function `computeFbrp()` (i.e. without 's). In the following example the F_{brp} is chosen to be $F_{0.1}$ and a B_{lim} proxy is chose so that is corresponds $0.25B_{F0.1}$.

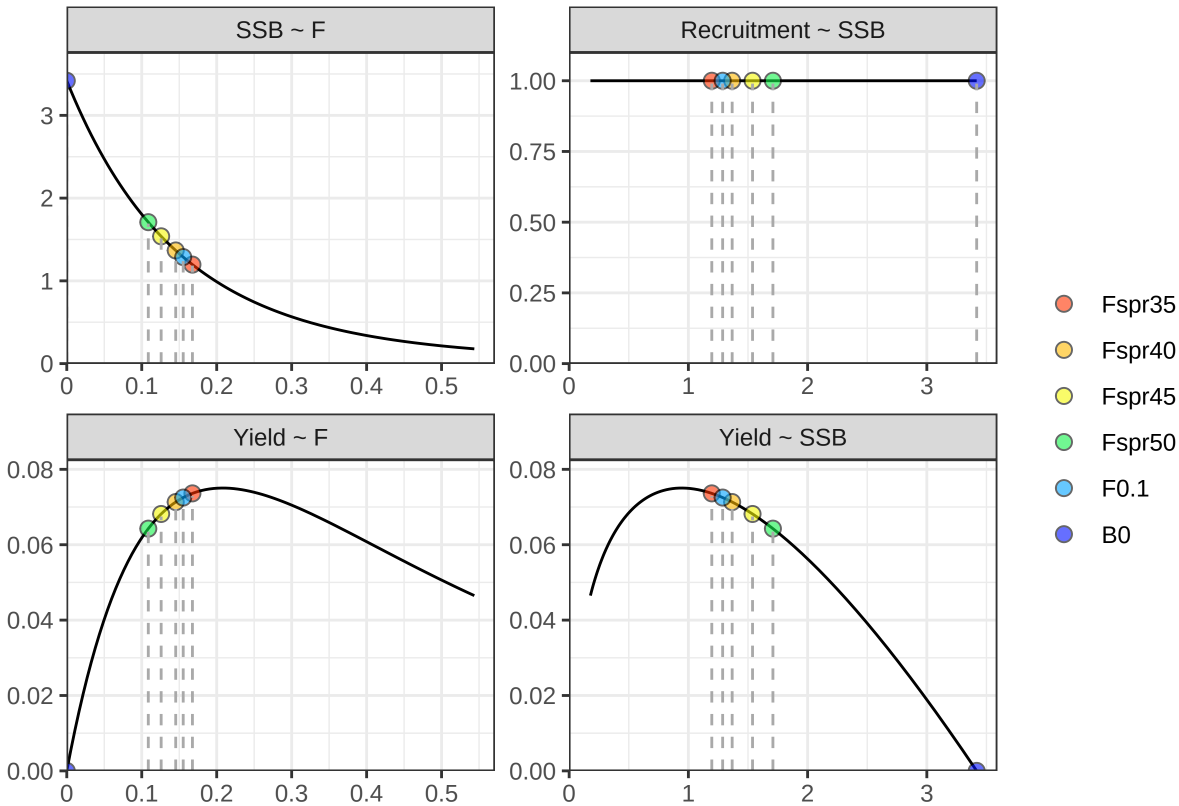

Figure 2: Estimated per-recruit reference points corresponding to the F_{BPR}'s $F_{spr35-50}$ and $F_{0.1}$.

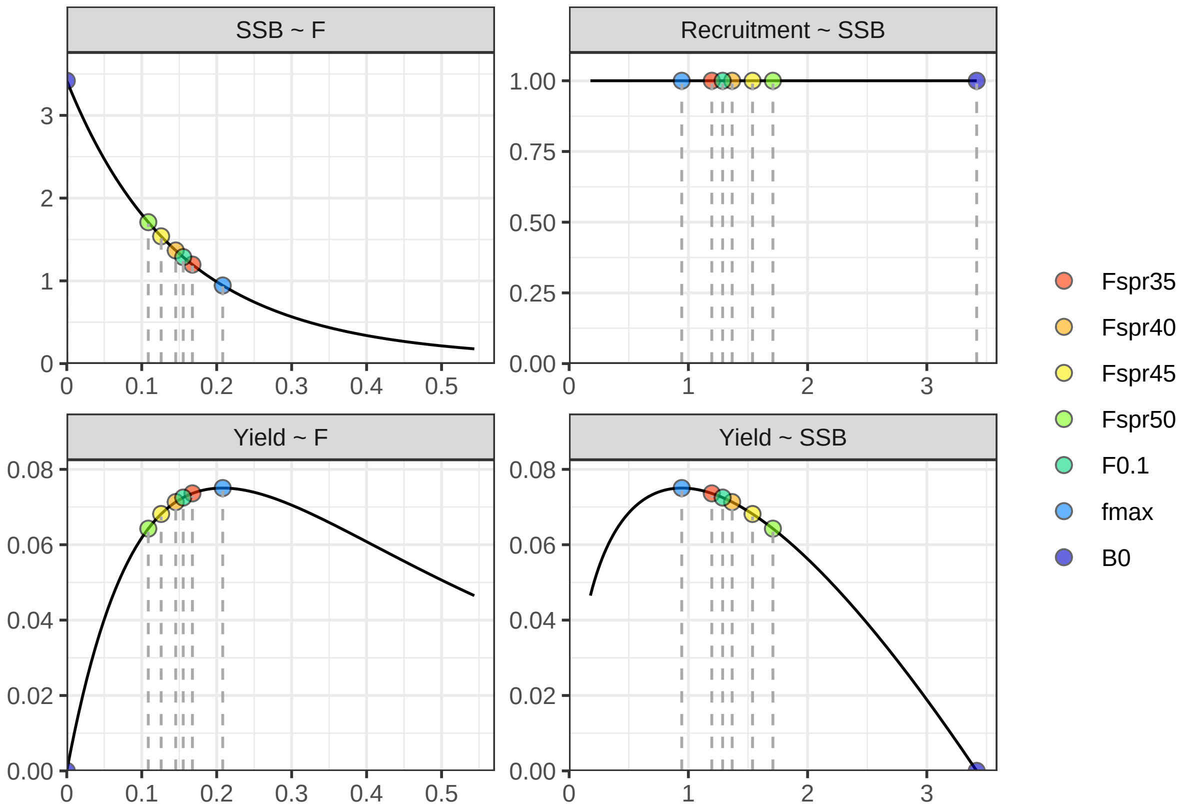

Figure 3: Estimated per-recruit reference points corresponding to the F_{BPR}, $F_{spr35-50}$ and $F_{0.1}$.

```
fbrp = computeFbrp(stock = stk, proxy = c("f0.1"), blim = 0.25, type = "btrg",
    verbose = FALSE)

Fbrp(fbrp)
   An object of class "FLPar"
   params
     F0.1   Btrg   Blim   Flim    Yeq     B0
   0.1553 1.2874 0.3219 0.4109 0.0725 3.4180
   units:  NA
```

It is also possible to add additional F_{BRP}. However, note that by convention the first in order of occurrence is used, e.g. to compute the ratio to approximate B_{lim}. It the example below B_{lim} is now computed as $0.25B_{spr40}$, i.e relative to the biomass per-recruit corresponding to F_{spr40}, spefied as `proxy = "sprx"` and `x=40`.

```
fbrp = computeFbrp(stock = stk, proxy = c("sprx", "f0.1"), x = 40, blim = 0.25,
    type = "btrg", verbose = FALSE)

Fbrp(fbrp)
   An object of class "FLPar"
   params
   Fspr40   Btrg   Blim   Flim    Yeq     B0
   0.1453 1.3672 0.3418 0.3985 0.0713 3.4180
   units:  NA
```

```
ploteq(fbrp, refpts = "fmax")
```

