

# MSE with Multiple Stocks and Fleets

## CDFW Case-studies

Adrian Hordyk      [adrian@bluematterscience.com](mailto:adrian@bluematterscience.com)

24 September, 2020

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	A Note on Package Names . . . . .	2
1.2	Installing and Loading <code>OMtool</code> . . . . .	2
<b>2</b>	<b>Rock Crab Case Study</b>	<b>2</b>
2.1	Set up <code>MM</code> Object . . . . .	3
2.2	Run the Historical Simulations . . . . .	4
2.3	Define Management Procedures . . . . .	5
2.4	Run Forward Projections . . . . .	7
2.5	Examine Results . . . . .	9
<b>3</b>	<b>Lobster Case Study</b>	<b>16</b>
<b>4</b>	<b>Halibut Case Study</b>	<b>16</b>

## 1 Introduction

This document describes the application of the newly developed multiple stock and multiple fleet extension for `DLMtool` to three case-study fisheries from the California Department of Fish and Wildlife (CDFW).

`DLMtool` is an R package designed to make management strategy evaluation (MSE) accessible to data-limited fisheries. The `DLMtool` operating model (OM) assumes a single stock (e.g., usually female component of the population) and a single fishing fleet (i.e., all fishing fleets are modeled together in aggregate). The `MSEtool` R package has been developed as an extension to `DLMtool`, and allows more complex features typical of more data-rich fisheries to be included in the modeling framework, such as quantitative stock assessments and complex spatial arrangements.

`MultiMSE` was developed to extend the modeling framework to include more complex biology (e.g., separate male and female populations, and hermaphroditism), interactions between multiple stocks (e.g., MICE models for evaluating ecosystem interactions), and options for fleet-specific management (e.g., separate recreational and commercial fleets). `MultiMSE` was developed as a proof-of-concept, to evaluate the viability and practicality of including more complex life-history patterns, and flexible management options within the `DLMtool` framework.

The aim of this project was to further develop the `MultiMSE` extension so that it can be used to evaluate the performance of management options for CDFW fisheries with sexually dimorphic growth (male and female stocks), stock complexes (multiple species managed as a single unit), and fleet-specific management for fisheries that are exploited by fleets with different characteristics.

The purpose of this document is to describe the application of the multi-stock & multi-fleet model to three CDFW case study fisheries: 1) Rock Crab, 2) Lobster, and 3) Halibut, and to identify any remaining issues with applying the model to these fisheries. This extension is under continued development and this document will be revised as changes are made to the model.

## 1.1 A Note on Package Names

As mentioned above, the multi-stock & multi-fleet model was initially developed as a proof-of-concept and included in the `MSEtool` R package. Early on in this project it became clear that the development of the multi-stock & multi-fleet model required a substantial re-structure of the code in the `DLMtool` and `MSEtool` packages.

The solution to this was to develop a new temporary package called `OMtool` which contains the newly developed multi-stock & multi-fleet model as well as the revised code for the operating model, simulations, and management procedures, from `DLMtool`.

The eventual goal, aimed to be complete in January 2021, is to release new major releases of the `MSEtool` and `DLMtool` packages that will contain stable versions of the model for creating operating models and running MSE for both single stock and multiple-stock fisheries.

For the time being, all multi-stock & multi-fleet analyses described in this document are done using the temporary package `OMtool`. To avoid clashes, it is recommended that the `DLMtool` and `MSEtool` are **not** loaded in the R session while using the `OMtool` package.

## 1.2 Installing and Loading OMtool

The `OMtool` package can be installed directly from GitHub. We recommend using the latest version of R (at least R 4.0.0).

The Rtools software is also required to install the `OMtool` package.

```
# install.packages('devtools') # run this if devtools isn't installed
library(devtools)
install_github('Blue-Matter/OMtool') # install from GitHub

# load package into R session (needs to be done for each new session)
library('OMtool')
```

As mentioned above, `OMtool` will be replaced at a later date with an updated version of `MSEtool`, which will be released on CRAN (Rtools isn't required for packages on CRAN).

## 2 Rock Crab Case Study

The rock crab case study involves three species, Brown, Red, and Yellow, which are managed as a single stock complex. Male and female rock crab have different growth patterns and mature at different sizes, and modeling all stocks together will involve 6 separate stocks.

Given that there are no interactions that are being modeled between the species, and the management options being explored by CDFW are focused on size regulations and effort controls, it may not be necessary to model the three stocks simultaneously. Wherever possible, we recommend limiting the number of stocks that are modeled together, as each additional stock significantly increases the run time of the model and may lead to increased convergence issues.

We first explore a case study that includes the female and male stocks for the Red rock crab. This analysis can be repeated for the Brown and Yellow rock crab case studies.

The case study can be developed further if CDFW wishes to explore management options that will affect the dynamics of the three species simultaneously.

