

Aplicación de Stochastic Production Model in Continuous Time (SPiCT) (Pedersen et al., 2017) en Sardina Austral de la XI Región

Mardones, M., Zúñiga, MJ., Pérez, MC

22 April, 2021

Contents

| | |
|--|----|
| Breve descripción del keystone paper (Pedersen & Berg, 2016) | 2 |
| Principales supuestos y requerimiento de datos para SPiCT. | 3 |
| Carga de librerías necesarias. | 4 |
| Main assumptions and input data for SPiCT | 5 |
| Incorporando la propia data | 6 |
| Ploteando Data de Sardina Austral en la XI | 7 |
| Ahora aplico el modelo | 9 |
| Interpretando los resultados | 14 |
| Diagnostico y residuos | 15 |
| Extrear parametros estimados | 16 |
| Análisis retrospectivo | 17 |
| Escenarios de Manejo | 18 |
| Fijando parametros | 23 |
| Brechas analíticas | 25 |

Breve descripción del keystone paper (Pedersen & Berg, 2016)

Este documento provee una guía (casi traducida de Pedersen et al., 2017) para el uso del modelo de producción estatocástico continuo en tiempo (SPiCT) y que está dividido en tres partes

Este es un documento vivo que estará en permanente cambio. Todos los avances y actualizaciones pueden ser seguidas y obtenidas de <https://github.com/DTUAqua/spict/commits/master>. El SPiCT package esta siendo activamente actualizado y se pueden reportar errores aquí: <https://github.com/DTUAqua/spict/releases>.

El modelo de excedente de producción tiene una larga historia como método para manejar las poblaciones de peces con datos limitados. Los avances recientes han arrojado modelos de producción excedente como modelos de espacio de estado que separan la variabilidad aleatoria de la dinámica del stock del error en los índices observados de biomasa. Pedersen & Berg (2016) presentan un modelo de producción excedente estocástico en tiempo continuo (SPiCT), que además de la dinámica del stock también modela la dinámica de las pesquerías. Esto permite que el error en el proceso de captura se refleje en la incertidumbre de los parámetros estimados del modelo y las cantidades de manejo.

La formulación del modelo de espacio de estado de tiempo continuo incluye la capacidad de proporcionar estimaciones de biomasa explotable y mortalidad por pesca en cualquier momento a partir de datos muestreados a intervalos arbitrarios y posiblemente irregulares.

Esta guía es un ejemplo de aplicación del modelo con los datos de la librería y también propone utilizar datos propios.

Principales supuestos y requerimiento de datos para SPiCT.

- Serie de tiempo en dato que contenga cambios forzantes en la población
- Contraste en los datos (Hilborn & Walters, 1989)

-Extreme observations or outliers in index and catch are commonly encountered problem in fisheries data (Chen et al. 1994).

Carga de librerías necesarias.

Lo primero es cargar TMB usando el GitHub usando devtools Package. Aquí se explica como; (<https://github.com/kaskr/adcomp/wiki/Download>)

```
install.packages("devtools")
install.packages("TMB")
#si hay problemas, instalarlo desde el github
devtools::install_github("kaskr/adcomp", subdir = "TMB")
```

Una vez cargado eso se llama de la misma forma al SPiCT package

```
devtools::install_github("DTUAqua/spict/spict")
#devtools::install_github("DTUAqua/spict/spict", ref = "1.2.8")
# aquí algunas dependencias también necesitan ser instaladas
install.packages("Rcpp")
install.packages("ellipse")
```

Aquí a veces hay problemas para acceder a GitHub por problemas en las credenciales. Para ello se debe obtener un token de GitHub (deben tener una cuenta) y hacer un proceso como lo planteado acá: <https://www.r-bloggers.com/using-travis-make-sure-you-use-a-github-pat/>

Una vez solucionando e instalando dependencias, llamamos las librerías:

```
library(usethis)
library(devtools)
library(ellipse)
library(spict) #comprobar esta versión de spict_v1.2.8
library(tidyverse)
library(patchwork)
```

Main assumptions and input data for SPiCT

- Catch data should be representative of both landings and bycatch. It is also possible to use landings only, but then the interpretation of the results changes. If available, seasonal catches should be used as input. Catches are assumed to be taken over a time interval (e.g. years or quarters), thus the associated time vector in SPiCT `inp$timeC` should reflect the beginning of each catch interval (e.g. 2002.25 and 2002.75 for the second and fourth quarter catches, respectively). Additionally, the vector `inp$dtC` should reflect the length of each time interval (e.g. 1 for annual and 0.25 for quarterly catches, respectively).
- Stock size indices should be in terms of biomass (not numbers) and representative of the part of the stock vulnerable to the commercial fleets, the so called exploitable stock biomass (ESB). In many cases, the gear selectivity of the commercial and scientific fleets do not coincide and thus the stock size indices have to be corrected to exclude individuals that are not represented in the commercial fleets.
- Biomass indices are assumed to be snapshots at given points in time. Therefore, the timing of survey indices `inp$timeI` has to be given as decimal years reflecting the timing of the survey (e.g. 1995.5 for the middle of the year). The timing of the survey will be matched to the closest model time which is dependent on `inp$dtEuler` (see below). Commercial CPUE index should be associated with the midpoint of the interval of the corresponding catches, i.e. middle of the year if they are based on yearly aggregated catches and effort.

Incorporando la propia data

Leo el directorio donde contengo los datos. Si estoy trabajando en el proyecto solo vizualisto con `'list.file()'`

```
getwd()
setwd('~')
```

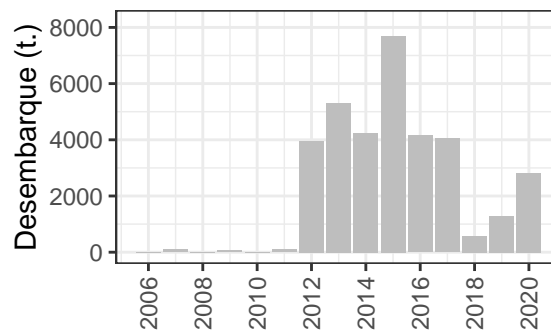
Usaré el ejemplar de datos de erizo de la zona X Norte. Este ejemplo no tiene mucho contraste de datos y veremos los problemas que ello acarrea.

