SPiCT scenarios for the Greater silver smelt (*Argentina silus*) in Subareas 1, 2, and 4, and Division 3.a (Northeast Arctic, North Sea, Skagerrak and Kattegat)

Alexandros Kokkalis, Elvar Halldor Hallfredsson, Lise Heggebakken

05/02/2020

Introduction

This working document present a series of different assessments using the surplus production model in continous time (SPiCT; Pedersen and Berg (2017)) available as an R package (https://github.com/DTUAqua/spict).

Read in the data

```
library(spict)

## Read in the data
dat <- readxl::read_xlsx("../data/GSS_indices270120_AK.xlsx")
#plot(dat$year,dat$catchTOT,type = "l",ylim = c(0,max(na.omit(dat$catchTOT))))
## Sum up the catches from each area to get the total catch
dat$catchTOT <- dat$catch1and2 + dat$catch3 + dat$catch4
#plot(dat$year,dat$catchTOT,type = "l", ylim = c(0,max(na.omit(dat$catchTOT))))
## run retro or not
runretro <- FALSE
```

Scenario 1

Table 1: Input data for Scenario 1

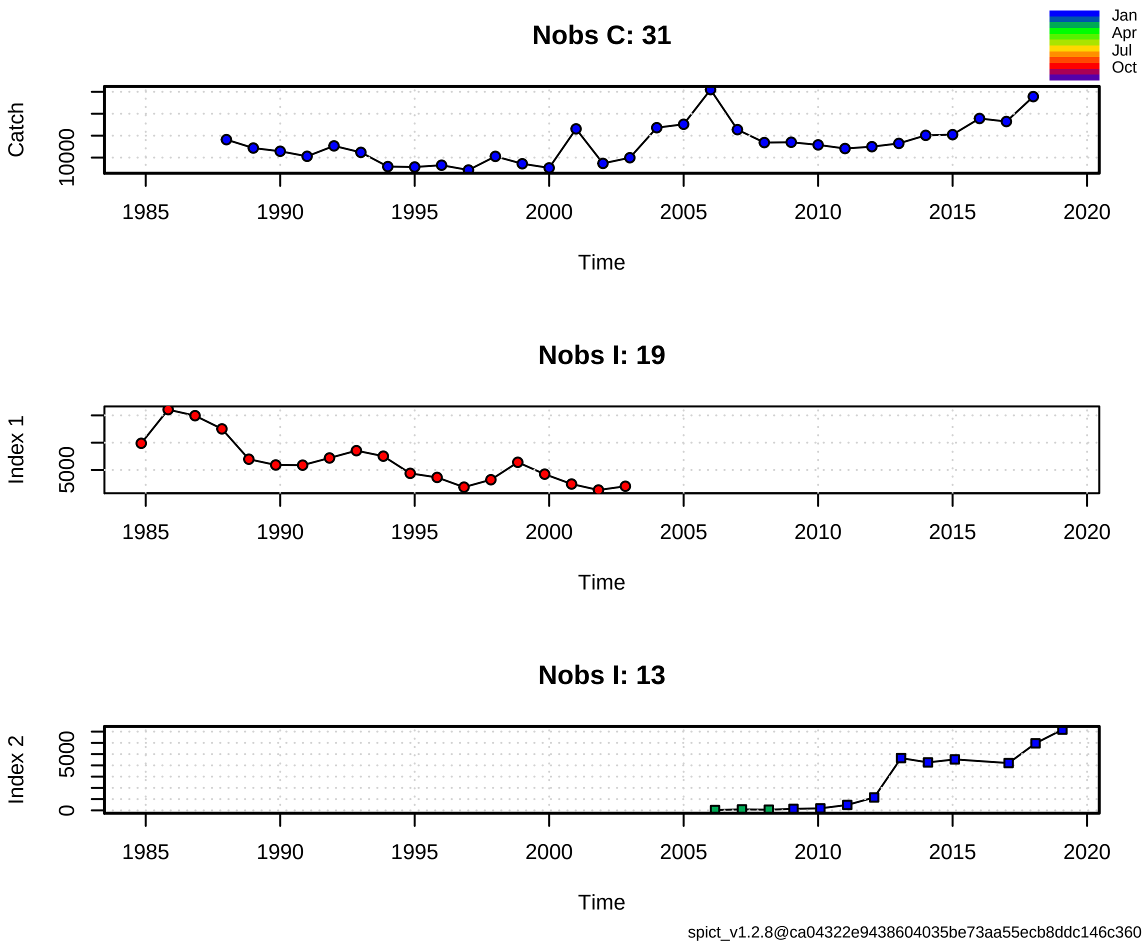
Input data	Name	Range	Notes
Catch	Total catch	1988-2018	
Biomass indices	Shrimp survey	1984–2002	Split in two periods
		2005–2018	
			Default priors

```
## Choose only the years where the survey was in October
w <- !is.na(dat$northsea_month) & dat$northsea_month == 10
## Choose only the years where the survey was in January or February
```

```r
v <- !is.na(dat$northsea_month) & dat$northsea_month %in% c(1, 2)


## Make the input list
inp_NS <- list(timeC = dat$year,                                   ## Timing of catch
               obsC = dat$catchTOT,                                ## Observed catches
               timeI = list(dat$year[w] + dat$northsea_month[w] / 12, ## Timing of survey index
                            dat$year[v] + dat$northsea_month[v] / 12),
               obsI = list(dat$northsea_SA[w],                     ## Observed indices
                           dat$northsea_SA[v]),
               optimiser.control = list(iter.max = 1e5,            ## Optimiser options
                                        eval.max = 1e5),           ## sometimes help converge
               priors = list(                                      ## List of priors (empty, i.e. de
                                                                   ## see possible priors with list.
                 ))
## Check input time series, remove missing and zero observations
inp_NS <- check.inp(inp_NS)
```

```
## Removing zero, negative, and NAs in  C  series
```

```r
## Plot input data
plotspict.data(inp_NS)
```

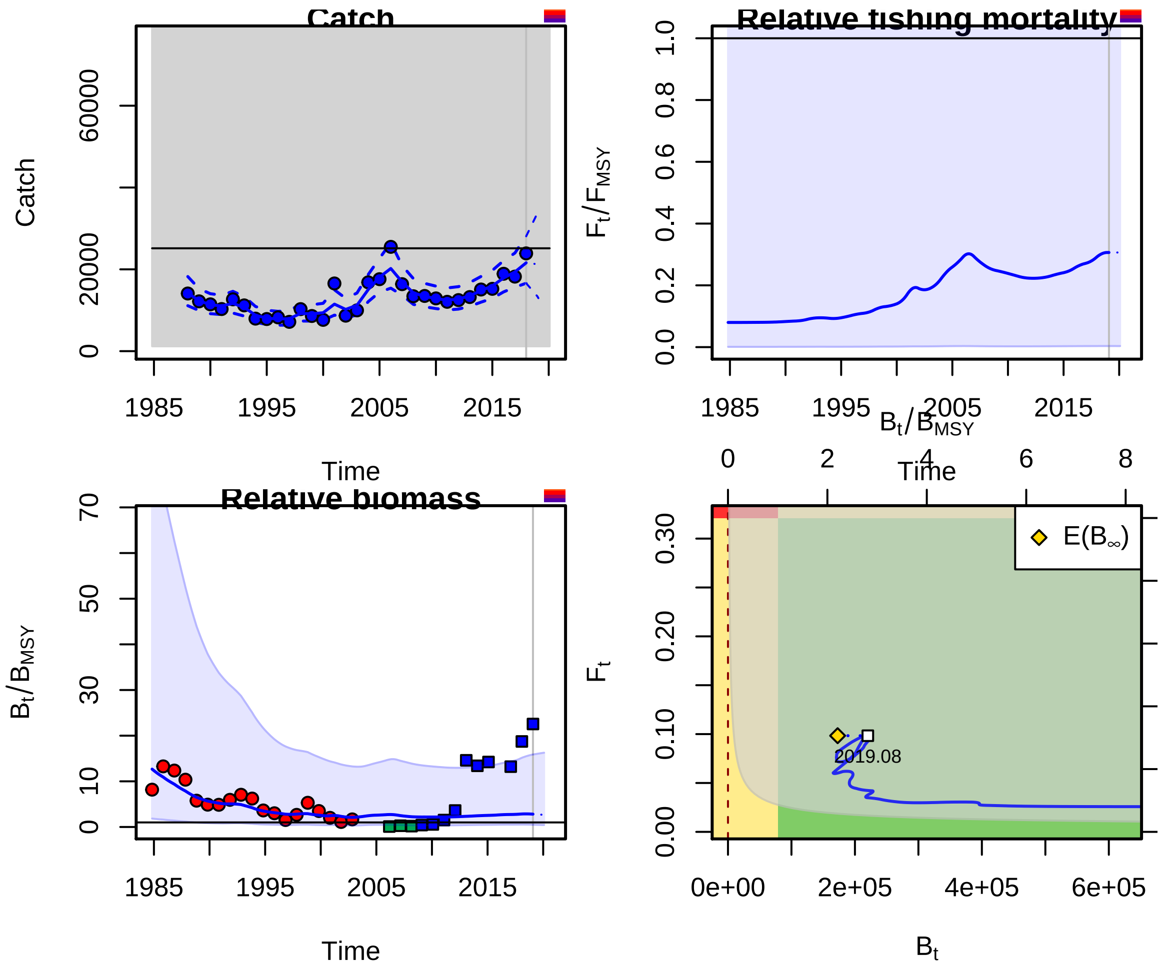


```r
## Fit spict
fit_NS <- fit.spict(inp_NS)

## Summary of the fit - in the vignette there is a line-by-line description of that summary
fit_NS
```

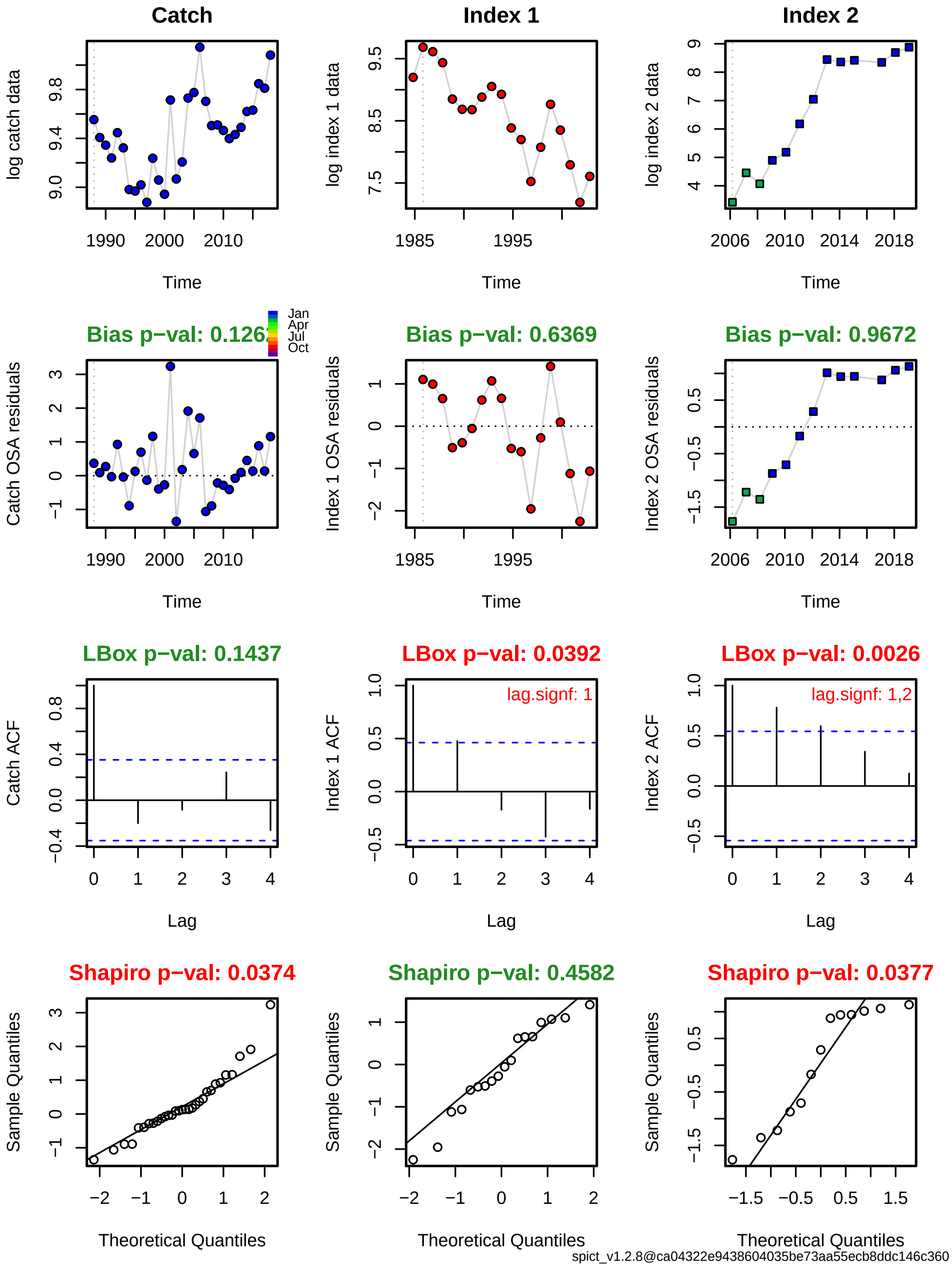
```
## Convergence: 0  MSG: relative convergence (4)
## Objective function at optimum: 57.3810526
## Euler time step (years):  1/16 or 0.0625
## Nobs C: 31,  Nobs I1: 19,  Nobs I2: 13
##
## Priors
##      logn  ~  dnorm[log(2), 2^2]
##  logalpha  ~  dnorm[log(1), 2^2]
##   logbeta  ~  dnorm[log(1), 2^2]
##
## Model parameter estimates w 95% CI
##             estimate        cilow        ciupp    log.est
##  alpha1 2.390221e+00    0.6561160 8.707540e+00  0.8713858
##  alpha2 1.185998e+01    3.5324126 3.981958e+01  2.4731699
##  beta   9.416977e-01    0.3601728 2.462137e+00 -0.0600710
##  r      1.999363e-01    0.0614380 6.506482e-01 -1.6097566
##  rc     6.376559e-01    0.0245182 1.658384e+01 -0.4499565
##  rold   5.361626e-01    0.0090055 3.192149e+01 -0.6233178
##  m      2.551782e+04 1202.3903049 5.415541e+05 10.1471325
##  K      2.797479e+05 4660.6208998 1.679152e+07 12.5416442
##  q1     1.544270e-02    0.0000792 3.010343e+00 -4.1706220
##  q2     4.063500e-03    0.0000136 1.210055e+00 -5.5057111
##  n      6.270977e-01    0.0459170 8.564407e+00 -0.4666529
##  sdb    1.576217e-01    0.0512043 4.852053e-01 -1.8475574
##  sdf    1.625876e-01    0.0768551 3.439552e-01 -1.8165384
##  sdi1   3.767507e-01    0.2468519 5.750051e-01 -0.9761716
##  sdi2   1.869391e+00    1.2648928 2.762781e+00  0.6256125
##  sdc    1.531084e-01    0.0978477 2.395782e-01 -1.8766094
##
## Deterministic reference points (Drp)
##            estimate        cilow        ciupp   log.est
##  Bmsyd 8.003635e+04  544.6157923 1.176209e+07 11.290236
##  Fmsyd 3.188279e-01    0.0122591 8.291918e+00 -1.143104
##  MSYd  2.551782e+04 1202.3903049 5.415541e+05 10.147132
## Stochastic reference points (Srp)
##            estimate        cilow        ciupp   log.est rel.diff.Drp
##  Bmsys 7.831263e+04  552.3440942 1.110335e+07 11.268464 -0.022010800
##  Fmsys 3.210608e-01    0.0122407 8.421108e+00 -1.136125  0.006954625
##  MSYs  2.514696e+04 1185.4484920 5.334435e+05 10.132492 -0.014747787
##
## States w 95% CI (inp$msytype: s)
##                    estimate       cilow        ciupp   log.est
##  B_2019.08      2.203227e+05 776.4156127 6.252077e+07 12.302849
##  F_2019.08      9.836480e-02   0.0003846 2.515559e+01 -2.319072
##  B_2019.08/Bmsy 2.813374e+00   0.4986872 1.587182e+01  1.034385
##  F_2019.08/Fmsy 3.063744e-01   0.0036177 2.594577e+01 -1.182947
##
```

```
## Predictions w 95% CI (inp$msytype: s)
##                    prediction        cilow        ciupp   log.est
##  B_2019.08      2.203227e+05 7.764156e+02 6.252077e+07 12.302849
##  F_2019.08      9.836480e-02 3.846000e-04 2.515559e+01 -2.319072
##  B_2019.08/Bmsy 2.813374e+00 4.986872e-01 1.587182e+01  1.034385
##  F_2019.08/Fmsy 3.063744e-01 3.617700e-03 2.594577e+01 -1.182947
##  Catch_2019.08  2.110211e+04 1.295520e+04 3.437222e+04  9.957128
##  E(B_inf)       1.727415e+05           NA           NA 12.059552
```

```r
## If the model converged, it reports convergence as 0
## Continue with plotting and diagnostics only if convergence was reached
converged <- fit_NS$opt$convergence == 0
if (converged) {
  ## Calculate the One Step Ahead (osa) residuals
  fit_NS <- calc.osa.resid((fit_NS))

  ## Make a plot showing relative F, relative B, Kobe plot catch
  par(mfrow = c(2,2),  ## 2x2 subplots
      mar = c(4.1, 4.1, 0.5, 0.5)) ## Change default margins for the plots
  plotspict.catch(fit_NS)
  plotspict.ffmsy(fit_NS)
  plotspict.bbmsy(fit_NS)
  plotspict.fb(fit_NS)
}
```




```r
if (converged) {
  plotspict.diagnostic(fit_NS)
}
```




```r
## If runretro is TRUE, run and plot the retrospective analysis
if (runretro & converged) {
  fit_NS <- retro(fit_NS)
  plotspict.retro(fit_NS)
}
```