## 2.1 Set up MOM Object

### 2.1.1 Import Stocks and create Stock Objects

Note that the some redundant slots have now been removed from the **Stock** and **Fleet** objects in **OMtool** (e.g., **M2**, **Mgrad**, **Amplitude**, **Period**, **SeYears**, etc). The Excel files received from CDFW have been slightly modified to account for these changes, and will be distributed with this document.

```
# directory where the OM Excel files are located
OMdir <- 'G:/My Drive/1_PROJECTS/CDFW_Multi_Fleet/RockCrabDLM'

# Import the Female and Male Red Crab Stocks
# (ignore any warnings about missing slots)
RedF_Stock <- XL2Stock(file.path(OMdir, 'OMRedF.xlsx'), msg=FALSE)
RedM_Stock <- XL2Stock(file.path(OMdir, 'OMRedM.xlsx'), msg=FALSE)

# a list of stock objects (female stock in first position)
Stocks <- list(RedF_Stock, RedM_Stock)
```

### 2.1.2 Import Fleet and create Fleet Object

A single fleet is imported:

```
Red_Fleet <- XL2Fleet(file.path(OMdir, 'OMRedF.xlsx'), msg=FALSE)
```

And a **Fleets** list is created, where the **Red\_Fleet** is applied to both the female and male stocks:

```
Fleets <- list(
  list(Red_Fleet), # female stock
  list(Red_Fleet) # male stock
)
```

### 2.1.3 Set up 2-sex model

For a 2-sex model it is necessary to set up the model so that recruitment is determined by the spawning stock biomass of the female population. This is done using an argument to the multi-OM object (**MOM**) called **SexPars**:

```
SexPars <- list()
# Stock (Row) spawn from SSB contributed by Stock (Column)
SexPars$SSBfrom <- matrix(c(1,0,
                           1,0),
                          byrow=TRUE,nrow=2)

SexPars$SSBfrom
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    1    0
```

**SexPars\$SSBfrom** is a 2x2 matrix specifying recruitment to females (row 1) and males (row 2) comes from female spawning biomass (column 1).

Note that it is important that the **SexPars\$SSBfrom** matches the order of the stocks in **Stocks** (ie., female in first position in list **Stocks**, and first column in **SexPars\$SSBfrom**).

### 2.1.4 Create MOM Object

The multi-stock operating model is an object of class MOM. Similar to the OM object, it requires a list of Stock, Fleet, Observation (Obs), and Implementation (Imp) objects, as well as the number of simulations to run.

For simplicity, we will use the generic observation object:

```
ObsList <- list(
  list(Generic_Obs), # Obs for female
  list(Generic_Obs) # Obs for male (the same)
)
```

And no implementation error:

```
ImpList <- list(
  list(Perfect_Imp),
  list(Perfect_Imp)
)
```

As this is for demonstration and testing purposes, the number of simulations is set quite low so that it runs quickly:

```
nsim <- 10
```

By default, the operating model calculates depletion in terms of spawning biomass relative to the equilibrium unfished spawning biomass ( $SB_0$ ). To calculate depletion in terms of vulnerable biomass, use the custom parameters argument:

```
cpars <- list()
# skip this line to calculate depletion in terms of SSB
cpars$control$D <- 'VB'
```

The MOM object is created using the new function:

```
MOM <- new("MOM",
  Stocks = Stocks,
  Fleets = Fleets,
  Obs = ObsList,
  Imps = ImpList,
  CatchFrac = NULL,
  SexPars=SexPars,
  cpars=cpars,
  nsim = nsim)
```

## 2.2 Run the Historical Simulations

The simulations for the historical and projection period can be run separately in `OMtool`. This is important as running the historical simulations can be quite time-consuming in some cases where the model has to optimize for depletion for a large number of stocks and fleets.

### 2.2.1 Simulate Historical Fishery

The simulation of the historical fishery is done with the `SimulateMOM` function (here we are using parallel processing; set `parallel` to `FALSE` if there are any problems with using parallel processing on your machine):

```
RCrabHist <- SimulateMOM(MOM, parallel = TRUE)
```

To avoid re-running the historical simulations each time, they can be saved to disk:

```
saveRDS(RCrabHist, file.path(OMdir, 'RCrabHist.rda'))
```

## 2.3 Define Management Procedures

The management options that have been proposed by CDFW for the rock crab fishery are static size limits, male only fishery (release of all female crabs), and effort reductions.

Permits for the crab fishery are limited, but there is currently latent effort in the fishery, so effort could increase. Modeling the changes in fishing effort in the future projections requires either a bio-economic model that describes how effort changes in response to catch rates, or simplifying assumptions for the fishing effort in the future years.

The latter approach is used in this case study. Two scenarios are included for future fishing effort:

1. Effort remains at the current level
2. Effort increases over the next 5 years to twice the current level

These scenarios are included as examples only, and without further discussion with CDFW and fishery stakeholders, they should not be considered credible scenarios for future effort.

### 2.3.1 Current Size Limit

The curE MP assumes that the current size limit of 4.25 inch (108 mm) remains the same and fishing effort remains at the current level.

A MP is defined that increases effort over the first 5 years of the projection period to twice the current level (current size limit remains unchanged):

```
incE <- function(x, Data, ...) {
  rec <- new('Rec') # create recommendation object
  curYr <- max(Data@Year) - Data@LHYear # current year
  if (curYr <= 5) {
    eff_inc <- 1 + 0.2 * curYr # effort increases 20% per year for first 5 years
  } else {
    eff_inc <- 2 # 100% increase after 5 years
  }
  rec@Effort <- eff_inc # populate Effort slot
  rec
}
class(incE) <- 'MP' # assign to class `MP`
```

### 2.3.2 Male Only Fishery & Current Size Limit

In a male only fishery, there is no retention of female catch. The MP does this by setting the retention curve to 0 for the female stock. Note that under this assumption, the female catch still suffers the discard mortality (if any) from crabs that are selected but not retained. To modify this assumption, the MP should change the selectivity curve instead of the retention curve.

