A Simple Example in FLBEIA

Marga Andres and FLBEIA team

Aim

FLBEIA provides a battery of tutorials for learning how to use this software. This tutorial of `FLBEIA` is a practical guide about one of the simplest possible implementation of `FLBEIA`. In this tutorial a simple example, named *one*, is run in `FLBEIA`. Then, the outputs of `FLBEIA` are explored, summarized and plotted. Once the user has understood the structure and outputs of `FLBEIA`, let's start playing! Several scenarios are run changing and adjusting several funcionalities that **FLBEIA** provides. Finally, another simple example but with three iterations it is introduced. In this last example some exercises are proposed to be resolved by the user.

Required packages to run this tutorial

To follow this tutorial you should have installed the following packages

- CRAN: ggplot2
- FLR: FLCore, FLAssess, FLash,FLBEIA, FLFleet, ggplotFL

If you are using Windows, please use 32-bit R version because some of the packages do not work in 64-bit.

```
install.packages(c("ggplot2"))
install.packages(c("FLCore", "FLBEIA", "FLFleets", "FLash", "FLAssess", "FLXSA", "ggplotFL"), repos="htt
```

Load all necessary packages.

```
# This chunk loads all necessary packages.
library(FLBEIA)
library(FLXSA)
library(FLash)
library(ggplotFL)
#library(FLBEIAShiny) # This is a beta version
```

EXAMPLE 1: single stock, single fleet, single season and one iteration.

Description

This example is one of the simplest possible implementation of **FLBEIA**. The Operating Model (OM) is formed by a single age-structured stock. Only one fleet which activity is performed in an unique metier and the time step is annual. The historic data go from 1990 to 2009 and projection period from 2010 to 2025.

- Operating model:

  - Population dynamics: Age structured population growth
  - SR model: Beverthon and Holt autoregressive/segmented regression
- Management Procedure:
  - Perfect observation
  - No assessment

Exploring the data

The neccesary `FLR` objects to run `FLBEIA` are available in the dataset called `one`.

```
data(one)
```

With the `ls()` command we can see the objects stored in `one`, which are those needed to call to run `FLBEIA`.

```
ls()
```

```
 [1] "fign"                "fig_nums"
 [3] "knit_print.FLQuant" "oneAdv"
 [5] "oneAdvC"            "oneAssC"
 [7] "oneBio"             "oneBioC"
 [9] "oneCv"              "oneCvC"
[11] "oneFl"              "oneFlC"
[13] "oneIndAge"          "oneIndBio"
[15] "oneMainC"           "oneObsC"
[17] "oneObsCIndAge"      "oneObsCIndBio"
[19] "oneSR"
```

```
# Show the class of each of the objects.
sapply(ls(), function(x) class(get(x)))
```

```
              fign           fig_nums
        "function"         "function"
knit_print.FLQuant             oneAdv
        "function"             "list"
           oneAdvC            oneAssC
            "list"             "list"
            oneBio            oneBioC
          "FLBiols"            "list"
             oneCv             oneCvC
            "list"             "list"
             oneFl             oneFlC
     "FLFleetsExt"             "list"
         oneIndAge          oneIndBio
            "list"             "list"
          oneMainC            oneObsC
            "list"             "list"
     oneObsCIndAge      oneObsCIndBio
            "list"             "list"
             oneSR
            "list"
```

The exact way to define the objects used to set the simulation is described in the **FLBEIA** manual. The manual can be download from Manual, within the 'doc' folder of the package installation or typing `help(package = FLBEIA)` in the R console. The description of objects stored in `one` is detailed below.

- `oneBio`: A FLBiols object that contains biological information of *stk1* relative to 'real' biological population within the OM.

- `oneBioC`: A list that controls the biological operating model. The growth function of stk1 is defined as Age Structured Populatin Growth `ASPG`. The function `ASPG` describes the evolution of an age structured population using an exponential survival equation for existing age classes and a stock-recruitment relationship to generate the recruitment. In FLBEIA there are other models available: `fixedPopulation` and `BDPG`.
- `oneMainC`: Controls the behaviour of FLBEIA function setting the initial and final year of the proyected period.
- `oneFL`: It is a FLFleetExt object that contains information relative to the fleet `_fl1_` (effort, costs, capacity…) and information relative to the metier `_met1_` (effort share, varaible costs, parameters of the Cobb Douglas function).
- `oneFlC`: The basic structure of the fleets.ctrl object; the effort dynamics is Simple Mixed Fisheries Behaviour `SMFB`. `SMFB` is a simplified version of the behavior of fleets that work in a mixed fisheries framework. The function is seasonal and assumes that effort share among metiers is given as input parameter. There are also other effort models defined in FLBEIA [`fixedEffort`, `SSFB`, `MaxProfit`, `MaxProfitSeq`]. See the Manual for detailed description. In this example the capacity, catchability and price are given as input data and are unchanged within the simulation. Summarizing, the fleet catches yearly exactly the adviced TAC and there is no exit-entry of vessels in the fishery.
- `oneCv`: All variables needed for the capital dynamics are stored (fuel costs, capital costs, number of vessels, invest share, number of vessels, maximum fishing days by vessel…). The capital model is defined in `oneFlC` as `fixedCapital`.
- `oneCvC`: Control list for the capital dynamics. All slots of this element are `fixedCovar`, i.e, covariates are given as input data and are unchanged within the simulation.
- `oneAdv`: List containing information on management advice; total allowable catch 'TAC' and quota share.
- `oneAdvC`: Controls the behaviour of advice management procedure. It is a list with one element per stock and one more element for each fleet. The selected model is `IcesHCR`. The function represents the HCR used by ICES to generate TAC advice in the MSY framework. It is a biomass based HCR, where the TAC advice depends on F in relation to several reference points.
- `oneAssC`: Defines the name of the assessment model to be used for each stock. In the current example there is no asssessment `NoAssessment`.
- `oneObsC`: These type of models are useful when no assessment model is used in the next step of the MP and management advice is just based on the population 'observed' in this step. A `perfecObs` model is defined. This function does not introduce any observation uncertainty in the observation of the different quantities stored in the `FLStock` or `FLFLeetsExt` objects.
- `oneSR`: The stock recruitment relationship, not needed in the case of population aggregated in biomass. The model is `bevholtAR1`, a regression stock recruitment model with autoregressive normal log residuals of first order. There are several SR models available in FLR and FLBEIA: `geomean`, `bevholt`, `ricker`, `segreg`, `shepherd`, `rickerAR1`, `segregAR1`, `cushing`, `bevholtSV`, `rickerSV`, `segregSV`, `shepherdSV`, `cushingSV`, `rickerCa`, `hockstick`, `redfishRecModel` and `ctRec`. All these models are detailed in the Manual.