Scenario 2

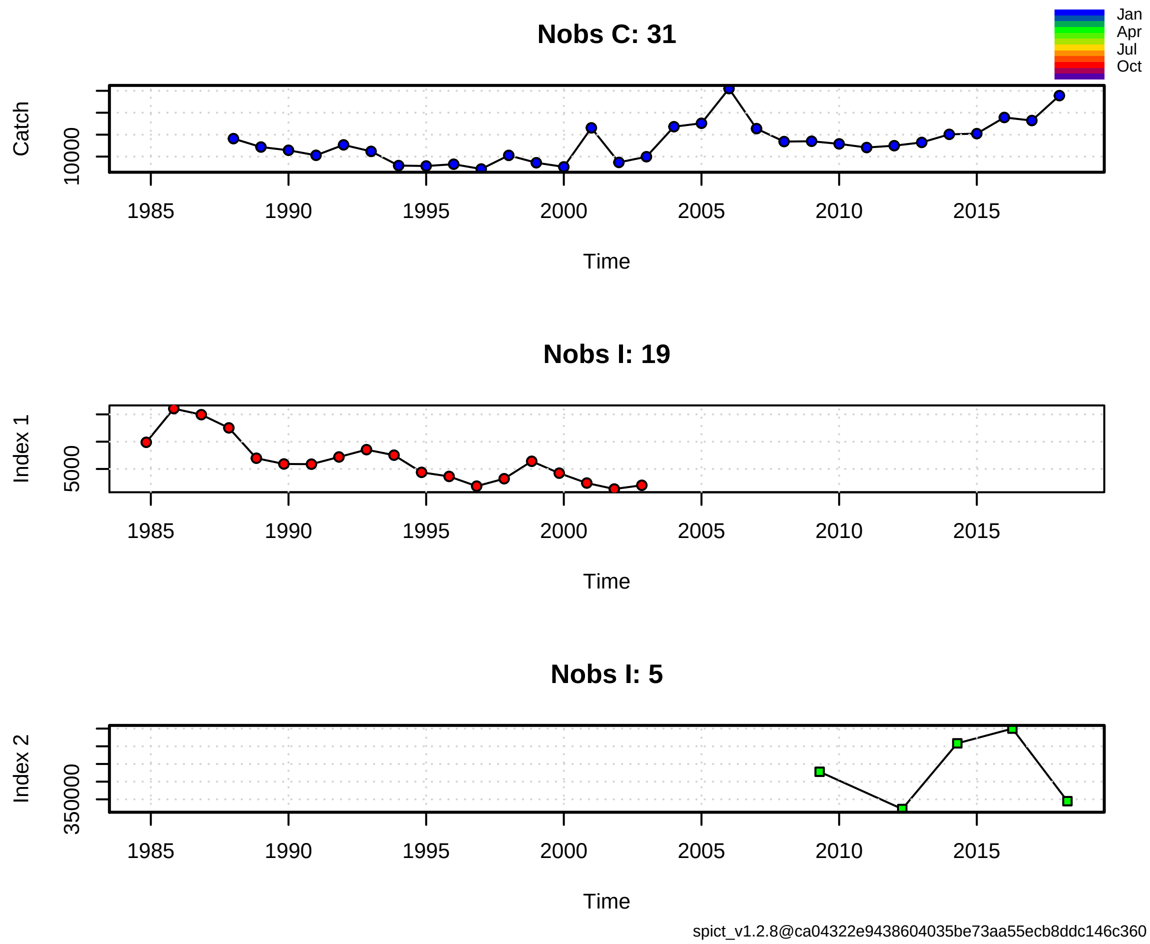
Table 2: Input data for Scenario 1

Input data	Name	Range	Notes
Catch	Total catch	1988-2018	
Biomass indices	Shrimp survey	1984–2002	Only october period
Biomass indices	Acoustic survey	2009–2018	Matlab
			Default priors

```
## Choose only the years where the survey was in October
w <- !is.na(dat$northsea_month) & dat$northsea_month == 10
## Choose only the years where the survey was in January or February
##v <- !is.na(dat$northsea_month) & dat$northsea_month %in% c(1, 2)


## Make the input list
inp_NS <- list(timeC = dat$year,                                     ## Timing of catch
               obsC = dat$catchTOT,                                  ## Observed catches
               timeI = list(dat$year[w] + dat$northsea_month[w] / 12, ## Timing of survey index
                            dat$year+3.5/12),
               obsI = list(dat$northsea_SA[w],                       ## Observed indices
                          dat$norwegian_seaAC),
               optimiser.control = list(iter.max = 1e5,              ## Optimiser options
                                        eval.max = 1e5),             ## sometimes help converge
               priors = list(                                        ## List of priors (empty, i.e. de
 logn=c(log(2),.5,1),
 logbkfrac=c(log(.5),1,1)
## see possible priors with list.possible.priors()
                 ))
## Check input time series, remove missing and zero observations
inp_NS <- check.inp(inp_NS)
```

```
## Removing zero, negative, and NAs in  C  series
## Removing zero, negative, and NAs in  I  series  2
```

```
## Plot input data
plotspict.data(inp_NS)
```



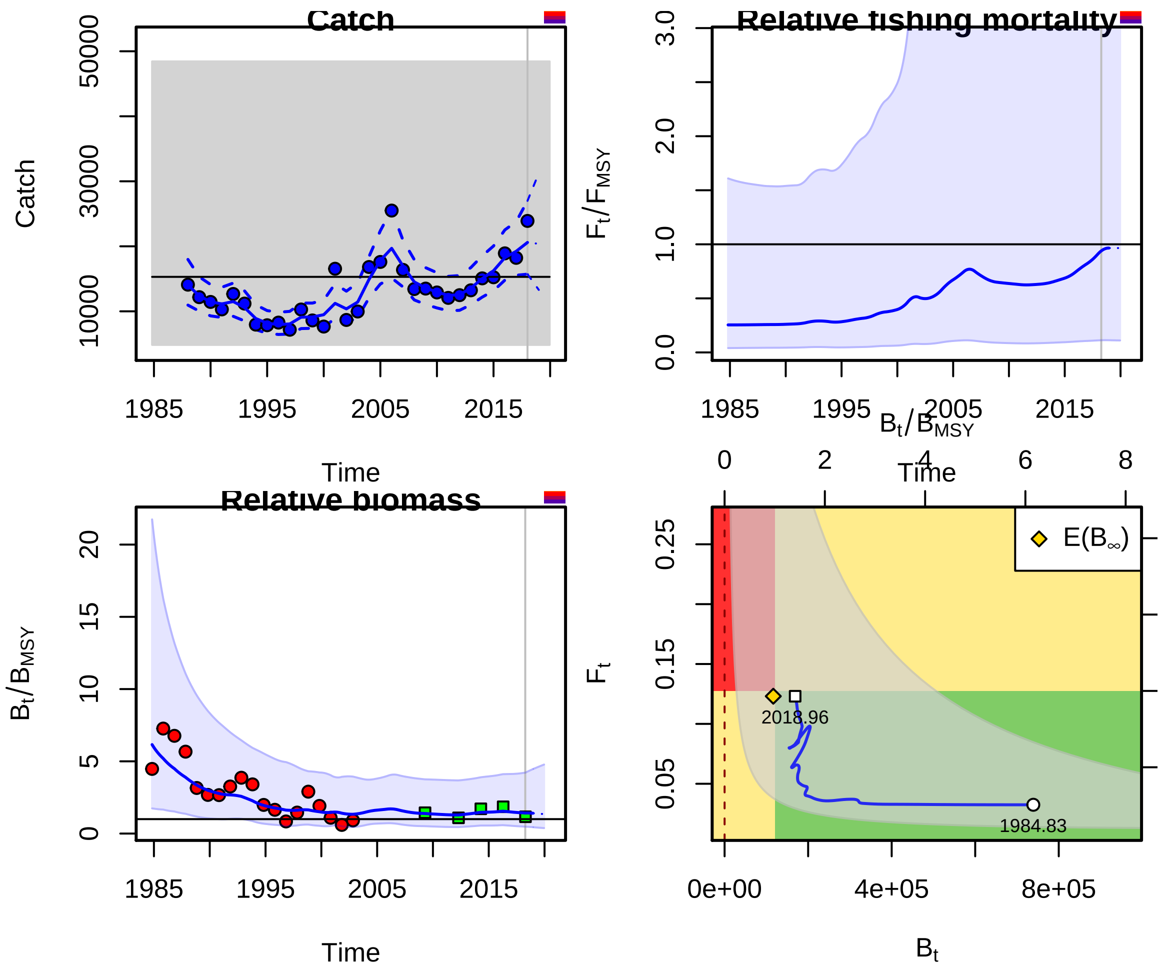

```
## Fit spict
fit_NS <- fit.spict(inp_NS)

## Summary of the fit - in the vignette there is a line-by-line description of that summary
fit_NS
```

```
## Convergence: 0  MSG: relative convergence (4)
## Objective function at optimum: 28.3888908
## Euler time step (years):  1/16 or 0.0625
## Nobs C: 31,  Nobs I1: 19,  Nobs I2: 5
##
## Priors
##       logn  ~  dnorm[log(2), 0.5^2]
##  logbkfrac  ~  dnorm[log(0.5), 1^2]
##   logalpha  ~  dnorm[log(1), 2^2]
##    logbeta  ~  dnorm[log(1), 2^2]
##
## Model parameter estimates w 95% CI
##             estimate        cilow        ciupp    log.est
##  alpha1 3.110579e+00 7.699815e-01 1.256615e+01  1.1348088
##  alpha2 1.551961e+00 3.382870e-01 7.119942e+00  0.4395195
##  beta   1.066262e+00 4.099298e-01 2.773440e+00  0.0641595
##  r      1.878088e-01 4.709680e-02 7.489285e-01 -1.6723309
```