The `plotAdvice()` function provide can then be used to show the estimated stock trajectories per recruit relative to the reference points. To compute those from the `FLStock` object, the recruitment is normalised by its geometric mean which is a assumed to approximate R_0 (i.e. expected mean recruitment in the absence of a stock recruitment relationship). The estimate of spawning biomass per recruit is computed as $SB/R = SSB/R_0$ and then expressed as the Spawning Ratio Potential (SRP) relative to SPR_0. The "observed" yield per recruit is first computed as $Y/R = landings/R_0$ and then expressed as the ratio to the equilibrium Yield corresponding to F_{BRP}.

```
plotAdvice(stk, fbrp)
```

3 Integrating stock recruitment (S-R) functions into reference point computations

The simplest S-R model is assuming a that the expected recruitment is constant with R_0 estimated in the form of the geometric mean. This `geomean` can therefore be interpreted as Null model S-R functions. To set this up in `FLSRTMB`, it is only required to create a standard `FLSR` object as input to the function `srrTMB()`:

```
object = as.FLSR(stk, model = geomean)

gm = srrTMB(object)
```

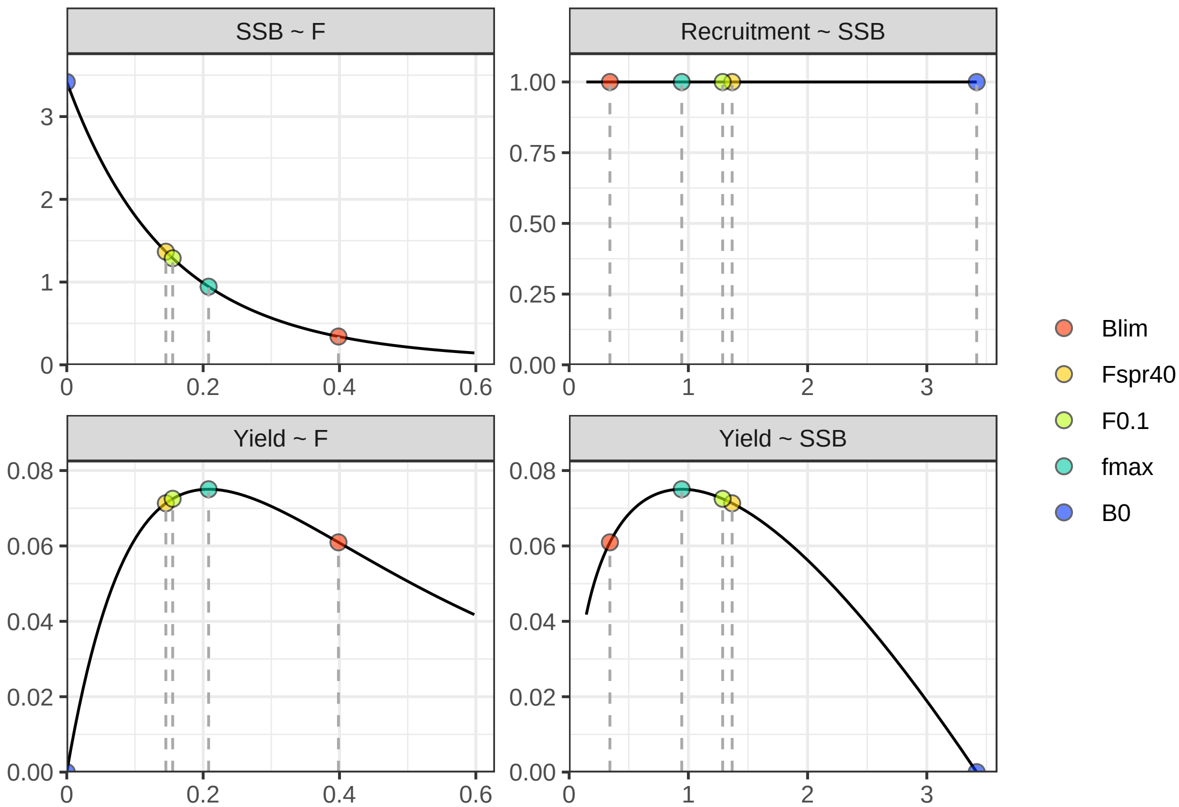

Figure 4: Estimated per-recruit reference points corresponding to the

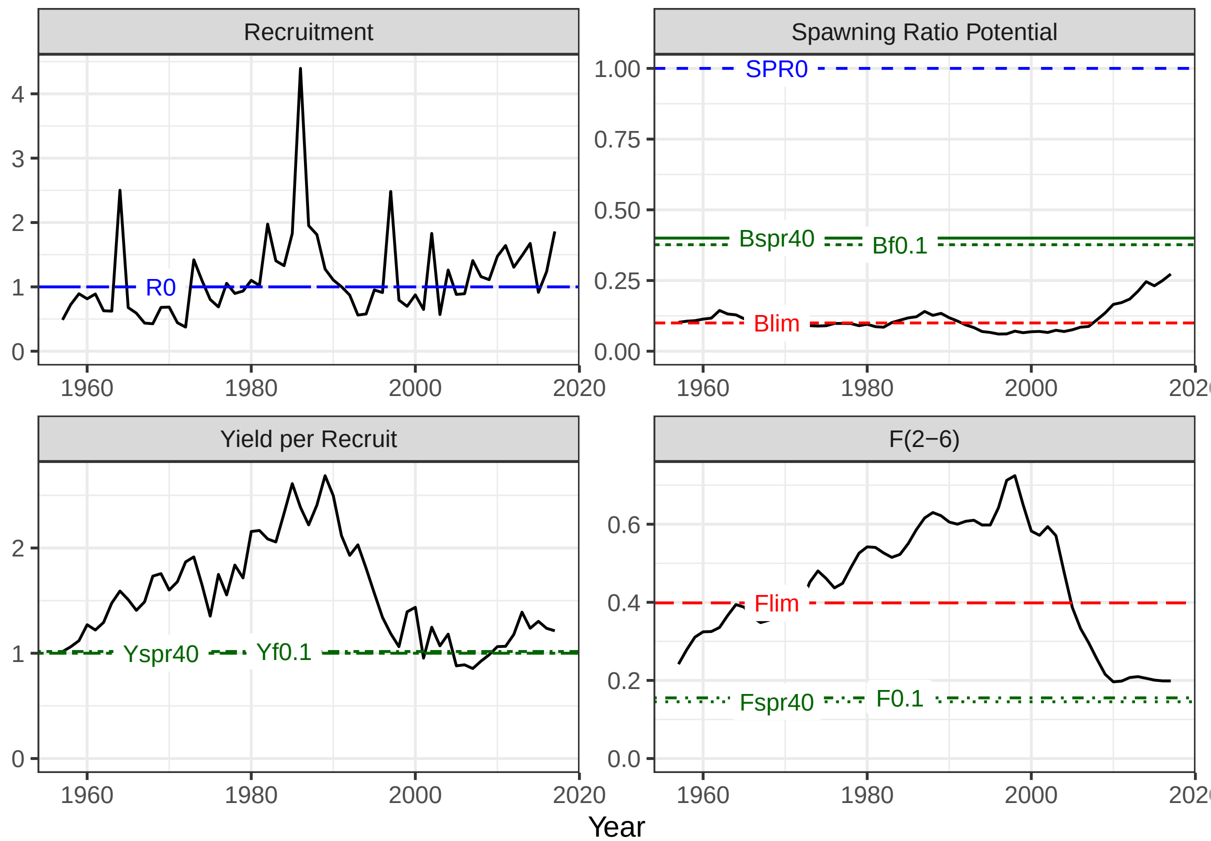

Figure 5: Stock advice plot showing the modelled quantities from a per-recruit perspestive relative to per-recruit based reference points

The reference points can now be re-calculated with `computeFbrp()` by simply specificying `sr=gm`, such that

```
fbrp = computeFbrp(stock = stk, sr = gm, proxy = c("sprx", "f0.1"), x = 40,
    blim = 0.25, type = "btrg", verbose = FALSE)
```

The only difference to the per-recruit representation is that that the reference points to recruitment, biomass and yield are now readily scaled by R_0 to the corresponding modelled quantities, which allows to add those for reference using the option `obs=TRUE`.

```
ploteq(fbrp, refpts = "fmax", obs = TRUE)
```

