Cross test using Operating Model based on Life History

L Kell

23 June, 2018

OEM

```
setGeneric("cpue", function(object, ...) standardGeneric("cpue"))
```

```
[1] "cpue"
```

```
#' @rdname cpue
#' @aliases cpue,FLStock-method
setMethod("cpue", signature(object = "FLStock"), function(object, sel = catch.sel(object),
    effort = c("f", "h"), mass = TRUE) {

    if (effort[1] == "h")
        E <- catch(object)%/%stock(object) else E <- fbar(object)

    cpue <- (catch.n(object) %*% sel)%/%E

    if (mass)
        cpue <- cpue * catch.wt(object)

    return(cpue)
})
```

```
[1] "cpue"
```

```
# survey
setGeneric("survey", function(object, ...) standardGeneric("survey"))
```

```
[1] "survey"
```

```
setMethod("survey", signature(object = "FLStock"), function(object, sel = stock.n(object) %=%
    1, wt = stock.wt(object), timing = 0.5, mass = FALSE) {

    timing = pmax(pmin(timing, 1), 0)

    stock.n = stock.n(object) * exp(-(harvest(object) * timing + m(object) *
        timing))

    res = stock.n %*% sel

    if (mass)
        res = res %*% wt

    return(res)
})
```

```
[1] "survey"
```

```
#### Indicators
#### ################################################################
```

```r
setGeneric("mnSwt", function(object, ...) standardGeneric("mnSwt"))
```

```
[1] "mnSwt"
```

```r
setGeneric("mnCwt", function(object, ...) standardGeneric("mnCwt"))
```

```
[1] "mnCwt"
```

```r
setGeneric("mnLwt", function(object, ...) standardGeneric("mnLwt"))
```

```
[1] "mnLwt"
```

```r
setGeneric("mnLen", function(object, ...) standardGeneric("mnLen"))
```

```
[1] "mnLen"
```

```r
setGeneric("wt2z", function(object, ...) standardGeneric("wt2z"))
```

```
[1] "wt2z"
```

```r
setGeneric("ln2z", function(object, Linf, ...) standardGeneric("ln2z"))
```

```
[1] "ln2z"
```

```r
setMethod("mnSwt", signature(object = "FLStock"), function(object) apply(stock.wt(object) *
    stock.n(object), 2:6, sum)/apply(stock.n(object), 2:6, sum))
```

```
[1] "mnSwt"
```

```r
setMethod("mnCwt", signature(object = "FLStock"), function(object) apply(catch.wt(object) *
    catch.n(object), 2:6, sum)/apply(catch.n(object), 2:6, sum))
```

```
[1] "mnCwt"
```

```r
setMethod("mnLwt", signature(object = "FLStock"), function(object) apply(landings.wt(object) *
    landings.n(object), 2:6, sum)/apply(landings.n(object), 2:6, sum))
```

```
[1] "mnLwt"
```

```r
setMethod("mnLen", signature(object = "FLStock"), function(object, a = 0.001,
    b = 3, wt = "stock.wt") mnLenFunc(object, a, b, wt))
```

```
[1] "mnLen"
```

```r
setMethod("wt2z", signature(object = "FLStock"), function(object, a = 0.001,
    b = 3, wt = "stock.wt") wt2zFunc(object, a, b, wt))
```

```
[1] "wt2z"
```

```r
setMethod("ln2z", signature(object = "numeric", Linf = "numeric"), function(object,
    Linf, Lc, k) ln2zFunc(object, Linf, Lc, k))
```

```
[1] "ln2z"
```

```r
setMethod("ln2z", signature(object = "numeric", Linf = "FLPar"), function(object,
    Linf) ln2zFunc(object, Linf["Linf"], Linf["Lc"], Linf["k"]))
```

```
[1] "ln2z"
```

```r
mnLenFunc <- function(object, a = 0.001, b = 3, wt = "stock.wt") {
    wt. = slot(object, wt)
    n. = slot(object, gsub(".wt", ".n", wt))
    apply((wt./a)^(1/b) * n., c(2, 6), sum)/apply(n., c(2, 6), sum)
}

wt2zFunc <- function(object, Linf, Lc, k, a = 0.001, b = 3, wt = "stock.wt") {
    mnSz <- mnSzStock(object, a, b, wt)
    k * (Linf - mnSz)/(mnSz - Lc)
}

ln2zFunc <- function(object, Linf, Lc, k) {
    k * (Linf - object)/(object - Lc)
}
################################################################################
```