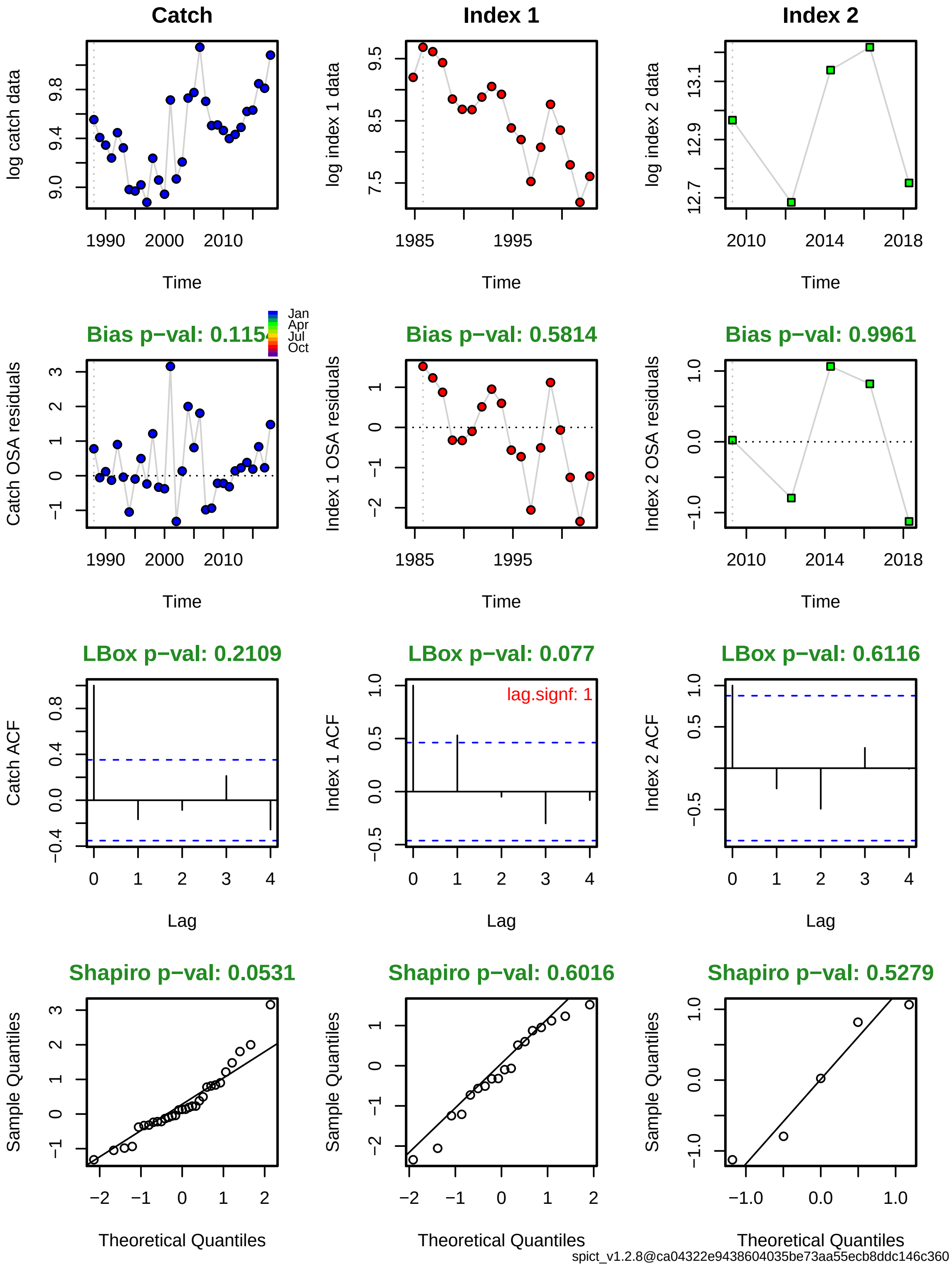
```
##  rc     2.588986e-01 4.339450e-02 1.544631e+00 -1.3513188
##  rold   4.165855e-01 1.347390e-02 1.288001e+01 -0.8756636
##  m      1.612555e+04 5.107054e+03 5.091651e+04  9.6881603
##  K      2.843817e+05 3.946527e+04 2.049218e+06 12.5580725
##  q1     1.840820e-02 1.227900e-03 2.759774e-01 -3.9949610
##  q2     2.469988e+00 1.496627e-01 4.076394e+01  0.9042132
##  n      1.450829e+00 5.912595e-01 3.560035e+00  0.3721351
##  sdb    1.309317e-01 3.852300e-02 4.450098e-01 -2.0330797
##  sdf    1.493363e-01 6.796260e-02 3.281410e-01 -1.9015548
##  sdi1   4.072732e-01 2.634923e-01 6.295118e-01 -0.8982710
##  sdi2   2.032009e-01 8.578800e-02 4.813095e-01 -1.5935603
##  sdc    1.592316e-01 1.058182e-01 2.396063e-01 -1.8373952
##
## Deterministic reference points (Drp)
##           estimate        cilow        ciupp   log.est
##  Bmsyd 1.245704e+05 1.438294e+04 1.078902e+06 11.732626
##  Fmsyd 1.294493e-01 2.169730e-02 7.723154e-01 -2.044466
##  MSYd  1.612555e+04 5.107054e+03 5.091651e+04  9.688160
## Stochastic reference points (Srp)
##           estimate        cilow        ciupp   log.est rel.diff.Drp
##  Bmsys 1.200232e+05 1.417972e+04 1.015927e+06 11.695440  -0.03788633
##  Fmsys 1.275264e-01 2.049930e-02 7.933447e-01 -2.059432  -0.01507846
##  MSYs  1.529738e+04 4.830986e+03 4.843934e+04  9.635437  -0.05413822
##
## States w 95% CI (inp$msytype: s)
##                    estimate        cilow        ciupp    log.est
##  B_2018.27      1.704629e+05 9476.1522142 3.066391e+06 12.0462727
##  F_2018.27      1.197088e-01    0.0067759 2.114883e+00 -2.1226934
##  B_2018.27/Bmsy 1.420250e+00    0.4788558 4.212352e+00  0.3508327
##  F_2018.27/Fmsy 9.386979e-01    0.1126937 7.819019e+00 -0.0632615
##
## Predictions w 95% CI (inp$msytype: s)
##                  prediction        cilow        ciupp    log.est
##  B_2019.02      1.684139e+05 8.707537e+03 3.257320e+06 12.0341797
##  F_2019.02      1.230908e-01 6.825200e-03 2.219921e+00 -2.0948329
##  B_2019.02/Bmsy 1.403178e+00 4.395614e-01 4.479258e+00  0.3387398
##  F_2019.02/Fmsy 9.652182e-01 1.128026e-01 8.259088e+00 -0.0354011
##  Catch_2019.00  2.037854e+04 1.329067e+04 3.124637e+04  9.9222378
##  E(B_inf)       1.165644e+05           NA           NA 11.6661995
```

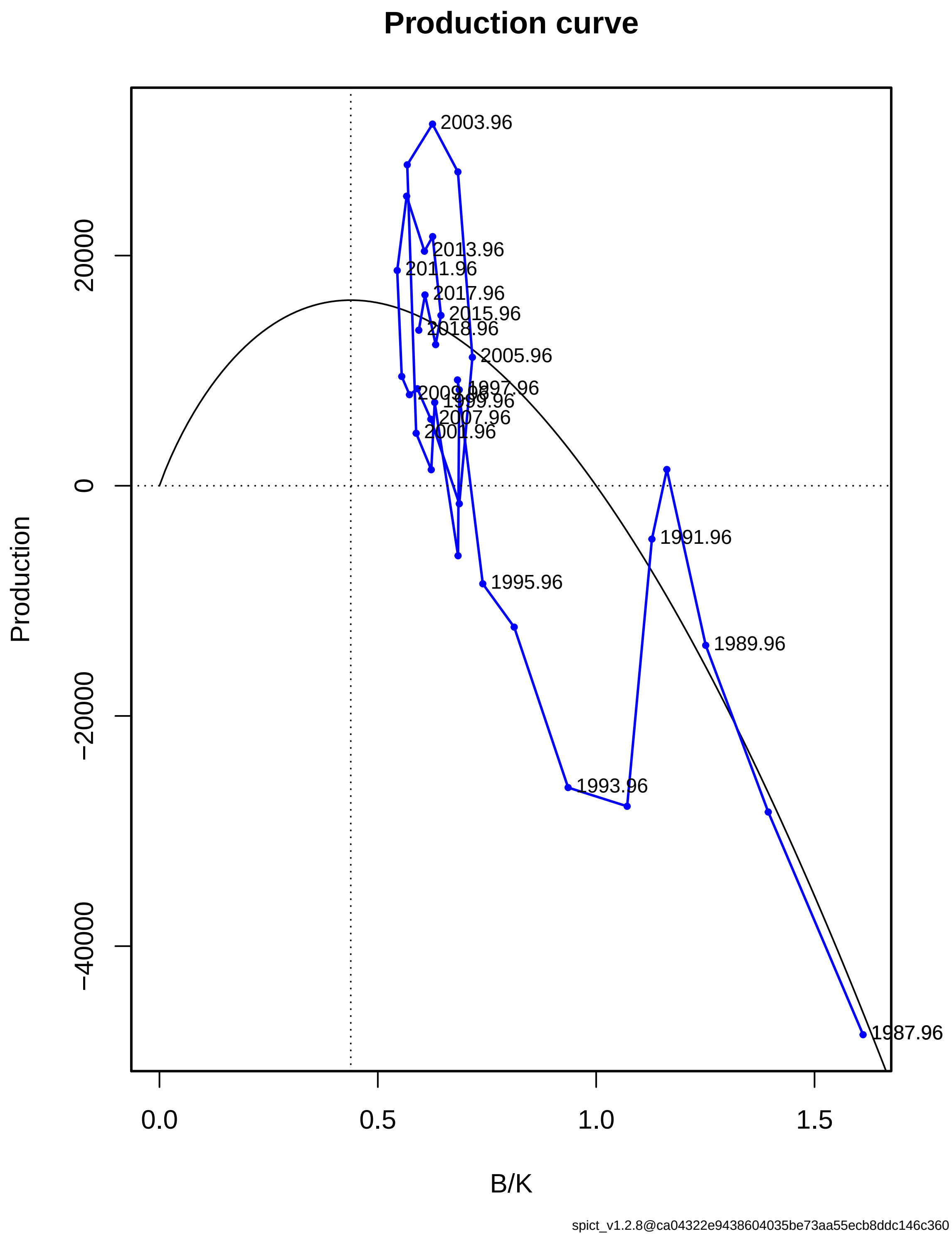
```r
## If the model converged, it reports convergence as 0
## Continue with plotting and diagnostics only if convergence was reached
converged <- fit_NS$opt$convergence == 0
if (converged) {
  ## Calculate the One Step Ahead (osa) residuals
  fit_NS <- calc.osa.resid((fit_NS))

  ## Make a plot showing relative F, relative B, Kobe plot catch
  par(mfrow = c(2,2),  ## 2x2 subplots
      mar = c(4.1, 4.1, 0.5, 0.5)) ## Change default margins for the plots
  plotspict.catch(fit_NS)
  plotspict.ffmsy(fit_NS)
  plotspict.bbmsy(fit_NS)
  plotspict.fb(fit_NS)
```

```
}
```




```r
if (converged) {
  plotspict.diagnostic(fit_NS)
  plotspict.production(fit_NS)
}
```


Catch
Index 1
Index 2
log catch data
log index 1 data
log index 2 data
Time
Bias p−val: 0.1154
Jan
Apr
Jul
Oct
Bias p−val: 0.5814
Bias p−val: 0.9961
Catch OSA residuals
Index 1 OSA residuals
Index 2 OSA residuals
LBox p−val: 0.2109
LBox p−val: 0.077
LBox p−val: 0.6116
lag.signf: 1
Catch ACF
Index 1 ACF
Index 2 ACF
Lag
Shapiro p−val: 0.0531
Shapiro p−val: 0.6016
Shapiro p−val: 0.5279
Sample Quantiles
Theoretical Quantiles
spict_v1.2.8@ca04322e9438604035be73aa55ecb8ddc146c360




```r
## If runretro is TRUE, run and plot the retrospective analysis
if (runretro & converged) {
  fit_NS <- retro(fit_NS)
  plotspict.retro(fit_NS)
}
```