In the code below we call to `FLBEIA` with the arguments stored in `one` dataset. It takes less than a minute to run up to 2025.

Run FLBEIA

```
s0 <- FLBEIA(biols = oneBio,        # FLBiols object with one FLBiol element for stk1.
               SRs = oneSR,         # A list with one FLSRSim object for stk1.
```

```
                BDs = NULL,          # No Biomass dynamics populations in this case.
             fleets = oneFl,         # FLFleets object with one fleet.
             covars = oneCv,         # Covar is and object with aditional data on fleet (number of vessel
            indices = NULL,          # Indices not used
             advice = oneAdv,        # A list with two elements 'TAC' and 'quota.share'
          main.ctrl = oneMainC,      # A list with one element to define the start and end of the simulat
         biols.ctrl = oneBioC,       # A list with one element to select the model to simulate the stock
        fleets.ctrl = oneFlC,        # A list with several elements to select fleet dynamic models and st
        covars.ctrl = oneCvC,        # A list with several data related to the fleet.
           obs.ctrl = oneObsC,       # A list with one element to define how the stock observed ("Perfect
        assess.ctrl = oneAssC,       # A list with one element to define how the stock assessment model u
        advice.ctrl = oneAdvC)       # A list with one element to define how the TAC advice is obtained (
```

FLBEIA returns a list with several objects, let's print the names of the objects and its class.

```
names(s0)
```

```
[1] "biols"         "fleets"
[3] "covars"        "advice"
[5] "stocks"        "indices"
[7] "fleets.ctrl"   "pkgs.versions"
```

```
sapply(s0, function(x) class(x))
```

```
        biols        fleets        covars
    "FLBiols" "FLFleetsExt"        "list"
       advice        stocks       indices
       "list"        "list"        "NULL"
  fleets.ctrl pkgs.versions
       "list"      "matrix"
```

Summarizing results

FLBEIA has several functions to summaryze the information in data frames. These data frames will allow to use methods available in R to analyze the results visually and numerically. You can summaryze information in two formats.

- Long format: where all the indicators are in the same column. There is one column, indicator, for the name of the indicator and a second one for the numeric value of the indicator.
- Wide format: where each column corresponds with one indicator.

The long format it is recommendable to work with ggplot2 functions, while the wide format is more efficient for memory allocation and speed of computations.

Long format.

```
s0_bio    <- bioSum(s0)           # Data frame (DF) of biological indicators.
s0_adv    <- advSum(s0)           # DF of management advice (TAC).
s0_flt    <- fltSum(s0)           # DF of economic indicators at fleet level.
s0_fltStk <- fltStkSum(s0)        # DF of indicators at fleet and stock level.
s0_mt     <- mtSum(s0)            # DF of indicators at fleet.
s0_mtStk  <- mtStkSum(s0)         # DF of indicators at fleet and metier level.
s0_vessel <- vesselSum(s0)        # DF of indicators at vessel level.
s0_vesselStk <- vesselStkSum(s0)  # DF of indicators at vessel and stock level.
```

```r
s0_npv  <- npv(s0, y0 = '2014')   # DF of net present value per fleet over the selected range of years.
s0_risk <- riskSum(s0, Bpa = c(stk1= 135000), Blim = c(stk1= 96000), Prflim = c(fl1 = 0))

# Exploring data frames
head(s0_bio); unique(s0_bio$indicator)
head(s0_adv); unique(s0_adv$indicator)
head(s0_flt); unique(s0_flt$indicator)
head(s0_fltStk); unique(s0_fltStk$indicator)
head(s0_mt); unique(s0_mt$indicator)
head(s0_mtStk); unique(s0_mtStk$indicator)
head(s0_vessel); unique(s0_vessel$indicator)
head(s0_vesselStk); unique(s0_vesselStk$indicator)
head(s0_risk); unique(s0_risk$indicator)
```

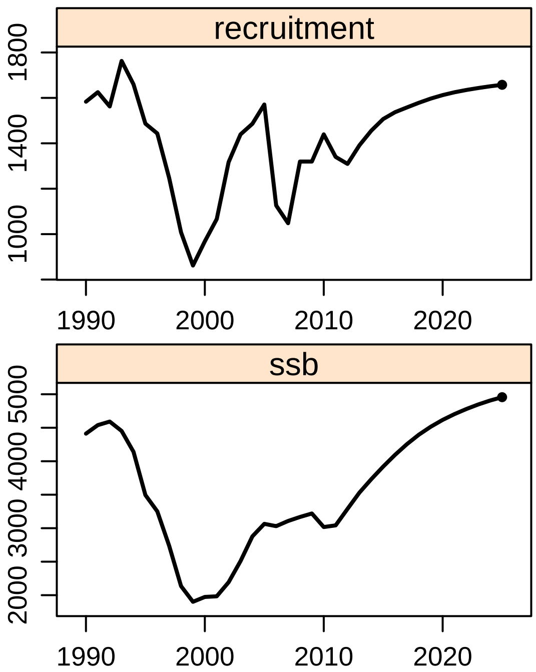
Wide format.

```r
s0_bio_l    <- bioSum(s0, long = FALSE, years = ac(2016:2020))
s0_adv_l    <- advSum(s0, long = FALSE, years = ac(2016:2020))
s0_flt_l    <- fltSum(s0, long = FALSE, years = ac(2016:2020))
s0_fltStk_l <- fltStkSum(s0, long = FALSE, years = ac(2016:2020))
s0_mt_l     <- mtSum(s0, long = FALSE, years = ac(2016:2020))
s0_mtStk_l  <- mtStkSum(s0, long = FALSE, years = ac(2016:2020))
s0_vessel_l <- vesselSum(s0, long = FALSE, years = ac(2016:2020))
s0_vesselStk_l <- vesselStkSum(s0, long = FALSE, years = ac(2016:2020))

# Exploring data frames
head(s0_bio_l, 2)
head(s0_adv_l, 2)
head(s0_flt_l, 2)
head(s0_fltStk_l, 2)
head(s0_mt_l, 2)
head(s0_mtStk_l, 2)
head(s0_vessel_l, 2)
head(s0_vesselStk_l, 2)
```