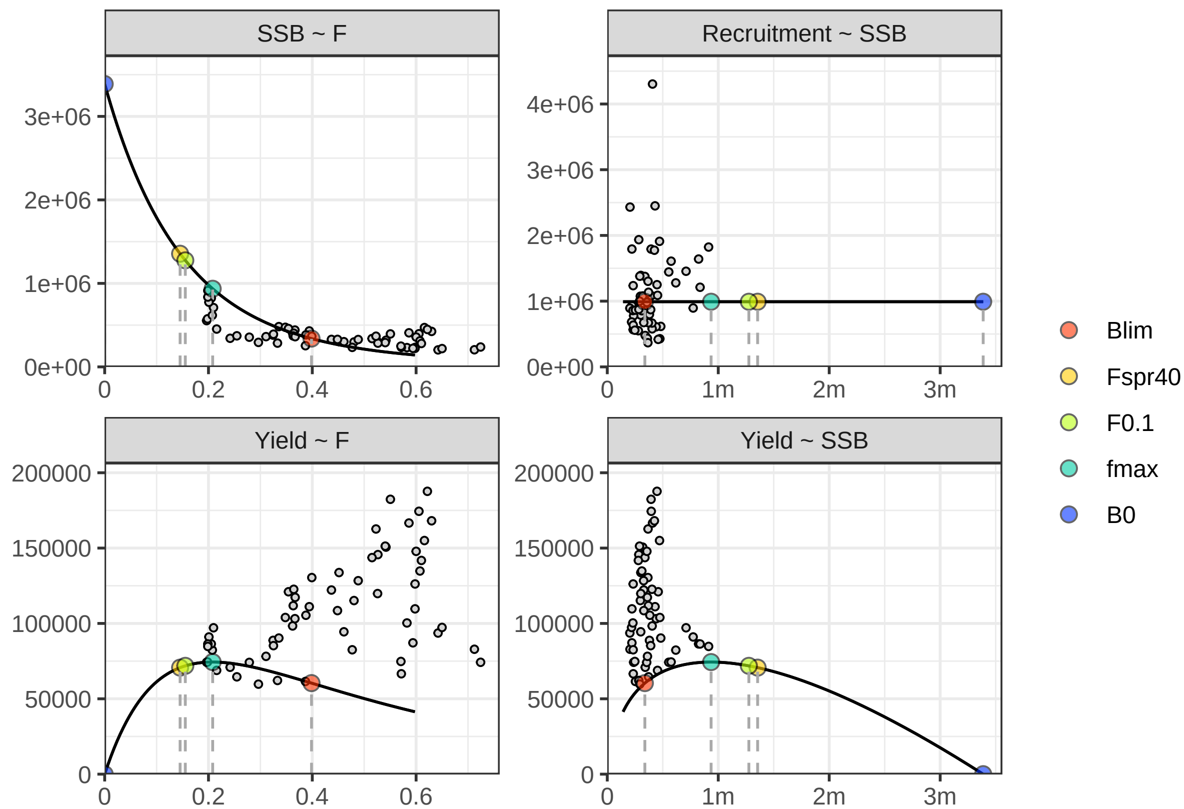

Figure 6: Estimated reference points relative to estimates of recruitment, SSB, F and landings

Similarly, the estimated time-series of recruitment, SSB, F and landings can now be directly compared to the reference points on absolute scale. Otherwise, the inference about the stock status remains the same as for the per-recruit analysis in the absensce a S-R relationship.

```
plotAdvice(stk, fbrp)
```

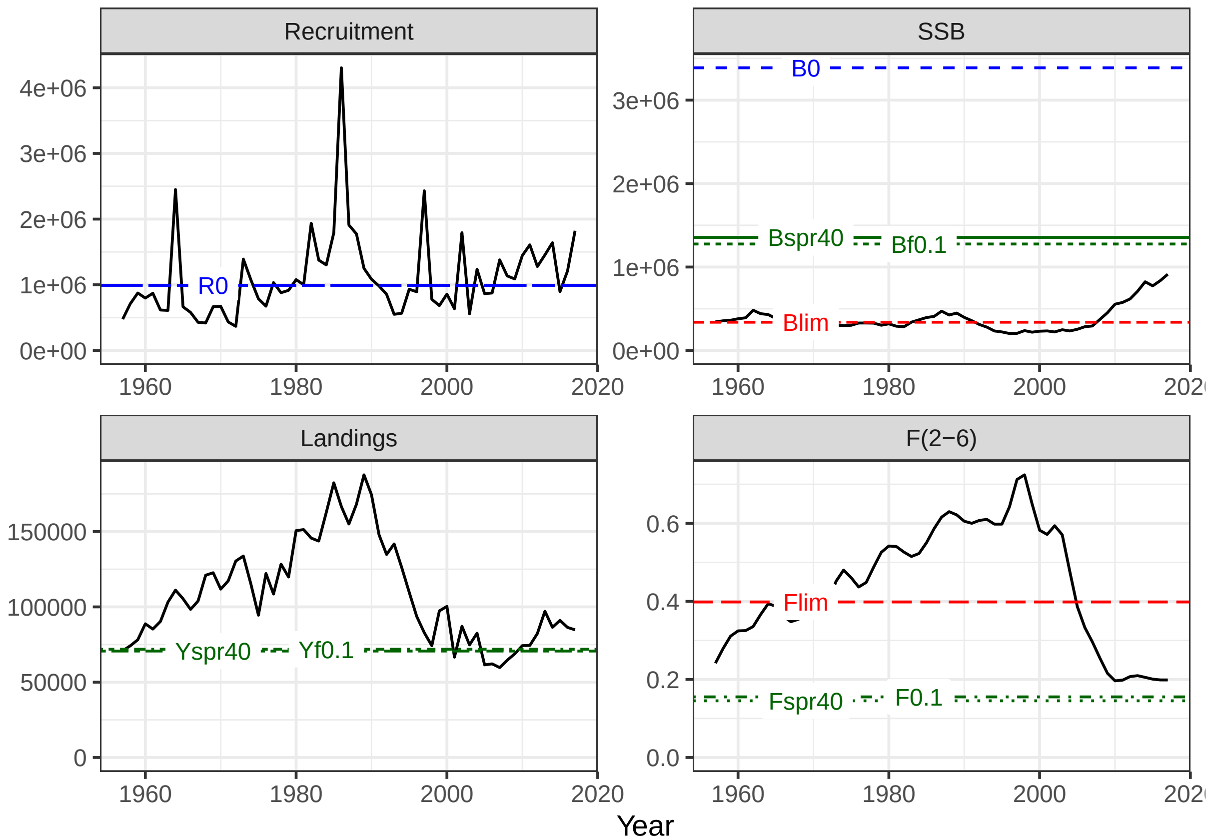

Figure 7: Stock advice plot showing modelled quantities and the corresponding reference points

The next step is to set fit alternative S-R functions with `srrTBM()`

The first one is a `model=bevholtSV` that is parameterised as a function of steepness s and SPR_0, which also requires to specify `spr0 = spr0y(stk)`, which computes the implicite values of SPR_0 in each year y as function of $w_{a,y}$, $mat_{a,y}$ and $M_{a,y}$. The estimates of s and R_0 are subsequently converted into the conventional `bevholt` parameter `a` and `b` given the mean SPR_0 for some reference years. For example, the default is use the geometric mean SPR_0 over the time-series whereas specifying `nyears=3` would use the geometric mean of SPR_0 over the 3 most recent years.

```
bh = srrTMB(as.FLSR(stk, model = bevholtSV), spr0 = spr0y(stk), verbose = FALSE)

bh@SV
          s    sigmaR      R0       rho      B0
  1 0.9632799 0.4766036 1144323 0.4089847 5824362
```

Calling bh@SV shows the maximum likelihood estimates of s, the recruitment standard deviation $sigmaR$, $R0$ and the post-hoc computed AR1 auto-correlation coeffecient rho.