Puedo tener un `.txt` ó un `.csv` pero deben ser transformados en formato lista.

```
sauxi <- read.table("Data_SAus_XI_2021.txt", sep=" ", header = T)
```

#un plot simple de indice y desembarque

```
bro <- ggplot(sauxi, aes(timeC, obsC)) +
  geom_bar(stat="identity", fill="gray") +
  ylim(0, 8200) +
  xlab("") +
  ylab("Desembarque (t.)") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
  scale_x_continuous(breaks = seq(from = 2006, to = 2020, by = 2))
bro
```



Convierto los datos como lista para ser lidos por las funciones del SPiCT

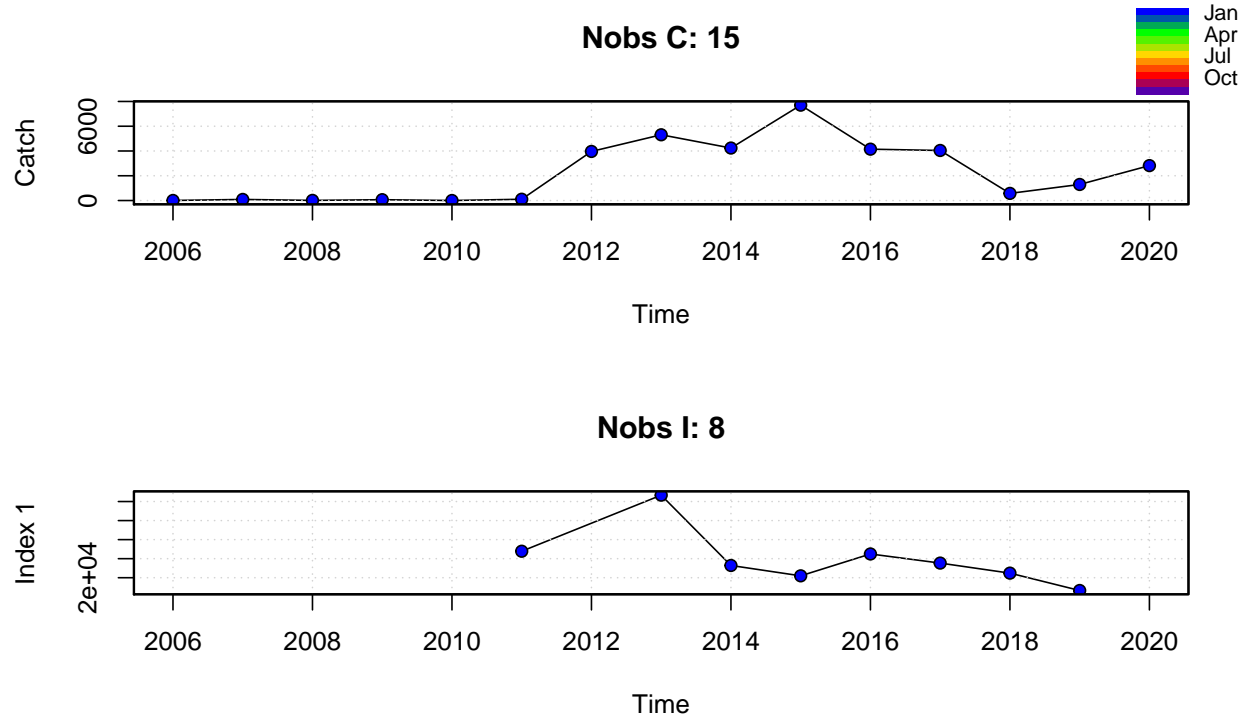
```
sauxi <- as.list(sauxi)
#compruebo
class(sauxi)
[1] "list"
```

Primero un vistazo a mis datos y compruebo el contraste, el que desde tya nos dirá que tan robusta es la estimación.

Tenga en cuenta que los datos están estructurados como una lista que contiene las entradas obsC (observaciones de captura), timeC (tiempo de oservaciones de captura), obsI (índice de observaciones) y timeI (tiempo de observaciones de índice). Si no se especifican los tiempos se supone que la primera observación se observa en el tiempo 1 y luego secuencialmente en adelante con un paso de tiempo de un año. Por lo tanto, se recomienda especificar siempre los tiempos de observación

Ploteando Data de Sardina Austral en la XI

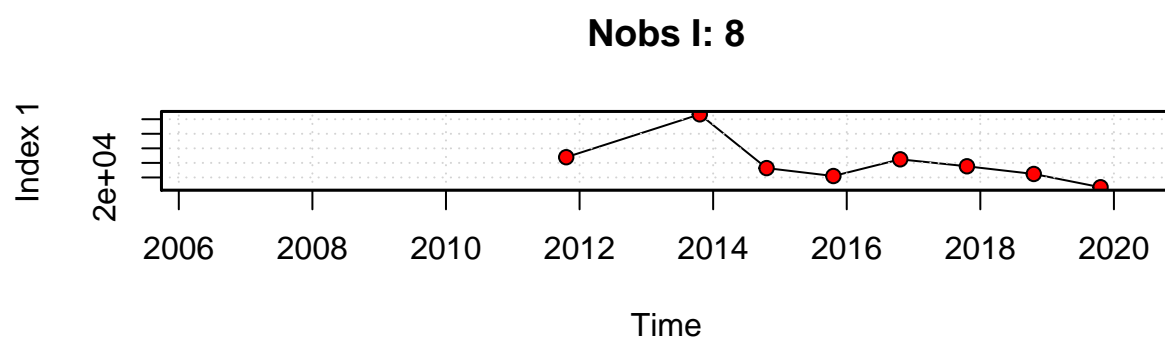
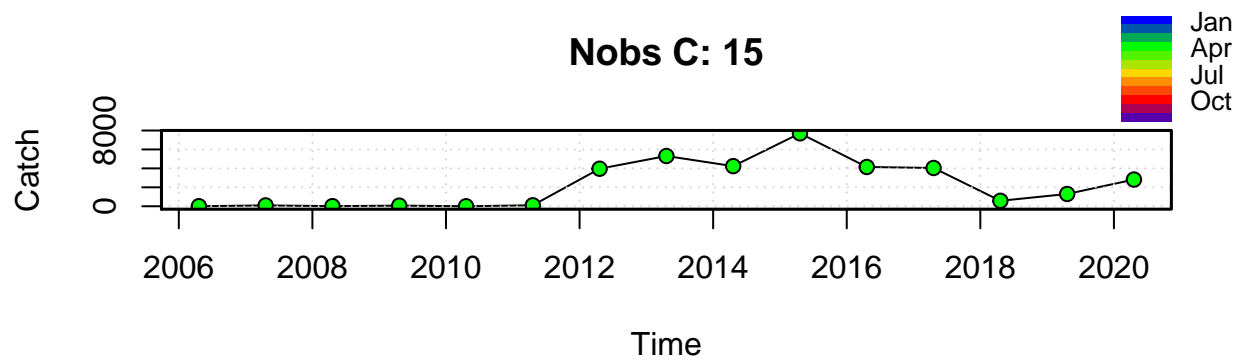
```
plotspict.data(sauxi)
Removing zero, negative, and NAs in I series 1
```