Two MPs are defined, one with effort remaining at current levels, and the other with an increase in effort:

```
MOnly_curE <- function(x, Data, ...) {
  rec <- new('Rec') # create recommendation object
  # set length-at-5% retention to maximum size class (ie nothing is retained)
  rec@LR5 <- max(Data@CAL_bins)
  rec@LFR <- rec@LR5 + 1
  rec@Effort <- 1
  rec
}
```

```

}
class(MOnly_curE) <- 'MP'

MOnly_incE <- function(x, Data, ...) {
  rec <- new('Rec') # create recommendation object
  # set length-at-5% retention to maximum size class (ie nothing is retained)
  rec@LR5 <- max(Data@CAL_bins)
  rec@LFR <- rec@LR5 + 1

  curYr <- max(Data@Year) - Data@LHYear # current year
  if (curYr <=5) {
    eff_inc <- 1 + 0.2 * curYr # effort increases 20% per year for first 5 years
  } else {
    eff_inc <- 2 # 100% increase after 5 years
  }
  rec@Effort <- eff_inc # populate Effort slot
  rec
}
class(MOnly_incE) <- 'MP'

```

### 2.3.3 Increased Size Limit

CDFW proposed evaluating the impact of an increased size limit for red crab - increasing from 108 mm to 140 mm.

As the stocks are independent, this MP can be applied separately to the Brown and Yellow crab stocks to evaluate the impact a 140 mm size limit would have on these stocks.

Again, two MPs are defined:

```

newSL_curE <- function(x, Data, ...) {
  rec <- new('Rec') # create recommendation object
  # set retention to ~138
  rec@LR5 <- 138
  rec@LFR <- 140

  rec@Effort <- 1 # populate Effort slot
  rec
}
class(newSL_curE) <- 'MP'

newSL_incE <- function(x, Data, ...) {
  rec <- new('Rec') # create recommendation object
  # set retention to ~138
  rec@LR5 <- 138
  rec@LFR <- 140

  curYr <- max(Data@Year) - Data@LHYear # current year
  if (curYr <=5) {
    eff_inc <- 1 + 0.2 * curYr # effort increases 20% per year for first 5 years
  } else {
    eff_inc <- 2 # 100% increase after 5 years
  }
  rec@Effort <- eff_inc # populate Effort slot
  rec
}

```

```
}
class(newSL_incE) <- 'MP'
```

### 2.3.4 Effort Limit

The final MP examines the impact of a reduction in fishing effort. The MP has been defined to reduced fishing effort in the future years by an arbitrary 25% from the current level.

Similar to the previous MPs, as the stocks are independent this MP can be applied separately to the Brown and Yellow crab stocks to evaluate the impact of reduced fishing effort on these stocks.

```
reduceE <- function(x, Data, ...) {
  rec <- new('Rec') # create recommendation object

  rec@Effort <- 0.75 # populate Effort slot
  rec
}
class(reduceE) <- 'MP'
```

## 2.4 Run Forward Projections

The forward projections are done with the ProjectMOM function. This function requires the historical simulations returned by SimulateMOM (class multiHist) and a list of management procedures.

Some of the MPs are stock-specific (ie. MOnly\_curE and MOnly\_incE) so the MPs must be specified as a stock-specific list:

```
MPlist <- list(
  # MPs for female stock
  c('curE', 'incE', 'newSL_curE', 'newSL_incE', 'MOnly_curE', 'MOnly_incE', 'reduceE'),
  # MPs for male stock
  c('curE', 'incE', 'newSL_curE', 'newSL_incE', 'curE', 'incE', 'reduceE')
)
```

