Blue whiting HCR Evaluation

Example

Laurence Kell

16 August, 2019

Introduction

This vignette provides an example of conditioning an Operating Model (OM) on a stock assessment, then runs example harvest control rules (HCRs).

Initial steps

To follow this vignette you will have to install a number of packages, either from CRAN or from www.flr-project.org where variety of tutorials are available.

Install FLR from https://www.flr-project.org/

```
install.packages(c("FLCore","FLFishery","FLasher","FLBRP","kobe","FLife"),
             repos="http://flr-project.org/R")
```

`devtools` needs to be installed and then loaded so that the `mydas` package can be installed from the GitHub repository.

```
install.packages("devtools",dependencies=TRUE)
```

```
library(devtools)

devtools::install_github("lauriekell/mydas-pkg")
```

Load libraries from CRAN

```
library(ggplot2)
library(plyr)
library(reshape)
library(rdrop2)
library(mvtnorm)
```

Load FLR libraries

```
library(FLCore)
library(FLasher)
library(FLBRP)
library(FLife)
library(mydas)
```

The data are in dropbox and can be loaded if you have read access

```
token<-drop_auth()
saveRDS(token, "token.RDS")

drop_download(path='bluewhiting/data/om.RData',local_path="data",overwrite=TRUE)

load(file.path("data","om.RData"))
```

This loads a few 'FLR' objects

```
ls()

is(om1)
```

'om1' is the blue whiting stock assessment amd 'om' is the assessment with a 1000 replicates with estimation error.

```
plot(om)
```

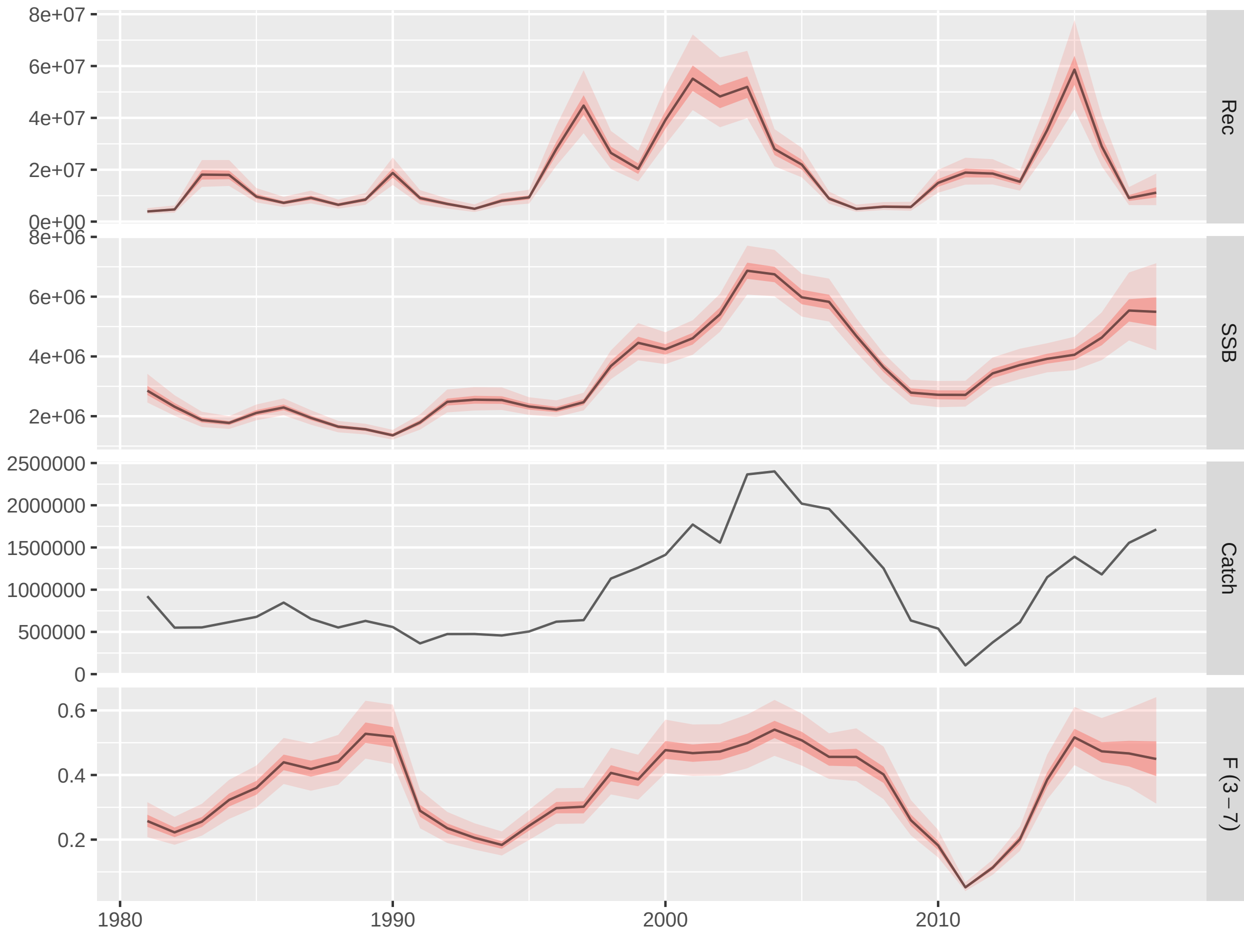

Figure 1 Blue whiting with assessment with estimation error.

Dynamics

The assessment includes the biological parameters and selection pattern. These can be extracted from the 'FLStock' object and plotted using 'ggplot2'.

```
dat=as.data.frame(FLQuants(om1,
                  stock.wt=stock.wt,
                  catch.wt=catch.wt,
                  m=m,
                  mat=mat,
                  sel=function(x)harvest(om1)%/%fbar(om1)),drop=T)

dat$qname=factor(dat$qname,levels=c("stock.wt","catch.wt","mat","m","sel"),
                           labels=c("Stock Mass",
                                    "Catch Mass",
                                    "Maturity",
                                    "M",
                                    "Selection"))
ggplot(dat)+
  geom_point( aes(age,data,col=ac(year),group=year))+
  geom_smooth(aes(age,data),method="loess")+
  facet_wrap(~qname,scale="free",ncol=2)+
  xlab("Age")+ylab("")+
  theme_bw(16)+
  theme(legend.position="none")
```