Scenario 3

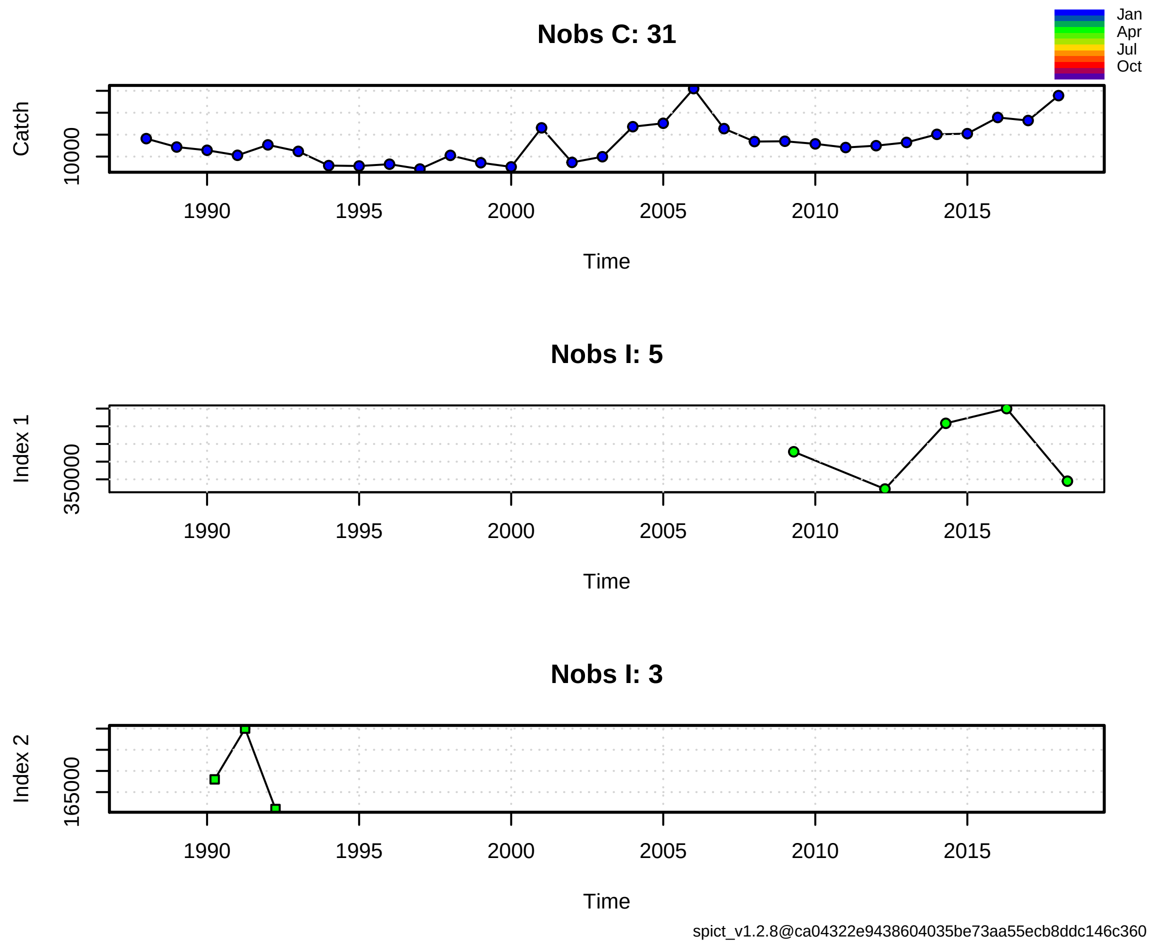
Table 3: Input data for Scenario 3

Input data	Name	Range	Notes
Catch	Total catch	1988-2018	
Biomass indices	Acoustic survey	2009–2018	Matlab
Biomass indices	Acoustic survey	1990-1993	Monstad
			Default priors

```
## Choose only the years where the survey was in October
w <- !is.na(dat$northsea_month) & dat$northsea_month == 10
## Choose only the years where the survey was in January or February
##v <- !is.na(dat$northsea_month) & dat$northsea_month %in% c(1, 2)


## Make the input list
inp_NS <- list(timeC = dat$year,                                    ## Timing of catch
               obsC = dat$catchTOT,                                 ## Observed catches
               timeI = list(dat$year+3.5/12,dat$year+3/12),## Timing of survey index
               obsI = list(dat$norwegian_seaAC,dat$Norwegian_seaAC_Monstad), ## Observed indices
               optimiser.control = list(iter.max = 1e3,             ## Optimiser options
                                        eval.max = 1e3),            ## sometimes help converge
               priors = list(                                       ## List of priors (empty, i.e. de
 #logn=c(log(2),.5,1),
 #logbkfrac=c(log(.5),1,1)
## see possible priors with list.possible.priors()
                ))
## Check input time series, remove missing and zero observations
inp_NS <- check.inp(inp_NS)
```

```
## Removing zero, negative, and NAs in  C  series
## Removing zero, negative, and NAs in  I  series  1
## Removing zero, negative, and NAs in  I  series  2
```

```
## Plot input data
plotspict.data(inp_NS)
```

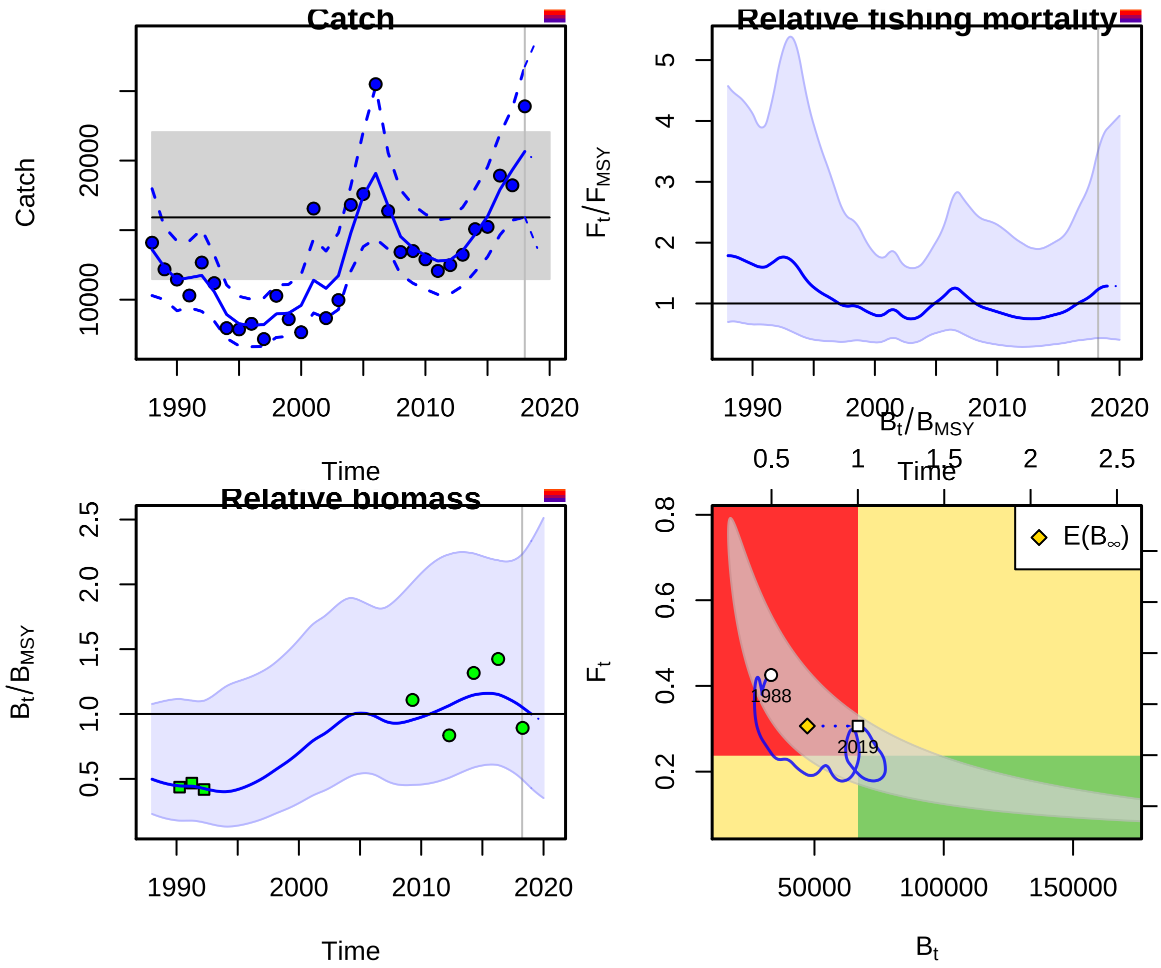



```
## Fit spict
fit_NS <- fit.spict(inp_NS)

## Summary of the fit - in the vignette there is a line-by-line description of that summary
fit_NS
```

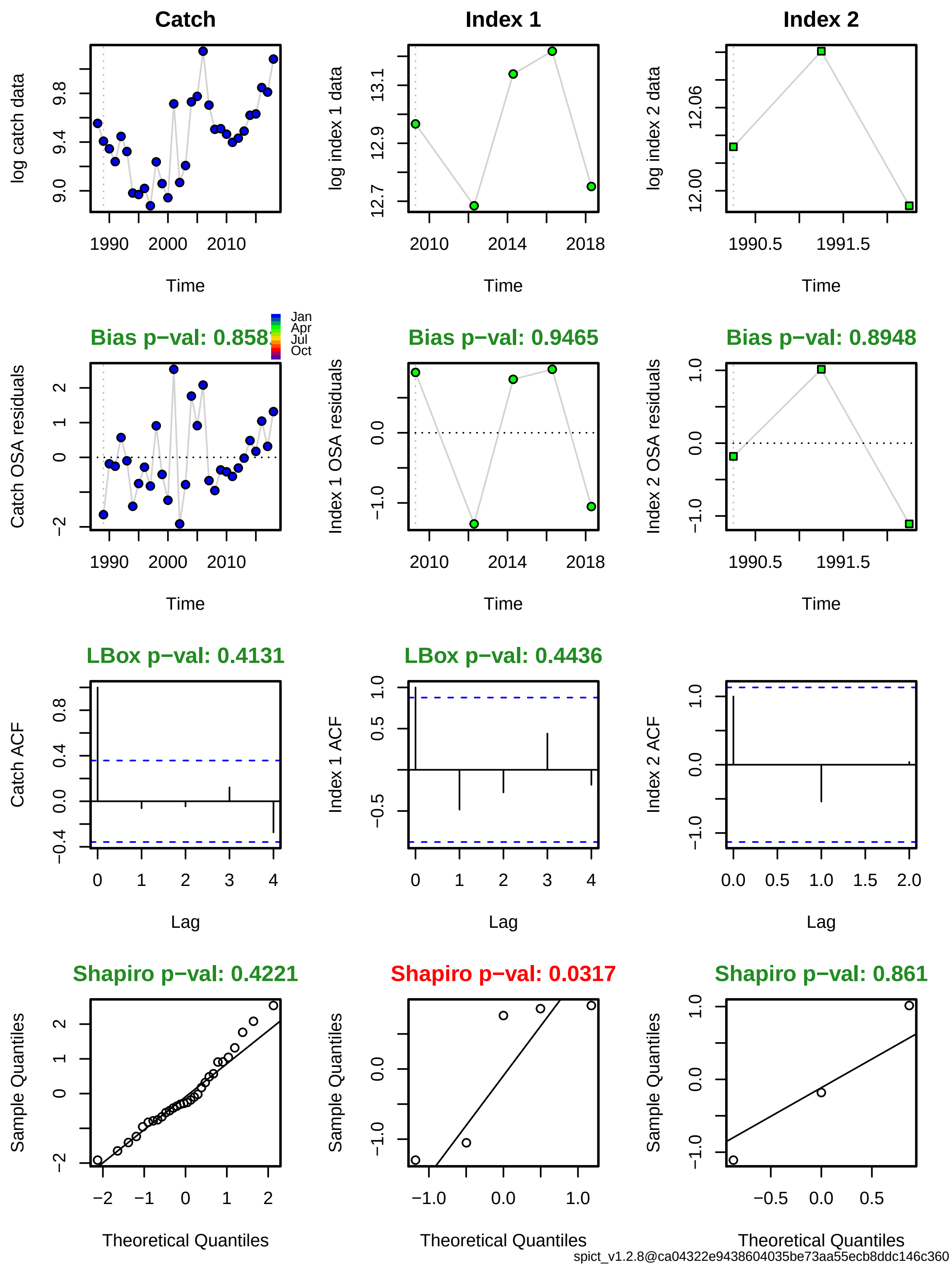
```
## Convergence: 0  MSG: relative convergence (4)
## Objective function at optimum: 14.3407429
## Euler time step (years):  1/16 or 0.0625
## Nobs C: 31,  Nobs I1: 5,  Nobs I2: 3
##
## Priors
##      logn  ~  dnorm[log(2), 2^2]
##  logalpha  ~  dnorm[log(1), 2^2]
##   logbeta  ~  dnorm[log(1), 2^2]
##
## Model parameter estimates w 95% CI
##             estimate        cilow        ciupp    log.est
##  alpha1 5.756276e+00 8.811767e-01 3.760280e+01  1.7502908
##  alpha2 1.431561e+00 1.678045e-01 1.221284e+01  0.3587658
##  beta   9.080295e-01 3.292256e-01 2.504415e+00 -0.0964785
##  r      4.896562e-01 4.112190e-02 5.830555e+00 -0.7140517
##  rc     4.770468e-01 1.828155e-01 1.244827e+00 -0.7401407
```