Notice that the curE and incE MPs are repeated for the male stock in the same position that the female stock has the no-retention MPs.

```
# load the saved historical simulations
RCrabHist <- readRDS(file.path(OMdir, 'RCrabHist.rda'))
RedCrab_MSE <- ProjectMOM(RCrabHist, MPs=MPlist)
```

```
## Bystock mode: you have specified a vector of MPs for each stock, but
## not a vector of MPs for each stock and fleet. The catch data for these
## fleets will be combined, a single MP will be used to set a single TAC
## for all fleets combined that will be allocated between the fleets
## according to recent catches
```

```
## ----- 1 / 7 MPs, Running MSE for:
```

```
##          RRCF
## Traps curE
##          RRCM
## Traps.1 curE
```

```
## -----
```

```
## .....
## ----- 2 / 7 MPs, Running MSE for:
```

```

##          RRCF
## Traps incE
##          RRCM
## Traps.1 incE

## -----
## .....

## ----- 3 / 7 MPs, Running MSE for:
##          RRCF
## Traps newSL_curE
##          RRCM
## Traps.1 newSL_curE

## -----
## .....

## ----- 4 / 7 MPs, Running MSE for:
##          RRCF
## Traps newSL_incE
##          RRCM
## Traps.1 newSL_incE

## -----
## .....

## ----- 5 / 7 MPs, Running MSE for:
##          RRCF
## Traps MOnly_curE
##          RRCM
## Traps.1 curE

## -----
## .....

## ----- 6 / 7 MPs, Running MSE for:
##          RRCF
## Traps MOnly_incE
##          RRCM
## Traps.1 incE

## -----
## .....

## ----- 7 / 7 MPs, Running MSE for:
##          RRCF
## Traps reduceE
##          RRCM
## Traps.1 reduceE

## -----
## .....

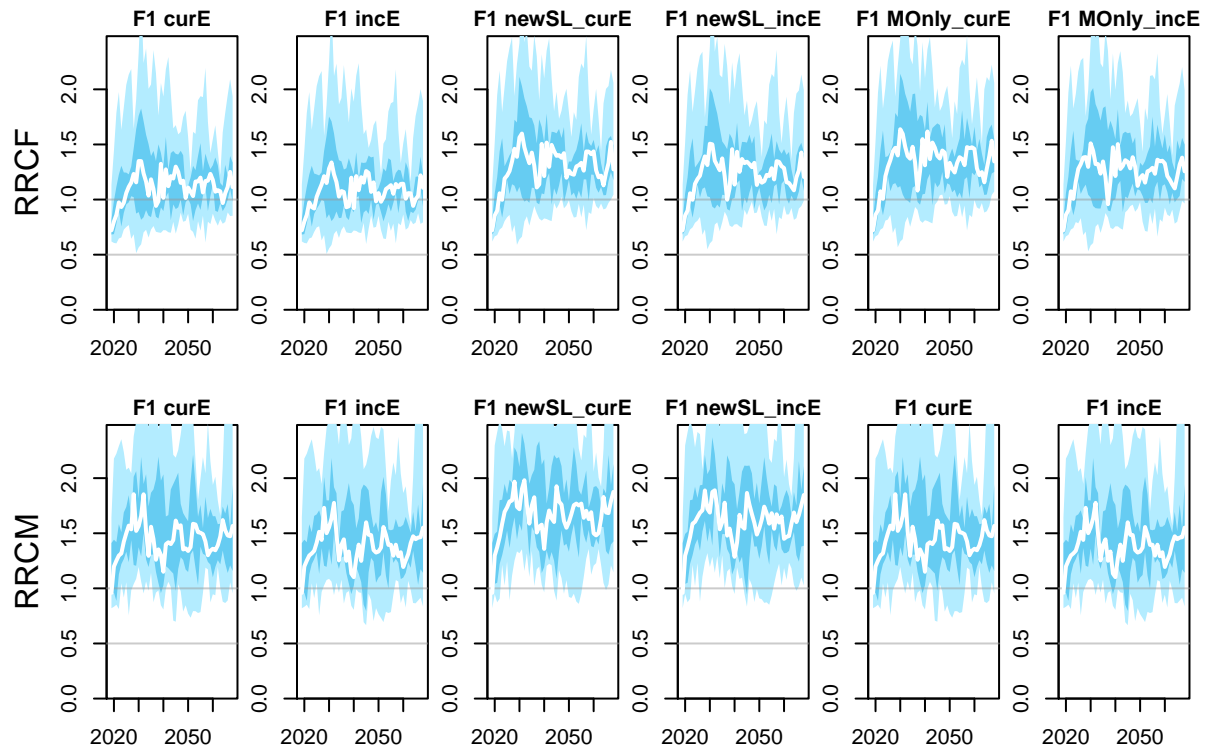
```