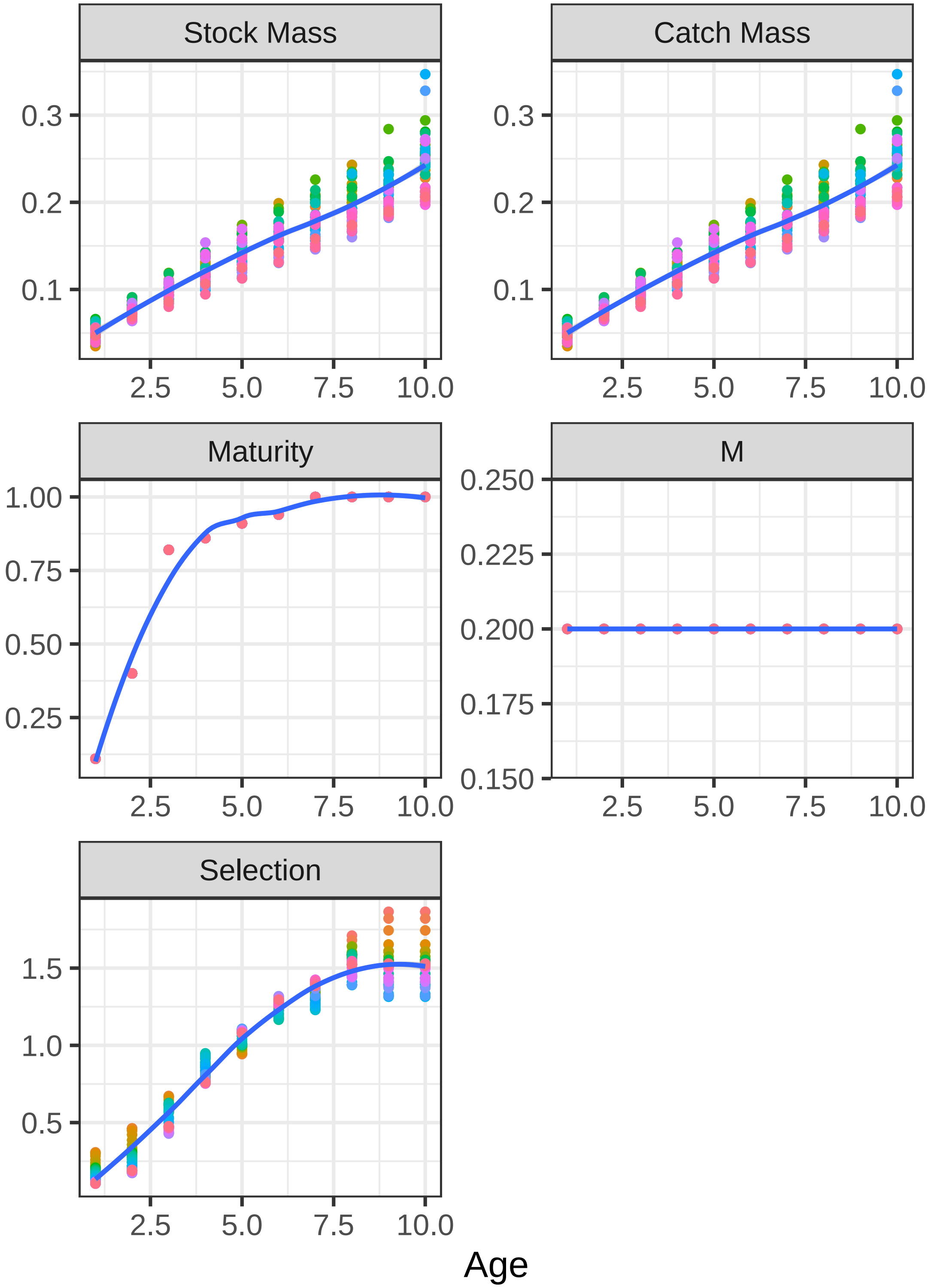

Figure 2 Vectors-at-age.

```
ggplot(data=subset(dat,(qname%in%c("Stock Mass","Selection"))))+
  geom_line(aes(year,data,group=age,col=ac(age)))+
  facet_wrap(~qname,scale="free",ncol=1)+
  xlab("Year")+ylab("")+
  theme_bw(18)+
```

```
  theme(legend.position="bottom")+
  scale_color_manual("Age",values=rainbow(10))
```

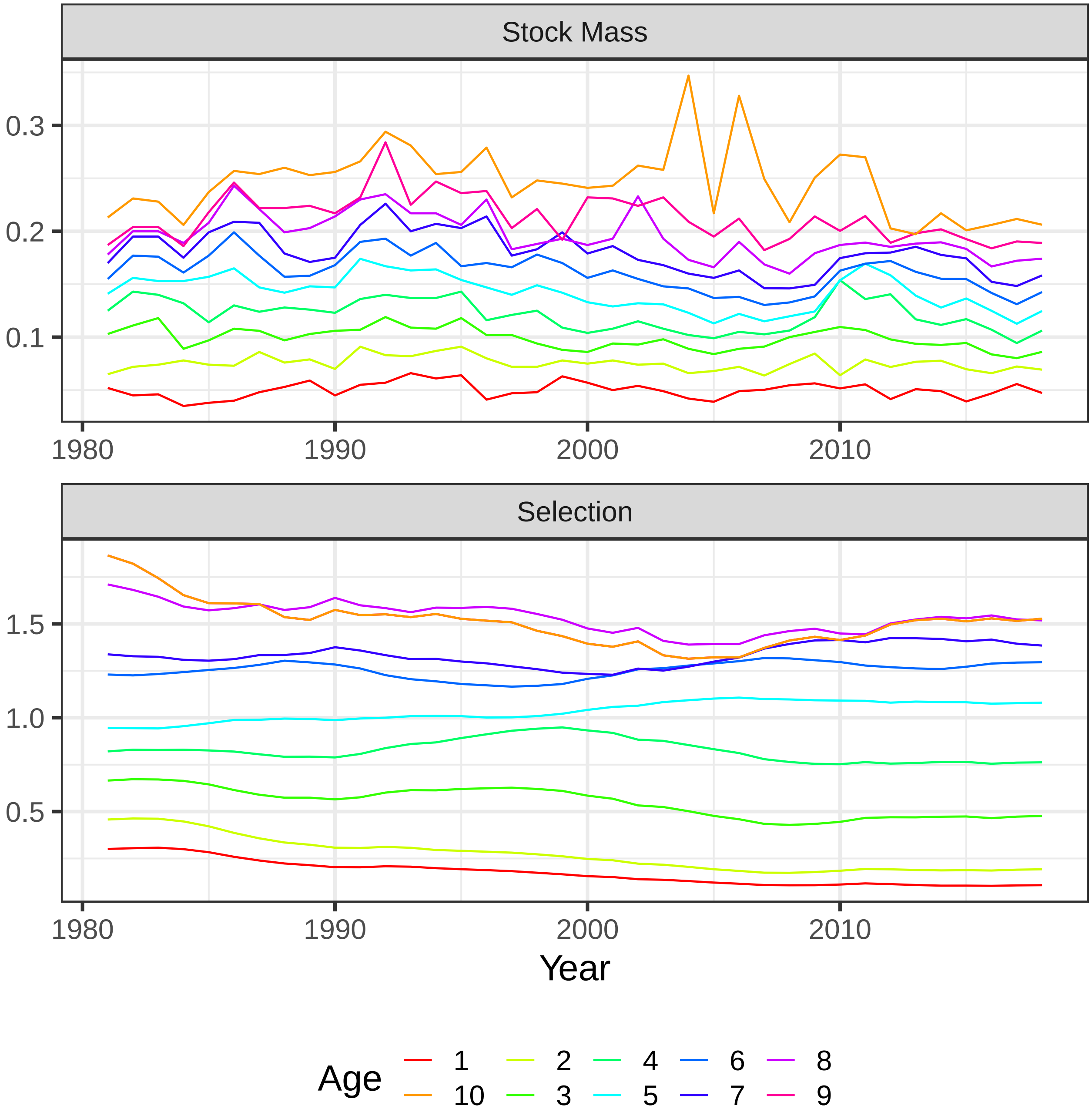

Figure 3 Time series of mass and selection-at-age

The dynamics are characterised by variability in recruitment that affects year class strength.

```
dat=stock.n(om1)%/%apply(stock.n(om1),1,mean)
ggplot(as.data.frame(dat))+
  geom_point(aes(year,age,size=data))+
  theme_bw(16)+
  theme(legend.position="none")+
  xlab("Year")+ylab("Age")
```

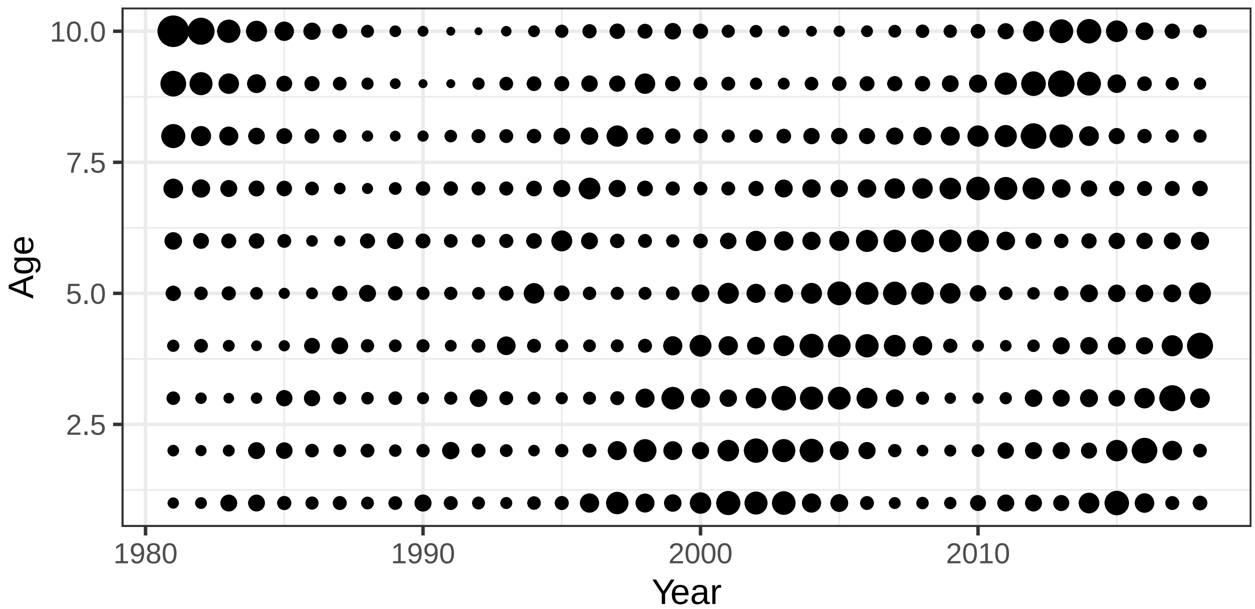

Figure 4 Cohort effects for stock numbers-at-age, scaled by mean by age.