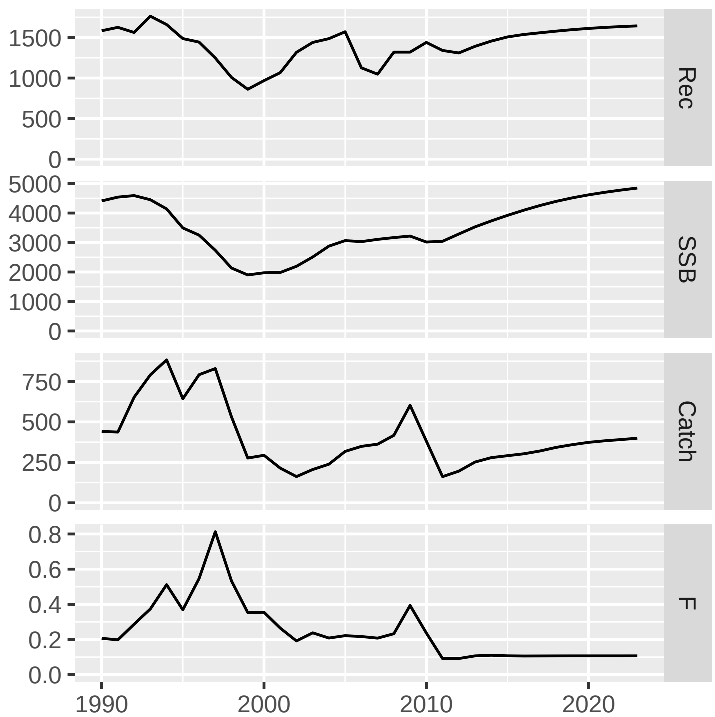
Plotting results

You can show results using the default plots in **FLCore** package.

```r
plot(s0$biols[[1]])
```




```r
plot(s0$stocks[[1]])
```



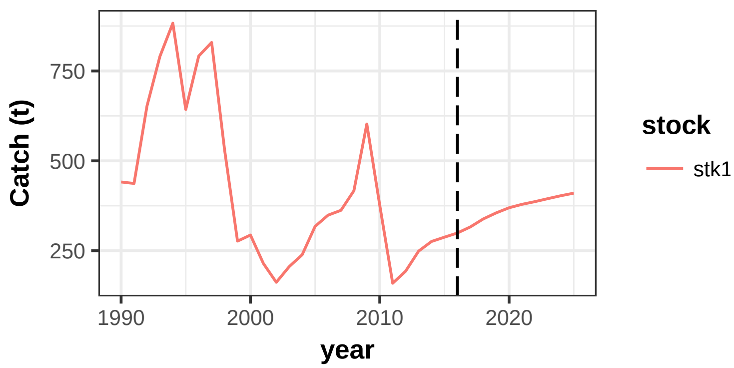
Additionally you can plot results using `plotFLBiols`, `plotFLFleets` and `plotCatchFl`. The plots will be load in your working directory by default.

```r
# set your own working directory.
# myWD <- "My working directory"
# setwd(myWD)
plotFLBiols(s0$biols, pdfnm="s0")
plotFLFleets(s0$fleets, pdfnm="s0")
```

```r
plotEco(s0, pdfnm="s0")
plotfltStkSum(s0, pdfnm="s0")
```

You can also use the function `ggplot` to plot additional figures.

```r
aux <- subset(s0_bio, indicator=="catch" )
p <- ggplot(data=aux, aes(x=year, y=value, color=stock))+
  geom_line()+
  geom_vline(xintercept = 2016, linetype = "longdash")+
  theme_bw()+
  theme(text=element_text(size=10),
        title=element_text(size=10,face="bold"),
        strip.text=element_text(size=10))+
  ylab("Catch (t)")
print(p)
```



Let's play

FLBEIA have several options to model the specific characteristics of a fishery. In particular, `FLBEIA` provides several options for population growth, effort dynamics, price models, capital dynamics, observation models and management advice. Let's play with these options.

Operating model: Change stock-recruitment relationship.

