Short-term Forecasts, Summer School in Quantative Fisheries Stock Assessment, Capo Granitola

Alessandro Ligas

July 13th, 2017

```
# STF
# Here we run the STF for 3 years, 2015, 2016, 2017
# You can change these as appropriate
# The first year of the STF should be the next one after the final year in your stock data
# For example, the final year in the HKE09_10_11 stk object is 2014 so the first year of the STF is 201

# Load the libraries
library(FLCore)
```

```
## Loading required package: MASS
```

```
## Loading required package: lattice
```

```
## FLCore (Version 2.6.3, packaged: 2017-07-05 12:26:15 UTC)
```

```
library(FLAssess)
```

```
## Loading required package: FLash
```

```
## Warning: replacing previous import 'ggplot2::%+%' by 'FLCore::%+%' when
## loading 'ggplotFL'
```

```
library(FLash)
library(ggplotFL)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:FLCore':
##
##     %+%
```

```
library(FLEDA)
library(FLXSA)
require(plyr)
```

```
## Loading required package: plyr
```

```
##
## Attaching package: 'plyr'
```

```
## The following object is masked from 'package:FLCore':
##
##     desc
```

```
require(FLBRP)
```

```
## Loading required package: FLBRP
```

```r
# load the stock object
load("HKEFbar0_3.RData")
hke<-HKE.new_xsa

stf_years <- c(2015,2016,2017)
no_stf_years <- length(stf_years)

# For the STF we would like to run a F0.1 scenario
# Use FLBRP to get F0.1       #SET RP ACCORDING TO THE LAST RUN
stk_brp <- brp(FLBRP(hke))
refpts(stk_brp)
```

```
## An object of class "FLPar"
##         quantity
## refpt    harvest    yield      rec        ssb        biomass    revenue
##   virgin 0.0000e+00 0.0000e+00 1.0000e+00 1.1603e+00 1.2168e+00         NA
##   msy    3.0035e-01 4.6147e-02 1.0000e+00 3.0193e-01 3.4690e-01         NA
##   crash  1.2052e+01 9.4880e-03 1.0000e+00 4.3705e-06 1.0933e-02         NA
##   f0.1   2.1604e-01 4.4382e-02 1.0000e+00 4.3362e-01 4.8134e-01         NA
##   fmax   3.0035e-01 4.6147e-02 1.0000e+00 3.0193e-01 3.4690e-01         NA
##   spr.30 2.6686e-01 4.5909e-02 1.0000e+00 3.4810e-01 3.9413e-01         NA
##   mey            NA         NA         NA         NA         NA         NA
##         quantity
## refpt    cost       profit
##   virgin         NA         NA
##   msy            NA         NA
##   crash          NA         NA
##   f0.1           NA         NA
##   fmax           NA         NA
##   spr.30         NA         NA
##   mey            NA         NA
## units:  NA
```

```r
f01 <- c(refpts(stk_brp)["f0.1","harvest"])
# f01=0.216 # please verify that f0.1 is similar to that estimated by the assessment


# We also need F status quo - the geometric mean of the last X years
# Here we use 3 years
no_stk_years <- dim(rec(hke))[2]
no_fbar_years <- 3 # Or set your own as appropriate
fbars <- fbar(hke)[,(no_stk_years - no_fbar_years + 1):no_stk_years]
fbar_status_quo <- exp(mean(log(c(fbars))))
ssb(hke)
```

```
## An object of class "FLQuant"
## , , unit = unique, season = all, area = unique
##
##      year
## age   2006   2007   2008   2009   2010   2011   2012   2013   2014
##   all 3716.8 3121.0 2695.7 2657.3 3077.7 2940.6 2355.4 2618.3 2911.4
##
## units:  kg
```

```r
catch(hke)
```

```
## An object of class "FLQuant"
## , , unit = unique, season = all, area = unique
##
##      year
## age   2006   2007   2008   2009   2010   2011   2012   2013   2014
##   all 4656.5 3829.7 3405.5 3664.3 3384.1 3757.4 2640.7 2895.4 3074.7
##
## units:  NA
```