The catch is mainly of younger age classes as shown by the exploitable numbers-at-age.

```
dat=dat%/%(harvest(om1)%/%fbar(om1))
ggplot(as.data.frame(dat))+
  geom_point(aes(year,age,size=data))+
  theme_bw(16)+
  theme(legend.position="none")+
  xlab("Year")+ylab("Age")
```

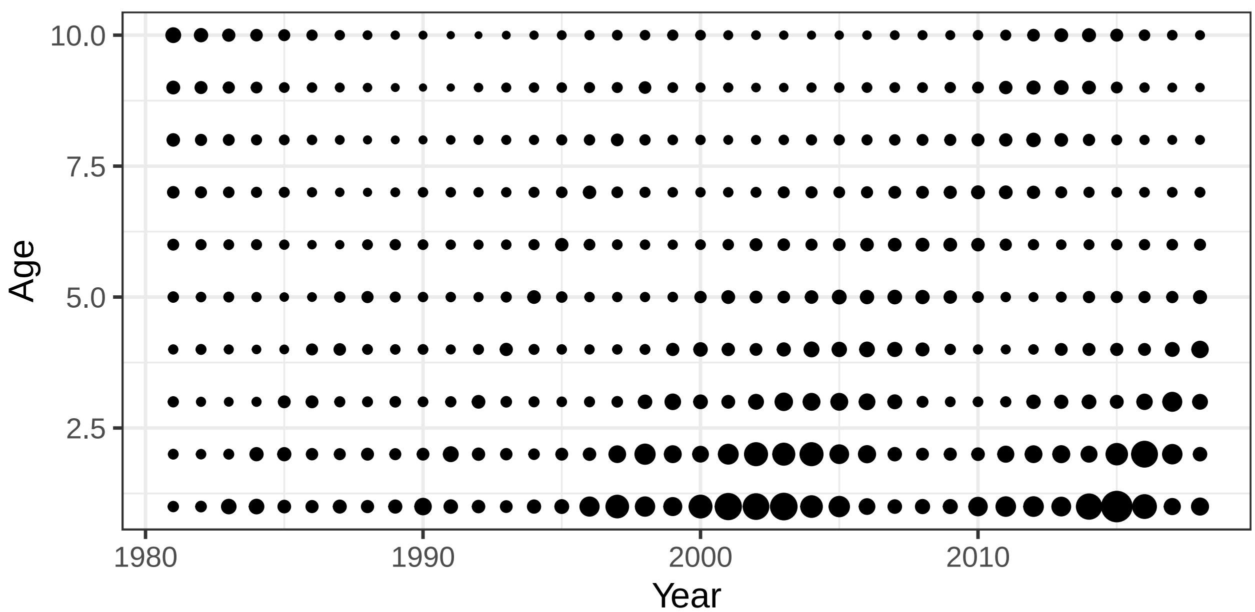

Figure 5 Cohort effects for exploitable numbers-at-age, stock numbers in Figure 4 scaled by selection pattern.

To explore whether dynamics are driven by recruitment of by SSB, the cross correlations between recruitment and SSB are plotted. If recruitment is driven by SSB then there should be a lag of +1 between SSB and N_1, however, the largest lags are negative indicating that SSB is driven by incoming year classes.

```
ccf(c(rec(om1)),c(ssb(om1)),main="")
```

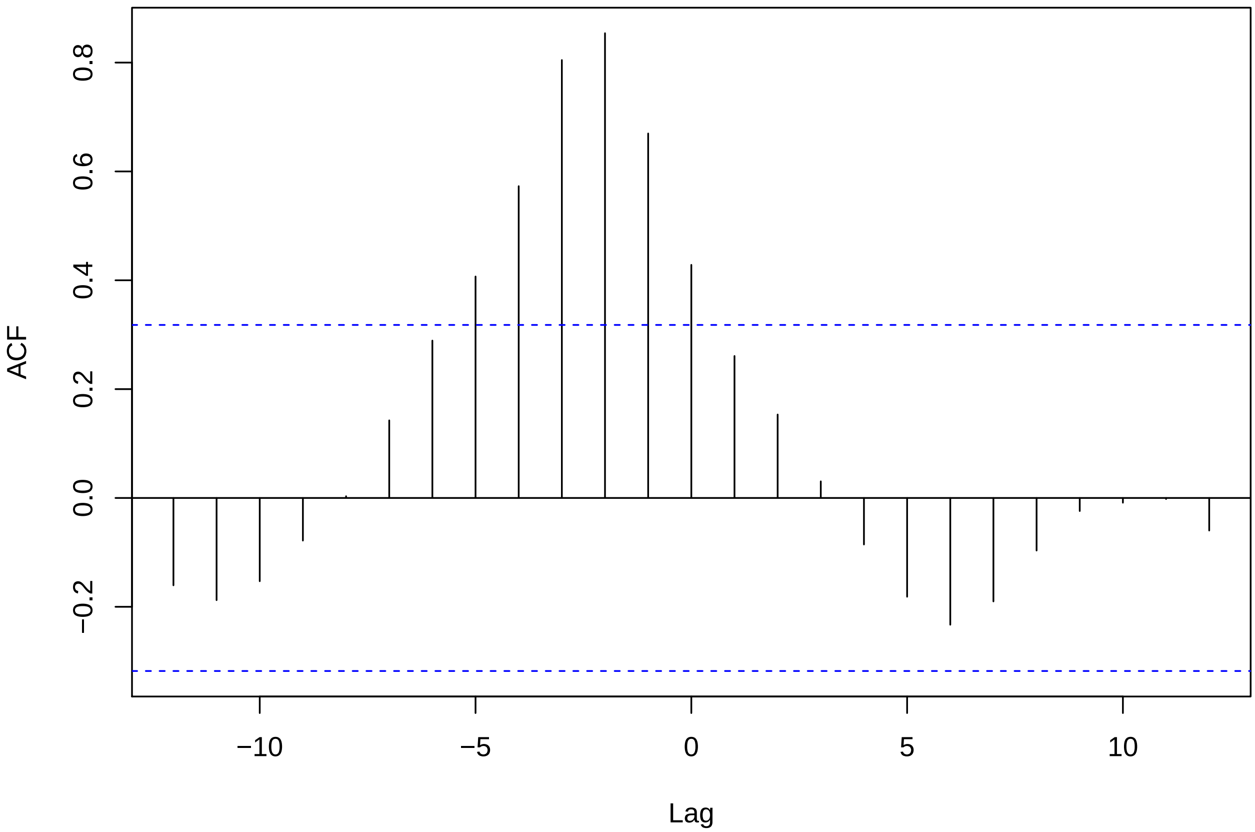

Figure 6 Cross correlation between recruitment and SSB.

A similar phenomenon is seen for exploitable biomass, but the lag is less due to the fishery exploiting immature ages.

```
esb    =function(x) apply((stock.n(x)%*%stock.wt(x))%/%
                                   (harvest(x)%/%fbar(x)),c(2,6),sum)
ccf(c(rec(om1)),c(esb(om1)),main="")
```