Firstly, we define some useful parameters.

```r
  first.yr          <- 1990
  proj.yr           <- 2010
  last.yr           <- 2025
  yrs <- c(first.yr=first.yr,proj.yr=proj.yr,last.yr=last.yr)

  stks <- c('stk1')
```

In example `one` the stock recruitment model is `bevholtAR1`.

```r
oneSR$stk1@model
```

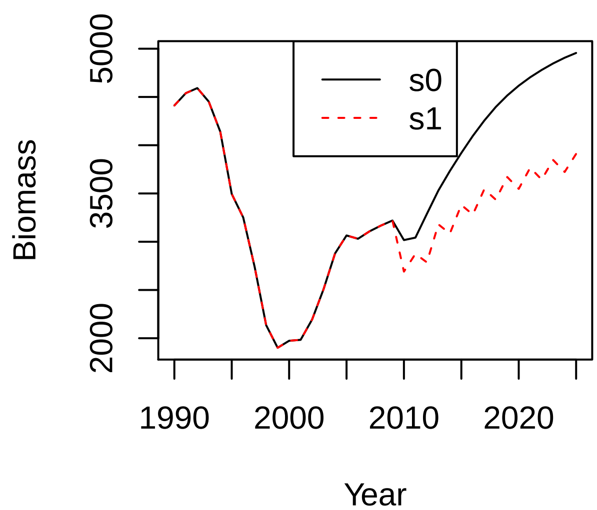
```
[1] "bevholtAR1"
```

Replace the stock - recruitment relationship `bevholtAR1` by `segreg` model in another simulation called s1. `segreg` is a segmented regression stock-recruitment model. For this new simulation you need to estimate parameters of `segreg` model using the function `fmle`. `fmle` is a method that fits the SR model using logl and R's optim model fitting through maximum likelihood estimation. For more details click here.

```r
ssb <-  ssb(oneBio[[1]])[,as.character(first.yr:(proj.yr-1)),1,1]
rec <-  oneBio[[1]]@n[1,as.character(first.yr:(proj.yr-1)),]
sr.segreg <- fmle(FLSR(model="segreg",ssb = ssb, rec = rec))

# Introduce the new model and its parameters in SR object.
SRs.segreg <- oneSR
SRs.segreg[[1]]@params[,,,] <- sr.segreg@params[,]
SRs.segreg[[1]]@model <- 'segreg'

#Run FLBEIA with the new SR function.
s1 <- FLBEIA(biols = oneBio, SRs = SRs.segreg , BDs = NULL, fleets = oneFl, covars = oneCv,
             indices = NULL, advice = oneAdv, main.ctrl = oneMainC, biols.ctrl = oneBioC,
             fleets.ctrl = oneFlC, covars.ctrl = oneCvC, obs.ctrl = oneObsC,
             assess.ctrl = oneAssC, advice.ctrl = oneAdvC)

plotFLBiols(s1$biols, pdfnm='s1')
plotFLFleets(s1$fleets, pdfnm='s1')
```

Compare biomass of s0 (“bevholtAR1”) with s1 (“segreg”),

```r
temp <- cbind(matrix(B_flbeia(s0)), matrix(B_flbeia(s1)))
matplot(temp, x = dimnames( B_flbeia(s1))$year, type = 'l', xlab = 'Year', ylab = 'Biomass')
legend('top', c('s0', 's1'), col = c('black','red'), lty = c(1,2))
```



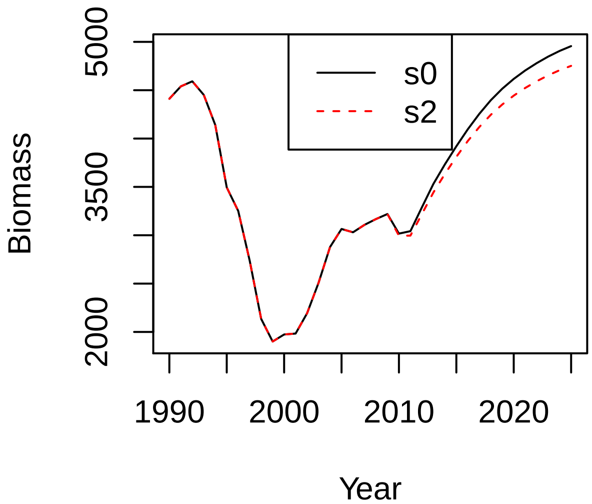
Natural mortality

In a simulation called s2, increase by 20% the natural mortality of the ages 7 to 12.

```r
oneBioM <- oneBio
oneBioM$stk1@m[7:12,,,,] <- oneBioM$stk1@m[7:12,,,,]*1.2

s2 <- FLBEIA(biols = oneBioM, SRs = oneSR , BDs = NULL, fleets = oneFl, covars = oneCv,
             indices = NULL, advice = oneAdv, main.ctrl = oneMainC, biols.ctrl = oneBioC,
             fleets.ctrl = oneFlC, covars.ctrl = oneCvC, obs.ctrl = oneObsC,
             assess.ctrl = oneAssC, advice.ctrl = oneAdvC)

plotFLBiols(s1$biols, pdfnm='s2')
plotFLFleets(s1$fleets, pdfnm='s2')
```

Compare the biomass of s0 with s2.

```
temp <- cbind(matrix(B_flbeia(s0)), matrix(B_flbeia(s2)))
matplot(temp, x = dimnames( B_flbeia(s0))$year, type = 'l', xlab = 'Year', ylab = 'Biomass')
legend('top', c('s0', 's2'), col = c('black','red'), lty = c(1,2))
```



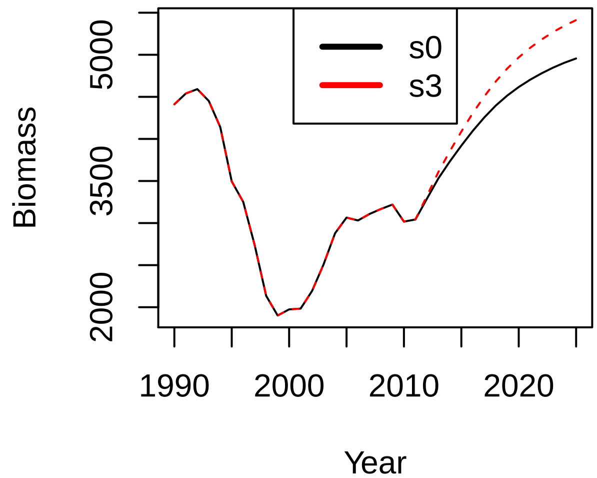
Harvest control rule

Change the target: the Fmsy value will be decreased by 20%.