```r
# Set up the future stock object.
# Here we use the default assumptions about what happens to weights, maturity and selection pattern in
# (e.g. weights are means of the last 3 years)
# NOTE: You may want to change some of these assumptions by hand
# See the help page for stf: ?stf for more details
stf_stk <- stf(hke, nyears = no_stf_years, wts.nyears = 3)

# Set up future recruitment to be mean of last X years
# Here we set as geometric mean of the last 3 years
no_rec_years <- 3 # Change number of years as appropriate
recs <- rec(hke)[,(no_stk_years - no_rec_years + 1):no_stk_years]
mean_rec <- exp(mean(log(c(recs))))

# We are going to run several F scenarios for the STF
# The scenarios are based on 'F status quo', which we calculated above as the mean F of the last X years
# An STF is for three years - you could change this but if you do you will have to hack the code below
# For a three year STF the F pattern is:
# year 1: fbar_status_quo
# year 2: fbar_status_quo * fbar_multiplier
# year 3: fbar_status_quo * fbar_multiplier
# The fbar_multiplier is the same for years 2 and 3

# We are going to run several STFs with different values for the fbar_multiplier
# The fbar_multiplier ranges from 0.1 to 2 by 0.1
fbar_multiplier <- seq(from = 0, to = 2, by = 0.1)

# We are going to build a data.frame that builds these scenarios
# Each column in the dataframe is a year
# Each row is a scenario
# Set up the fbar scenarios - note that if you project for more than 3 years you will need to add more
fbar_scenarios <- cbind(rep(fbar_status_quo,length(fbar_multiplier)),
                        fbar_multiplier*fbar_status_quo,
                        fbar_multiplier*fbar_status_quo)
# Add the F0.1 scenario as a final scenario
fbar_scenarios <- rbind(fbar_scenarios, c(fbar_status_quo,f01,f01))

# There are various results we want to extract from the STF
# Make an empty matrix in which to store the results
stf_results <- matrix(NA,nrow = nrow(fbar_scenarios),ncol = 10)
# Change the column names to reflect years
colnames(stf_results) <- c('Ffactor','Fbar','Catch_2014','Catch_2015','Catch_2016','Catch_2017','SSB_20

# Store the FLStock each time
```

```
stk_stf <- FLStocks()
# Loop over the scenarios
for (scenario in 1:nrow(fbar_scenarios)) {
  cat("Scenario: ", scenario, "\n")
  # Make a target object withe F values for that scenario
  ctrl_target <- data.frame(year = stf_years,
                            quantity = "f",
                            val = fbar_scenarios[scenario,])
  # Set the control object - year, quantity and value for the moment
  ctrl_f <- fwdControl(ctrl_target)