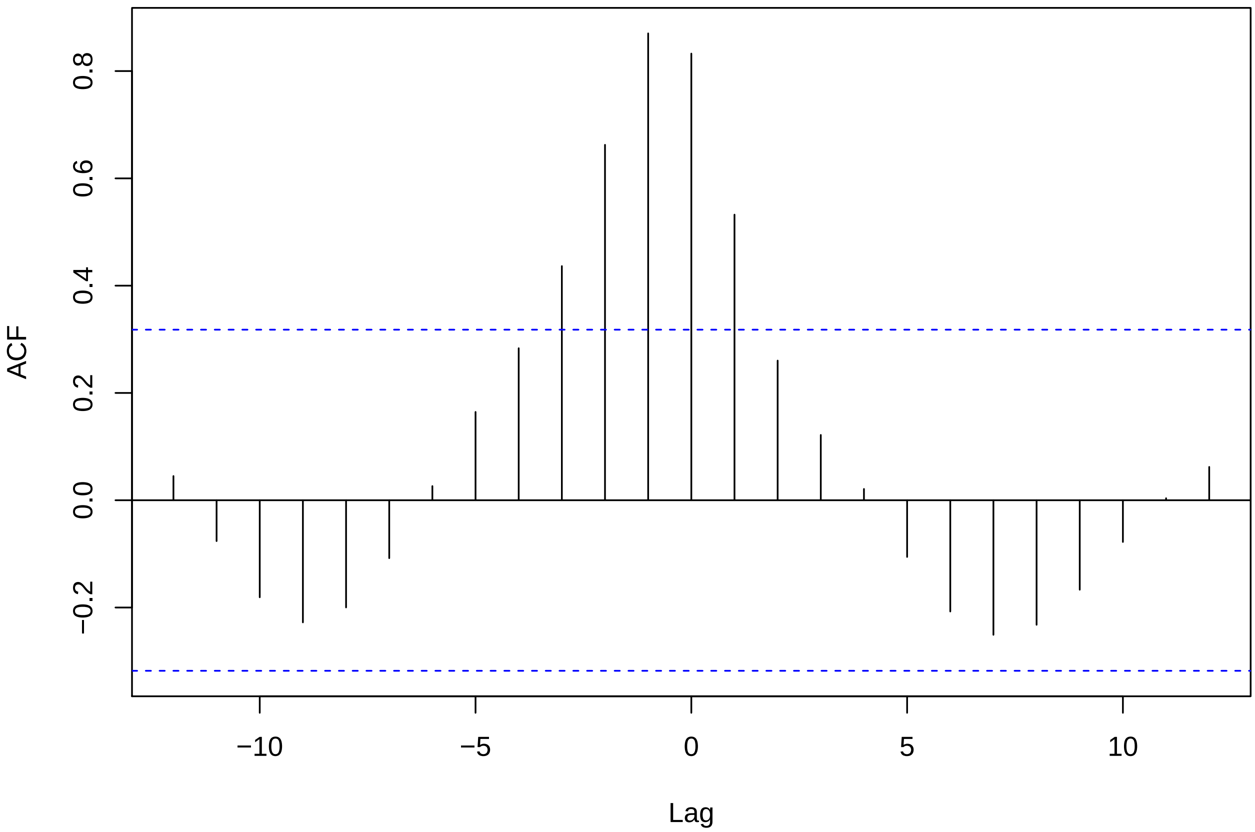

Figure 7 Cross correlation between recruitment and exploitable biomass.

Stock recruitment relationships

A Beverton and Holt stock recruitment relationship is fitted to the SSB-recruitment pairs.

```
sr=fmle(as.FLSR(om1,model="bevholt"),control=list(silent=TRUE))

ggplot(model.frame(FLQuants(rec    =fitted(sr),
                            ssb    =ssb(sr),
                            rec.obs=rec(sr)),drop=T))+
  geom_line( aes(ssb,rec))+
  geom_point(aes(ssb,rec.obs),col="red")+
  theme_bw(16)+
  expand_limits(x=0,y=0)+
  xlab("SSB")+ylab("Recruitment")
```

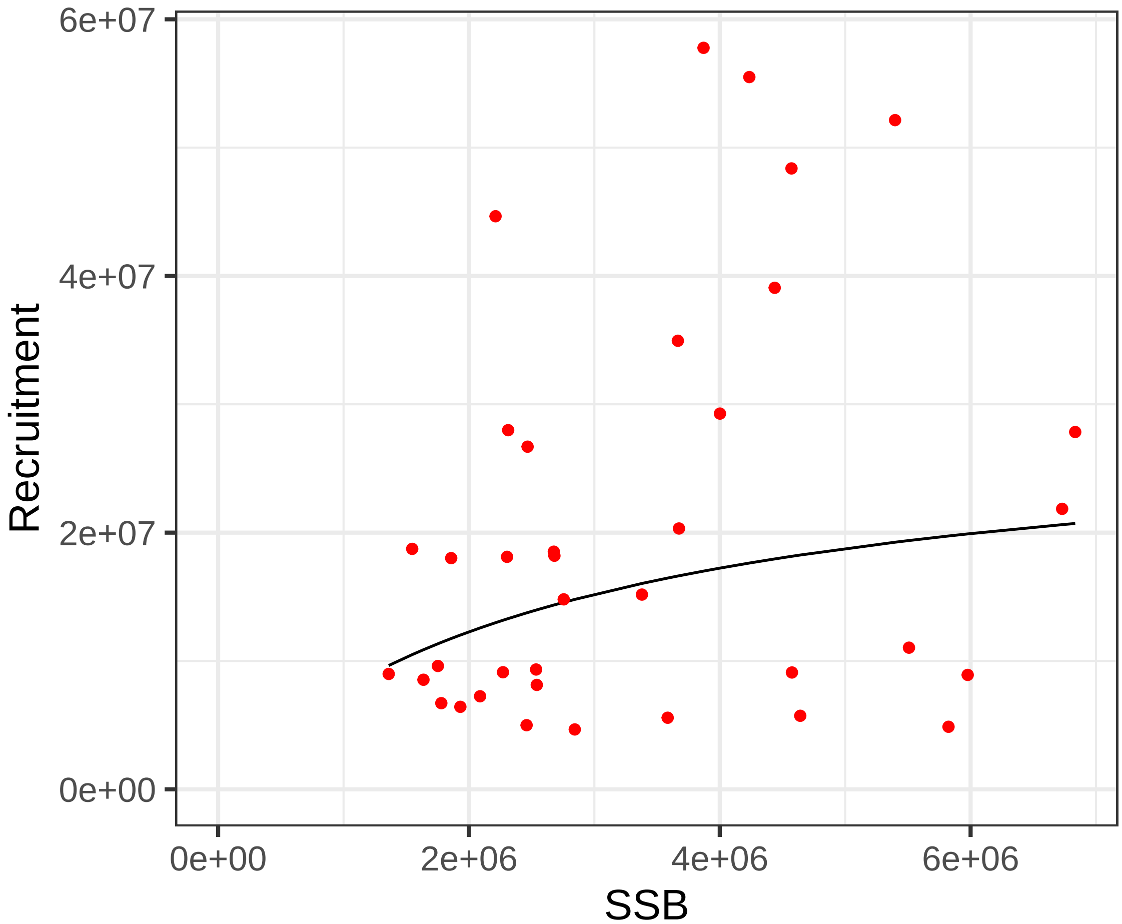

Figure 8 Beverton and Holt Stock recruitment relationship

There are stong patterns in the residuals and large recruitments can be found at low values of SSB.

```
ggplot(as.data.frame(residuals(sr),drop=TRUE))+
      geom_point(aes(year,data),col="grey25")+
      geom_line( aes(year,data),col="grey75")+
    geom_polygon(aes(year,data,group=regime),
          fill="lavender",col="blue",
          lwd=.25,alpha=.5,
          data=FLife:::rod(residuals(sr)))+
      xlab("Year")+ylab("Residual")+
      theme_bw(16)+
      theme(axis.text.x=element_text(angle=-30),legend.position="bottom")
```

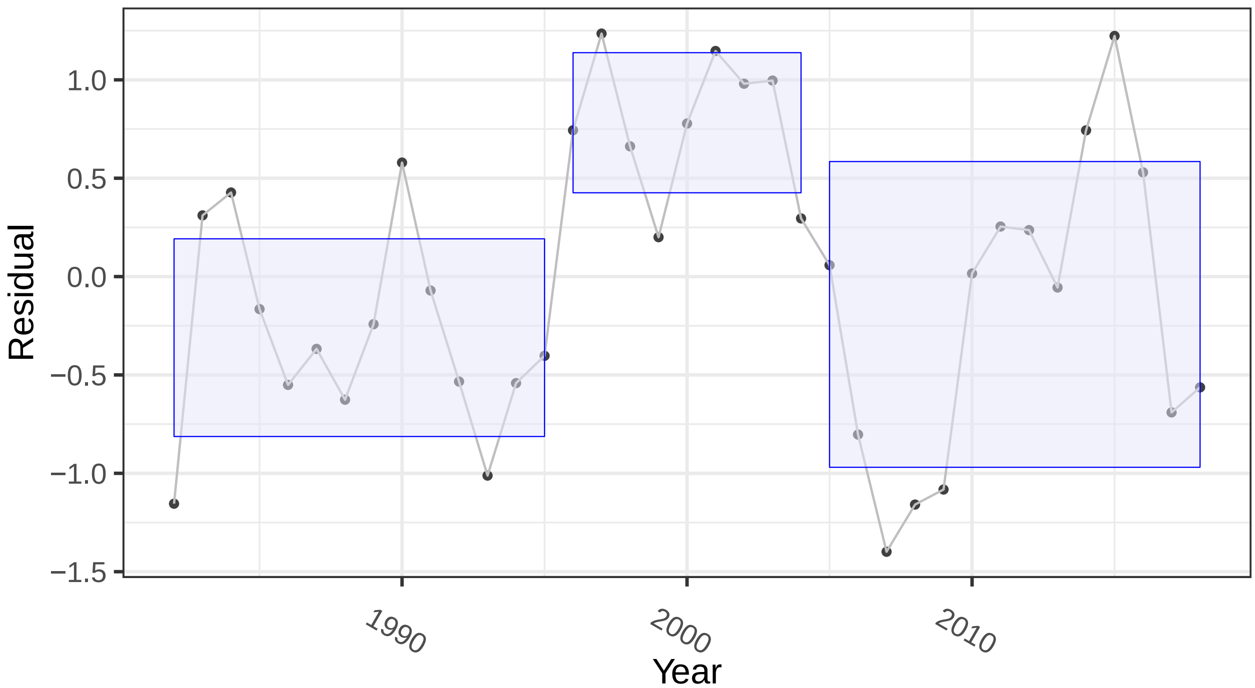

Figure 9 Resiuals from the Beverton and Holt fit, boxes indicate regimes derived from the STARS algorithm.