Similarly, the Ricker model `model=rickerSV` is parameterised as a function of steepness s and SPR_0, but s is in this case not restricted to an bound at one to enable obtaining the same unconstraint fits as the equivalent a, b formulation of the model.

```
ri = srrTMB(as.FLSR(stk, model = rickerSV), spr0 = spr0y(stk), verbose = FALSE)
```

Finally, `FLSRTMB` also allows to fit a hockey-stick `model=segreg`, which is formulated as function of SPR_0. This formulation enables to invoke contraits for the location of the break-point. For example, by specifying `lplim=0.05` and `uplim=0.2` the location of the break-point $b = B_{lim}$ is constrained to fall between $0.05 - 0.2B_0$, which is in the specific case of the hockey-stick identical to the spawning ratio potential $SRP_{0.05-0.2}$.

```
hs = srrTMB(as.FLSR(stk, model = segreg), spr0 = spr0y(stk), lplim = 0.05,
    uplim = 0.2)
```

The three S-R fit can be summarised in single $FLSRs$ to enable a quick comparison with `plotsrs`.

```
srs = FLSRs(bh = bh, ri = ri, hs = hs)

plotsrs(srs)
```

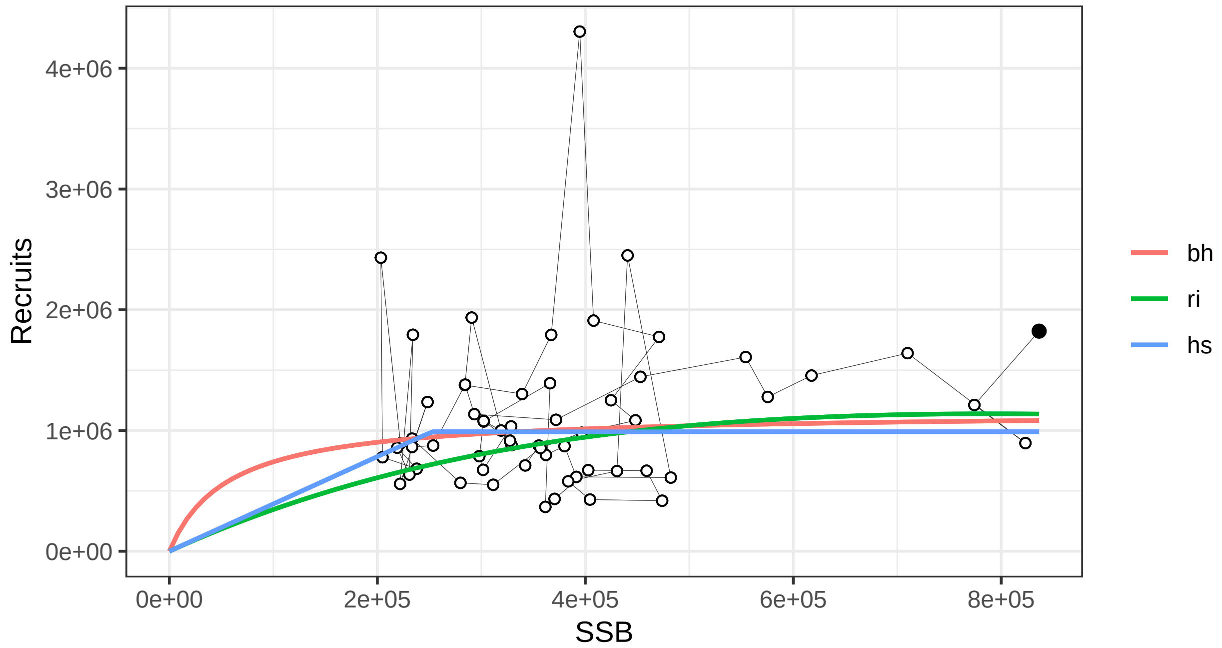

Figure 8: Comparison of the S-R relationship fitted by assuming Beverton-Holt (bh), Ricker (ri) and Hockey-Stick (hs) function. Open circles show the observed S-R pairs with the solid do denoting the final assessment year

Clearly, the hockey-stick fails to identify a clear break point in the data and therefore is located towards the lower specified bound, `lplim=0.05`.

```
hs@SV[["BlimB0"]]
   NULL
```

The stock shows a considerable variation and by providing a vector of `spr0=spr0y(stk)` the model effectly assumes time-varying SPR_{0_y} and thus B_{0_y}.

```
plot(spr0y(stk))+theme_bw()+
  ylab(expression(SPR[0]))+xlab("Year")+
  geom_hline(yintercept = mean(spr0y(stk)),linetype="dashed")
```

An alternative is to set SPR_0 to its mean or change the bounds *lplim* and *uplim*, which determine the "plausible" range of SRP. For this example, the lower limit of *lplim* is increase to 0.07.

```
hs1 = srs$hs
hs2 = srrTMB(as.FLSR(stk, model = segreg), spr0 = mean(spr0y(stk)), lplim = 0.05,
    uplim = 0.2)
hs3 = srrTMB(as.FLSR(stk, model = segreg), spr0 = mean(spr0y(stk)), lplim = 0.07,
    uplim = 0.2)

plotsrs(FLSRs(plim0.05 = hs1, muSPR0 = hs2, plim0.07 = hs3))
```

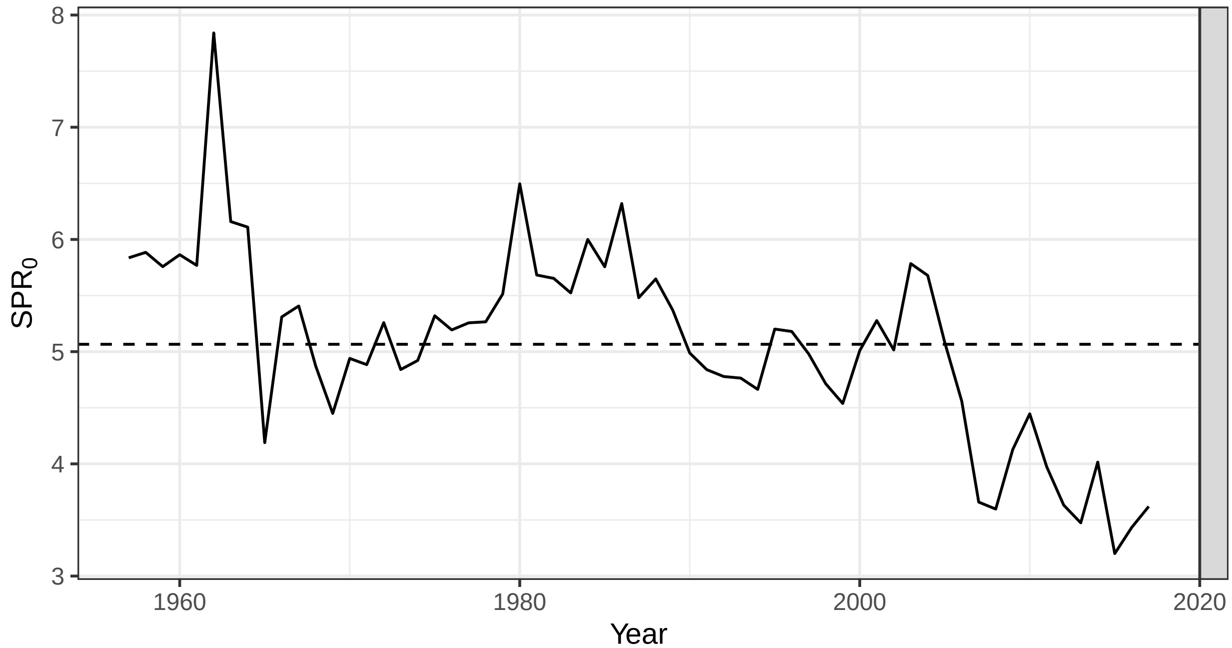

Figure 9: Annual SPR_{0_y} as function of time-varying w_{a_y} (here), mat_{a_y} and M_{a_y}.