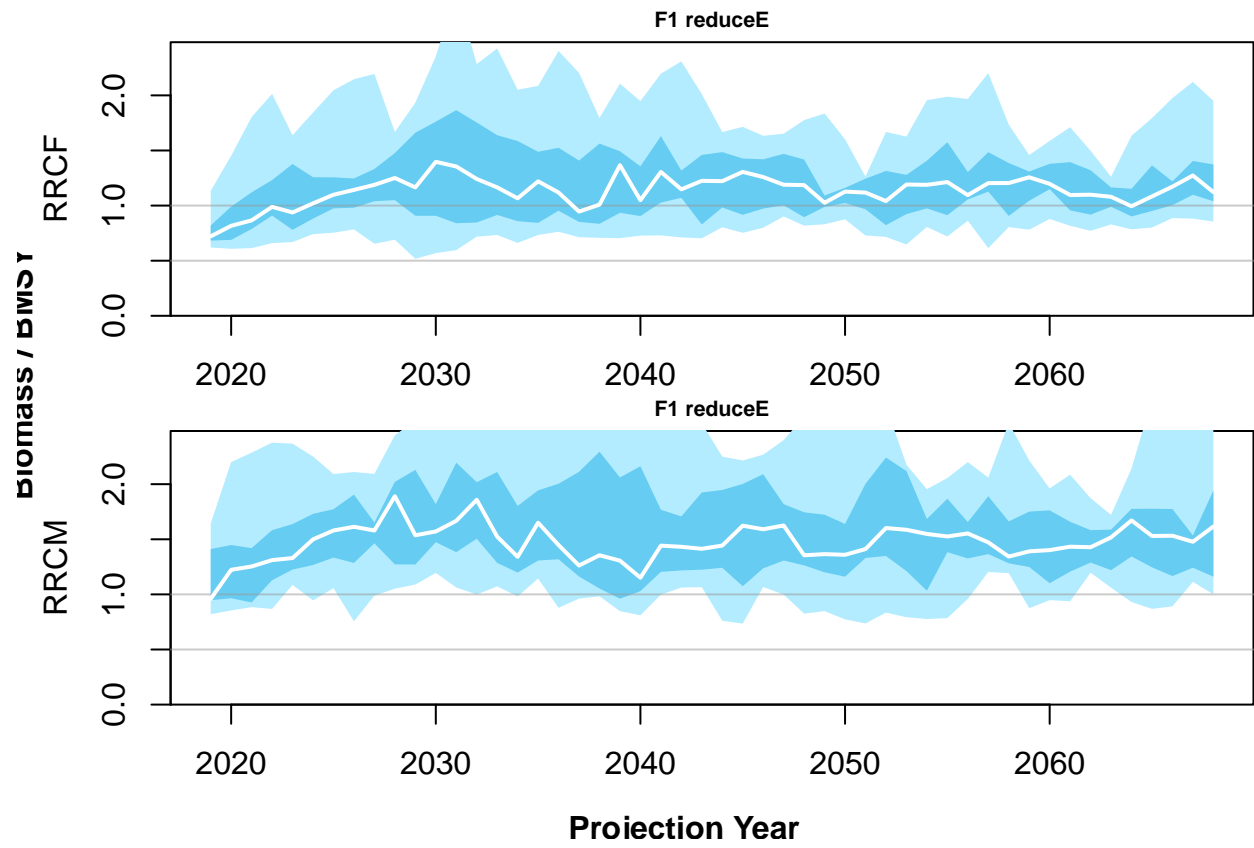


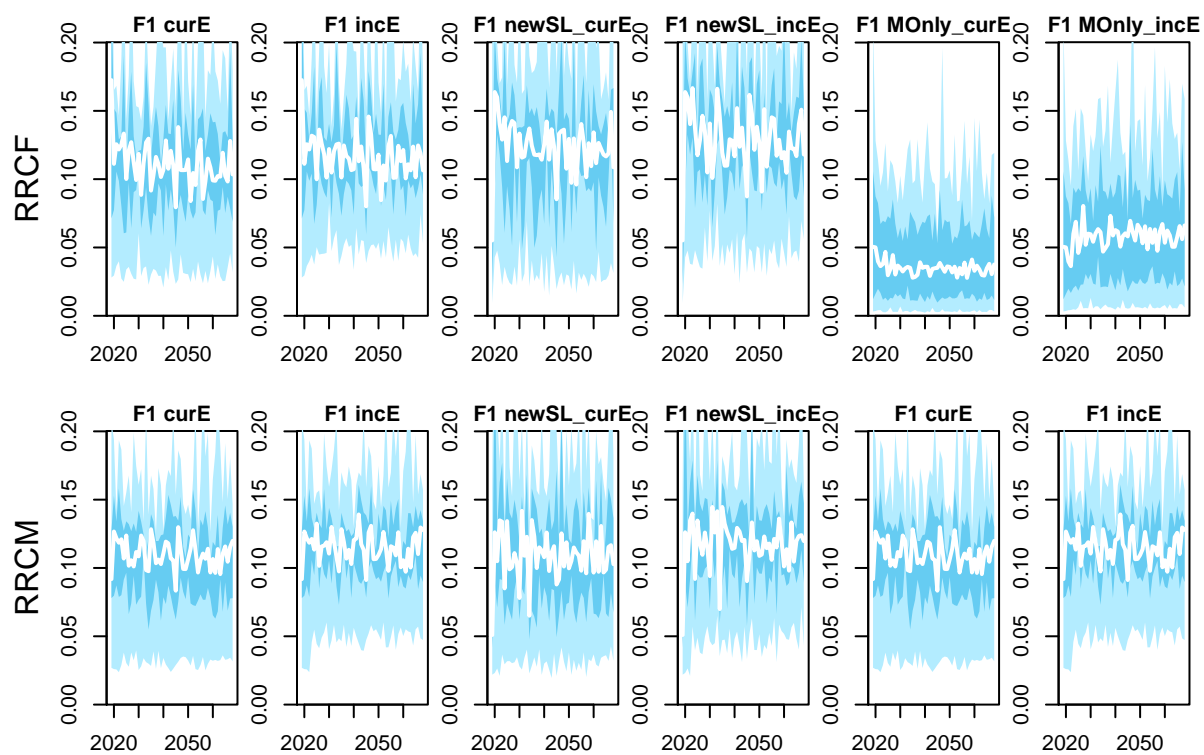
```
saveRDS(RedCrab_MSE, file.path(OMdir, 'RedCrab_MSE.rda'))
```