Combining growth and selection-at-age with the stock recruitment relationship allows the expected dynamics to be derived. These, along with reference points, can be plotted as equilibrium curves.

```
eql=FLBRP(om1,sr=sr)

plot(eql9)+
  theme_bw(16)
```

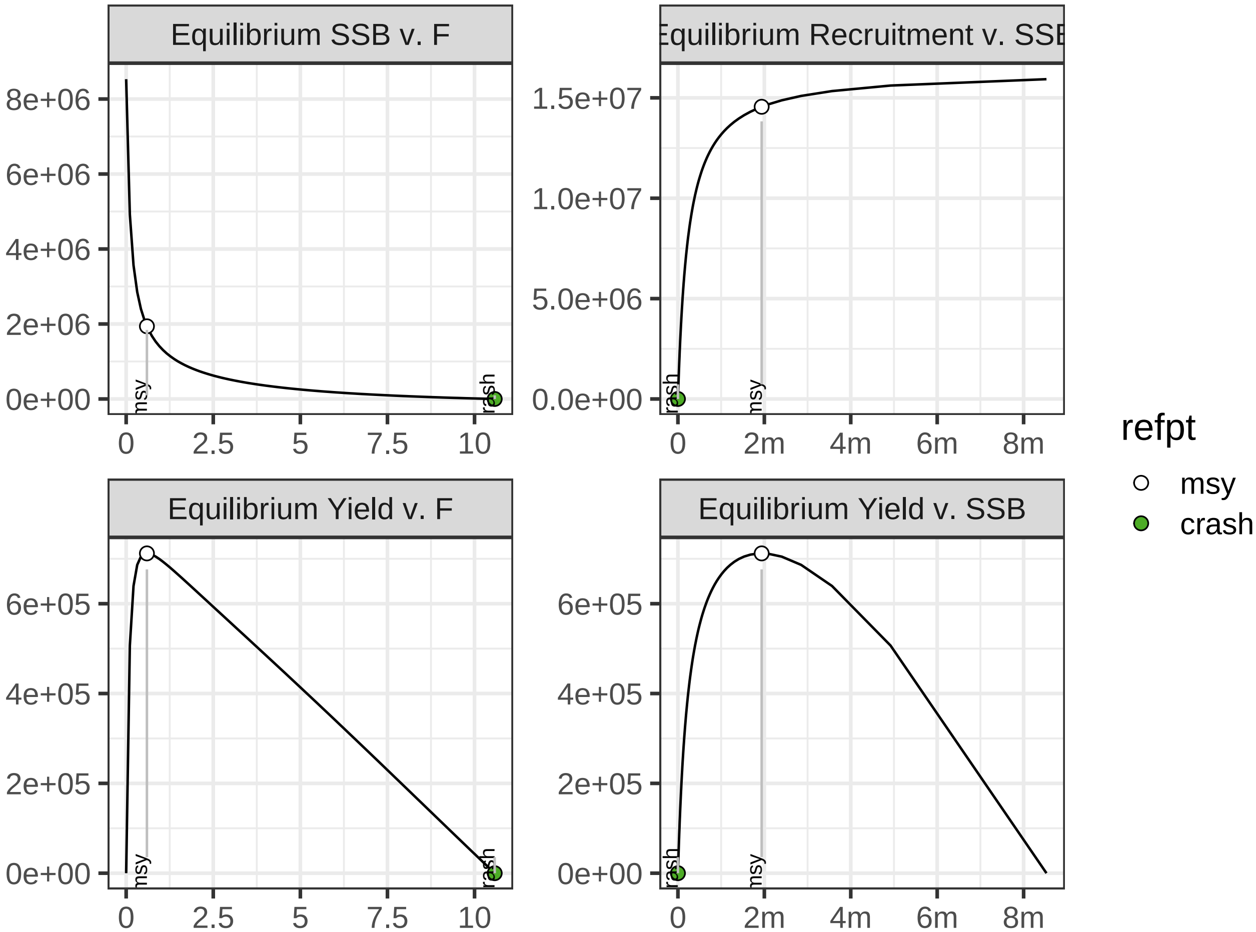

Figure 10 Equilibrium curves for Beverton and Holt stock recruitment relationship, i.e. expected dynamics, with reference points.

Other stock recruitment relationships can be explored, since there is little evidence of a relationship between SSB and recruitment a Beverton and Holt functional form was fitted with steepness=0.9. The impact on the dynamics can be evaluated by inspection of the equilibrium curves.

```
sr9=fmle(as.FLSR(om1,model="bevholtSV"),
                fixed=c(s=0.9,spr0=spr0(eql)),
                        control=list(silent=TRUE))
eql9=FLBRP(om1,sr=ab(sr9))
refpts(eql9)=refpts(eql9)[c(1,2,3)]

plot(eql9)+
  theme_bw(16)
```

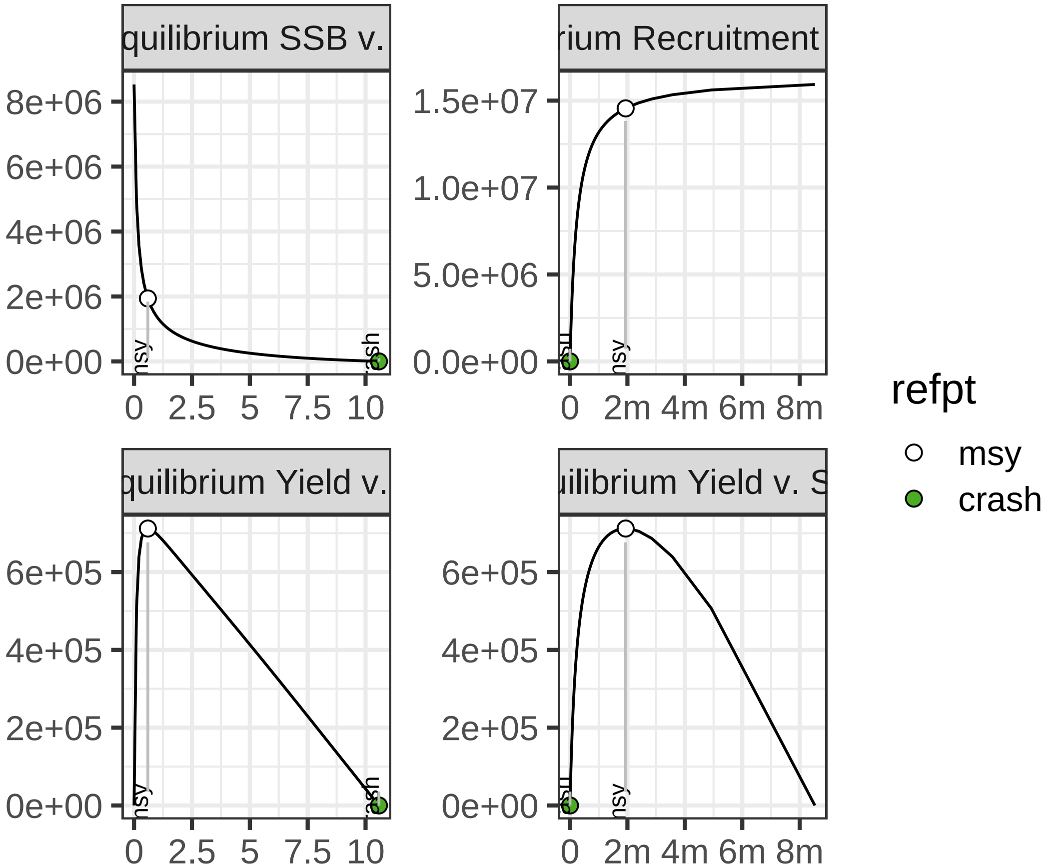

Figure 11 Equilibrium curves and reference points for Beverton and Holt stock recruitment relationship with steepness constrained with a steepess of 0.9.