spict_v1.2.8@d9ece0a31623f1a26d3cb4328499f16136822d14

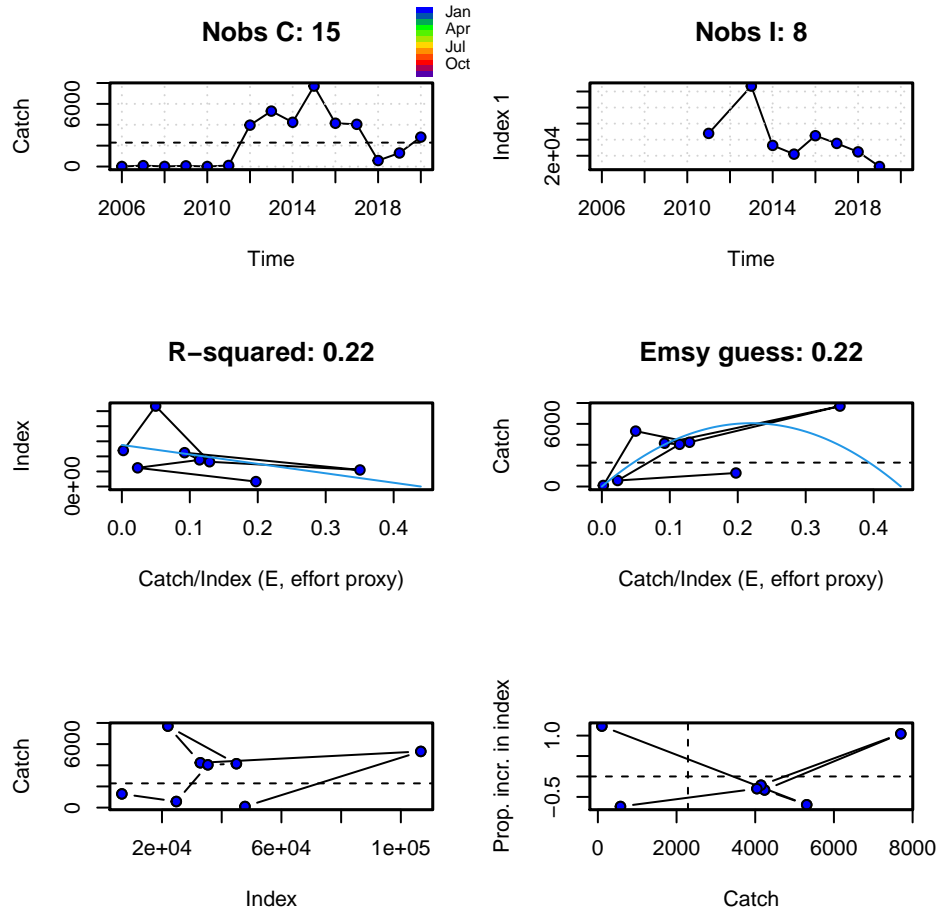
Además, el color de los puntos individuales muestra cuándo se realizó la observación en algún mes y se muestran los colores correspondientes en la leyenda de color en la esquina superior derecha. Con fines ilustrativos, intentemos cambiar un poco los datos.

```
inpshift <- sauxi
inpshift$timeC <- inpshift$timeC + 0.3 #April
inpshift$timeI <- inpshift$timeI + 0.8 #August
plotspict.data(inpshift)
Removing zero, negative, and NAs in I series 1
```



spict_v1.2.8@d9ece0a31623f1a26d3cb4328499f16136822d14

```
plotspict.ci(sauxi)
Removing zero, negative, and NAs in I series 1
```