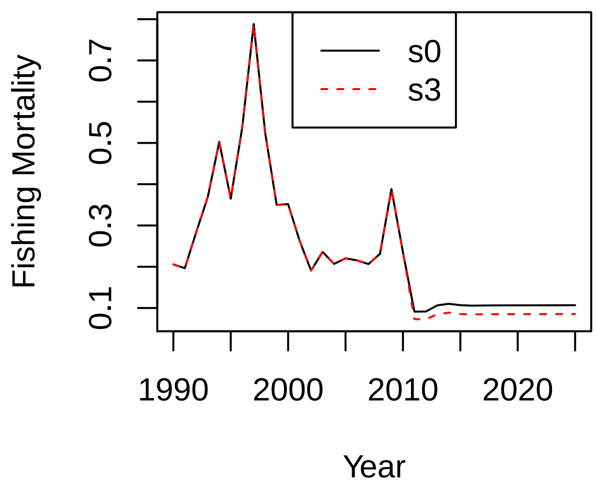
```
oneAdvC$stk1$ref.pts[3]
oneAdvC2 <- oneAdvC
oneAdvC2$stk1$ref.pts[3] <- oneAdvC$stk1$ref.pts[3]*0.8
s3 <- FLBEIA(biols = oneBio, SRs = oneSR , BDs = NULL, fleets = oneFl, covars = oneCv,
             indices = NULL, advice = oneAdv, main.ctrl = oneMainC, biols.ctrl = oneBioC,
             fleets.ctrl = oneFlC, covars.ctrl = oneCvC, obs.ctrl = oneObsC,
             assess.ctrl = oneAssC, advice.ctrl = oneAdvC2)
```

Compare de biomass and the fishing mortality of s0 with s3.

```
temp <- cbind(matrix(B_flbeia(s0)), matrix(B_flbeia(s3)))
matplot(temp, x = dimnames( B_flbeia(s0))$year, type = 'l', xlab = 'Year', ylab = 'Biomass')
legend('top', c('s0', 's3'), lwd = 3, col = c('black','red') )
```



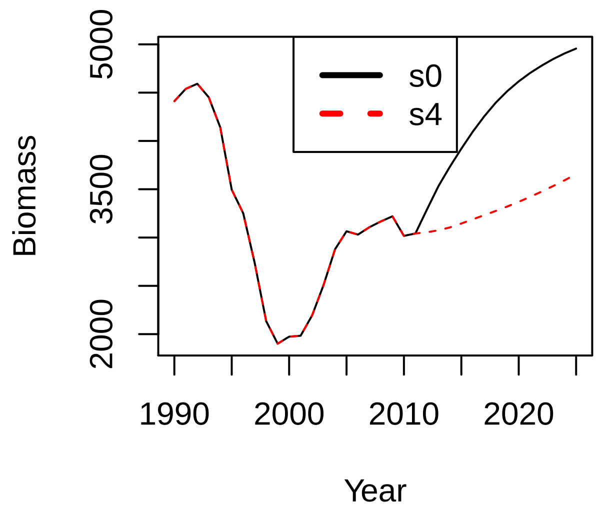

```
temp <- cbind(matrix(F_flbeia(s0)), matrix(F_flbeia(s3)))
matplot(temp, x = dimnames( B_flbeia(s0))$year, type = 'l', xlab = 'Year', ylab = 'Fishing Mortality')
legend('top', c('s0', 's3'), col = c('black','red'), lty = c(1,2) )
```



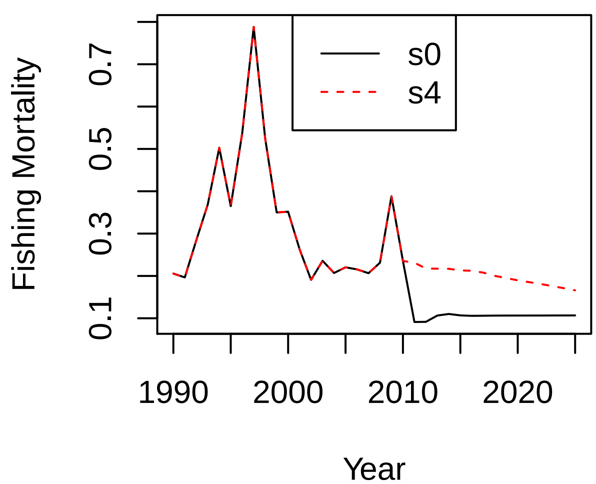
Now change the harvest control rule (HCR). The HCR applied in s0 was `IcesHCR`, which is the HCR used by ICES to generate TAC advice in the MSY framework. Now, run the model with `fixedAdvice`. This function is used when the advice is fixed and independent to the stock status, and therefore TAC values should be given as input in the advice object.

```
HCR.models       <- 'fixedAdvice'
oneAdvC1     <- create.advice.ctrl(stksnames = stks, HCR.models =  HCR.models)

s4 <- FLBEIA(biols = oneBio, SRs = oneSR , BDs = NULL, fleets = oneFl, covars = oneCv,
            indices = NULL, advice = oneAdv, main.ctrl = oneMainC, biols.ctrl = oneBioC,
            fleets.ctrl = oneFlC, covars.ctrl = oneCvC, obs.ctrl = oneObsC,
            assess.ctrl = oneAssC,  advice.ctrl = oneAdvC1)

plotFLBiols(s4$biols,pdfnm= 's4')
plotFLFleets(s4$fleets,pdfnm= 's4')
```