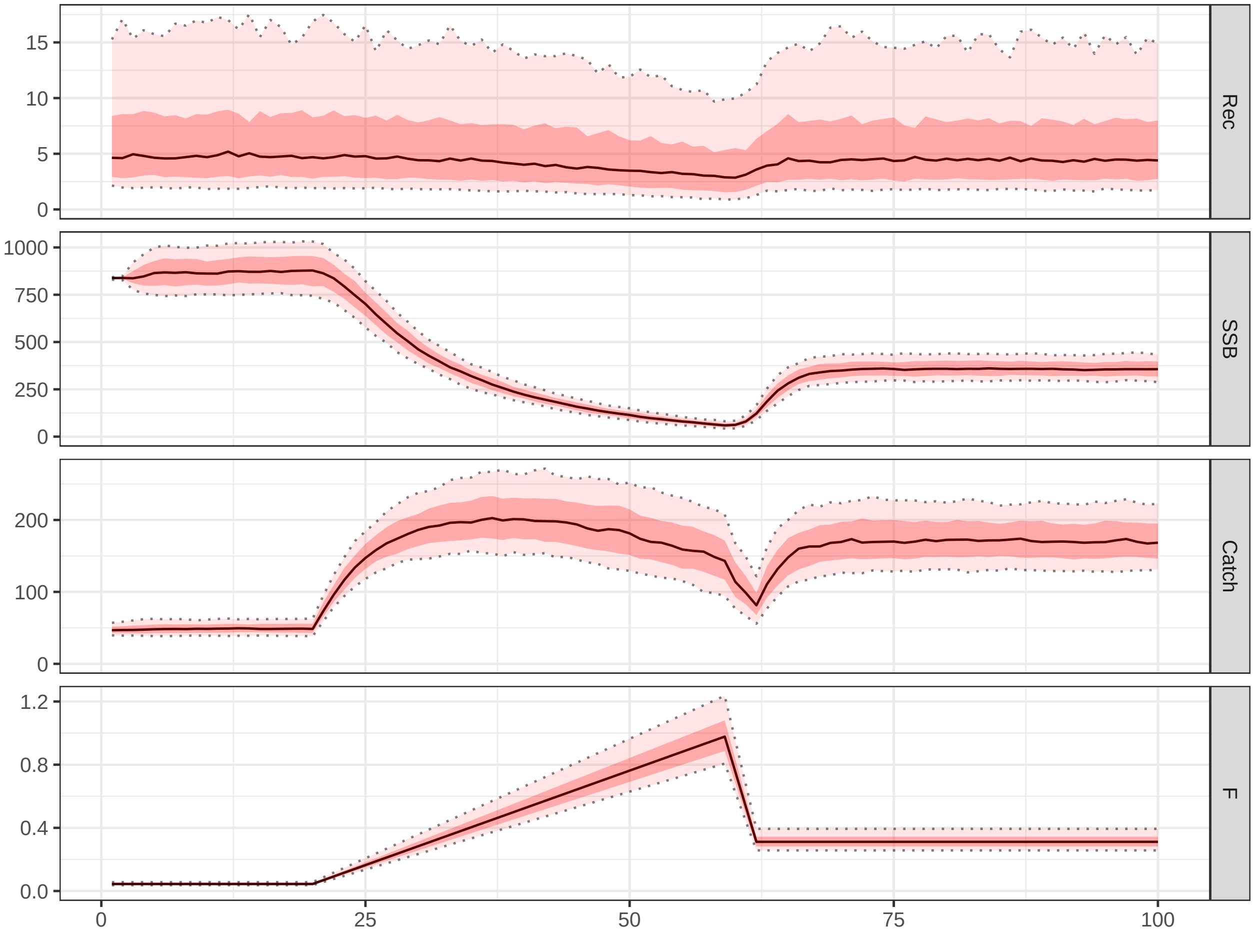

Figure 1, Operating Model Time series.