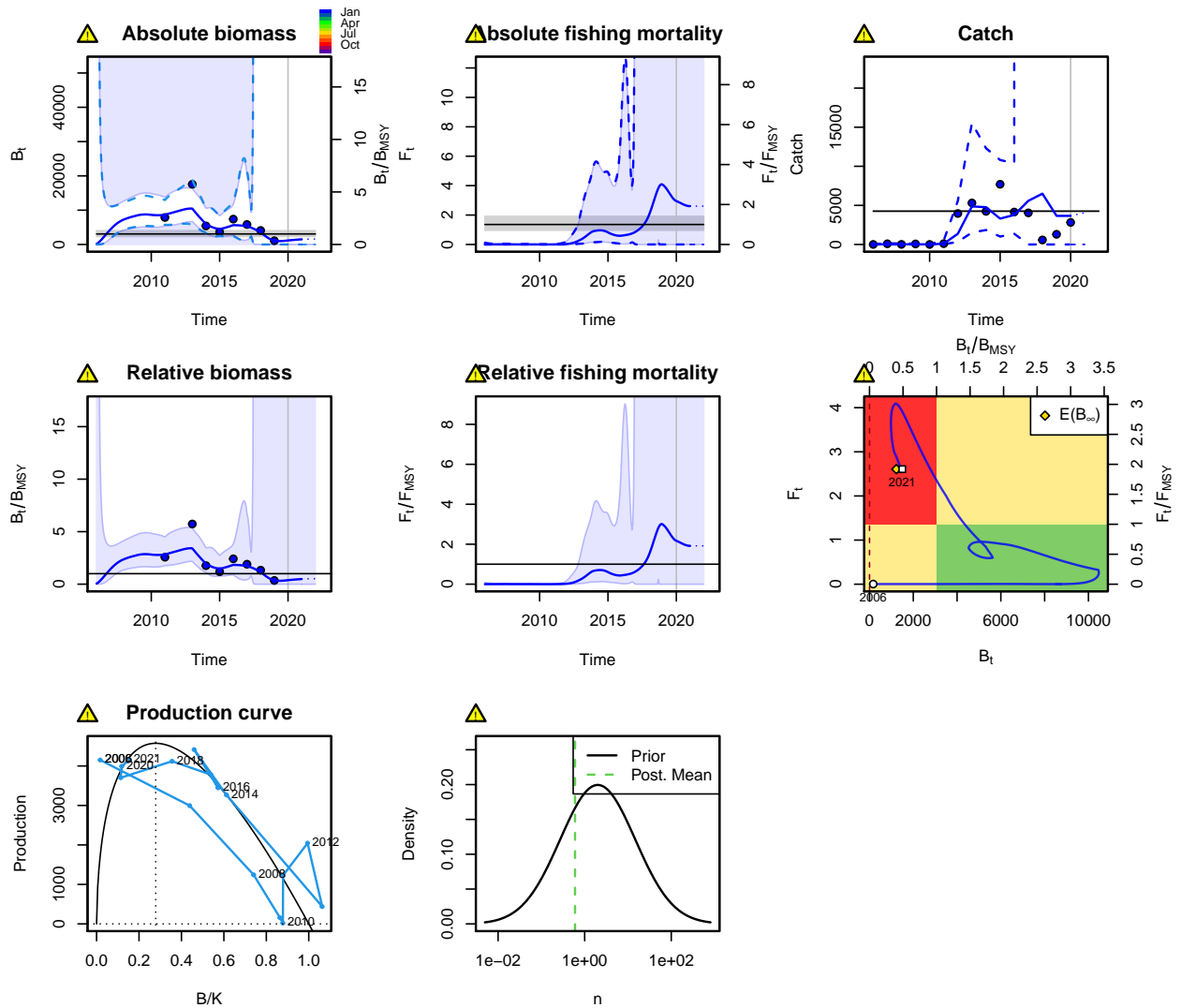



Las dos gráficas superiores provienen de `plotspict.data`, con la línea horizontal discontinua que representa una suposición de RMS. Esta suposición proviene de una regresión lineal entre el índice y la captura dividida por el índice (fila central, izquierda). Se espera que esta regresión tenga una pendiente negativa. Se puede hacer una gráfica similar mostrando captura versus $\text{catch} / \text{index}$ (fila central, derecha) para encontrar aproximadamente el esfuerzo óptimo (o proxy de esfuerzo). El proporcional el aumento en el índice en función de la captura (fila inferior, derecha) debería mostrar incrementos principalmente positivos en índice de capturas bajas y viceversa.

Ahora aplico el modelo

```
resau <- fit.spict(sauxi)
  Removing zero, negative, and NAs in I series 1

plot(resau)
```



spict_v1.2.8@d9ece0a31623f1a26d3cb4328499f16136822d14

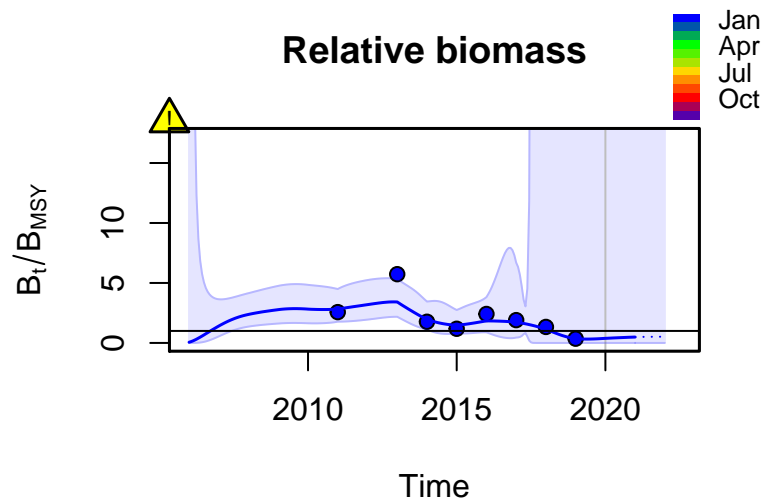
Los plot pueden ser explicados como sigue;

- Estimates (biomass, fishing mortality, catch, production) are shown using blue lines.
- 95% CIs of absolute quantities are shown using dashed blue lines.
- 95% CIs of relative biomass and fishing mortality are shown using shaded blue regions.
- Estimates of reference points (BMSY , FMSY , MSY) are shown using black lines.
- 95% CIs of reference points are shown using grey shaded regions.
- The end of the data range is shown using a vertical grey line.
- Predictions beyond the data range are shown using dotted blue lines.
- Data are shown using points colored by season. Different index series use different point characters (not shown here)

(Revisar guía Pedersen & Berg, 2017)

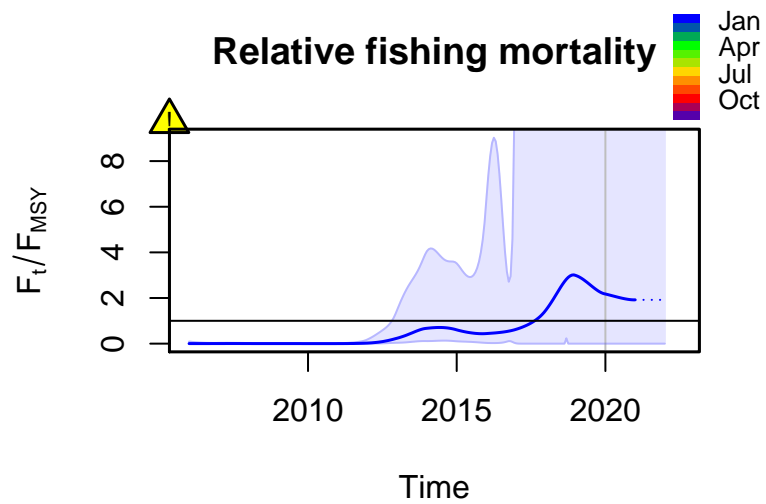
Cada uno de los plot se pueden sacar por separado

```
a <-plotspict.bbmsy(resau)
```