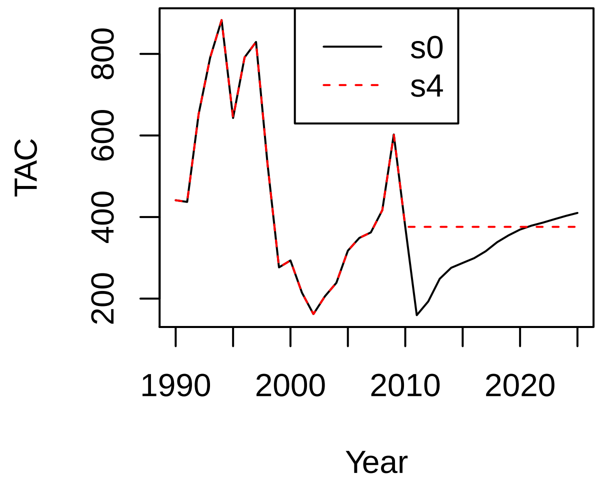
```
temp <- cbind(matrix(B_flbeia(s0)), matrix(B_flbeia(s4)))
matplot(temp, x = dimnames( B_flbeia(s0))$year, type = 'l', xlab = 'Year', ylab = 'Biomass')
legend('top', c('s0', 's4'), lwd = 3, col = c('black','red'), lty = c(1,2) )
```




```r
temp <- cbind(matrix(F_flbeia(s0)), matrix(F_flbeia(s4)))
matplot(temp, x = dimnames( B_flbeia(s4))$year, type = 'l', xlab = 'Year',
        ylab = 'Fishing Mortality')
legend('top', c('s0', 's4'), col = c('black','red'), lty = c(1,2))
```



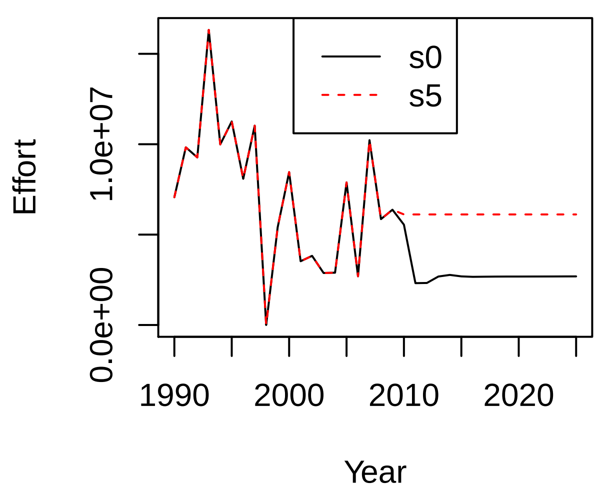

```r
temp <- cbind(matrix(s0$advice$TAC), matrix(s4$advice$TAC))
matplot(temp, x = dimnames(s0$advice$TAC)$year, type = 'l', xlab = 'Year',
        ylab = 'TAC')
legend('top', c('s0', 's4'), col = c('black','red'), lty = c(1,2))
```



Effort function.

The effort dynamics follows a Simple Mixed Fisheries Behaviour `SMFB`. Change that function by Fixed Effort `fixedEffort`. In this function all the parameters are given as input except discards and landings (total and at age). The only task of this function is to update the discards and landings (total and at age) according to the catch production function specified in `fleets.ctrl` argument.

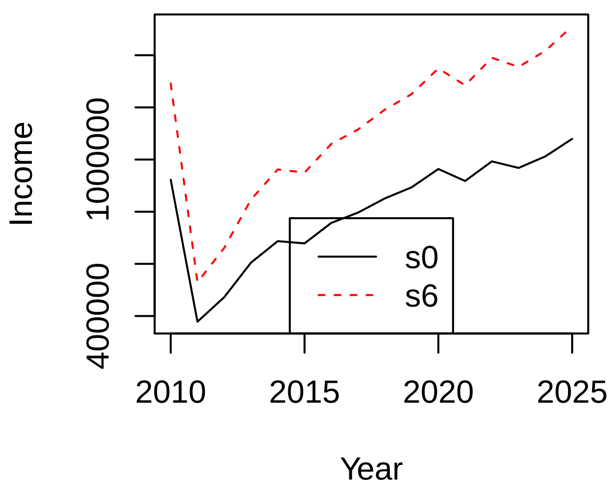
```
oneFlC1 <- oneFlC
oneFlC1$fl1$effort.model <- 'fixedEffort'
s5 <- FLBEIA(biols = oneBio, SRs = oneSR , BDs = NULL, fleets = oneFl, covars = oneCv,
             indices = NULL, advice = oneAdv, main.ctrl = oneMainC, biols.ctrl = oneBioC,
             fleets.ctrl = oneFlC1, covars.ctrl = oneCvC, obs.ctrl = oneObsC,
             assess.ctrl = oneAssC, advice.ctrl = oneAdvC)

s0_flt    <- fltSum(s0); s0_eff <- subset(s0_flt, indicator == 'effort')
s5_flt    <- fltSum(s5); s5_eff <- subset(s5_flt, indicator == 'effort')

temp <- cbind(s0_eff$value, s5_eff$value)
matplot(temp, x = dimnames( B_flbeia(s0))$year, type = 'l', xlab = 'Year', ylab = 'Effort')
legend('top', c('s0', 's5'), col = c('black','red'), lty = c(1,2))
```



Price

Now, increase by 40% the average price of the stock 1.

```
oneFl1 <- oneFl
oneFl1$fl1@metiers$met1@catches$stk1@price <- oneFl$fl1@metiers$met1@catches$stk1@price*1.4

s6 <- FLBEIA(biols = oneBio, SRs = oneSR , BDs = NULL, fleets = oneFl1, covars = oneCv,
             indices = NULL, advice = oneAdv, main.ctrl = oneMainC, biols.ctrl = oneBioC,
             fleets.ctrl = oneFlC, covars.ctrl = oneCvC, obs.ctrl = oneObsC,
             assess.ctrl = oneAssC, advice.ctrl = oneAdvC)

s0_prof<- subset(s0_flt, indicator == 'income')
s6_flt <- fltSum(s6); s6_prof <- subset(s6_flt, indicator == 'income')

temp <- cbind(s0_prof$value, s6_prof$value)
matplot(temp[21:36,], x = dimnames(B_flbeia(s6))$year[21:36], type = 'l',
        xlab = 'Year', ylab = 'Income')
legend('bottom', c('s0', 's6'),col = c('black','red'), lty = c(1,2) )
```