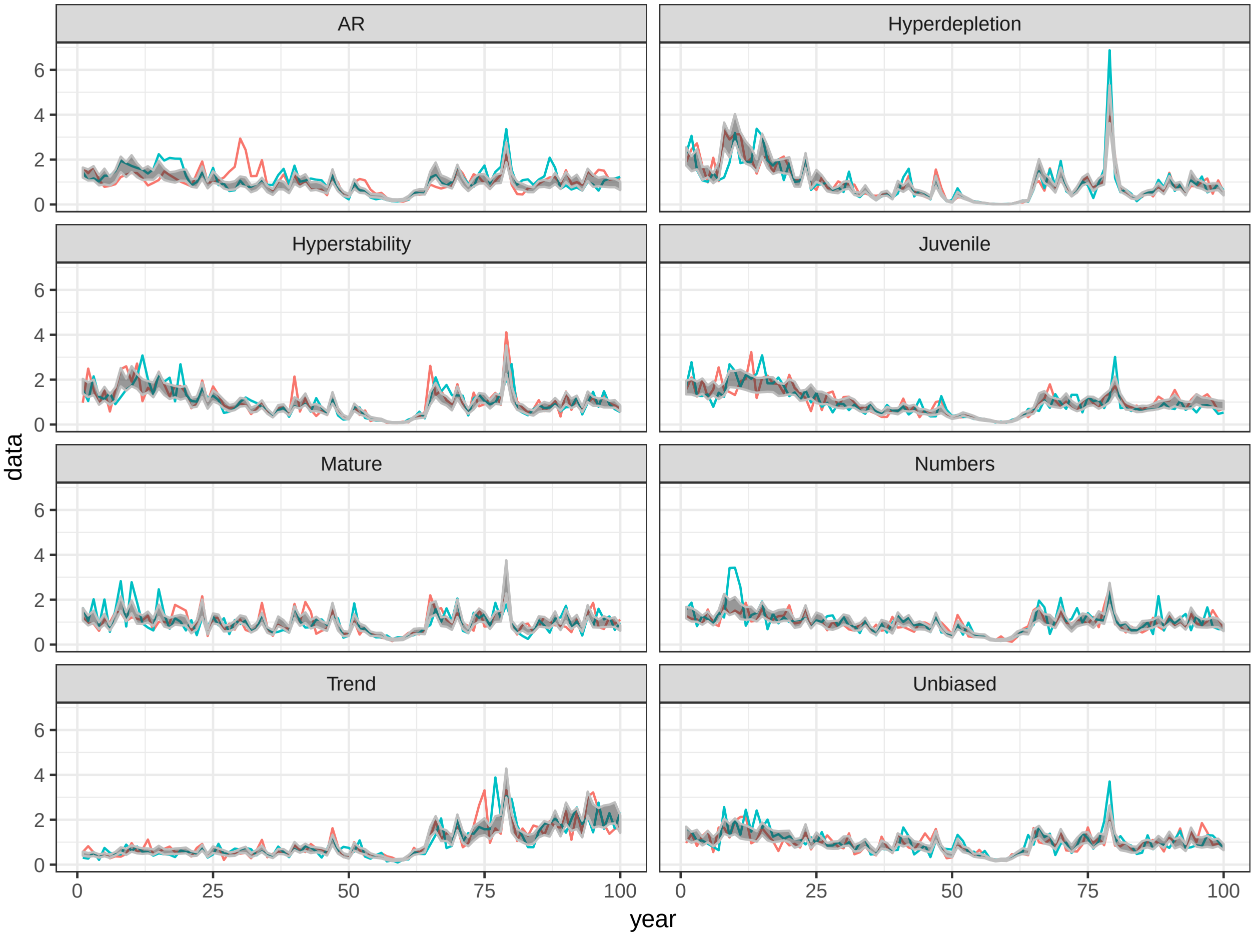

Figure 2, Commercial indices of abundance.