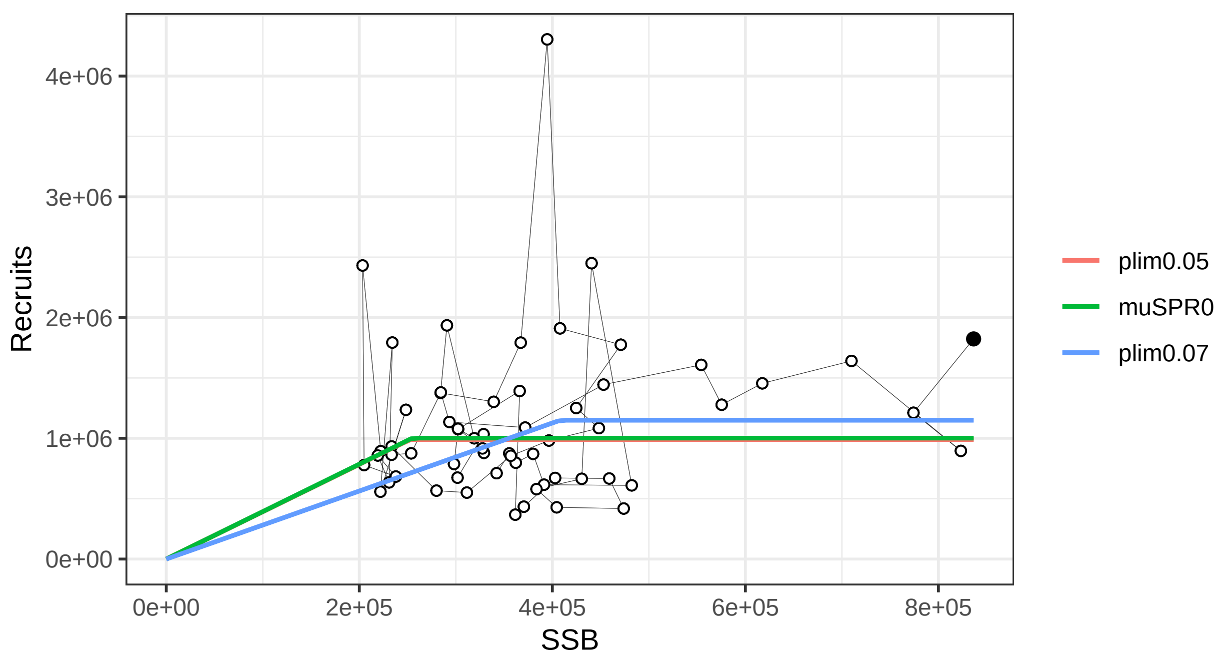

Figure 10: Comparison of the Hockey-Stick specified with (1) SRP_{5-20} and time-varying $SPR_{0,y}$, (2) the same but with the mean of $SPR_{0,y}$ (3) SRP_{7-20} and mean $SPR_{0,y}$.

Here models (1) `plim0.05` (2) `muSPR0` produce the same results. In option (3) the break-point is still located close to $plim = 0.07$. Therefore, the data hold no information about a break-point and the choice of "plausible" SPR_0 specification (mean vs time-varying) and the SRP bounds determine the estimate of the break-point. For this demo, option (2) is used instead of (1) for subsequent illustrations.

```
hsblim(hs1)
   An object of class "FLPar"
   params
       Blim       R0       B0   SRPlim
   2.53e+05 9.90e+05 5.04e+06 5.02e-02
   units:  NA
hsblim(hs2)
   An object of class "FLPar"
   params
       Blim       R0       B0   SRPlim
   2.55e+05 1.00e+06 5.07e+06 5.03e-02
   units:  NA
hsblim(hs3)
   An object of class "FLPar"
   params
       Blim       R0       B0   SRPlim
   4.08e+05 1.15e+06 5.82e+06 7.01e-02
   units:  NA
```

```
# Extract Blim
blim = c(params(hs)["b"])
# check break-point relative to B0
hsblim(hs)["SRPlim"]
   An object of class "FLPar"
   params
   SRPlim
   0.0502
   units:  NA
hsblim(hs3)["SRPlim"]
   An object of class "FLPar"
   params
   SRPlim
   0.0701
   units:  NA
```

The function `plotsrs` provides following options to illustrate the S-R:

- no S-R observations `path=FALSE`
- with S-R observations `path=TRUE`
- Projected through to $B_0 = R_0 SPR_0$
- Relative to SSB_0 and R_0 (permist comparson accross stocks)

```
p1 = plotsrs(srs, path = FALSE)
p2 = plotsrs(srs, path = TRUE)
p3 = plotsrs(srs, b0 = TRUE)
```

```
p4 = plotsrs(srs, b0 = TRUE, rel = TRUE)

ggarrange(p1, p2, p3, p4, ncol = 2, nrow = 2, common.legend = TRUE, legend = "right")
```

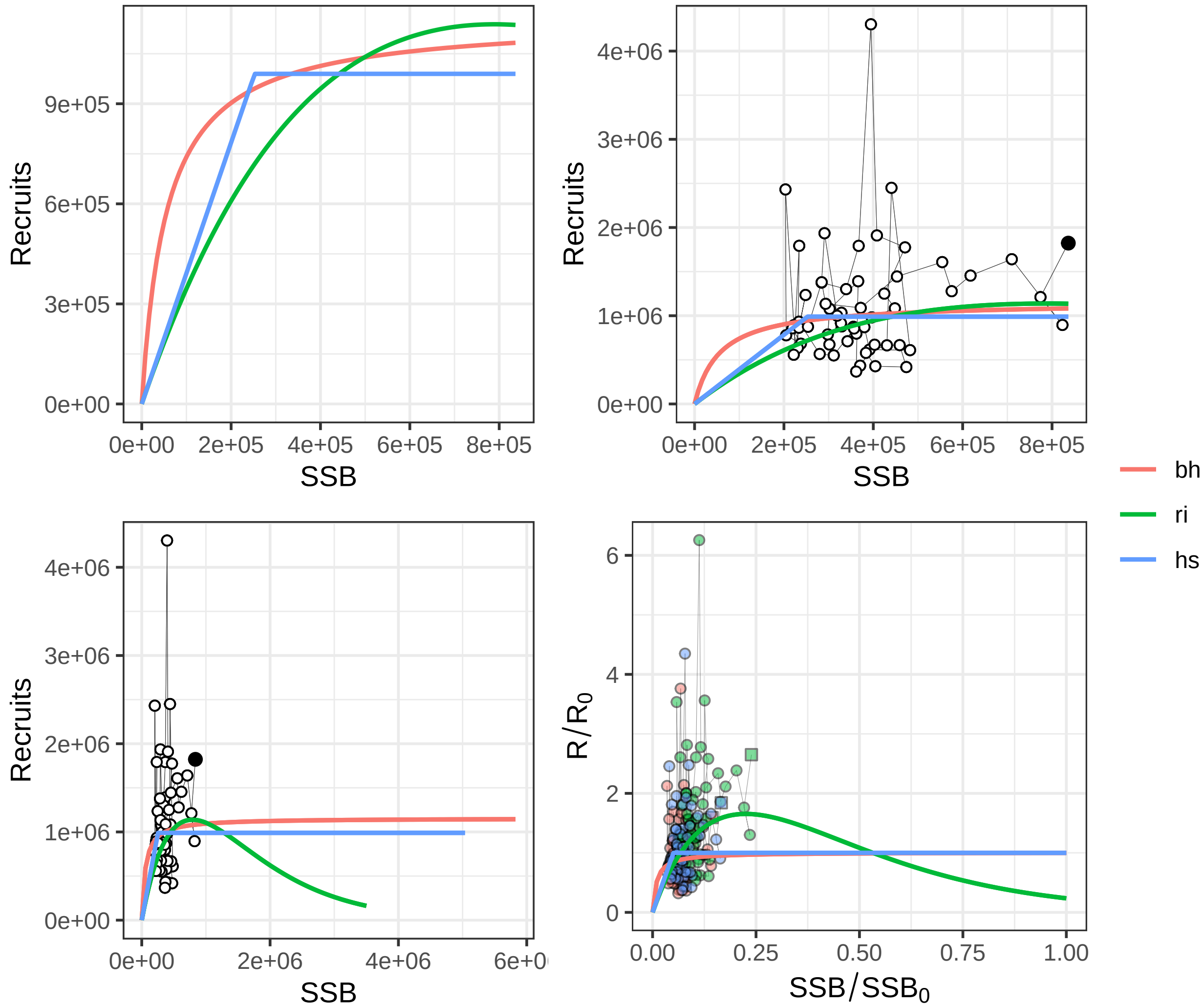

Figure 11: Comparison of the S-R relationship fitted by assuming Beverton-Holt (bh), Ricker (ri) and Hockey-Stick (hs) function.