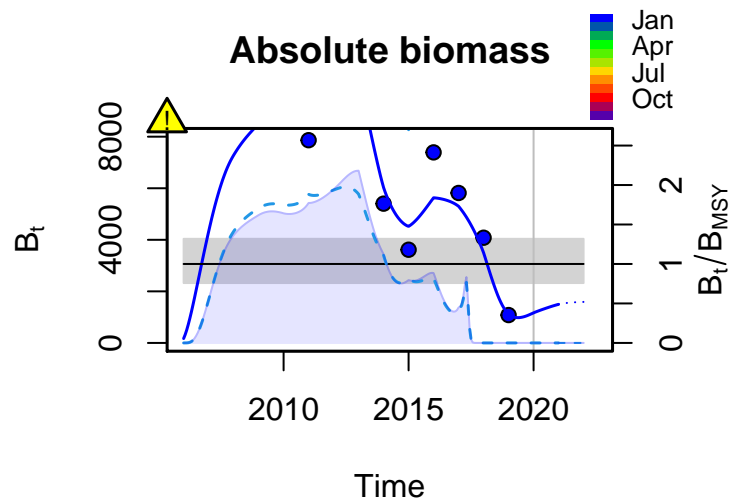
spict_v1.2.8@d9ece0a31623f1a26d3cb4328499f16136822d14

```
ab <- plotspict.ffmsy(resau)
```



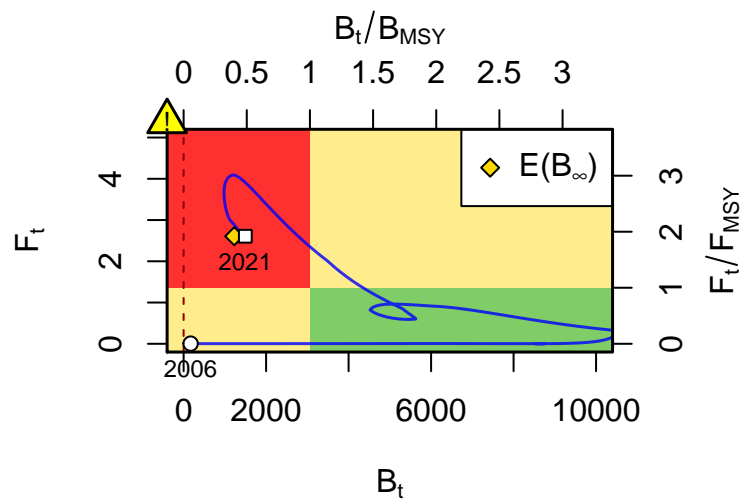
spict_v1.2.8@d9ece0a31623f1a26d3cb4328499f16136822d14

```
b<- plotspict.biomass(resau, ylim=c(0, 8000))
```



spict_v1.2.8@d9ece0a31623f1a26d3cb4328499f16136822d14

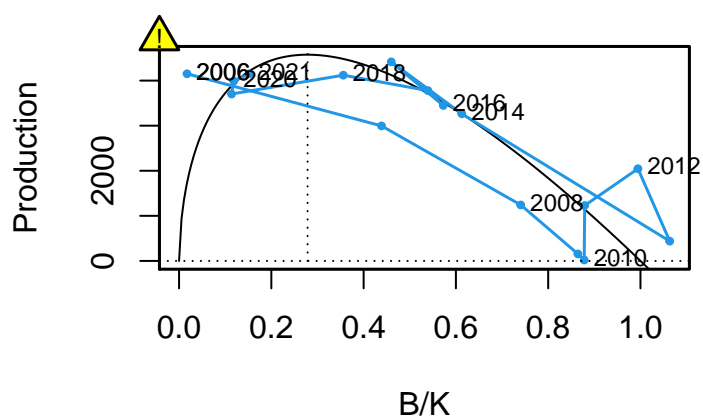
```
d <- plotspict.fb(resau, ylim=c(0, 5), xlim=c(0, 10000))
```



spict_v1.2.8@d9ece0a31623f1a26d3cb4328499f16136822d14

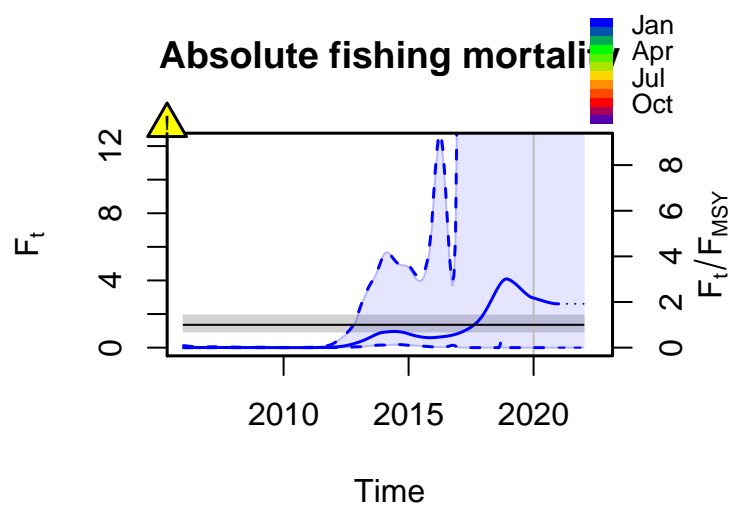
```
da <- plotspict.production(resau)
```

Production curve



spict_v1.2.8@d9ece0a31623f1a26d3cb4328499f16136822d14

```
c <- plotspict.f(resau)
```



spict_v1.2.8@d9ece0a31623f1a26d3cb4328499f16136822d14

```
#puedo hacer un layout con los graficos que quiera informar con Patchwork
a+b+c
integer(0)
```