Assessment Error

The assessment error can be evaluated by performing a retrospective analysis and comparing the difference between the most recent stock assessment and those conducted with the same data in previous years.

```
drop_download(path='bluewhiting/data/retro.RData',local_path="data",overwrite=TRUE)

load(file.path("data","retro.RData"))
```

```
plot(retro)+
  theme_bw(16)+
  theme(legend.position="bottom")+
  xlab("Year")
```

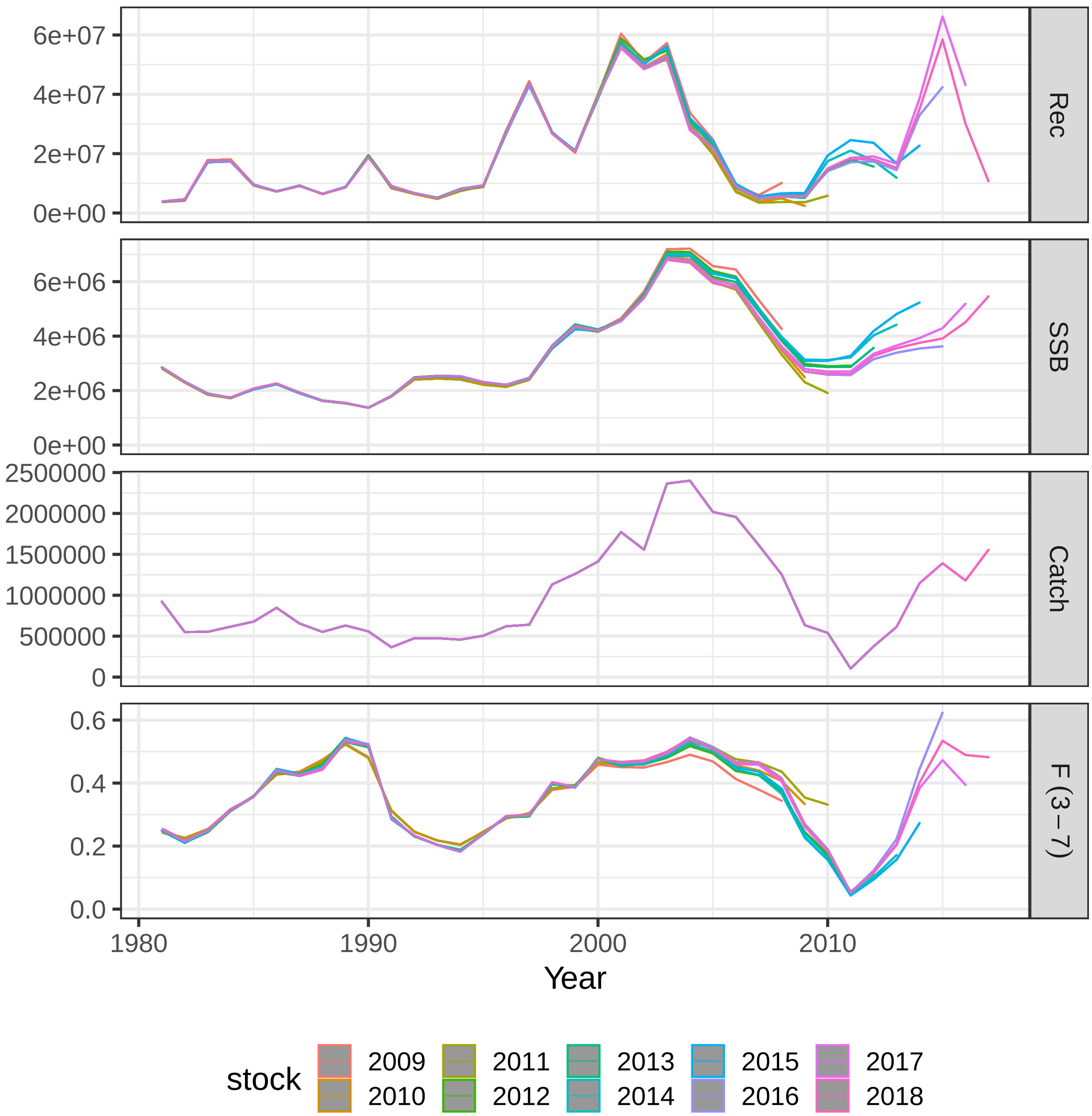

Figure 12 Retrospective analysis.

```
assEr=laply(retro, function(x){
  yr=ac(dims(fbar(x))$maxyear)

  c(f   =c(catch(x   )[,yr]/computeStock(x   )[,yr])/
         c(catch(om1)[,yr]/computeStock(om1)[,yr]),
    ssb=c(ssb(x)[,yr])/c(ssb(om1)[,yr]))})
```

```
library(grid)
source('/home/laurence/Desktop/flr/kobe/R/kobe-phaseMar.R')
source('/home/laurence/Desktop/flr/kobe/R/kobe-phase.R')

dt2=as.data.frame(rmvnorm(1000,c(1,1),cov(assEr)))
```

```
names(dt2)=c("stock","harvest")

kobePhaseMar(dt2,quadcol=c("white","white","white","white"),
  xlab="F",ylab="SSB")
```

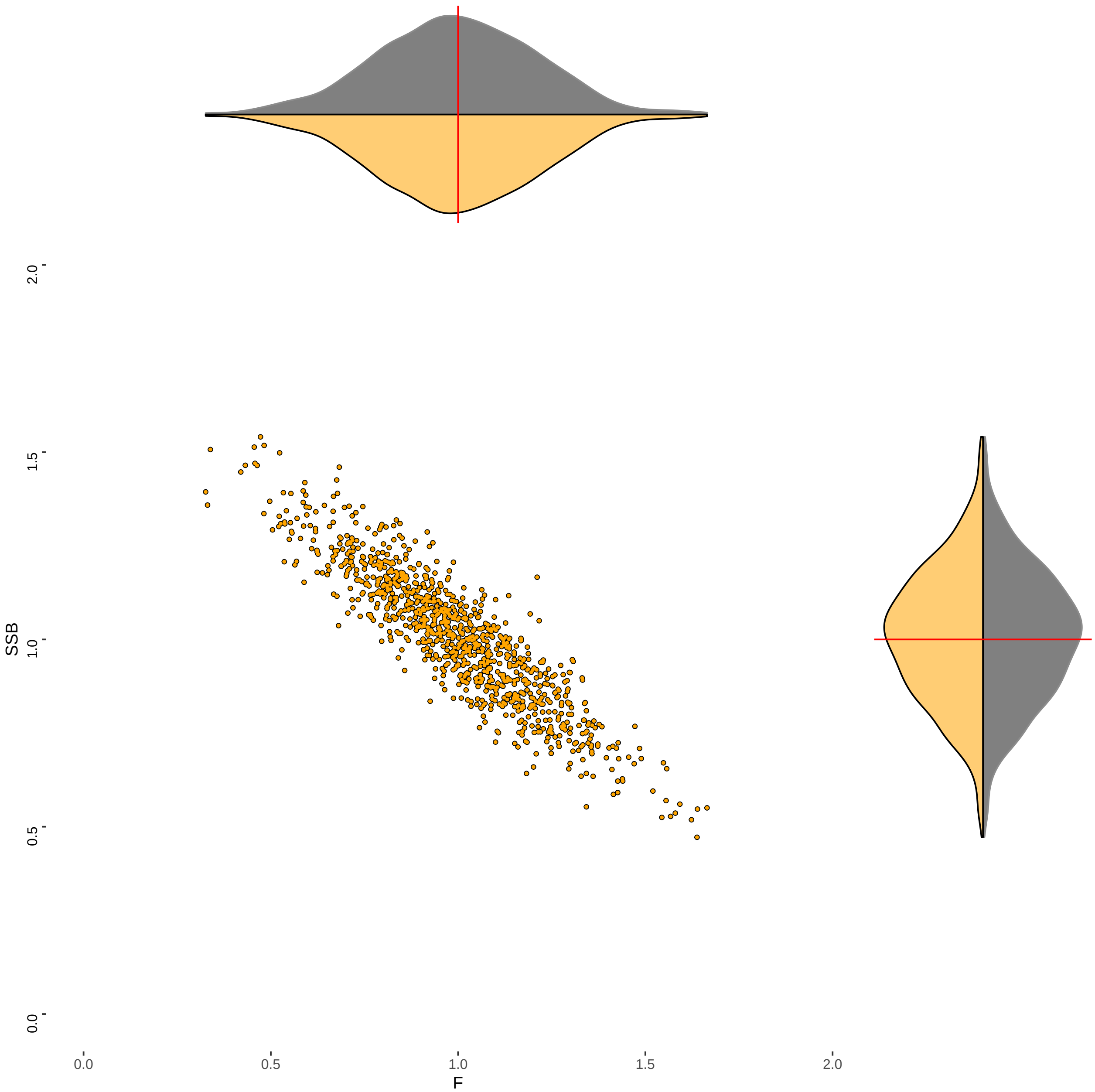

Figure 13 Simulation of assessment error based on retrospective analysis.

Simulations

The objects to run the simulations are now available, i.e. om the 'FLStock' object, eql the 'FLBRP' object with the stock recruitment relationship.