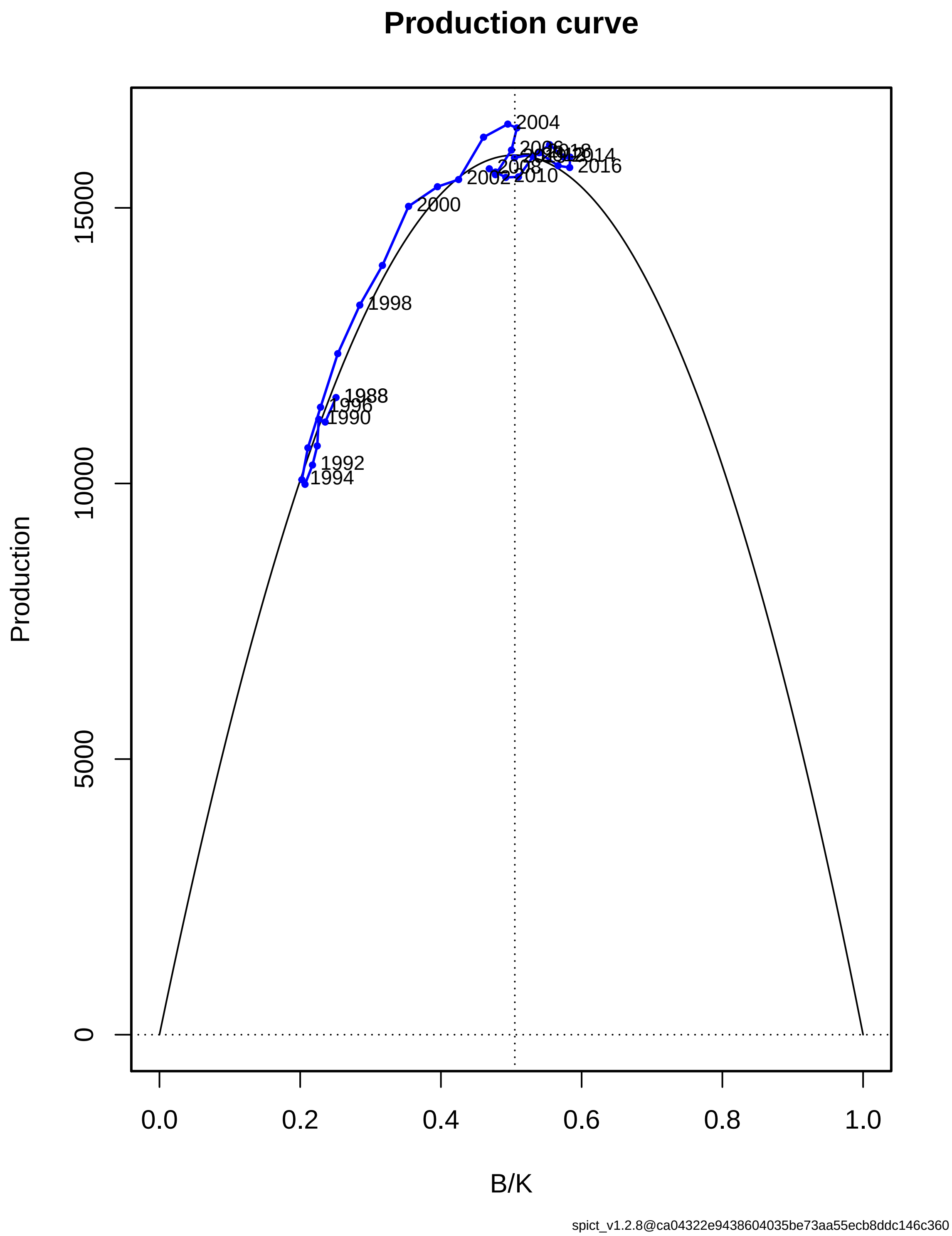
```
## rold   4.650705e-01 1.956400e-02 1.105556e+01 -0.7655663
## m      1.595670e+04 1.147293e+04 2.219278e+04  9.6776338
## K      1.324612e+05 4.637432e+04 3.783552e+05 11.7940448
## q1     5.779729e+00 1.367156e+00 2.443413e+01  1.7543568
## q2     5.761715e+00 1.062269e+00 3.125138e+01  1.7512351
## n      2.052865e+00 1.303502e-01 3.233023e+01  0.7192361
## sdb    3.434090e-02 5.769200e-03 2.044128e-01 -3.3714171
## sdf    1.768985e-01 8.492990e-02 3.684577e-01 -1.7321790
## sdi1   1.976759e-01 1.014676e-01 3.851061e-01 -1.6211263
## sdi2   4.916120e-02 1.558680e-02 1.550557e-01 -3.0126513
## sdc    1.606291e-01 1.055958e-01 2.443441e-01 -1.8286574
##
## Deterministic reference points (Drp)
##           estimate        cilow        ciupp   log.est
## Bmsyd 6.689782e+04 2.193877e+04 2.039913e+05 11.110922
## Fmsyd 2.385234e-01 9.140780e-02 6.224133e-01 -1.433288
## MSYd  1.595670e+04 1.147293e+04 2.219278e+04  9.677634
## Stochastic reference points (Srp)
##           estimate        cilow        ciupp   log.est rel.diff.Drp
## Bmsys 6.679055e+04 2.190420e+04 2.036585e+05 11.109317 -0.001606076
## Fmsys 2.382187e-01 9.101280e-02 6.235185e-01 -1.434566 -0.001279081
## MSYs  1.591072e+04 1.148177e+04 2.204808e+04  9.674749 -0.002889347
##
## States w 95% CI (inp$msytype: s)
##                     estimate        cilow        ciupp    log.est
## B_2018.25      7.035036e+04 1.489201e+04 3.323375e+05 11.1612432
## F_2018.25      2.935164e-01 6.110920e-02 1.409802e+00 -1.2258218
## B_2018.25/Bmsy 1.053298e+00 4.969773e-01 2.232369e+00  0.0519263
## F_2018.25/Fmsy 1.232130e+00 4.324268e-01 3.510755e+00  0.2087444
##
## Predictions w 95% CI (inp$msytype: s)
##                   prediction        cilow        ciupp    log.est
## B_2019.00      6.682726e+04 1.260149e+04 3.543933e+05 11.1098663
## F_2019.00      3.064146e-01 6.034140e-02 1.555978e+00 -1.1828162
## B_2019.00/Bmsy 1.000550e+00 4.288084e-01 2.334608e+00  0.0005494
## F_2019.00/Fmsy 1.286274e+00 4.273219e-01 3.871792e+00  0.2517500
## Catch_2019.00  1.987688e+04 1.371934e+04 2.879807e+04  9.8973127
## E(B_inf)       4.723742e+04           NA           NA 10.7629416
```

```r
## If the model converged, it reports convergence as 0
## Continue with plotting and diagnostics only if convergence was reached
converged <- fit_NS$opt$convergence == 0
if (converged) {
  ## Calculate the One Step Ahead (osa) residuals
  fit_NS <- calc.osa.resid((fit_NS))

  ## Make a plot showing relative F, relative B, Kobe plot catch
  par(mfrow = c(2,2),  ## 2x2 subplots
      mar = c(4.1, 4.1, 0.5, 0.5)) ## Change default margins for the plots
  plotspict.catch(fit_NS)
  plotspict.ffmsy(fit_NS)
  plotspict.bbmsy(fit_NS)
  plotspict.fb(fit_NS)
}
```




```
if (converged) {
  plotspict.diagnostic(fit_NS)
  plotspict.production(fit_NS)
}
```


Catch
Index 1
Index 2
log catch data
log index 1 data
log index 2 data
Time
Bias p−val: 0.858
Jan
Apr
Jul
Oct
Bias p−val: 0.9465
Bias p−val: 0.8948
Catch OSA residuals
Index 1 OSA residuals
Index 2 OSA residuals
LBox p−val: 0.4131
LBox p−val: 0.4436
Catch ACF
Index 1 ACF
Index 2 ACF
Lag
Shapiro p−val: 0.4221
Shapiro p−val: 0.0317
Shapiro p−val: 0.861
Sample Quantiles
Theoretical Quantiles
spict_v1.2.8@ca04322e9438604035be73aa55ecb8ddc146c360




```r
  ## If runretro is TRUE, run and plot the retrospective analysis
  #runretro=TRUE
if (runretro & converged) {
    fit_NS <- retro(fit_NS)
    plotspict.retro(fit_NS)
```

```
}
```

Scenario 4

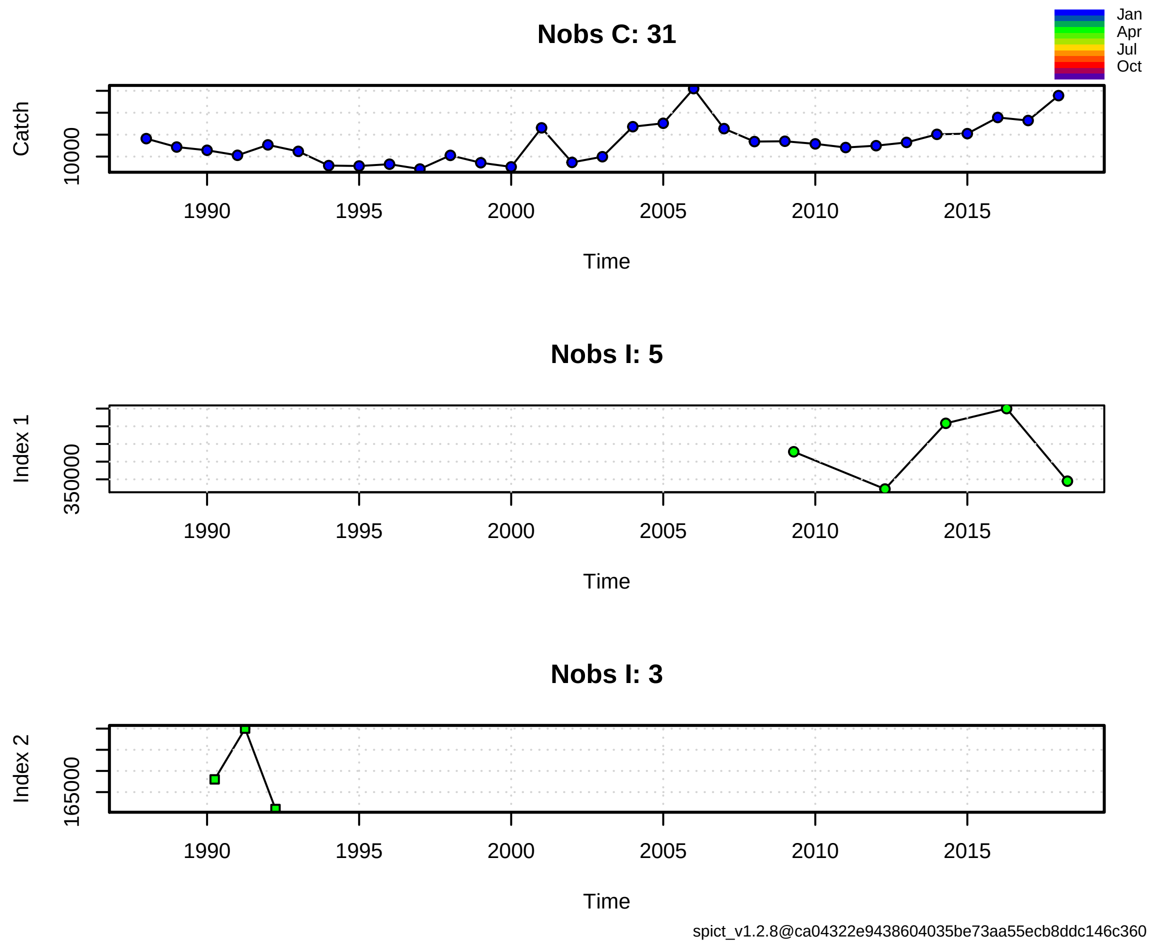
Table 4: Input data for Scenario 4

Input data	Name	Range	Notes
Catch	Total catch	1988-2018	
Biomass indices	Acoustic survey	2012–2018	StoX
Biomass indices	Acoustic survey	1990-1993	Monstad
			Default priors

```
## Choose only the years where the survey was in October
w <- !is.na(dat$northsea_month) & dat$northsea_month == 10
w[1:4]<-FALSE
## Choose only the years where the survey was in January or February
##v <- !is.na(dat$northsea_month) & dat$northsea_month %in% c(1, 2)


## Make the input list
inp_NS <- list(timeC = dat$year,                                    ## Timing of catch
               obsC = dat$catchTOT,                                 ## Observed catches
               timeI = list(dat$year+3.5/12, dat$year+3/12),
               obsI = list(dat$norwegian_seaAC,dat$Norwegian_seaAC_Monstad),
               optimiser.control = list(iter.max = 1e3,             ## Optimiser options
                                        eval.max = 1e3),            ## sometimes help converge
               priors = list(                                       ## List of priors (empty, i.e. de
 logn=c(log(2),.5,1)
 #logbkfrac=c(log(.5),1,1)
## see possible priors with list.possible.priors()
                 ))
## Check input time series, remove missing and zero observations
inp_NS <- check.inp(inp_NS)
```

```
## Removing zero, negative, and NAs in  C  series
## Removing zero, negative, and NAs in  I  series  1
## Removing zero, negative, and NAs in  I  series  2
```

```
## Plot input data
plotspict.data(inp_NS)
```

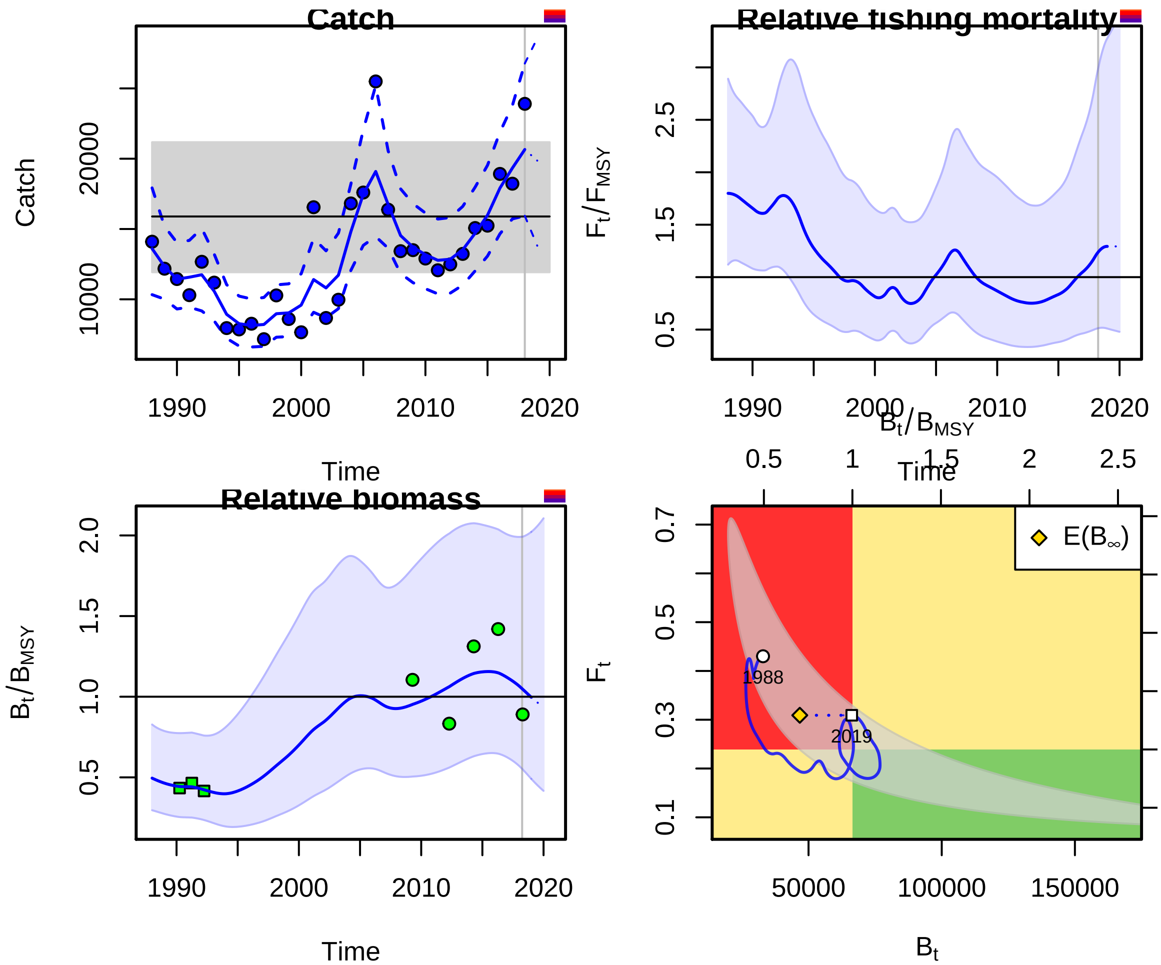



```r
## Fit spict
fit_AC <- fit.spict(inp_NS)

## Summary of the fit - in the vignette there is a line-by-line description of that summary
fit_AC
```