Interpretando los resultados

```
capture.output(summary(resau))
[1] "Convergence: 1 MSG: false convergence (8)"
[2] "WARNING: Model did not obtain proper convergence! Estimates and uncertainties are most likely
[3] "Gradient at current parameter vector"
[4] "      logm      logK      logq      logn      logsdb      logsdf      logsdi      "
[5] "    94210892    30981523  148809579  105008096  -22609261 -107114733  -33262054    "
[6] "      logsdc      "
[7] "    151993020      "
[8] ""
[9] "Objective function: 43.5489368"
[10] "Euler time step (years): 1/16 or 0.0625"
[11] "Nobs C: 15, Nobs I1: 8"
[12] ""
[13] "Priors"
[14] "      logn ~ dnorm[log(2), 2^2]"
[15] " logalpha ~ dnorm[log(1), 2^2]"
[16] " logbeta ~ dnorm[log(1), 2^2]"
[17] ""
[18] "Model parameter estimates w 95% CI "
[19] "      estimate      cilow      ciupp      log.est      "
[20] " alpha      1.1029166      0.4193654 2.900633e+00  0.0979581      "
[21] " beta       0.6808931      NaN      NaN      -0.3843500      "
[22] " r          1.0004326      NaN      NaN      0.0004325      "
[23] " rc         3.3415450      3.1070211 3.593771e+00  1.2064333      "
[24] " rold       2.4935040      NaN      NaN      0.9136890      "
[25] " m          4574.5278512 4099.5514093 5.104535e+03  8.4282588      "
[26] " K          9830.2753082 9438.2294775 1.023861e+04  9.1932222      "
[27] " q          6.0781924      5.7750617 6.397234e+00  1.8047073      "
[28] " n          0.5987844      NaN      NaN      -0.5128536      "
[29] " sdb        0.3547421      0.2374278 5.300221e-01  -1.0363642      "
[30] " sdf        1.7423598      NaN      NaN      0.5552404      "
[31] " sdi        0.3912510      0.2223583 6.884263e-01  -0.9384060      "
[32] " sdc        1.1863607      NaN      NaN      0.1708904      "
[33] " "
[34] "Deterministic reference points (Drp)"
[35] "      estimate      cilow      ciupp      log.est      "
[36] " Bmsyd 2737.971696      NaN      NaN      7.9149727      "
[37] " Fmsyd  1.670772      1.553511  1.796886  0.5132861      "
[38] " MSYd  4574.527851 4099.551409 5104.535344  8.4282588      "
[39] "Stochastic reference points (Srp)"
[40] "      estimate      cilow      ciupp      log.est      rel.diff.Drp      "
[41] " Bmsys 3062.470056 2322.7395087 4037.78504 8.0269771  0.1059597      "
[42] " Fmsys  1.358318  0.9644282  1.91308 0.3062474  -0.2300302      "
[43] " MSYs  4261.200088 4200.7129934 4322.55815 8.3573061  -0.0735304      "
[44] ""
[45] "States w 95% CI (inp$msytype: s)"
[46] "      estimate      cilow      ciupp      log.est      "
[47] " B_2020.00      1166.5337877      0 1.505254e+306  7.0617921      "
[48] " F_2020.00      2.9534788      0 1.505254e+306  1.0829837      "
[49] " B_2020.00/Bmsy  0.3809127      0 1.505254e+306 -0.9651850      "
[50] " F_2020.00/Fmsy  2.1743642      0 1.505254e+306  0.7767363      "
```

```

[51] ""
[52] "Predictions w 95% CI (inp$msytype: s)"
[53] "          prediction cilow          ciupp    log.est  "
[54] " B_2021.00      1496.0631531      0 1.505254e+306  7.3105924  "
[55] " F_2021.00      2.6067907      0 1.505254e+306  0.9581198  "
[56] " B_2021.00/Bmsy  0.4885152      0 1.505254e+306 -0.7163847  "
[57] " F_2021.00/Fmsy  1.9191309      0 1.505254e+306  0.6518724  "
[58] " Catch_2021.00  4056.4986533      0 1.505254e+306  8.3080755  "
[59] " E(B_inf)      1226.7173856      NA          NA  7.1120971  "

```

Línea 1: Convergencia del ajuste del modelo, que tiene el código 0 si el ajuste fue exitoso. Si este no es el caso no se obtuvo convergencia y los resultados informados no deben utilizarse. En caso de no convergencia, los resultados se seguirán informando para ayudar al diagnóstico del problema.

Diagnostico y residuos

Antes de proceder con los resultados para una evaluación, es muy importante que los residuos del modelo se verifiquen y se identifiquen posibles deficiencias del modelo.

Los residuos se pueden calcular usando `calc.osa.resid ()`.

OSA significa un paso adelante, que son los residuos adecuados para los modelos de espacio de estado. Más información sobre los residuos de OSA está contenido en Pedersen y Berg (2017).

```

resiaus <- calc.osa.resid(resau)
plotspict.diagnostic(resiaus)

```

Extrear parametros estimados

Lo primero es ver los estimados en una lista

```
list.quantities(resau)
[1] "Bmsy"          "Bmsy2"          "BmsyB0"
[4] "Bmsyd"         "Bmsys"          "Cp"
[7] "Emsy"          "Emsy2"          "Fmsy"
[10] "Fmsyd"         "Fmsys"          "gamma"
[13] "isdb2"         "isdc2"          "isde2"
[16] "isdf2"         "isdi2"          "K"
[19] "logalpha"      "logB"           "logBBmsy"
[22] "logbeta"       "logbkfrac"      "logBl"
[25] "logBlBmsy"     "logBlK"         "logBmsy"
[28] "logBmsyd"      "logBmsyPluslogFmsy" "logBmsys"
[31] "logBp"         "logBpBmsy"      "logBpK"
[34] "logCp"         "logCpred"       "logEmsy"
[37] "logEmsy2"      "logEp"          "logF"
[40] "logFFmsy"      "logFFmsynotS"   "logFl"
[43] "logFlFmsy"     "logFmsy"        "logFmsyd"
[46] "logFmsys"      "logFnotS"       "logFp"
[49] "logFpFmsy"     "logFs"          "logIp"
[52] "logIpred"      "logK"           "logm"
[55] "logMSY"        "logMSYd"        "logMSYs"
[58] "logn"          "logq"           "logq2"
[61] "logr"          "logrc"          "logrold"
[64] "logsdb"        "logsdc"         "logsdf"
[67] "logsdi"        "m"              "MSY"
[70] "MSYd"          "MSYs"           "p"
[73] "q"             "r"              "rc"
[76] "rold"          "sdb"            "sdc"
[79] "sde"           "sdf"            "sdi"
[82] "seasonsplinefine"
```