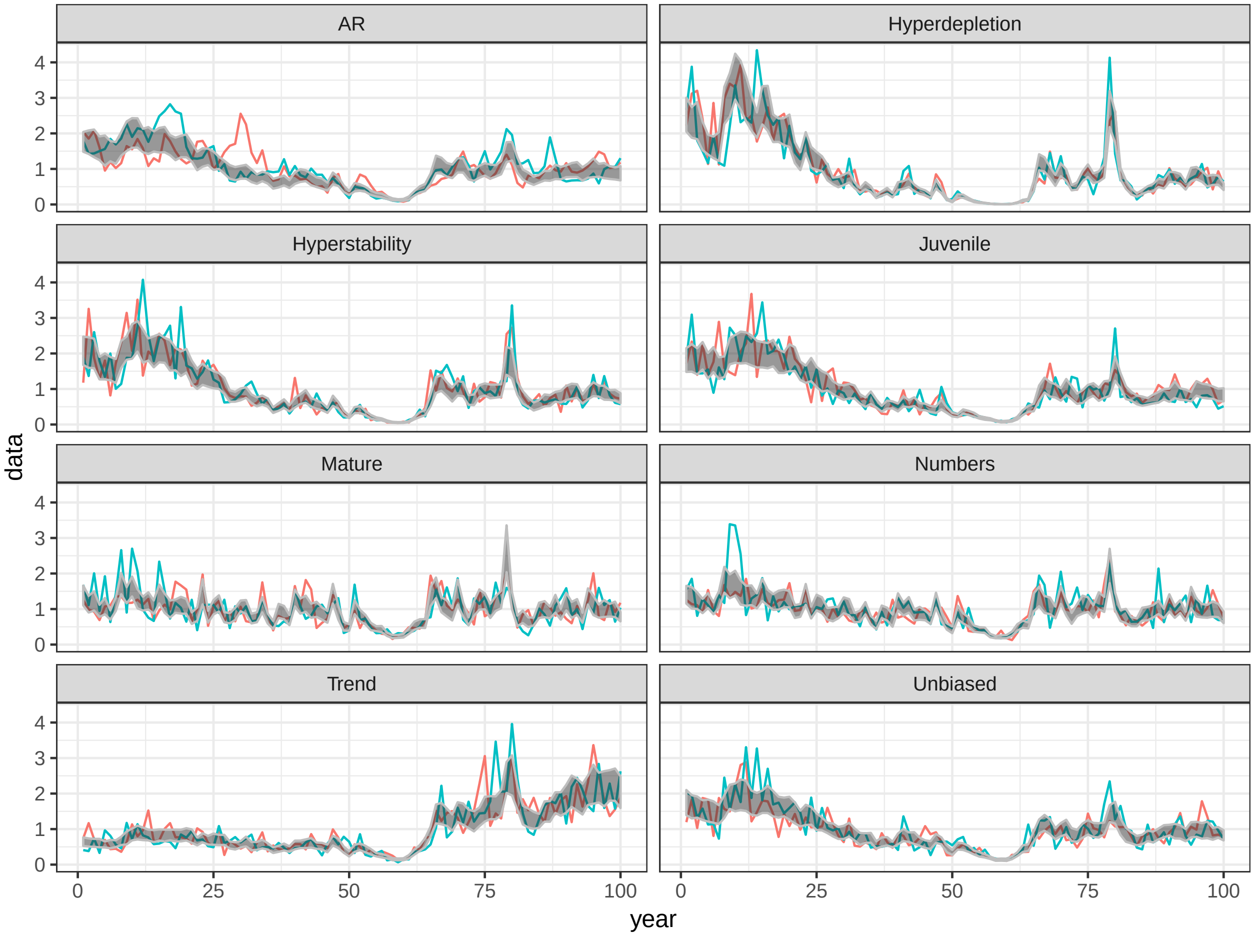

Figure 3, Survey indices of abundance.