Similar to `FLSRs`,the `computeFbrp` output in the form `FLBRP` objects can also be compiled in `FLBRS` to enable comparison. Note that in the case of the hockey-stick its breakpoint is used directly as input of an absolute value for blim, using the option `type="value"`.

```
brps = FLBRPs(bh = computeFbrp(stk, sr = srs[["bh"]], proxy = c("f0.1",
    "sprx", "msy"), x = 40, blim = 0.25, type = "btrg", verbose = FALSE),
    ri = computeFbrp(stk, sr = srs[["ri"]], proxy = c("f0.1", "sprx", "msy"),
        x = 40, blim = 0.25, type = "btrg", verbose = FALSE), hs = computeFbrp(stk,
        sr = srs[["hs"]], proxy = c("f0.1", "sprx", "msy"), x = 40, blim = blim,
        type = "value", verbose = FALSE))
```

```r
# plot
ploteq(brps)
```

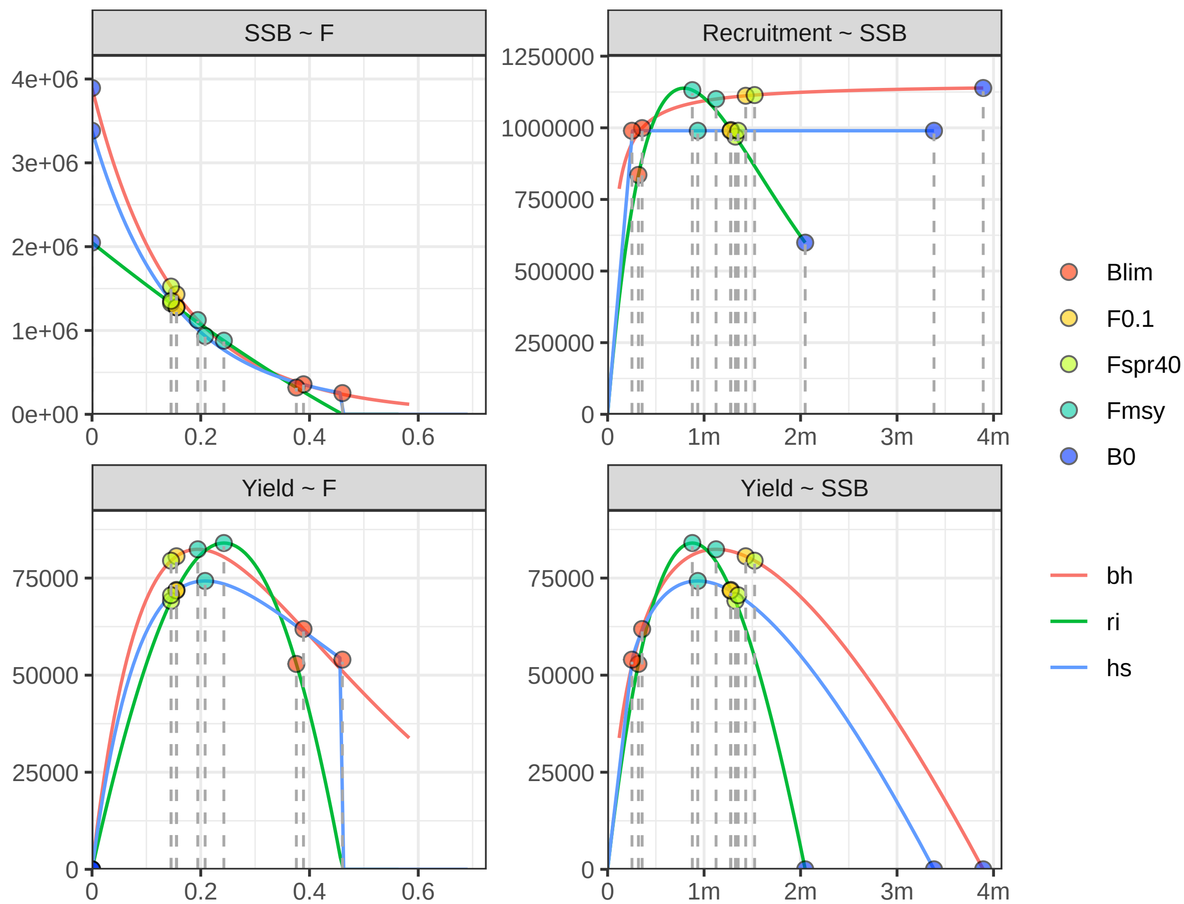

Figure 12: Estimated reference points at equilibrium recruitment, SSB, F and landings

The same plot can be produce with estimates from the assessment estimates.

```
# plot
ploteq(brps, obs = TRUE)
```

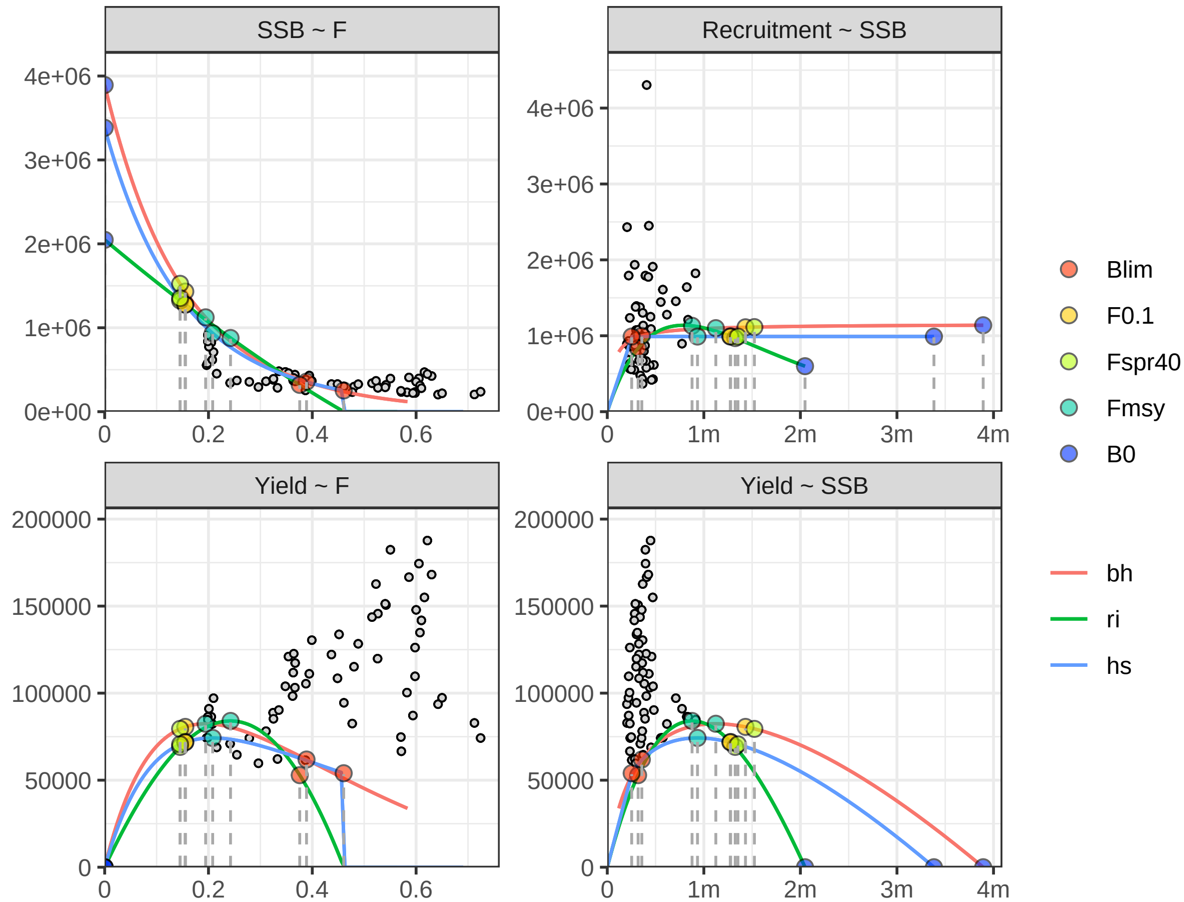

Figure 13: Estimated reference points relative to estimates of recruitment, SSB, F and landings