Now, we can estimate economics indicators

```
rev_s0 <- revenue_flbeia(s0$fleets$fl1)
rev_s6 <- revenue_flbeia(s6$fleets$fl1)
```

Visualizing results with flbeiaApp

Currently is under development, but soon you will be able to run the `flbeiaApp` to built an interactive web applications for visualizing data. Currently this a preliminary version or beta versi?n of the `flbeiaApp`.

```
one_simul <- list(s0, s1, s2, s3, s4, s5, s6)
scenarios <- c('s0', 's1', 's2', 's3', 's4', 's5', 's6')
names(one_simul) <- scenarios

RefPts <- data.frame(stock = rep(names(one_simul[[1]][[1]]), each = 6*length(one_simul)),
                     scenario = rep(names(one_simul), each = 6),
                     indicator = rep(c('Bmsy','Fmsy', 'Bpa', 'Blim', 'Fpa', 'Flim'), length(one_simul)),
                     value = rep(c(max(ssb(one_simul[[1]][[1]][['stk1']])[,1:12],na.rm = TRUE)*0.5,
                               0.27,
```

```
                                 max(ssb(one_simul[[1]][[1]][['stk1']])[,1:12],na.rm = TRUE)*0.25,
                                 max(ssb(one_simul[[1]][[1]][['stk1']])[,1:12],na.rm = TRUE)*0.15,
                                 0.35, 0.5), length(one_simul)))


flbeiaApp(one_simul , RefPts = RefPts,  years = ac(1990:2025), npv.y0 = '2009', npv.yrs = ac(2010:2025))
```

EXAMPLE 2: single stock, single fleet, single season and three iterations.

The second example is the same as the first example, but with three iterations. The Operating Model (OM) is formed by a single age-structured stock. Only one operates fleet which activity is performed in an unique metier and the time step is annual. The historic data go from 1990 to 2009 and projection period from 2010 to 2025.

- Operating model:
  - Population dynamics: Age structured population growth
  - SR model: Beverthon and Holt autoregressive/segmented regression
- Management Procedure:
  - Perfect observation
  - No assessment

Load data

```
rm(list =ls())  # Clean the environment
data(oneIt)
ls()
```

Run FLBEIA

Call to `FLBEIA` function with the arguments stored in `oneIt` dataset. The uncertainty is in the parameters of the SR. Check it.

```
opts_chunk$set(message=FALSE)

s0_it <- FLBEIA(biols = oneItBio, SRs = oneItSR , BDs = NULL, fleets = oneItFl, covars = oneItCv,
            indices = NULL, advice = oneItAdv, main.ctrl = oneItMainC, biols.ctrl = oneItBioC,
            fleets.ctrl = oneItFlC, covars.ctrl = oneItCvC, obs.ctrl = oneItObsC, assess.ctrl = oneItAs
            advice.ctrl = oneItAdvC)
```

Summarizing results

```
s0_it_bio    <- bioSum(s0_it)
s0_it_adv    <- advSum(s0_it)
s0_it_flt    <- fltSum(s0_it)
s0_it_fltStk <- fltStkSum(s0_it)
s0_it_mt     <- mtSum(s0_it)
s0_it_mtStk  <- mtStkSum(s0_it)
```

```
s0_it_vessel <- vesselSum(s0_it)
s0_it_vesselStk <- vesselStkSum(s0_it)
s0_it_npv  <- npv(s0_it, y0 = '2014')
s0_it_risk <- riskSum(s0_it, Bpa = c(stk1= 135000), Blim = c(stk1= 96000), Prflim = c(fl1 = 0))
```

Create summary data frames (biological, economic, and catch) with quantiles(5,50,95). There are several predeterminated functios to sumarize results of `FLBEIA` outputs taken into account the uncertainty.

```
proj.yr     <- 2009
s0_it_bioQ   <- bioSumQ(s0_it_bio)
s0_it_fltQ   <- fltSumQ(s0_it_flt)
```

Plotting results

Create several plots and save them in the working directory using 'pdf' format.

```
plotFLBiols(s0_it$biols, pdfnm='s0_it')
plotFLFleets(s0_it$fleets,pdfnm='s0_it')
plotEco(s0_it, pdfnm='s0_it')
```

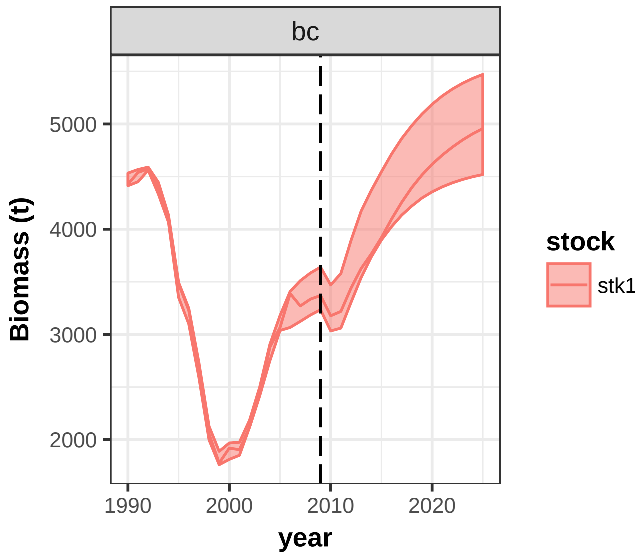
```
pdf
  2
```

```
plotfltStkSum(s0_it, pdfnm='s0_it')
```