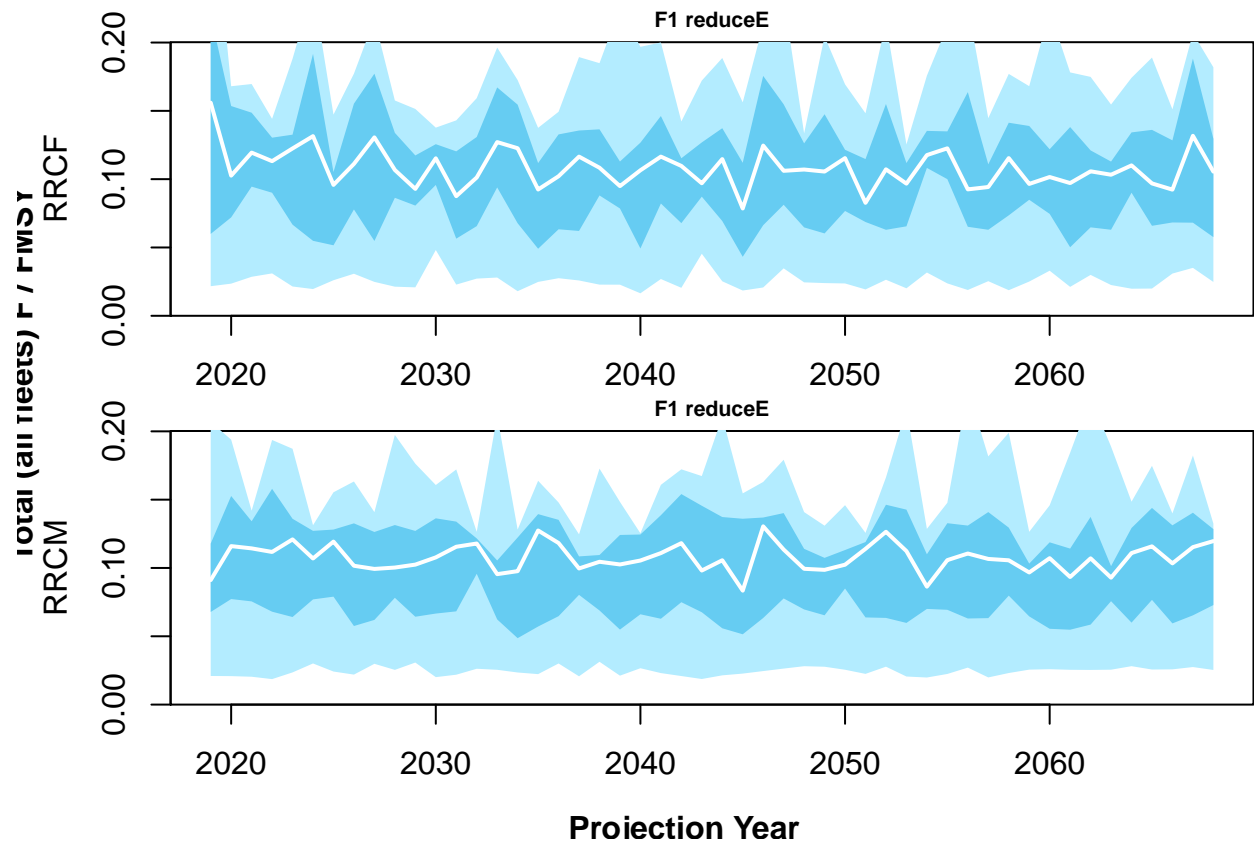
## 2.5 Examine Results

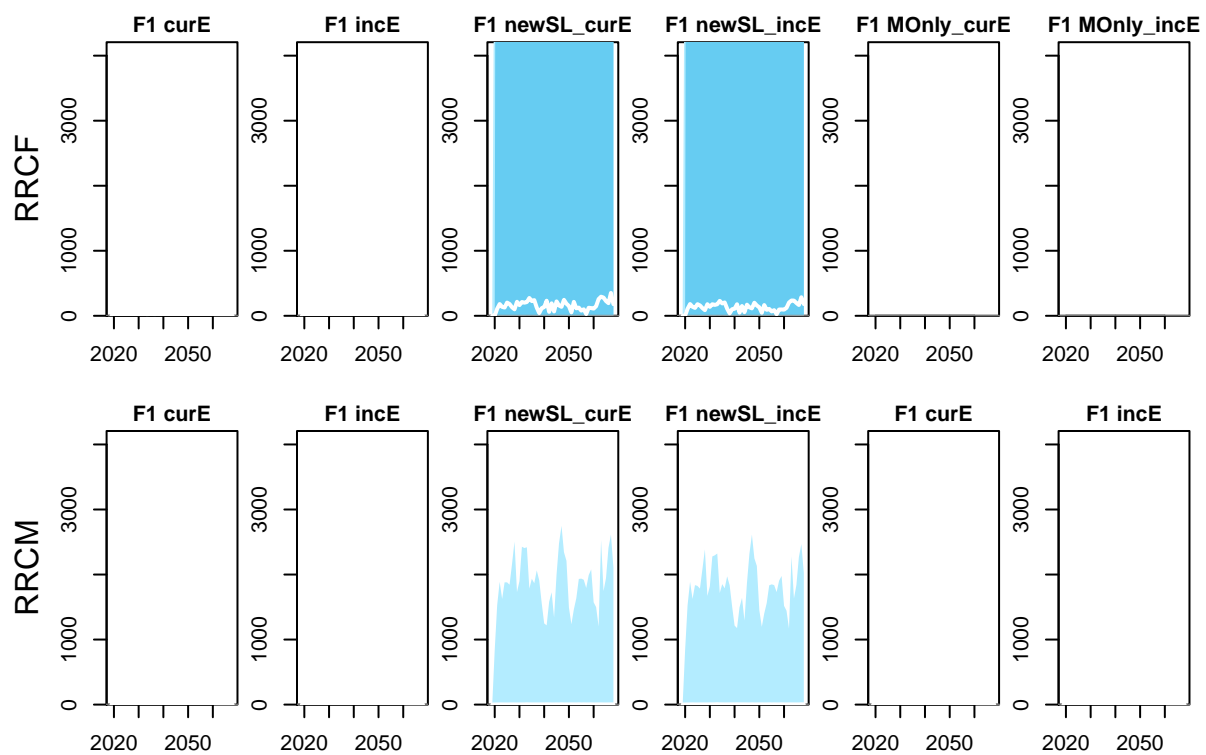