  # Run the forward projection. We include an additional argument, maxF.
  # By default the value of maxF is 2.0
  # Here we increase it to 10.0 so that F is not limited
  stk_stf_fwd <- fwd(stf_stk, ctrl = ctrl_f, sr = list(model="mean", params=FLPar(a = mean_rec)), maxF =
  ## Check it has worked - uncomment out to check scenario by scenario
  #plot(stk_stf_fwd)
  # Store the result - if you want to, comment out if unnecessary
  stk_stf[[as.character(scenario)]] <- stk_stf_fwd

  # Fill results table
  stf_results[scenario,1] <- fbar_scenarios[scenario,2] / fbar_scenarios[scenario,1] # fbar status quo
  stf_results[scenario,2] <- fbar(stk_stf_fwd)[,ac(2017)] # final stf year
  stf_results[scenario,3] <- catch(stk_stf_fwd)[,ac(2014)] # last 'true' year
  stf_results[scenario,4] <- catch(stk_stf_fwd)[,ac(2015)] # 1st stf year
  stf_results[scenario,5] <- catch(stk_stf_fwd)[,ac(2016)] # 2nd stf year
  stf_results[scenario,6] <- catch(stk_stf_fwd)[,ac(2017)] # final stf year
  stf_results[scenario,7] <- ssb(stk_stf_fwd)[,ac(2016)] # 2nd stf year
  stf_results[scenario,8] <- ssb(stk_stf_fwd)[,ac(2017)] # final stf year
  # Change in SSB
  stf_results[scenario,9] <- (ssb(stk_stf_fwd)[,ac(2017)]-ssb(stk_stf_fwd)[,ac(2016)])/ssb(stk_stf_fwd)
  stf_results[scenario,10] <- (catch(stk_stf_fwd)[,ac(2016)]-catch(stk_stf_fwd)[,ac(2014)])/catch(stk_s
}
```

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

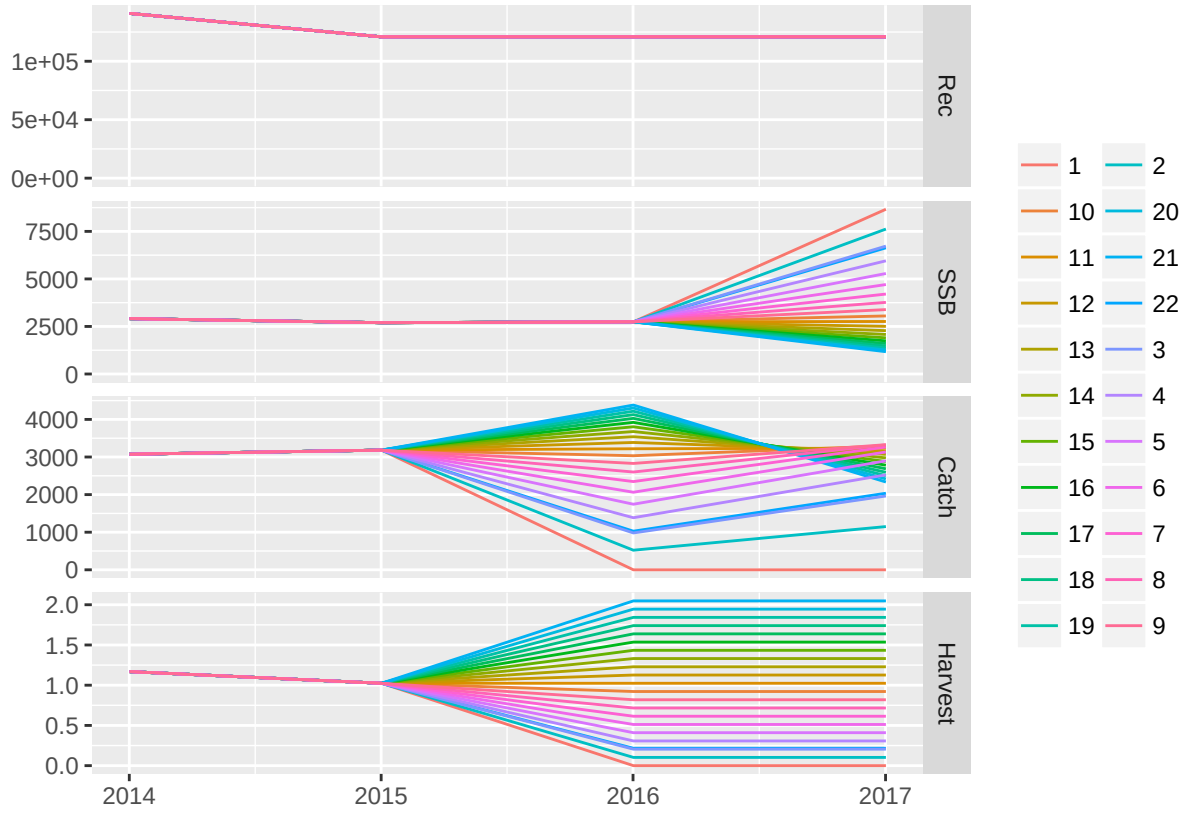
```
## Scenario:  22
```

```
#}
# Look at the table of results
stf_results
```

```
##         Ffactor      Fbar Catch_2014 Catch_2015 Catch_2016 Catch_2017
##  [1,] 0.0000000 0.0000000   3074.654   3182.886     0.0000      0.000
##  [2,] 0.1000000 0.1023917   3074.654   3182.886   521.4372   1150.034
##  [3,] 0.2000000 0.2047834   3074.654   3182.886   980.3961   1963.352
##  [4,] 0.3000000 0.3071751   3074.654   3182.886  1385.5753   2524.711
##  [5,] 0.4000000 0.4095668   3074.654   3182.886  1744.3786   2898.392
##  [6,] 0.5000000 0.5119585   3074.654   3182.886  2063.1139   3133.078
##  [7,] 0.6000000 0.6143502   3074.654   3182.886  2347.1619   3265.573
##  [8,] 0.7000000 0.7167419   3074.654   3182.886  2601.1170   3323.646
##  [9,] 0.8000000 0.8191336   3074.654   3182.886  2828.9075   3328.192
## [10,] 0.9000000 0.9215253   3074.654   3182.886  3033.8964   3294.885
## [11,] 1.0000000 1.0239170   3074.654   3182.886  3218.9667   3235.439
## [12,] 1.1000000 1.1263087   3074.654   3182.886  3386.5937   3158.567
## [13,] 1.2000000 1.2287004   3074.654   3182.886  3538.9057   3070.710
## [14,] 1.3000000 1.3310921   3074.654   3182.886  3677.7359   2976.597
## [15,] 1.4000000 1.4334838   3074.654   3182.886  