`FLSRTMB` provides also the option fix s or use informative priors, such as those that can be derived from `FishLife`; Thorson (2020). This can be done

```
# Fixed steepness
s = c(seq(0.8, 0.95, 0.05))
fixs = FLSRs(lapply(as.list(s), function(x) {
    srrTMB(as.FLSR(stk, model = bevholtSV), spr0 = spr0y(stk), s = x, s.est = FALSE)
}))
names(fixs) = paste0("s=", s)
# with prior with mean s=0.85 and s.logitsd = 0.3
s.pr = srrTMB(as.FLSR(stk, model = bevholtSV), spr0 = spr0y(stk), s = 0.8,
    s.logitsd = 0.3)
# uncontrained estimate
s.est = srrTMB(as.FLSR(stk, model = bevholtSV), spr0 = spr0y(stk), s = 0.8)
```

```
# combine
bhs = FLSRs(c(s.est = s.est, s.pr = s.pr, fixs))
# add s estimate
names(bhs)[1:2] = c(paste0("s.est(", round(s.est@SV[["s"]], 2), ")"), paste0("s.pr(",
    round(s.pr@SV[["s"]], 2), ")"))

plotsrs(bhs)
```

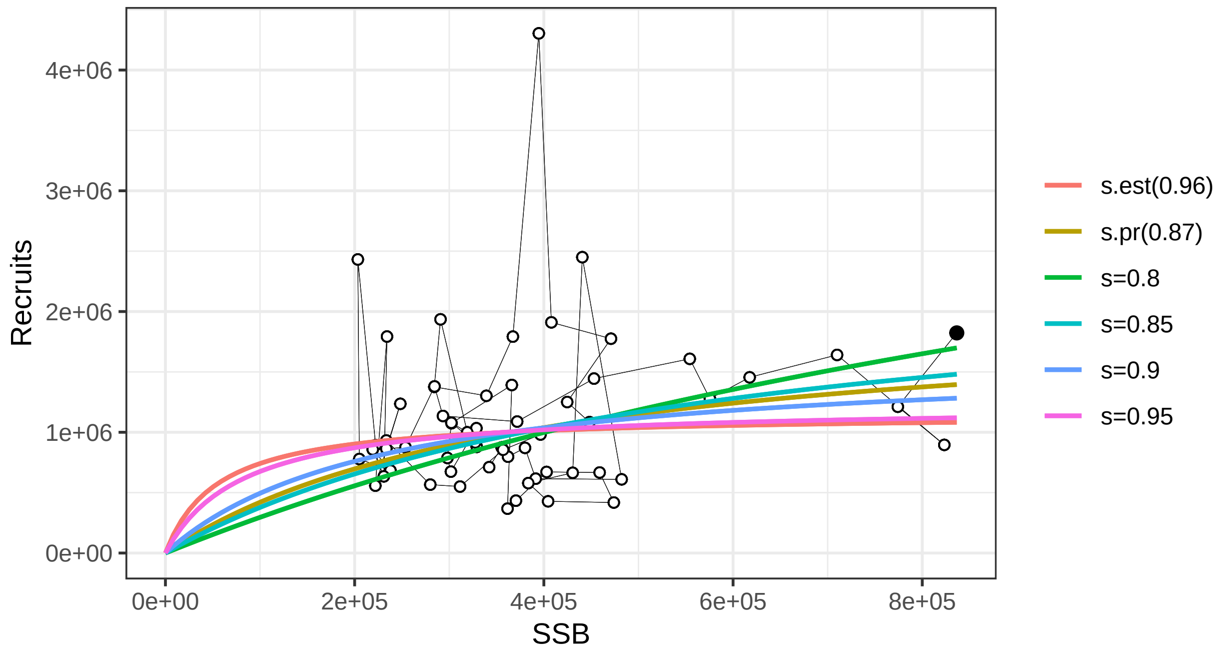

Figure 14: Comparison of alternative parameterisation of the Beverton Holt S-R

```
bh.brps = FLBRPs(lapply(bhs, function(x) {
    computeFbrp(stk, x, proxy = c("f0.1", "msy"), blim = 0.25, type = "btrg",
        verbose = FALSE)
}))

ploteq(bh.brps, obs = TRUE, panels = 4)
```

```
plotAdvice(stk, bh.brps[[1]]) + ggtitle(paste0(stk@name, ": BevHolt with s = ",
    round(bhs[[1]]@SV[["s"]], 3)))
```

Another option to illustrate the stock status against the reference point estimates is the "Advice Rule" plot `plotAR()`. For the variety option please see the available examples `?plotAR`. Here we consider 4 options of illustration: (1) Basic plot with a precautionary biomass B_{pa} add that expressed relative $B_(lim)$, adding a $B_{trigger}$ as fraction of the target Biomass reference point $Btrg$, (3) using kobe type color-coding with de facto fishing closure at $Blim$ and (4) showing the quatative relative to the targer reference points. The input can be either the output of `Fbrp()` (easy to manipulate) or the `FLBRP` output from `computeFbrp()`.

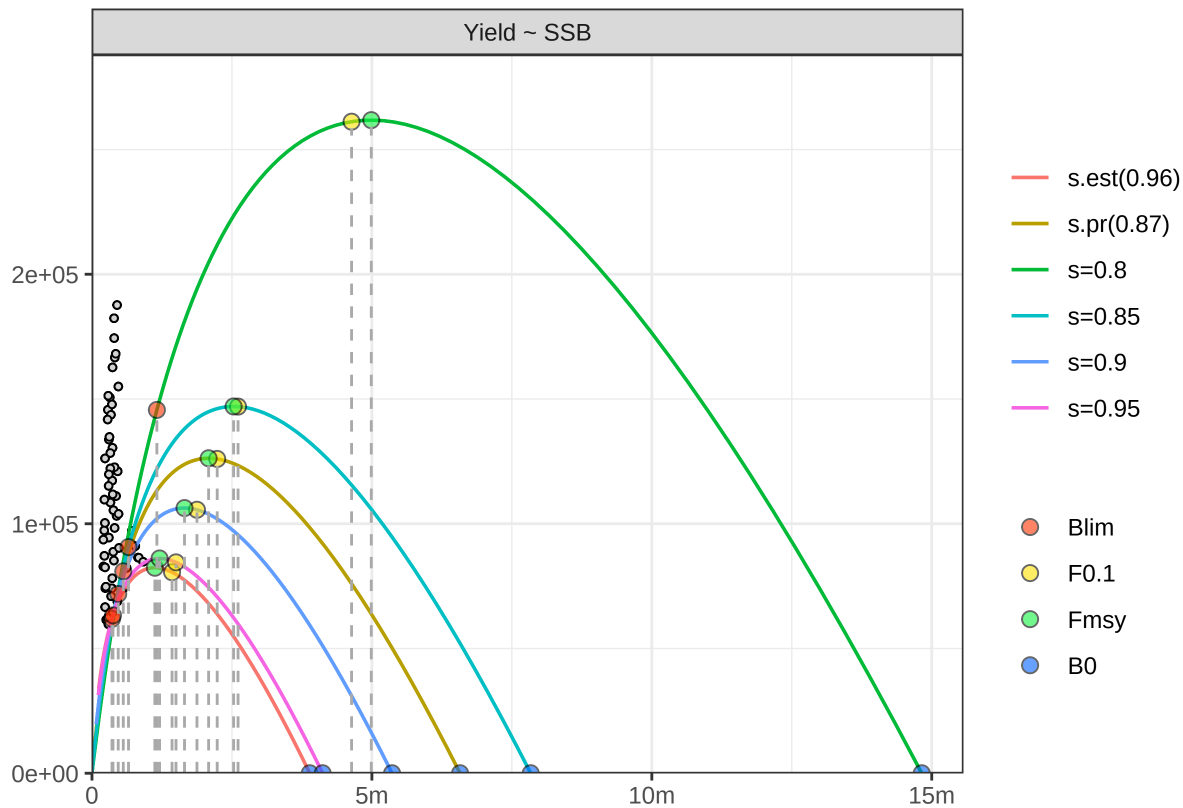

Figure 15: Comparison of equilibriun curves and reference points for alternative parameterisation of the Beverton Holt S-R