```
plot(RedCrab_MSE)
```

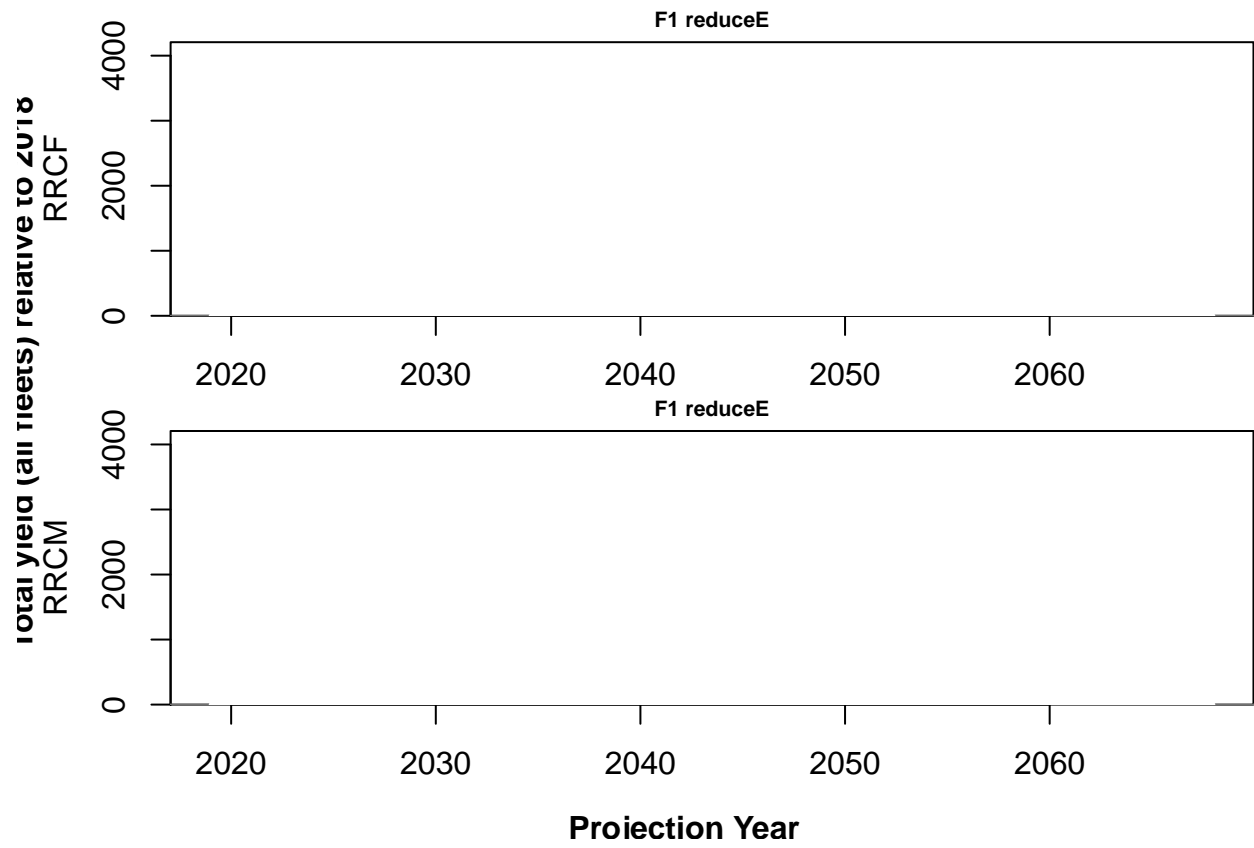


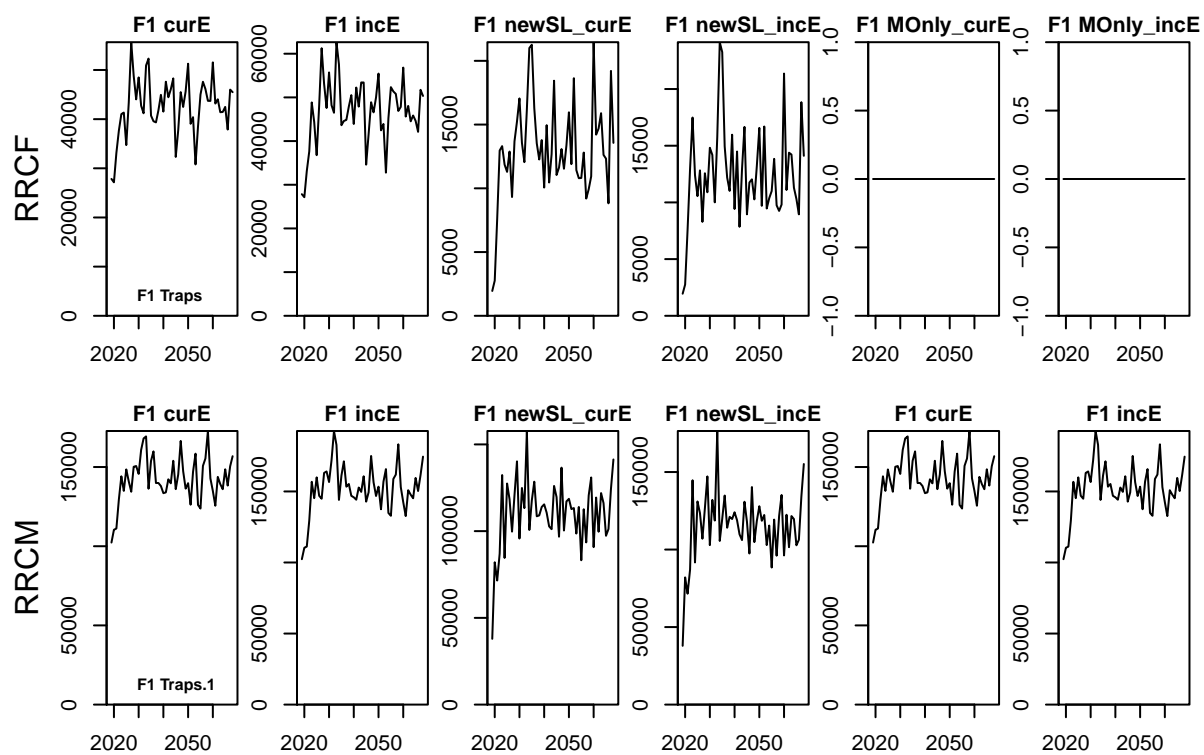