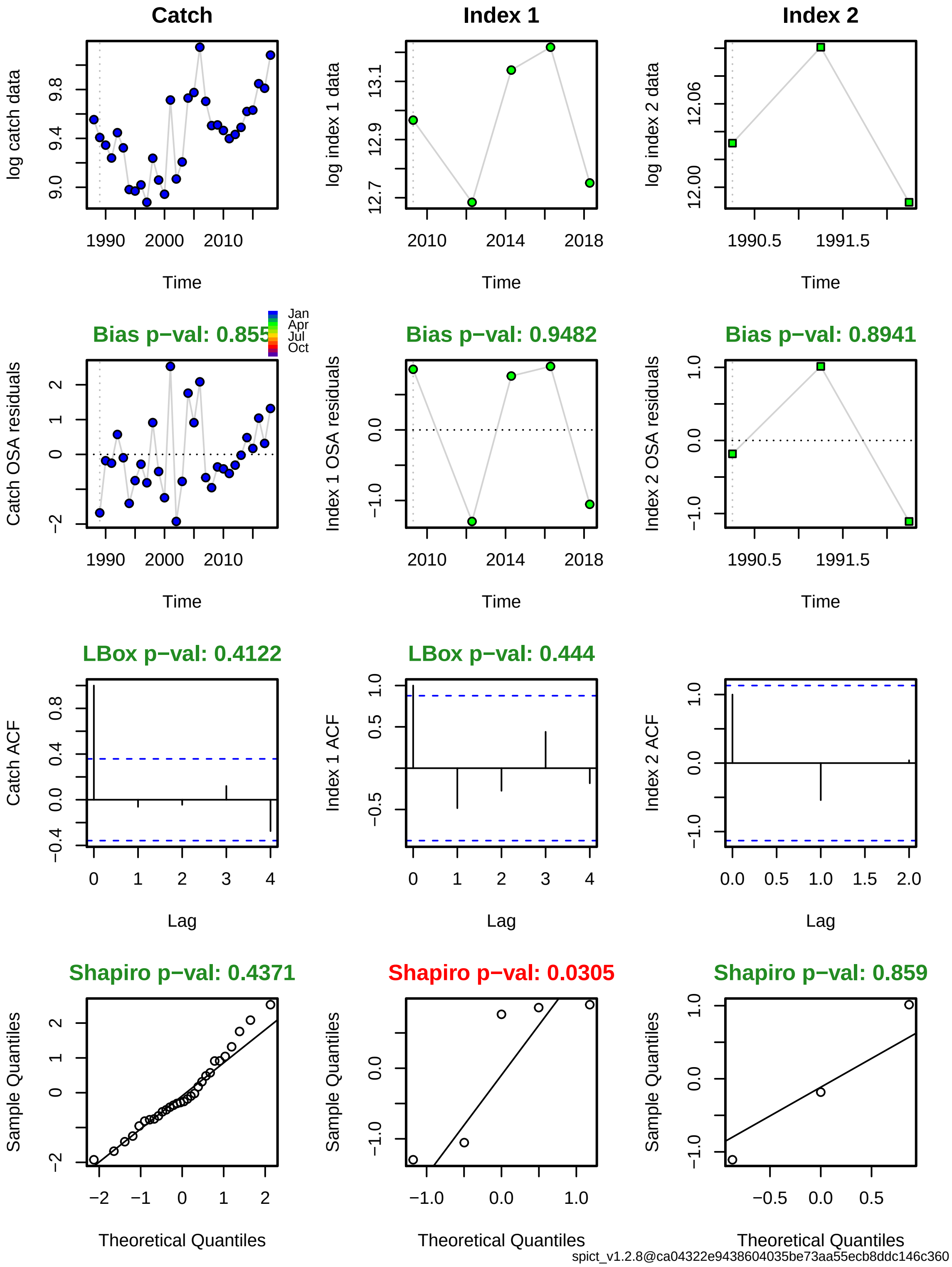
```
## Convergence: 0  MSG: relative convergence (4)
## Objective function at optimum: 12.9546007
## Euler time step (years):  1/16 or 0.0625
## Nobs C: 31,  Nobs I1: 5,  Nobs I2: 3
##
## Priors
##      logn  ~  dnorm[log(2), 0.5^2]
##  logalpha  ~  dnorm[log(1), 2^2]
##   logbeta  ~  dnorm[log(1), 2^2]
##
## Model parameter estimates w 95% CI
##             estimate       cilow       ciupp    log.est
##  alpha1 5.735290e+00 9.167939e-01 3.587889e+01  1.7466383
##  alpha2 1.426235e+00 1.733854e-01 1.173193e+01  0.3550379
##  beta   9.053337e-01 3.469476e-01 2.362401e+00 -0.0994517
##  r      4.802329e-01 1.475322e-01 1.563209e+00 -0.7334841
##  rc     4.787391e-01 1.996902e-01 1.147734e+00 -0.7365994
```

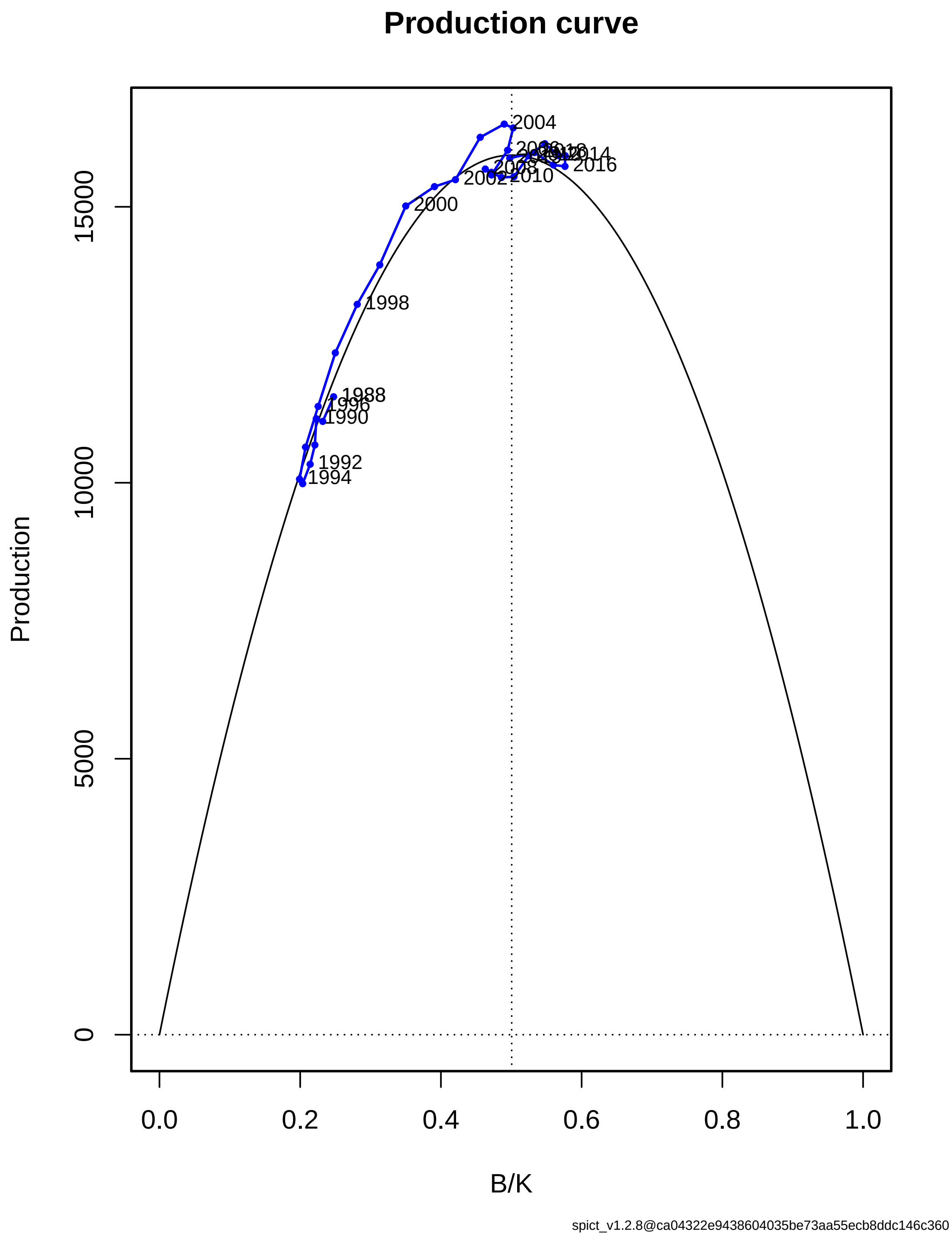
```
##  rold   4.772546e-01 1.191177e-01 1.912159e+00 -0.7397051
##  m      1.593563e+04 1.194744e+04 2.125514e+04  9.6763129
##  K      1.329867e+05 5.068315e+04 3.489414e+05 11.7980040
##  q1     5.828785e+00 1.983238e+00 1.713094e+01  1.7628085
##  q2     5.828383e+00 1.963658e+00 1.729937e+01  1.7627397
##  n      2.006240e+00 7.762974e-01 5.184869e+00  0.6962625
##  sdb    3.446170e-02 6.040500e-03 1.966066e-01 -3.3679081
##  sdf    1.772791e-01 8.862120e-02 3.546316e-01 -1.7300302
##  sdi1   1.976476e-01 1.013611e-01 3.854000e-01 -1.6212698
##  sdi2   4.915040e-02 1.551090e-02 1.557466e-01 -3.0128701
##  sdc    1.604967e-01 1.065456e-01 2.417668e-01 -1.8294819
##
## Deterministic reference points (Drp)
##            estimate        cilow        ciupp   log.est
##  Bmsyd 6.657334e+04 2.523644e+04 1.756194e+05 11.106060
##  Fmsyd 2.393696e-01 9.984510e-02 5.738668e-01 -1.429747
##  MSYd  1.593563e+04 1.194744e+04 2.125514e+04  9.676313
## Stochastic reference points (Srp)
##            estimate        cilow        ciupp   log.est rel.diff.Drp
##  Bmsys 6.646671e+04 2.519372e+04 1.753541e+05 11.104456 -0.001604306
##  Fmsys 2.390763e-01 9.966970e-02 5.734692e-01 -1.430972 -0.001226657
##  MSYs  1.589059e+04 1.192444e+04 2.117590e+04  9.673482 -0.002834767
##
## States w 95% CI (inp$msytype: s)
##                     estimate        cilow        ciupp    log.est
##  B_2018.25      6.975339e+04 2.066381e+04 2.354616e+05 11.1527212
##  F_2018.25      2.960679e-01 8.687730e-02 1.008966e+00 -1.2171663
##  B_2018.25/Bmsy 1.049448e+00 5.530032e-01 1.991566e+00  0.0482647
##  F_2018.25/Fmsy 1.238382e+00 5.132862e-01 2.987789e+00  0.2138060
##
## Predictions w 95% CI (inp$msytype: s)
##                   prediction        cilow        ciupp    log.est
##  B_2019.00      6.621767e+04 1.790426e+04 2.449015e+05 11.1007026
##  F_2019.00      3.091462e-01 8.612220e-02 1.109719e+00 -1.1739409
##  B_2019.00/Bmsy 9.962532e-01 4.908775e-01 2.021931e+00 -0.0037539
##  F_2019.00/Fmsy 1.293086e+00 5.101852e-01 3.277381e+00  0.2570315
##  Catch_2019.00  1.986372e+04 1.381284e+04 2.856524e+04  9.8966500
##  E(B_inf)       4.670922e+04           NA           NA 10.7516968
```

```
## If the model converged, it reports convergence as 0
## Continue with plotting and diagnostics only if convergence was reached
converged <- fit_AC$opt$convergence == 0
if (converged) {
  ## Calculate the One Step Ahead (osa) residuals
  fit_AC <- calc.osa.resid((fit_AC))

  ## Make a plot showing relative F, relative B, Kobe plot catch
  par(mfrow = c(2,2),  ## 2x2 subplots
      mar = c(4.1, 4.1, 0.5, 0.5)) ## Change default margins for the plots
  plotspict.catch(fit_AC)
  plotspict.ffmsy(fit_AC)
  plotspict.bbmsy(fit_AC)
  plotspict.fb(fit_AC)
}
```




```r
if (converged) {
  plotspict.diagnostic(fit_AC)
  plotspict.production(fit_AC)
}
```


Catch
Index 1
Index 2
log catch data
log index 1 data
log index 2 data
Time
Bias p−val: 0.855
Jan
Apr
Jul
Oct
Bias p−val: 0.9482
Bias p−val: 0.8941
Catch OSA residuals
Index 1 OSA residuals
Index 2 OSA residuals
LBox p−val: 0.4122
LBox p−val: 0.444
Catch ACF
Index 1 ACF
Index 2 ACF
Lag
Shapiro p−val: 0.4371
Shapiro p−val: 0.0305
Shapiro p−val: 0.859
Sample Quantiles
Theoretical Quantiles
spict_v1.2.8@ca04322e9438604035be73aa55ecb8ddc146c360




```r
  ## If runretro is TRUE, run and plot the retrospective analysis
  ## runretro=TRUE
if (runretro & converged) {
    fit_AC <- retro(fit_AC)
    plotspict.retro(fit_AC)
```

```
  }
```