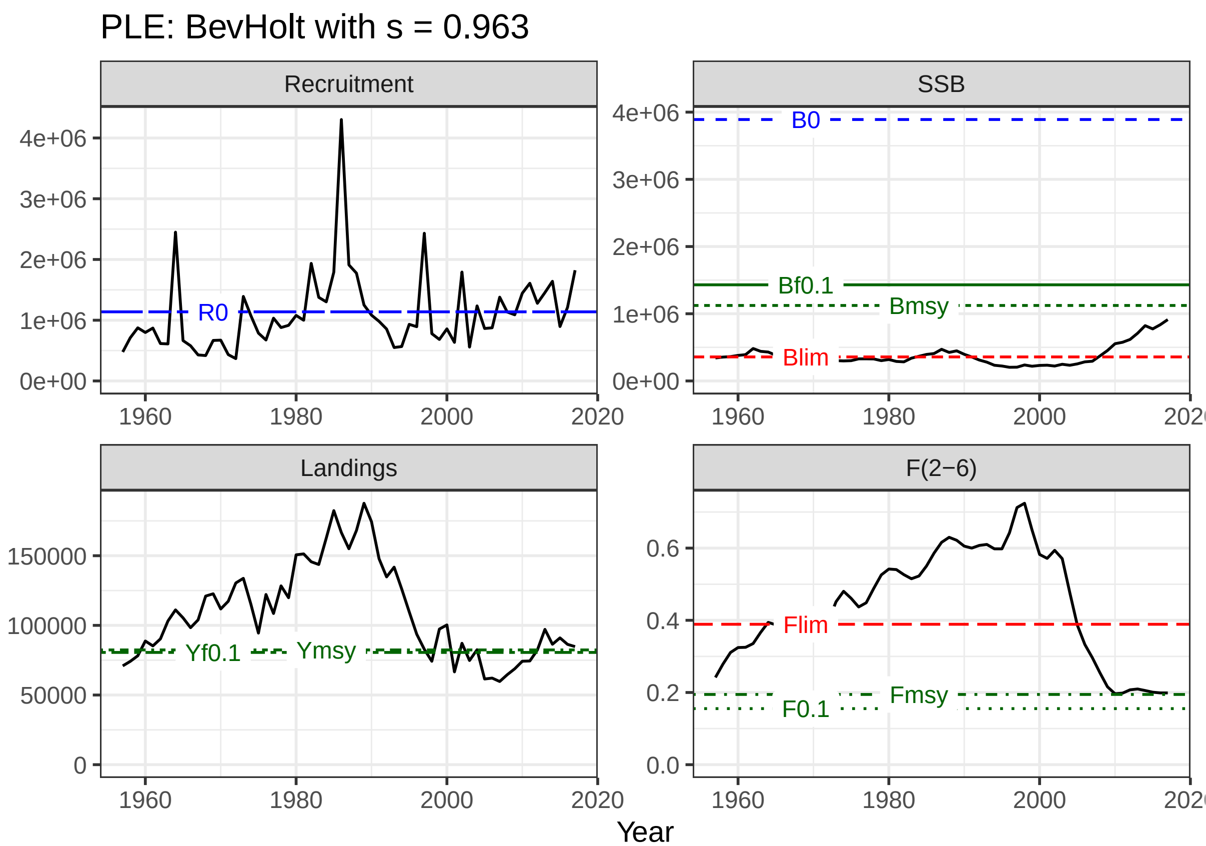

Figure 16: Stock advice plot showing modelled quantities and the corresponding reference points for a Beverton S-R model with estimated s

```
pars = Fbrp(bh.brps[[1]])
pars
   An object of class "FLPar"
   params
       F0.1      Btrg      Blim      Flim       Yeq       B0
   1.55e-01 1.43e+06 3.58e+05 3.89e-01 8.06e+04 3.89e+06
   units:  NA
p1 = plotAR(bh.brps[[1]], obs = stk, kobe = FALSE, bpa = 1.4)
p2 = plotAR(bh.brps[[1]], obs = stk, kobe = FALSE, bpa = 1.4, btrigger = 0.7)
p3 = plotAR(bh.brps[[1]], obs = stk, kobe = TRUE, bpa = 1.4, btrigger = 0.7,
    bclose = 1, fmin = 0.01)
p4 = plotAR(bh.brps[[1]], obs = stk, kobe = TRUE, bpa = 1.4, btrigger = 0.7,
    rel = TRUE)

ggarrange(p1, p2, p3, p4, ncol = 2, nrow = 2)
```

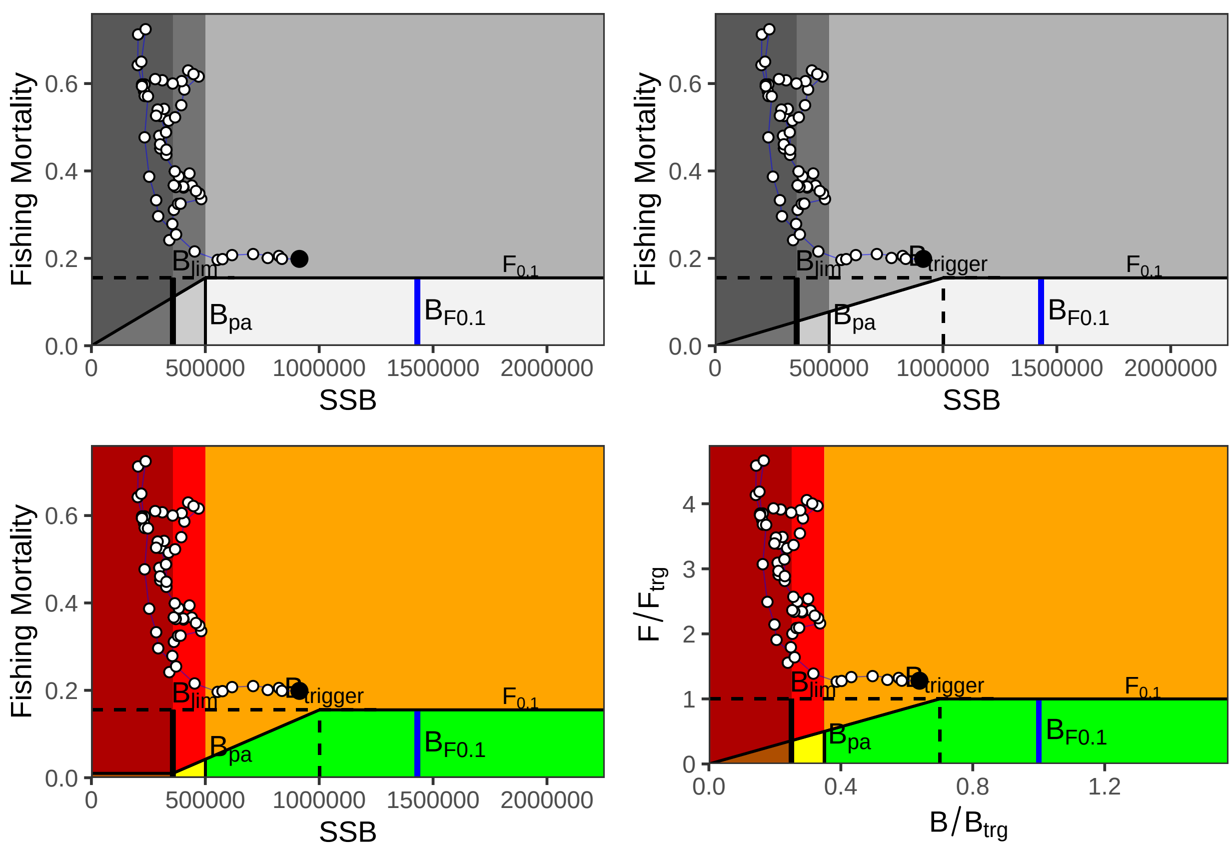

Figure 17: Stock advice plot showing modelled quantities and the corresponding reference points for a Beverton S-R model with estimated s

4 TO BE FINISHED