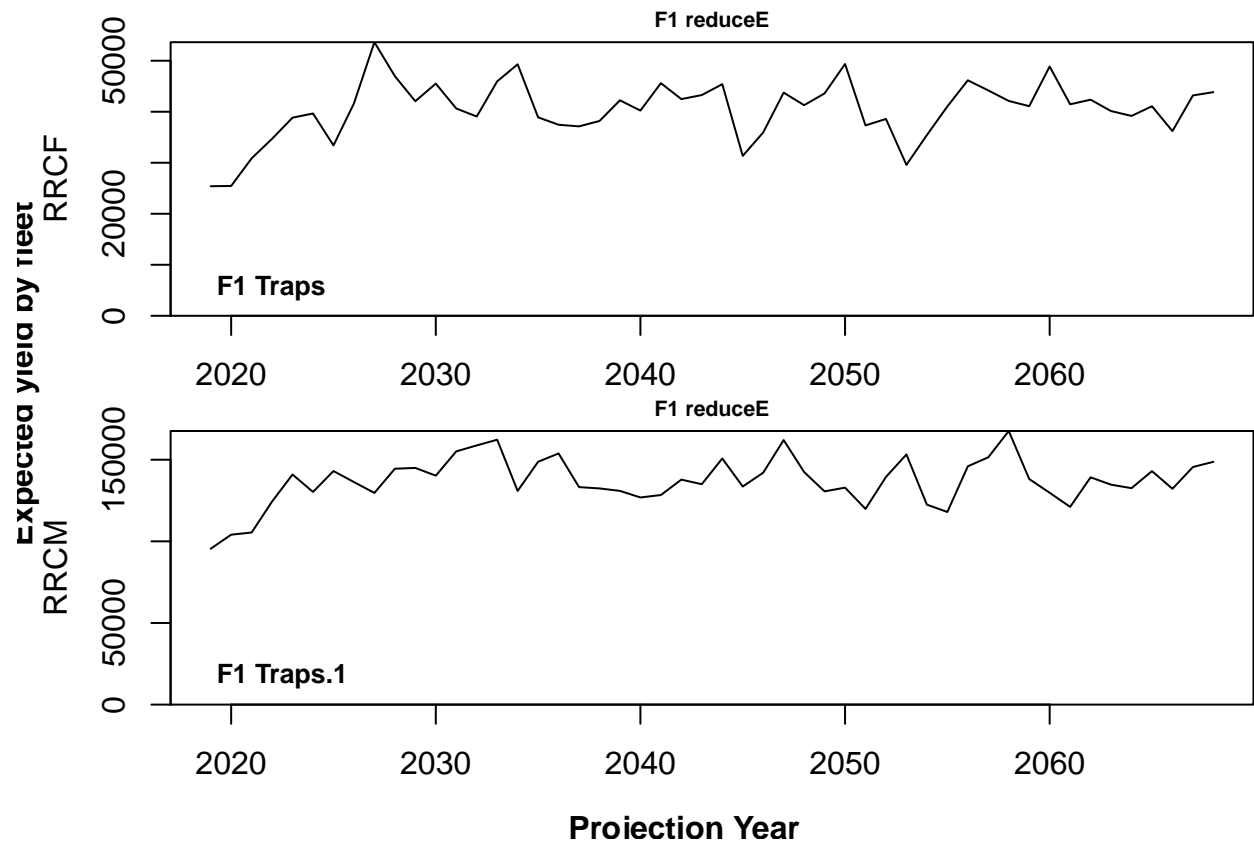












3 Lobster Case Study

4 Halibut Case Study