##Scenario 5

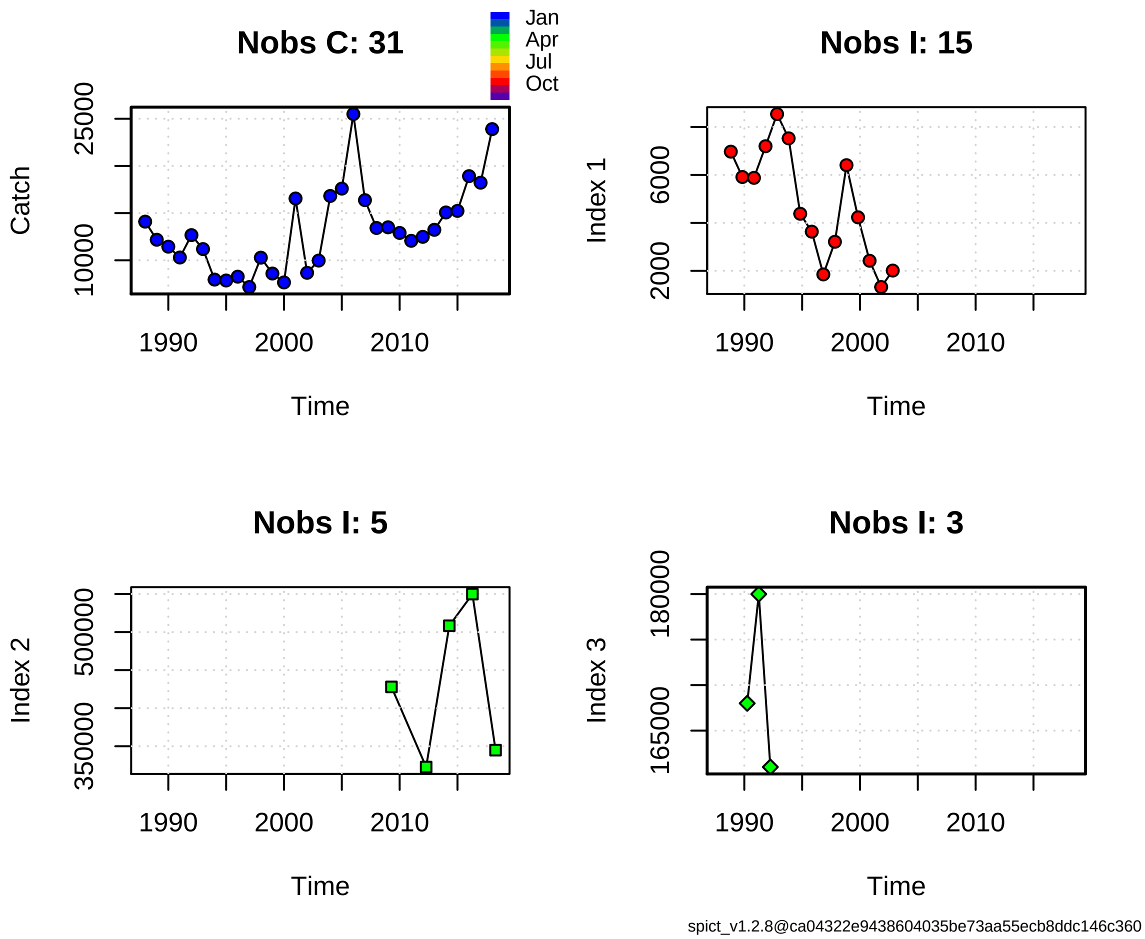
Table 5: Input data for Scenario 5

Input data	Name	Range	Notes
Catch	Total catch	1988-2018	
Biomass indices	Shrimp survey	1984–2002	Only october period
Biomass indices	Acoustic survey	2012–2018	StoX
Biomass indices	Acoustic survey	1990-1993	Monstad
			Default priors

```
## Choose only the years where the schrimp survey was in October
w <- !is.na(dat$northsea_month) & dat$northsea_month == 10
w[1:4]<-FALSE #remove years before 1988 in schrip survey
## Choose only the years where the survey was in January or February
v <- !is.na(dat$northsea_month) & dat$northsea_month %in% c(1)#c(1, 2)
## Make the input list
inp_NS <- list(timeC = dat$year,                                        ## Timing of catch
               obsC = dat$catchTOT,                                     ## Observed catches dat$year[v] +
               timeI = list(dat$year[w] + dat$northsea_month[w] / 12,dat$year+3.5/12, dat$year+3/12),
               obsI = list(dat$northsea_SA[w],dat$norwegian_seaAC,dat$Norwegian_seaAC_Monstad),
               optimiser.control = list(iter.max = 1e3,                 ## Optimiser options
                                        eval.max = 1e3),                ## sometimes help converge

               priors = list(                                           ## List of priors (empty, i.e. de
 logn=c(log(2),.5,1)
 #logbkfrac=c(log(.5),1,1)
## see possible priors with list.possible.priors()
                 ))
## Check input time series, remove missing and zero observations
inp_NS <- check.inp(inp_NS)
```

```
## Removing zero, negative, and NAs in  C  series
## Removing zero, negative, and NAs in  I  series  2
## Removing zero, negative, and NAs in  I  series  3
```

```
## Plot input data
plotspict.data(inp_NS)
```



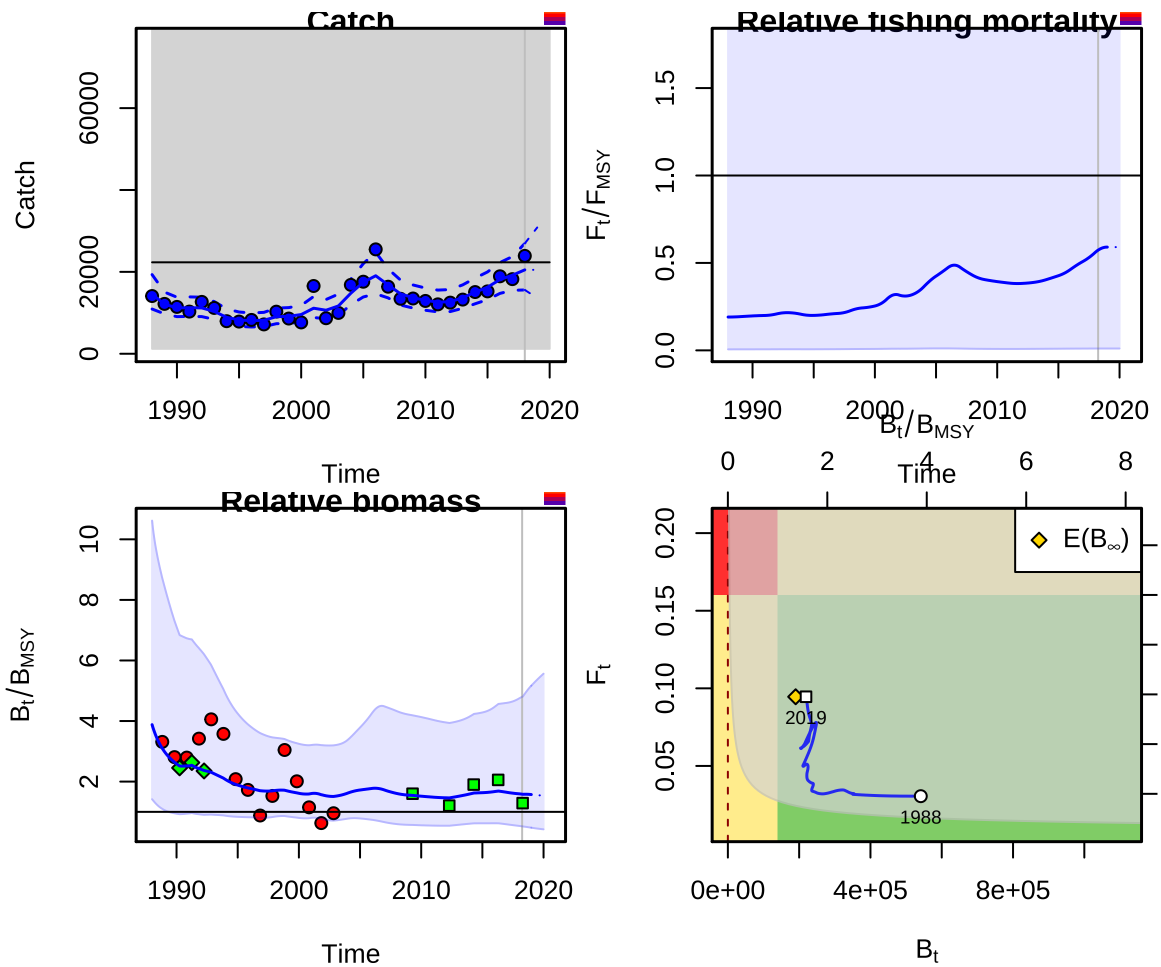

```r
## Fit spict
fit_NS <- fit.spict(inp_NS)

## Summary of the fit - in the vignette there is a line-by-line description of that summary
fit_NS
```

```
## Convergence: 0  MSG: relative convergence (4)
## Objective function at optimum: 24.9215913
## Euler time step (years):  1/16 or 0.0625
## Nobs C: 31,  Nobs I1: 15,  Nobs I2: 5,  Nobs I3: 3
##
## Priors
##      logn  ~  dnorm[log(2), 0.5^2]
##  logalpha  ~  dnorm[log(1), 2^2]
##   logbeta  ~  dnorm[log(1), 2^2]
##
## Model parameter estimates w 95% CI
##            estimate       cilow        ciupp    log.est
##  alpha1 4.067296e+00    1.0756622 1.537927e+01  1.4029784
##  alpha2 1.876796e+00    0.4335163 8.125102e+00  0.6295662
##  alpha3 5.116063e-01    0.0455970 5.740308e+00 -0.6701999
##  beta   1.151900e+00    0.4271710 3.106187e+00  0.1414125
##  r      3.336071e-01    0.0378613 2.939511e+00 -1.0977913
```

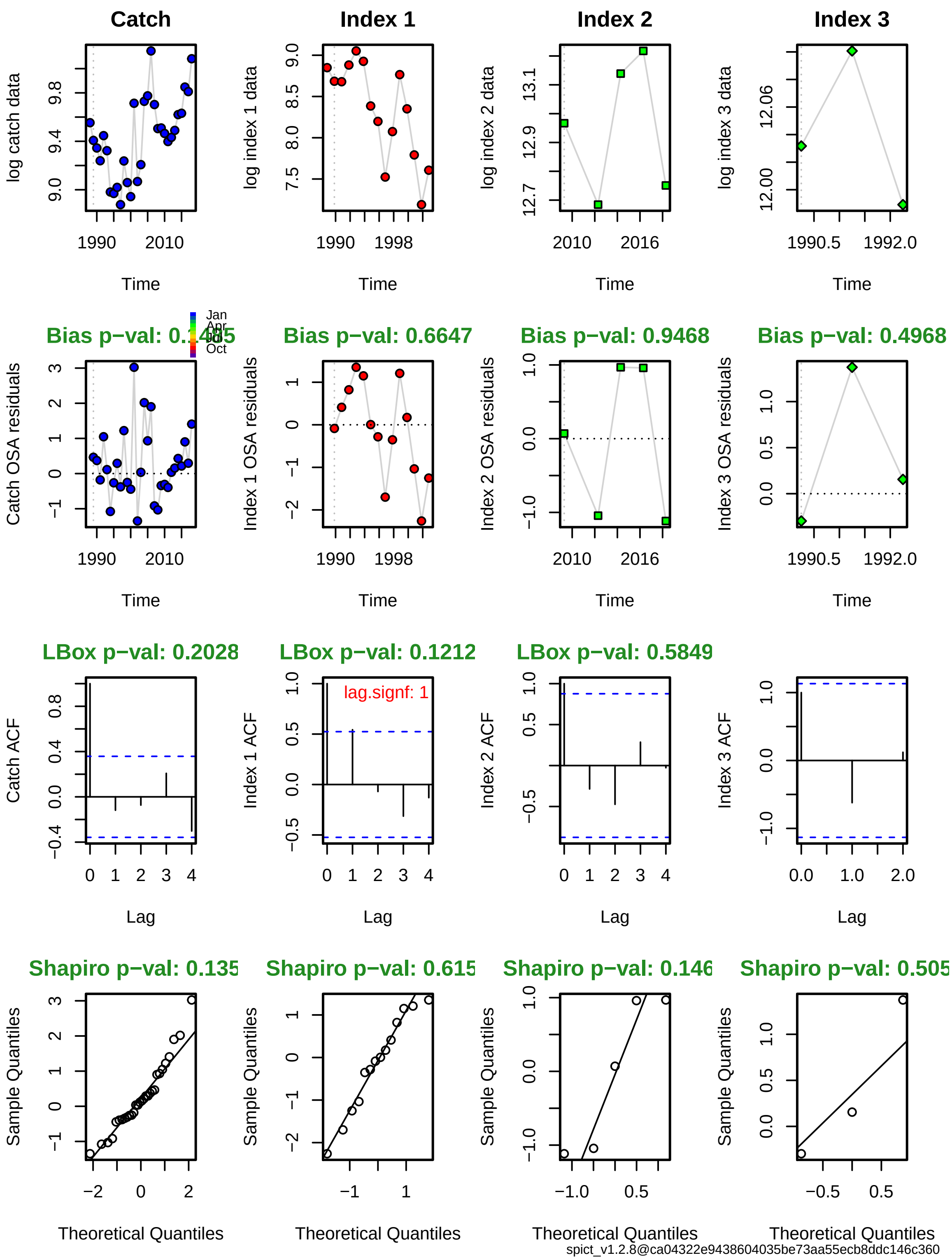
```
##  rc     3.261782e-01    0.0342093 3.110035e+00 -1.1203116
##  rold   3.190729e-01    0.0200195 5.085410e+00 -1.1423358
##  m      2.322878e+04 1204.9317666 4.478063e+05 10.0531472
##  K      2.824040e+05 3754.5623194 2.124136e+07 12.5510939
##  q1     1.509110e-02    0.0001121 2.031102e+00 -4.1936497
##  q2     1.917505e+00    0.0105035 3.500575e+02  0.6510251
##  q3     4.908128e-01    0.0033168 7.262877e+01 -0.7116925
##  n      2.045552e+00    0.6838253 6.118933e+00  0.7156675
##  sdb    1.063386e-01    0.0326205 3.466507e-01 -2.2411267
##  sdf    1.436393e-01    0.0642778 3.209857e-01 -1.9404499
##  sdi1   4.325107e-01    0.2685314 6.966241e-01 -0.8381483
##  sdi2   1.995759e-01    0.0923852 4.311355e-01 -1.6115605
##  sdi3   5.440350e-02    0.0081871 3.615130e-01 -2.9113266
##  sdc    1.654581e-01    0.1120487 2.443257e-01 -1.7990374
##
## Deterministic reference points (Drp)
##            estimate        cilow        ciupp   log.est
##  Bmsyd 1.424300e+05 1767.4629765 1.147764e+07 11.866606
##  Fmsyd 1.630891e-01    0.0171047 1.555017e+00 -1.813459
##  MSYd  2.322878e+04 1204.9317666 4.478063e+05 10.053147
## Stochastic reference points (Srp)
##            estimate        cilow        ciupp   log.est rel.diff.Drp
##  Bmsys 1.394924e+05 1766.4994028 1.101508e+07 11.845765  -0.02105921
##  Fmsys 1.601566e-01    0.0162056 1.582799e+00 -1.831603  -0.01831005
##  MSYs  2.233202e+04 1200.8787732 4.152950e+05 10.013777  -0.04015576
##
## States w 95% CI (inp$msytype: s)
##                     estimate        cilow        ciupp    log.est
##  B_2018.25      2.208996e+05 1085.1370218 4.496816e+07 12.3054634
##  F_2018.25      9.179950e-02    0.0004545 1.854234e+01 -2.3881485
##  B_2018.25/Bmsy 1.583596e+00    0.5227232 4.797521e+00  0.4596981
##  F_2018.25/Fmsy 5.731857e-01    0.0106308 3.090459e+01 -0.5565455
##
## Predictions w 95% CI (inp$msytype: s)
##                   prediction        cilow        ciupp    log.est
##  B_2019.00      2.192208e+05 9.924901e+02 4.842139e+07 12.2978346
##  F_2019.00      9.456920e-02 4.576000e-04 1.954474e+01 -2.3584230
##  B_2019.00/Bmsy 1.571561e+00 4.780774e-01 5.166116e+00  0.4520692
##  F_2019.00/Fmsy 5.904798e-01 1.066420e-02 3.269490e+01 -0.5268199
##  Catch_2019.00  2.049855e+04 1.362291e+04 3.084438e+04  9.9281092
##  E(B_inf)       1.901538e+05           NA           NA 12.1555886
```