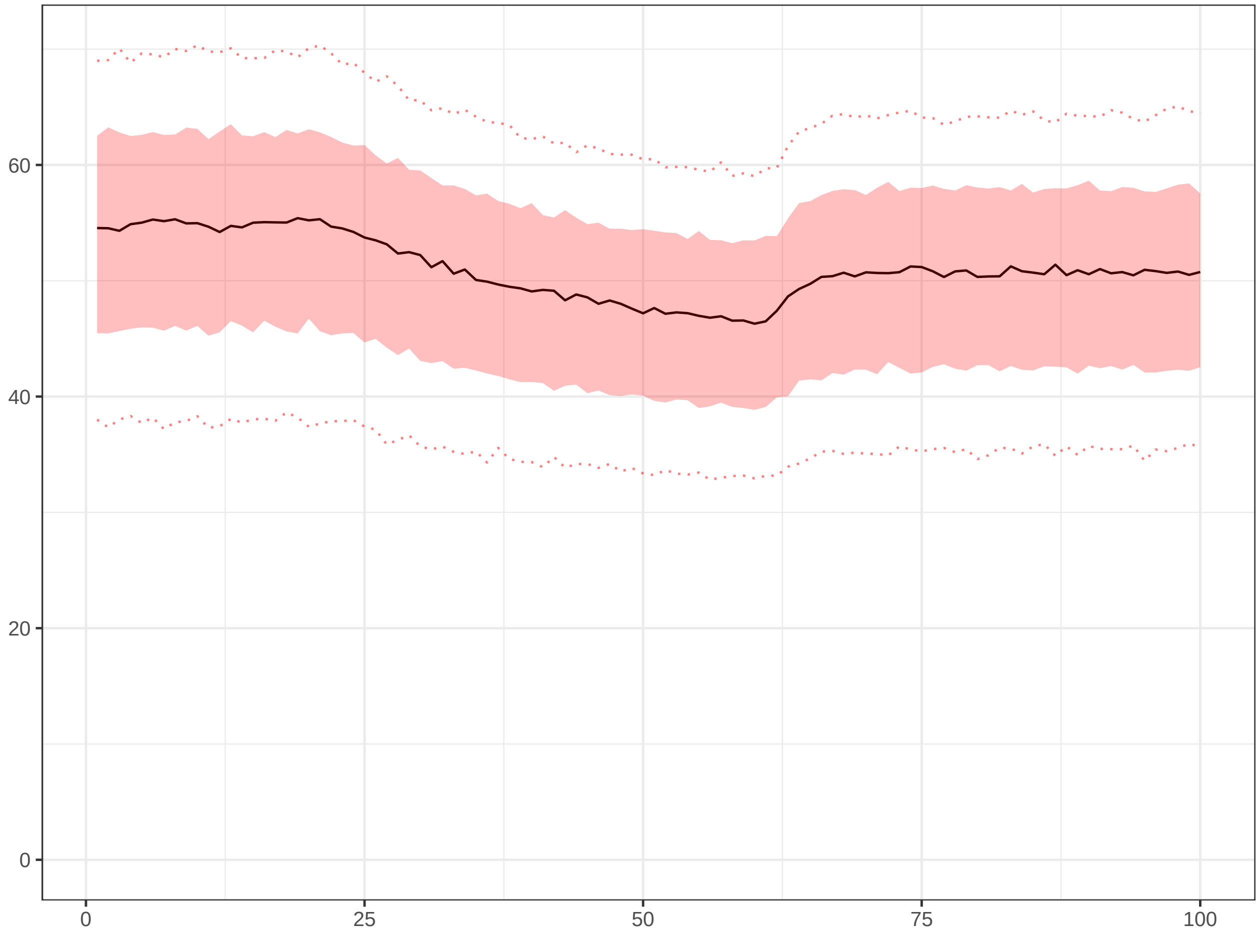

Figure 4, Index of mean length.

Priors

Lmax Amax Selectivity (S50; S95) K or B0 r Lc Lopt k Linf to Lm50 A50 M h alpha-B-H M/K Fmsy/M Fmsy/K Bmsy/K (or B0) Depletion Fecundity at age/length Length-weight relationship

```
An object of class "FLPar"
iters:  500

params
             linf                 k                t0
  41.59229(2.7446)    0.43990(0.0578)   -0.91428(0.3090)
              a50             ato95                 a
   1.18600(0.4800)    1.00000(0.0000)    0.02225(0.0000)
                b                 s                 v
   2.92000(0.0000)    0.80000(0.0000) 1000.00000(0.0000)
units:  NA
```

```
An object of class "FLPar"
iters:  500

params
              linf                  k                  t0
```

```
  41.59229( 2.7446)    0.43990( 0.0578)   -0.91428( 0.3090)
                  a                   b               ato95
   0.02225( 0.0000)    2.92000( 0.0000)    1.00000( 0.0000)
                a50                asym                  bg
   1.18600( 0.4800)    1.00000( 0.0000)    2.92000( 0.0000)
                 m1                  m2                  a1
 164.97836(26.2678)   -1.61000( 0.0000)    1.18600( 0.4800)
                 sl                  sr                   s
   2.00000( 0.0000) 5000.00000( 0.0000)    0.80000( 0.0000)
                  v                 l50                fmsy
1000.00000( 0.0000)   25.00000( 0.0000)    0.44473( 0.0648)
                msy                bmsy
 169.56749(23.8202)  222.67512( 5.8705)
units:  NA
```

```
An object of class "FLQuant"
iters:  500

, , unit = unique, season = all, area = unique

     year
quant 1
  all 1.0331(0.0796)

units:  NA
```

```
An object of class "FLQuant"
iters:  500

, , unit = unique, season = all, area = unique

     year
age   1
  all 0.45414(0.0506)

units:  NA
```

Stock Assessment Models

Catch only

DB-SRA: Depletion-based Stock reduction analysis (Dick and MacCall 2011)

data catch only

parameters: a50, m, Fmsy/m, bmsy/k, depletion

outputs: K, B0, bmsy, fmsy, msy Cfmsy

Length

lbspr

data length

paramters linf, m/k, W=alˆb

outputs

F/M, selectivilty at length, SPR

Catch and Cpue

biodyn