Ahora los saco por separado

```
get.par('MSY', resau)
      ll      est      ul      sd      cv
MSY 4200.28 4261.2 4322.121 31.08189 0.007294163
```

```
get.par('logBmsy', resau)
      ll      est      ul      sd      cv
logBmsy 7.750503 8.026977 8.303452 0.1410584 0.01757304
```

Y a su vez puedo pedir el dato absoluto con exp=T

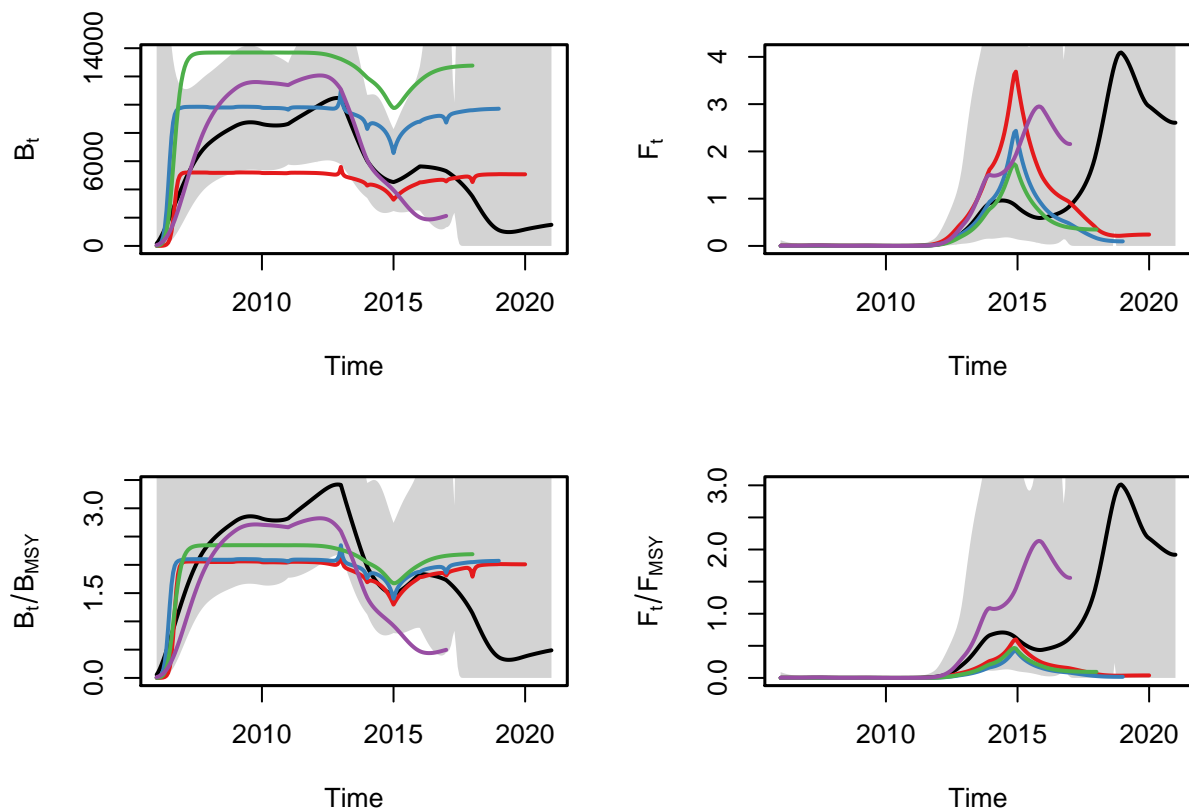
```
get.par('logBmsy', resau, exp=TRUE)
      ll      est      ul      sd      cv
logBmsy 2322.74 3062.47 4037.785 0.1410584 0.141763
```


Analisis retrospectivo

Evaluar robustez de la estimación. En este caso será para 10 años.

De forma predeterminada, retro crea 5 escenarios con series de tiempo de captura e índice que se acortan del 1 al 5 al final observaciones.

```
retrosau <- retro(resau, nretroyear = 4)
plotspict.retro(retrosau)
```



spict_v1.2.8@d9ece0a31623f1a26d3cb4328499f16136822d14

Escenarios de Manejo

El paquete tiene una función que ejecuta varios escenarios de gestión predefinidos, que se pueden presentar en un tabla de predicción. Para realizar los cálculos necesarios para producir la ejecución de la tabla de pronóstico:

```
manesau <- manage(resau)
```

Puedo ver los resultados en la tabla con mansummary

```
mansummary(manesau)
```

```
Observed interval, index: 2011.00 - 2019.00
```

```
Observed interval, catch: 2006.00 - 2021.00
```

```
Fishing mortality (F) prediction: 2022.00
```

```
Biomass (B) prediction: 2022.00
```

```
Catch (C) prediction interval: 2021.00 - 2022.00
```

Predictions

| | C | B | F | B/Bmsy | F/Fmsy | perc.dB | perc.dF |
|-----------------------|--------|--------|-------|--------|--------|---------|---------|
| 1. Keep current catch | 3028.0 | 582.6 | 5.173 | 0.213 | 3.808 | -61.1 | 98.4 |
| 2. Keep current F | 4056.5 | 1590.4 | 2.607 | 0.581 | 1.919 | 6.3 | 0.0 |
| 3. Fish at Fmsy | 3060.3 | 2807.1 | 1.362 | 1.025 | 1.003 | 87.6 | -47.7 |
| 4. No fishing | 9.4 | 5671.7 | 0.003 | 2.071 | 0.002 | 279.1 | -99.9 |
| 5. Reduce F 25% | 3665.1 | 2108.9 | 1.967 | 0.770 | 1.448 | 41.0 | -24.5 |
| 6. Increase F 25% | 4242.4 | 1189.6 | 3.306 | 0.434 | 2.434 | -20.5 | 26.8 |
| 7. MSY advice rule | 3019.9 | 2850.2 | 1.331 | 1.041 | 0.980 | 90.5 | -48.9 |