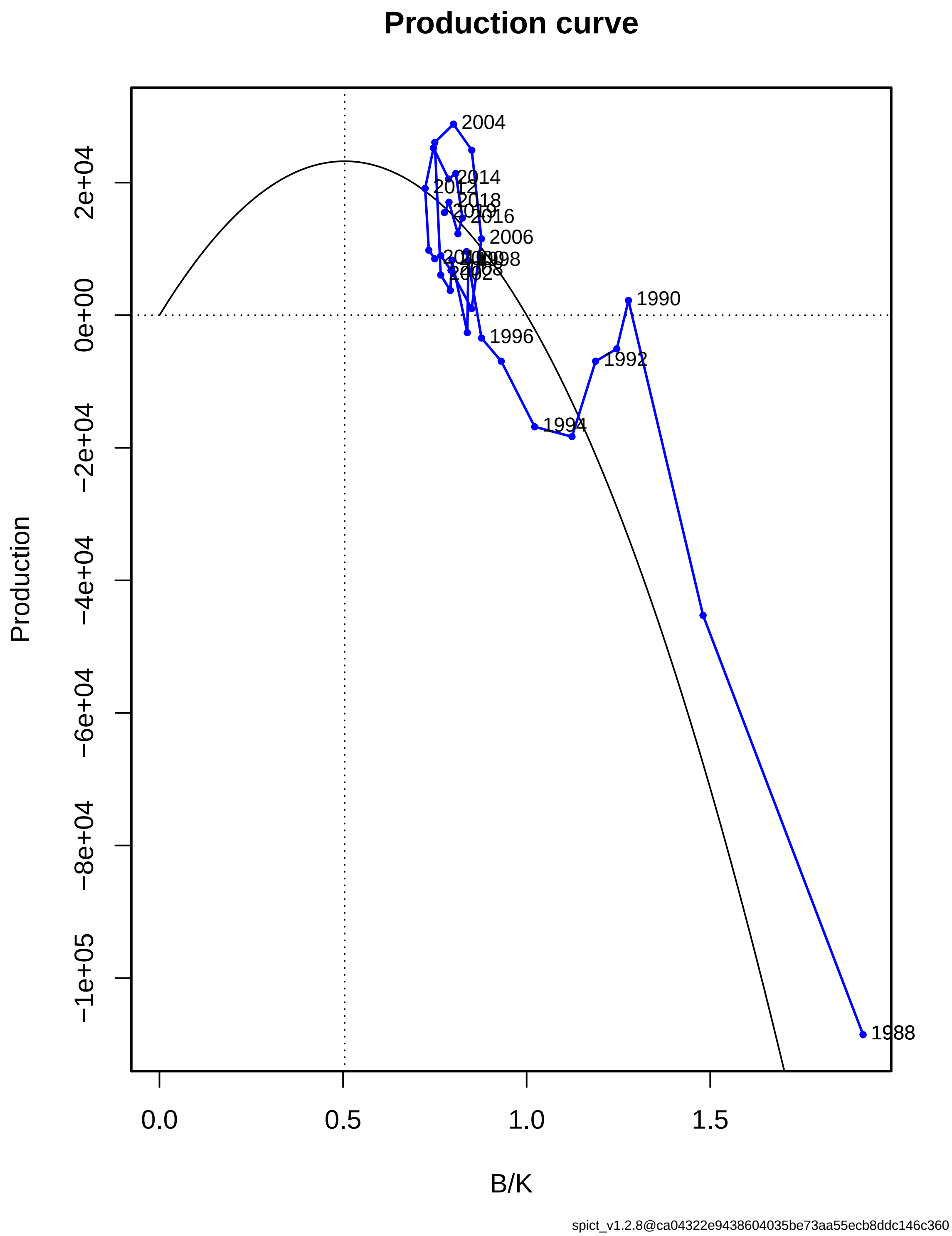
```r
## If the model converged, it reports convergence as 0
## Continue with plotting and diagnostics only if convergence was reached
converged <- fit_NS$opt$convergence == 0
if (converged) {
  ## Calculate the One Step Ahead (osa) residuals
  fit_NS <- calc.osa.resid((fit_NS))

  ## Make a plot showing relative F, relative B, Kobe plot catch
  par(mfrow = c(2,2),  ## 2x2 subplots
      mar = c(4.1, 4.1, 0.5, 0.5)) ## Change default margins for the plots
  plotspict.catch(fit_NS)
  plotspict.ffmsy(fit_NS)
```

```
  plotspict.bbmsy(fit_NS)
  plotspict.fb(fit_NS)
}
```



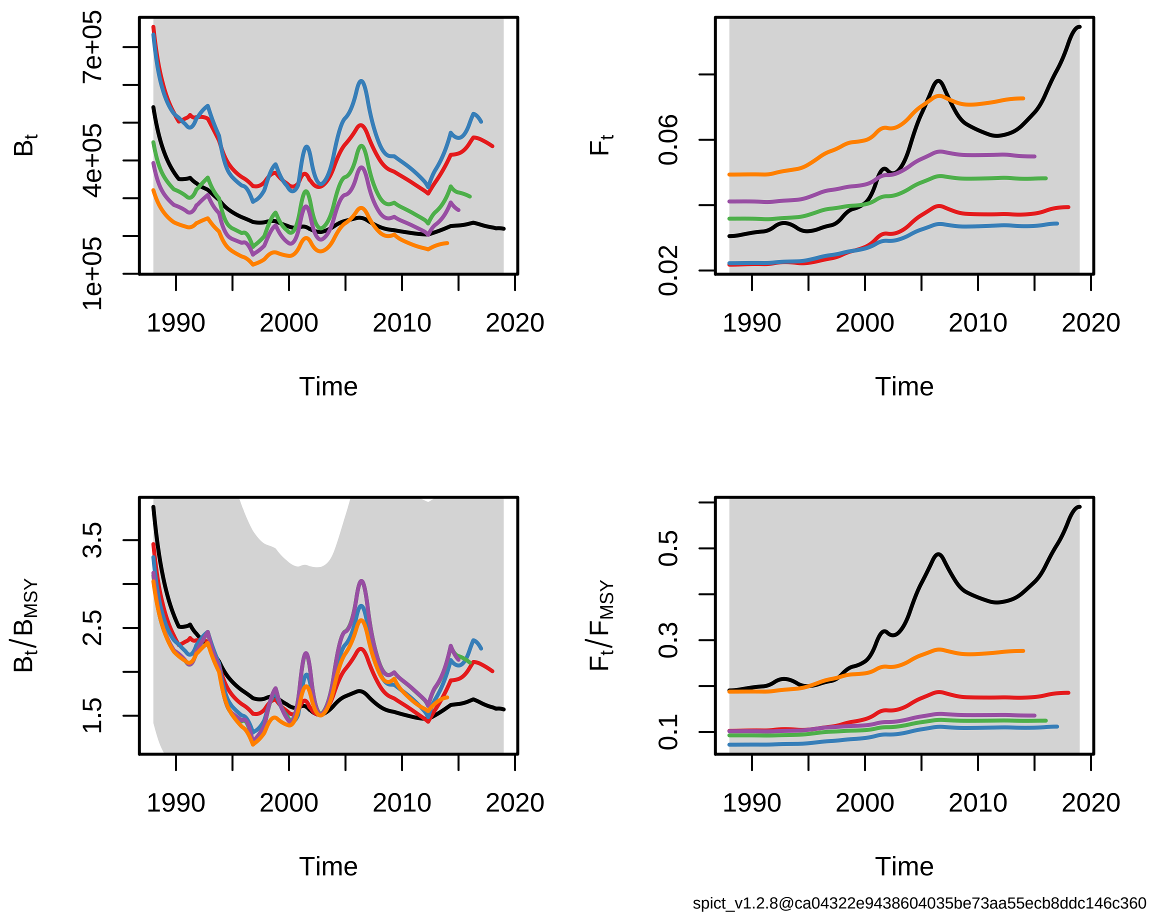

```
if (converged) {
  plotspict.diagnostic(fit_NS)
  plotspict.production(fit_NS)
}
```


Catch
Index 1
Index 2
Index 3
log catch data
log index 1 data
log index 2 data
log index 3 data
Time
Jan
Apr
Jul
Oct
Bias p−val: 0.1485
Bias p−val: 0.6647
Bias p−val: 0.9468
Bias p−val: 0.4968
Catch OSA residuals
Index 1 OSA residuals
Index 2 OSA residuals
Index 3 OSA residuals
LBox p−val: 0.2028
LBox p−val: 0.1212
LBox p−val: 0.5849
lag.signf: 1
Catch ACF
Index 1 ACF
Index 2 ACF
Index 3 ACF
Lag
Shapiro p−val: 0.135
Shapiro p−val: 0.615
Shapiro p−val: 0.146
Shapiro p−val: 0.505
Sample Quantiles
Theoretical Quantiles
spict_v1.2.8@ca04322e9438604035be73aa55ecb8ddc146c360



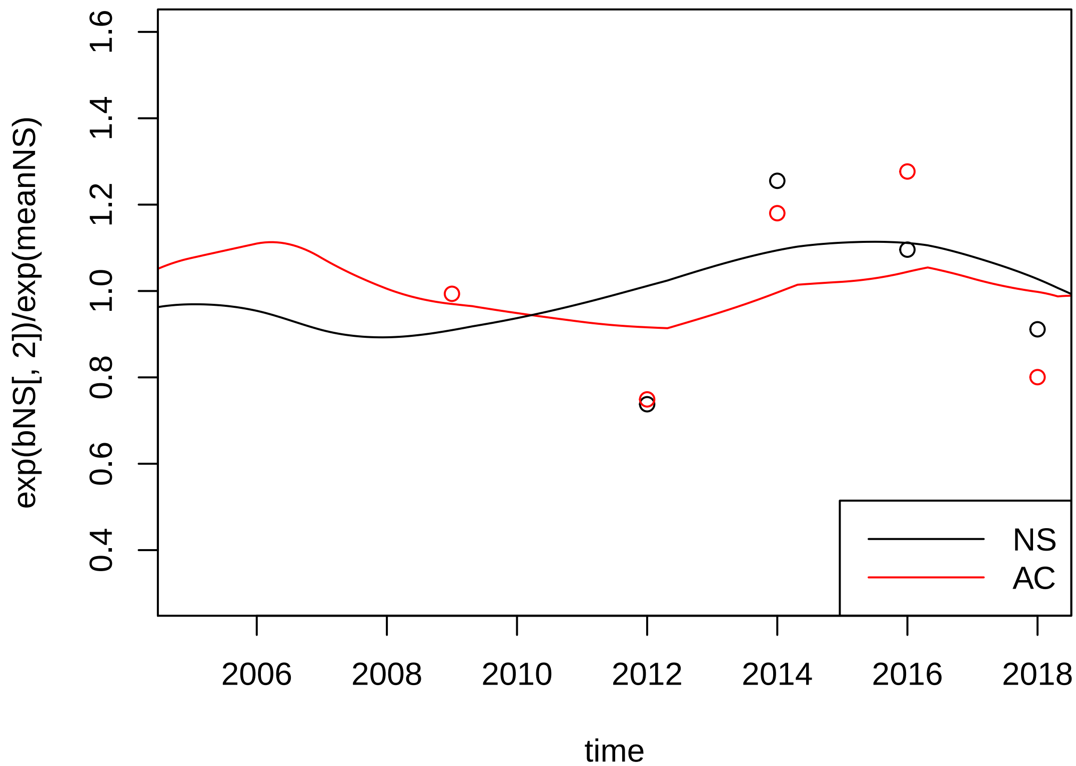

```
## If runretro is TRUE, run and plot the retrospective analysis
  #runretro=TRUE
if (runretro & converged) {
    fit_NS <- retro(fit_NS)
    plotspict.retro(fit_NS)
```

```
}
```



Biomass comparsin scenario 4 and 5

```
bAC <- get.par(parname = "logBBmsy",fit_AC)
time <- as.numeric(rownames(bAC))
bNS <- get.par(parname = "logBBmsy",fit_NS)



meanAC <- mean(bAC[,2][time > 2005 & time < 2019])
meanNS <- mean(bNS[,2][time > 2005 & time < 2019])
ylim <- c(0.3, 1.6)
plot(time, exp(bNS[,2]) / exp(meanNS),col=2,type="l", ylim = ylim, xlim=c(2005,2018))
lines(time, exp(bAC[,2]) / exp(meanAC))
legend("bottomright",,c("NS", "AC"), col = 1:2, seg.len = 4, lty = 1)
points(dat$year, dat$norwegian_sea_AC_stox / mean(dat$norwegian_sea_AC_stox, na.rm = TRUE))
points(dat$year, dat$norwegian_seaAC / mean(dat$norwegian_seaAC ,na.rm = TRUE), col="red")
```



Referneces

Pedersen, Martin W., and Casper W. Berg. 2017. "A stochastic surplus production model in continuous time." *Fish and Fisheries* 18 (2): 226–43. https://doi.org/10.1111/faf.12174.