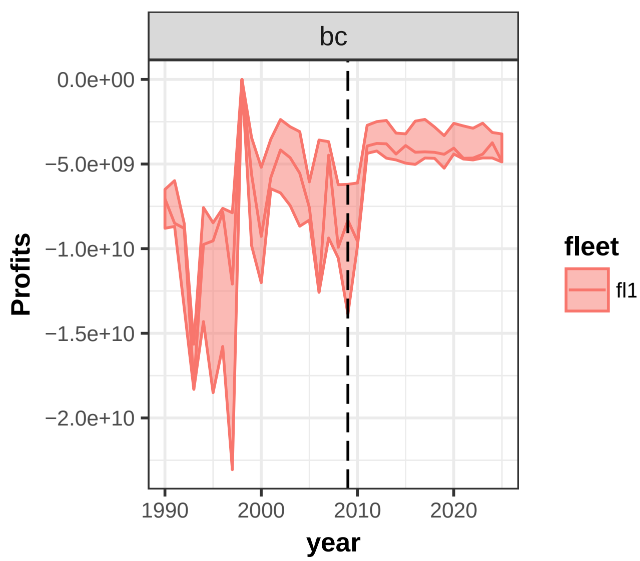
```
pdf
  2
```

See the biomass and the profit time series with its uncertainty (quantiles). In this case `ggplot` function is used.

```
aux <-  subset(s0_it_bioQ, indicator=="biomass")
p <- ggplot(data=aux , aes(x=year, y=q50, color=stock))+
      geom_line()+
      geom_ribbon(aes(x=year, ymin=q05, ymax=q95, fill=stock), alpha=0.5)+
      facet_wrap(~scenario, scales="free")+
      geom_vline(xintercept = proj.yr, linetype = "longdash")+
      theme_bw()+
      theme(text=element_text(size=10),
            title=element_text(size=10,face="bold"),
            strip.text=element_text(size=10))+
      ylab("Biomass (t)")
print(p)
```




```
aux <-  subset(s0_it_fltQ, indicator=="profits")
aux$year <- as.numeric(as.character(aux$year))
p1 <- ggplot(data=aux , aes(x=year, y=q50, color=fleet))+
      geom_line()+
      geom_ribbon(aes(x=year, ymin=q05, ymax=q95, fill=fleet), alpha=0.5)+
      facet_wrap(~scenario, scales="free")+
      geom_vline(xintercept = proj.yr, linetype = "longdash")+
      theme_bw()+
      theme(text=element_text(size=10),
            title=element_text(size=10,face="bold"),
            strip.text=element_text(size=10))+
      ylab("Profits")
print(p1)
```



Visualizing results with flbeiaApp

Apply the `flbeiaApp` (beta versi?n) to built an interactive web applications for visualizing data.

```
RefPts <- data.frame(stock = rep(names(one_simul[[1]][[1]]), each = 6*length(one_simul)),
                     scenario = rep(names(one_simul), each = 6),
                     indicator = rep(c('Bmsy','Fmsy', 'Bpa', 'Blim', 'Fpa', 'Flim'),
                     length(one_simul)),value = rep(c(max(ssb(one_simul[[1]][[1]][['stk1']])
                     [,1:12],na.rm = TRUE)*0.5, 0.27,
                     max(ssb(one_simul[[1]][[1]][['stk1']])[,1:12],na.rm = TRUE)*0.25,
                     max(ssb(one_simul[[1]][[1]][['stk1']])[,1:12],na.rm = TRUE)*0.15,
                     0.35, 0.5), length(one_simul)))

RefPts <- data.frame(stock = rep(c('stk1'), each = 6), scenario = 'bc',
                     indicator = rep(c('Bmsy','Fmsy', 'Bpa', 'Blim', 'Fpa', 'Flim')
                      , 2), value = c(max((ssb(s0_it[[1]][[1]])), na.rm = TRUE)*0.75,
                      0.27, max((ssb(s0_it[[1]][[1]])),na.rm = TRUE)*0.5,
                      max((ssb(s0_it[[1]][[1]])),na.rm = TRUE)*0.25,
                      0.35, 0.5))


flbeiaApp(list(bc = s0_it), RefPts = RefPts, npv.y0 = '2012', npv.yrs = ac(2013:2020))
```

EXERCISES

Run the following scenarios and compare them with the baseline scenario (s0_it). The comparison should be done numerically and/or plotting scenarios.

- *s1_it*: Population growth: Change stock-recruitment relationship.
- *s2_it*: Natural mortality: Increase the natural mortility of ages 2 to 6 as 25%.
- *s3_it*: Harvest control rule: apply `FroeseHCR`.
- *s4_it*: Effort dinamic: Apply `MaxProfit`.
- *s6_it*: Decrease a 15% the price of the recruits.
- *s6_it*: The legislation has changed and the mesh size of the vessel will be changed, decreasing the catchability of vessels a 10%. What will be the biological and economic impacts of this legislation?
- *s7_it*: Change the effort dynamic from `SMFB` to `fixed effort` and compare de SSB resulted from this simulation with de baseline scenario.

More information

- You can submit bug reports, questions or suggestions on this tutorial at https://github.com/flr/doc/issues.
- Or send a pull request to https://github.com/flr/doc/
- For more information on the FLR Project for Quantitative Fisheries Science in R, visit the FLR webpage, http://flr-project.org.
- You can submit bug reports, questions or suggestions specific to **FLBEIA** to flbeia@azti.es.

Software Versions

- R version 3.4.1 (2017-06-30)
- FLCore: 2.6.5
- FLBEIA: 1.15.1
- FLFleet: 2.6.0
- FLash: 2.5.8

- FLAssess: 2.6.1
- FLXSA: 2.5.20140808
- ggplotFL: 2.6.2
- ggplot2: 2.2.1
- **Compiled**: Mon Oct 2 11:32:44 2017

License

Author information

FLBEIA team. AZTI. Marine Reserach Unit. Txatxarramendi Ugartea z/g, 48395, Sukarrieta, Basque Country, Spain. **Mail** flbeia@azti.es