Just make sure all objects have enough iters to hold outputs.

```r
om=propagate(om,1000)
```

The ICES reference points and HCR parameters are then loaded as 'FLPar' objects.

```r
### ICES reference points
refpts=FLPar(c(
  blim    =1500000,
  bpa     =2250000,
  flim    =0.88,
  fpa     =0.53,
  msytrig =2250000,
  fmsy    =0.32,
  bmgtlow =1500000,
  bmgt    =2250000,
  fmgstlow=0.05,
  fmgt    =0.32),units="NA")
```

```r
par1=array(c(ftar=refpts["fmsy"],btrig=refpts["msytrig"],fmin=FLPar(0.05),blim=refpts["blim"]),
           dim=c(4),
           dimnames=list(c("ftar","btrig","fmin","blim")))
par1
```

```
    ftar    btrig     fmin     blim 
3.20e-01 2.25e+06 5.00e-02 1.50e+06 
```

```r
par2=array(c(0.20,2250000,
             0.05,1500000,
             0.32,5200000,
             0.20,4000000),
          dim=c(4,2),
          dimnames=list(c("ftar","btrig","fmin","blim"),c("lower","upper")))[c(1,3,2,4),1:2]

par2
```

```
         lower   upper
ftar  2.00e-01 3.2e-01
fmin  5.00e-02 2.0e-01
btrig 2.25e+06 5.2e+06
blim  1.50e+06 4.0e+06
```

First run a simple projection for F_{MSY} for comparison with the ICES assessment and the HCR outcomes.

```r
fmsy=fwd(om,fbar=FLQuant(1,dimnames=list(year=2001:2018))%=%0.32,
                         sr=eql9,residuals=exp(residuals(sr9)))

plot(fmsy)
```

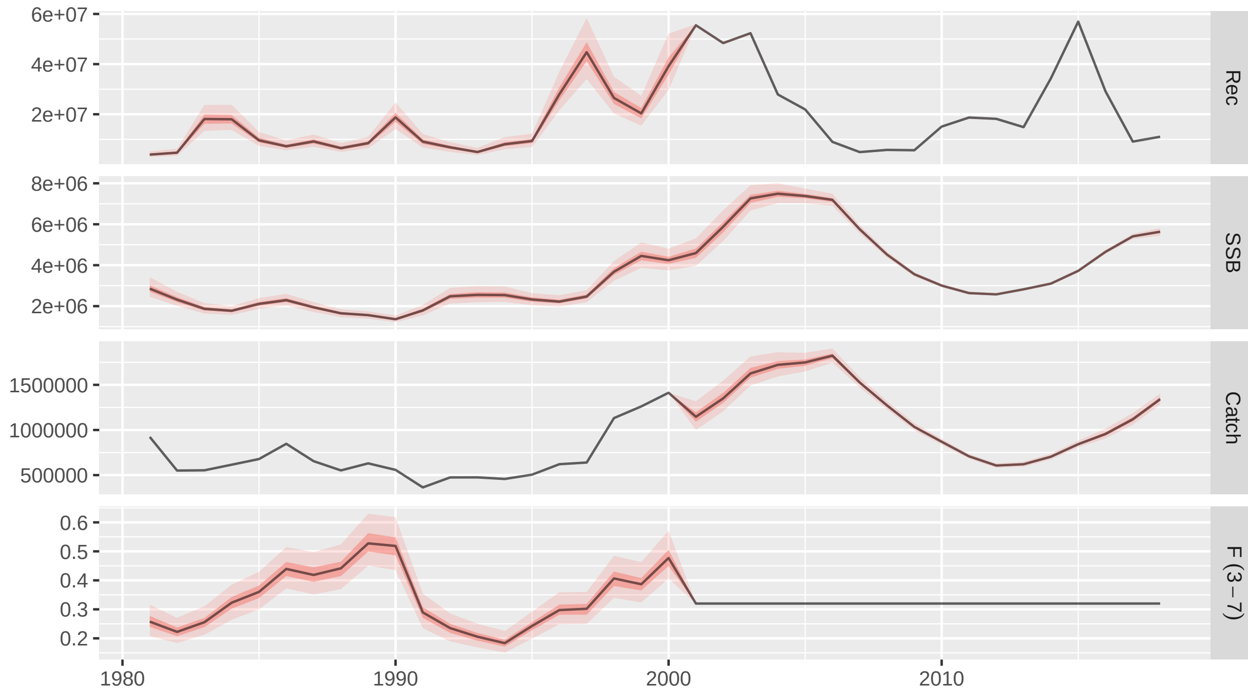

Figure 14 Timeseries from F_{MSY} projection.

HCRs

Run a HCR without assessment error and compare to the projection.

```
source('/home/laurence/Desktop/sea++/mydas/pkg/R/hcrICES.R')

HCR1NoErr=hcrICES(om,eql9,
              sr_deviances=propagate(exp(residuals(sr9)),1000),
              params=par1,
              start=2001,end=2018)
```

```
==2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017,
```

```
plot(FLStocks("Fmsy"=fmsy,"HCR"=HCR1NoErr[[1]]))
```

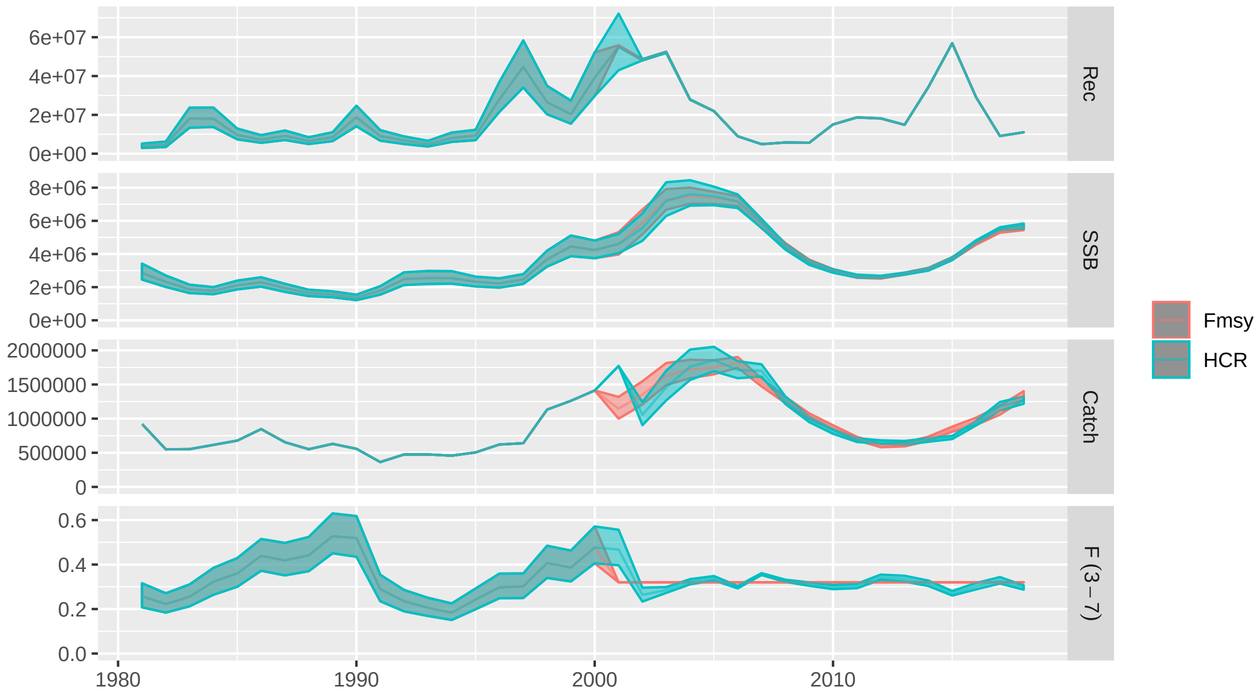

Figure 15 Comparison of timeseries for F_{MSY} projection and HCR without assessment error.

