CINAR MSE Workshop: Model-based Estimation Model

Gavin Fay

January 2022

Replacing our assessment with an estimation model

In the previous examples we used our survey index as the estimate of abundance.

Here we will instead use a model-based estimator, and fit a Schaefer model to the data from the OM to estimate MSY reference points for use in the control rule. (hint we can reuse the code we developed to condition the OM)

1. Build a function for fitting an estimation model to the index.i time series (don't fit to the future time steps that haven't happened yet)
2. Calculate the TAC based on the last year's (model-estiamted) biomass and the estimate of FMSY (r/2) from the fitted model.
3. run a FMSY based control rule

Stretch-goal

4. Compare the performance of the EM-based strategy when the OM is the Pella-Tomlinson model.

```
#install.packages('ggplot2')
library(tidyverse)
library(ggdist)
library(Hmisc)
library(mvtnorm)
```

first load packages

load the functions we created yesterday (sourced from a script to make things easy and reduce clutter in this document)

```
source("mse-functions.R")
```

First task is to create a function for the estimation moodel that will use the index data and call the 'assess()' function.

Setting up the Operating Model load data

```
data.years <- 1991:2013
harvest <- c(0.1,3,15,52,76,139,95,93,84,93,86,103,104,
             92,46,67,59,30,54,59,47,33,44)
index <- c(NA,NA,NA,NA,NA,NA,NA,NA,935,NA,1057,NA,678,NA,
           420,NA,554,NA,458,NA,474,NA,280)
```

Load the conditioned OM

```r
schaefer_fit <- readRDS("schaefer_om.rds")
pars.iter <- schaefer_fit$pars
biomass.iter <- schaefer_fit$bio
```

Applying the Management Strategy Define the years for the projection:

```r
proj.years <- 2014:2034
```

conduct the evaluation with the model-based estimation method

```r
project.emhcr <- em_evaluate(pars.iter, biomass.iter, control.pars,
                        data.years, proj.years, iterations=3, true.m = 2, est.m = 2)
```

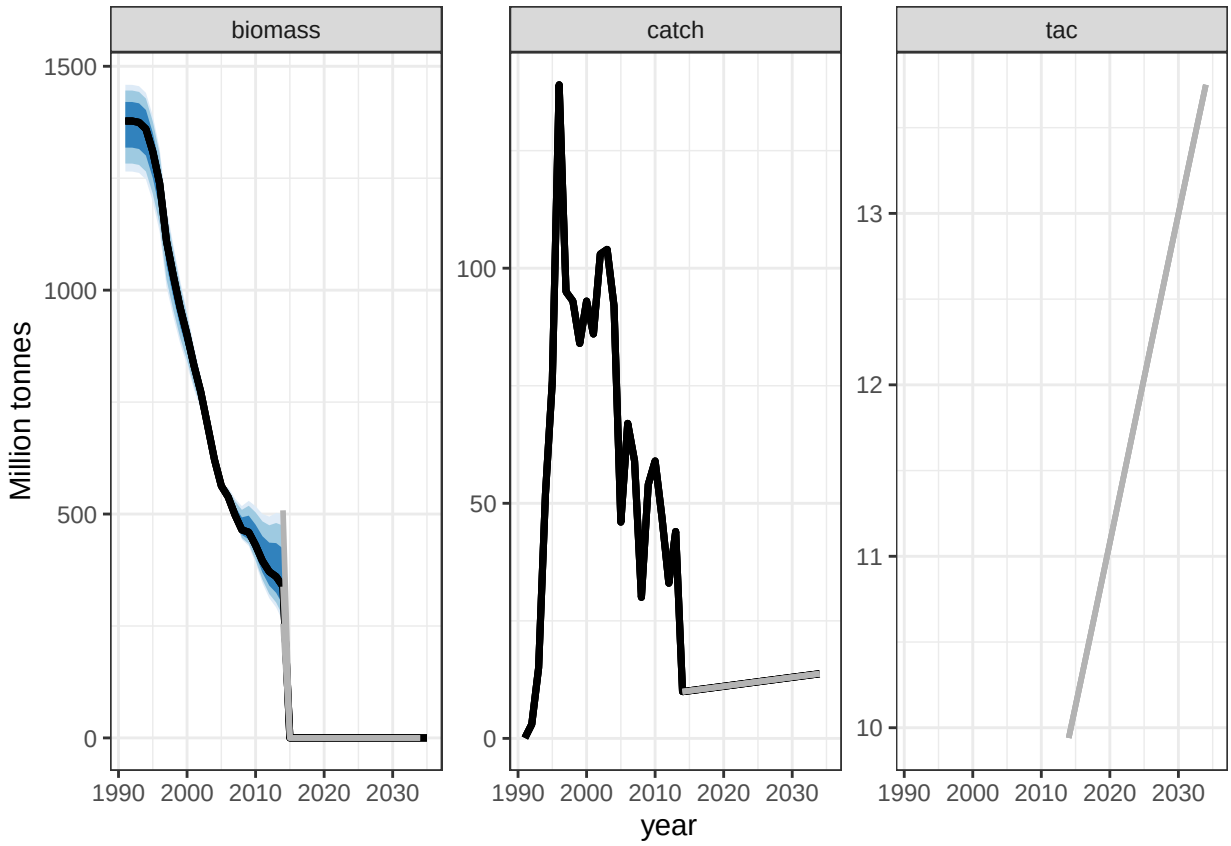
```
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
```

```
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
```

```
## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced

## Warning in rnorm(1, -0.5 * sigma^2, sigma): NAs produced
```

Plot the trajectories:

```
projection.plot(project.emhcr)
```


biomass
catch
tac
Million tonnes
1500
1000
500
0
100
50
0
13
12
11
10
1990
2000
2010
2020
2030
year