3804.6659   2879.663
## [16,] 1.5000000 1.5358754   3074.654   3182.886  3921.0627   2782.375
## [17,] 1.6000000 1.6382671   3074.654   3182.886  4028.1102   2686.471
## [18,] 1.7000000 1.7406588   3074.654   3182.886  4126.8354   2593.150
## [19,] 1.8000000 1.8430505   3074.654   3182.886  4218.1316   2503.207
## [20,] 1.9000000 1.9454422   3074.654   3182.886  4302.7770   2417.143
## [21,] 2.0000000 2.0478339   3074.654   3182.886  4381.4510   2335.239
## [22,] 0.2109933 0.2160397   3074.654   3182.886  1027.4257   2036.073
##       SSB_2016 SSB_2017 Change_SSB_2016-2017(%) Change_Catch_2014-2016(%)
##  [1,] 2742.799 8664.653             215.9055061                -100.000000
##  [2,] 2742.799 7622.295             177.9020540                 -83.040788
##  [3,] 2742.799 6725.372             145.2010493                 -68.113616
##  [4,] 2742.799 5952.073             117.0072557                 -54.935574
##  [5,] 2742.799 5283.972              92.6489102                 -43.265865
##  [6,] 2742.799 4705.497              71.5582178                 -32.899323
##  [7,] 2742.799 4203.475              53.2549560                 -23.660954
##  [8,] 2742.799 3766.760              37.3326894                 -15.401323
##  [9,] 2742.799 3385.908              23.4471785                  -7.992669
## [10,] 2742.799 3052.917              11.3066293                  -1.325614
## [11,] 2742.799 2760.997               0.6634918                   4.693610
## [12,] 2742.799 2504.383              -8.6924425                  10.145507
## [13,] 2742.799 2278.173             -16.9398495                  15.099300
## [14,] 2742.799 2078.198             -24.2307636                  19.614611
## [15,] 2742.799 1900.905             -30.6946968                  23.742879
## [16,] 2742.799 1743.265             -36.4421091                  27.528567
## [17,] 2742.799 1602.691             -41.5673328                  31.010176
## [18,] 2742.799 1476.969             -46.1510379                  34.221115
## [19,] 2742.799 1364.205             -50.2623121                  37.190431
## [20,] 2742.799 1262.773             -53.9604152                  39.943434
## [21,] 2742.799 1171.278             -57.2962604                  42.502227
## [22,] 2742.799 6634.673             141.8942368                 -66.584027
```

```
# export this if necessary
write.csv(stf_results, file="stf_results.csv")
```

```r
# Plotting
# Plotting is not necessary for the report but here is a crude one anyway
plot(window(stk_stf, start=2014, end=2017))
```




```
########     END OF SCRIPT ##############################################################
```