Add assessment error.

```
err =rlnorm(1000,FLQuant(0,dimnames=list(year=2000:2018)),0.3)

HCR1=hcrICES(om,eql9,
             sr_deviances=propagate(exp(residuals(sr9)),1000),
             params=par1,
             start=2001,end=2018,
             err=err)
```

```
==2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017,
```

```
plot(FLStocks("Without \nAssessment Error"=HCR1NoErr[[1]],"With \nAssessment Error"=HCR1[[1]]))
```

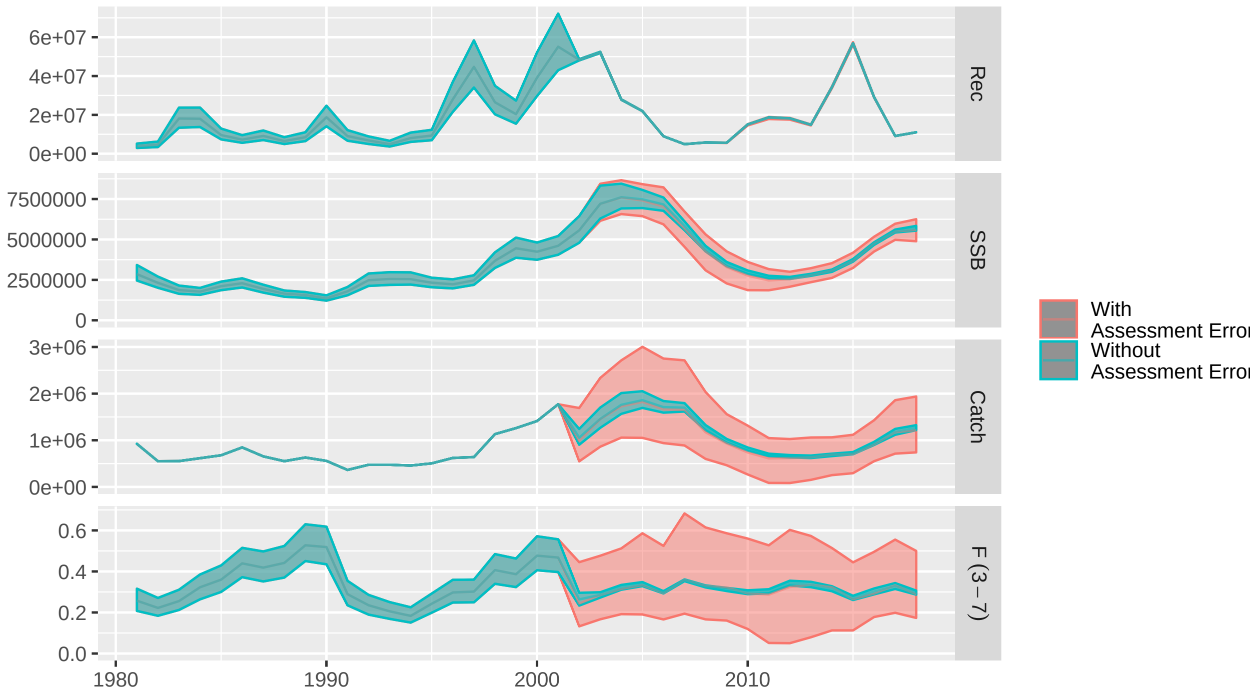

Figure 16 Comparison of timeseries for HCR I with and without assessment error.

Compare the two HCRs.

```
HCR2=hcrICES(om,eql9,
             sr_deviances=propagate(exp(residuals(sr9)),1000),
             params=par2,
             start=2001,end=2018,
             err=err)
```

```
==2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017,
```

```
plot(FLStocks("HCR1"=HCR1[[1]],"HCR2"=HCR2[[1]]))
```

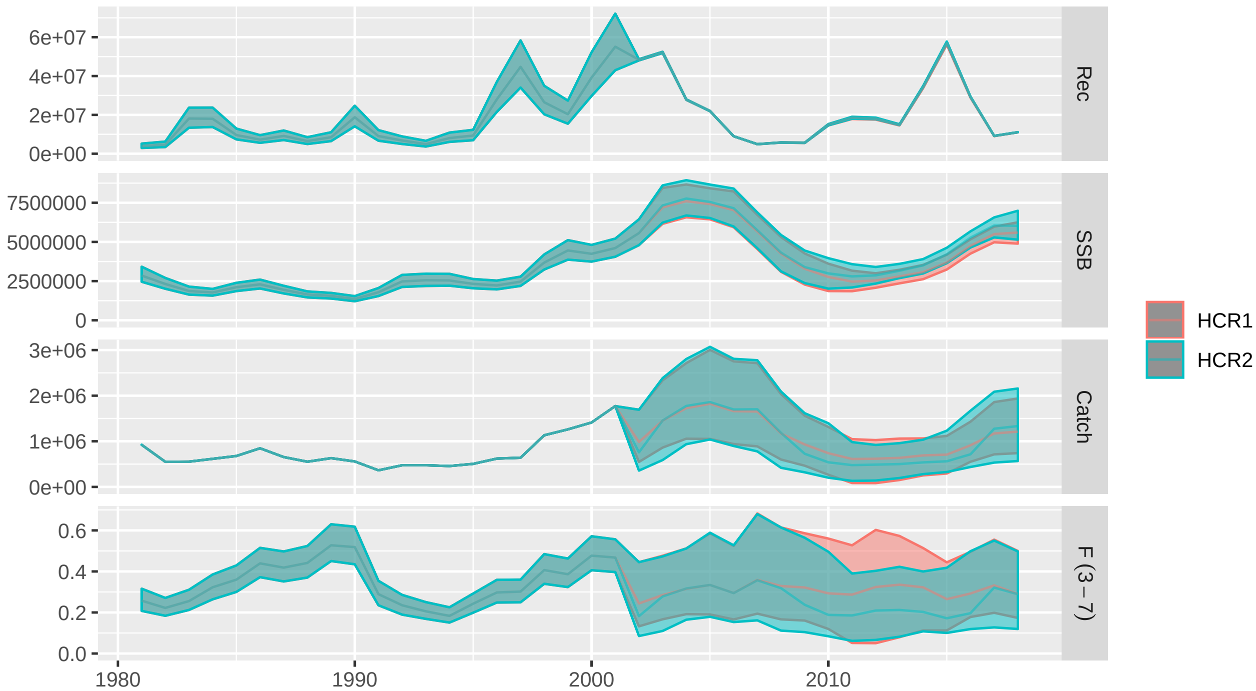

Figure 17 Comparison of timeseries for HCR I and HCR II.

Evaluate the effect of bounds.

```
HCR1Bnd=hcrICES(om,eql9,
              sr_deviances=propagate(exp(residuals(sr9)),1000),
              params=par1,
              start=2001,end=2018,
              err=err,
              bndTac=c(0.80,1.25))
```

```
==2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017,
```

```
plot(FLStocks("HCR1"=HCR1[[1]],"HCR1 with Bounds"=HCR1Bnd[[1]]))
```

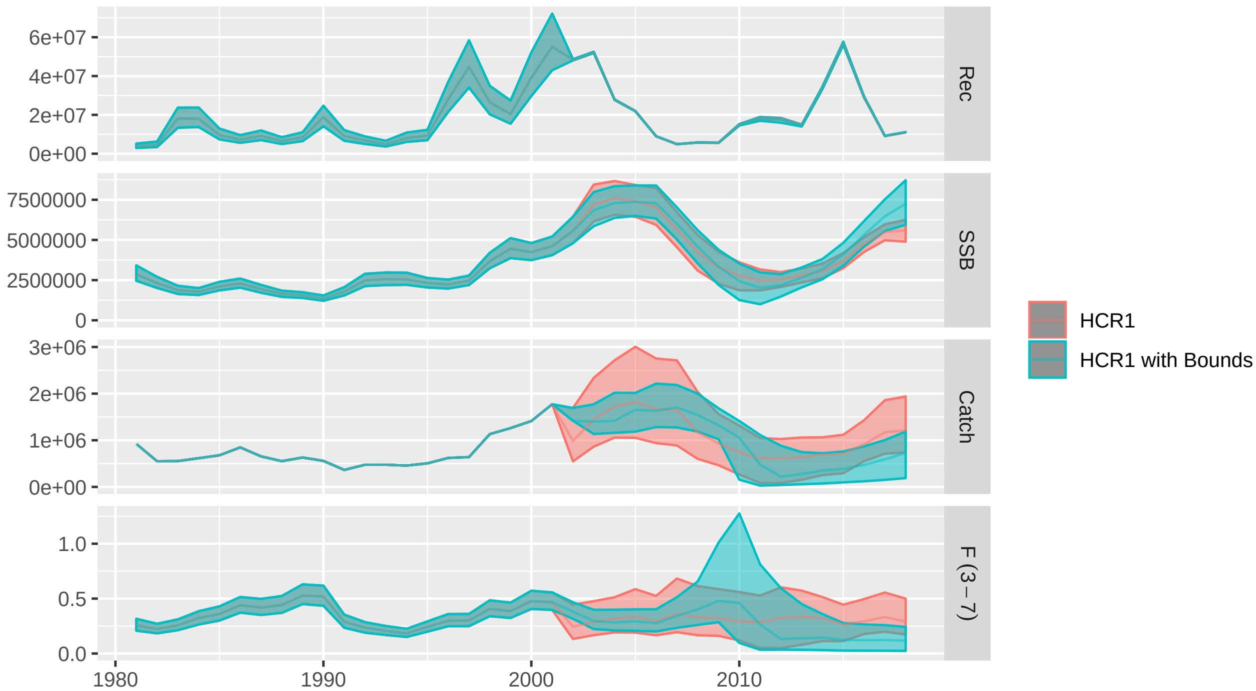

Figure 18 Comparison of timeseries for HCR I with and without bounds of -80 and +125%.