95% CIs of absolute predictions

| | C.lo | C.hi | B.lo | B.hi | F.lo |
|-----------------------|-------|---------------|------|---------------|-------|
| 1. Keep current catch | 419.6 | 2.184980e+04 | 0.5 | 6.624774e+05 | 0.083 |
| 2. Keep current F | 0.0 | 1.505254e+306 | 0.0 | 1.505254e+306 | 0.000 |
| 3. Fish at Fmsy | 0.0 | 5.892460e+43 | 0.0 | 1.505254e+306 | 0.000 |
| 4. No fishing | 0.0 | 2.941391e+26 | 0.0 | 3.103908e+14 | 0.000 |
| 5. Reduce F 25% | 0.0 | 2.166891e+16 | 0.0 | 1.167121e+218 | 0.000 |
| 6. Increase F 25% | 0.0 | 2.268799e+24 | 0.0 | 3.013542e+88 | 0.000 |
| 7. MSY advice rule | 0.0 | 2.362555e+46 | 0.0 | 1.505254e+306 | 0.000 |

F.hi

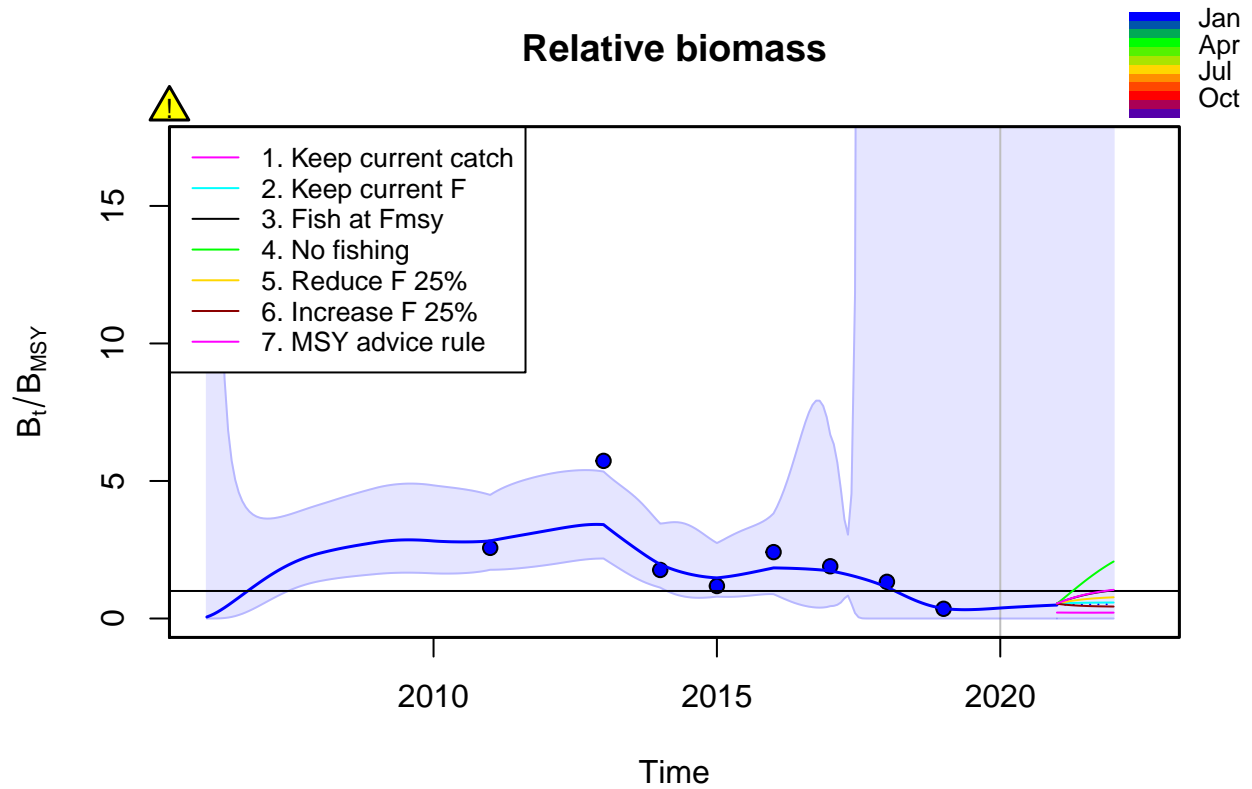
| | |
|-----------------------|---------------|
| 1. Keep current catch | 3.205270e+02 |
| 2. Keep current F | 1.505254e+306 |
| 3. Fish at Fmsy | 1.505254e+306 |
| 4. No fishing | 1.364730e+45 |
| 5. Reduce F 25% | 8.098227e+205 |
| 6. Increase F 25% | 8.009807e+60 |
| 7. MSY advice rule | 1.505254e+306 |

95% CIs of relative predictions

| | B/Bmsy.lo | B/Bmsy.hi | F/Fmsy.lo | F/Fmsy.hi |
|-----------------------|-----------|---------------|-----------|---------------|
| 1. Keep current catch | 0 | 4.072440e+02 | NaN | NaN |
| 2. Keep current F | 0 | 1.505254e+306 | 0 | 1.505254e+306 |
| 3. Fish at Fmsy | 0 | 1.505254e+306 | 0 | 1.505254e+306 |
| 4. No fishing | 0 | 1.133903e+11 | 0 | 1.008145e+45 |
| 5. Reduce F 25% | 0 | 4.262779e+214 | 0 | 5.958412e+205 |
| 6. Increase F 25% | 0 | 1.100682e+85 | 0 | 5.886879e+60 |
| 7. MSY advice rule | 0 | 1.505254e+306 | 0 | 1.505254e+306 |

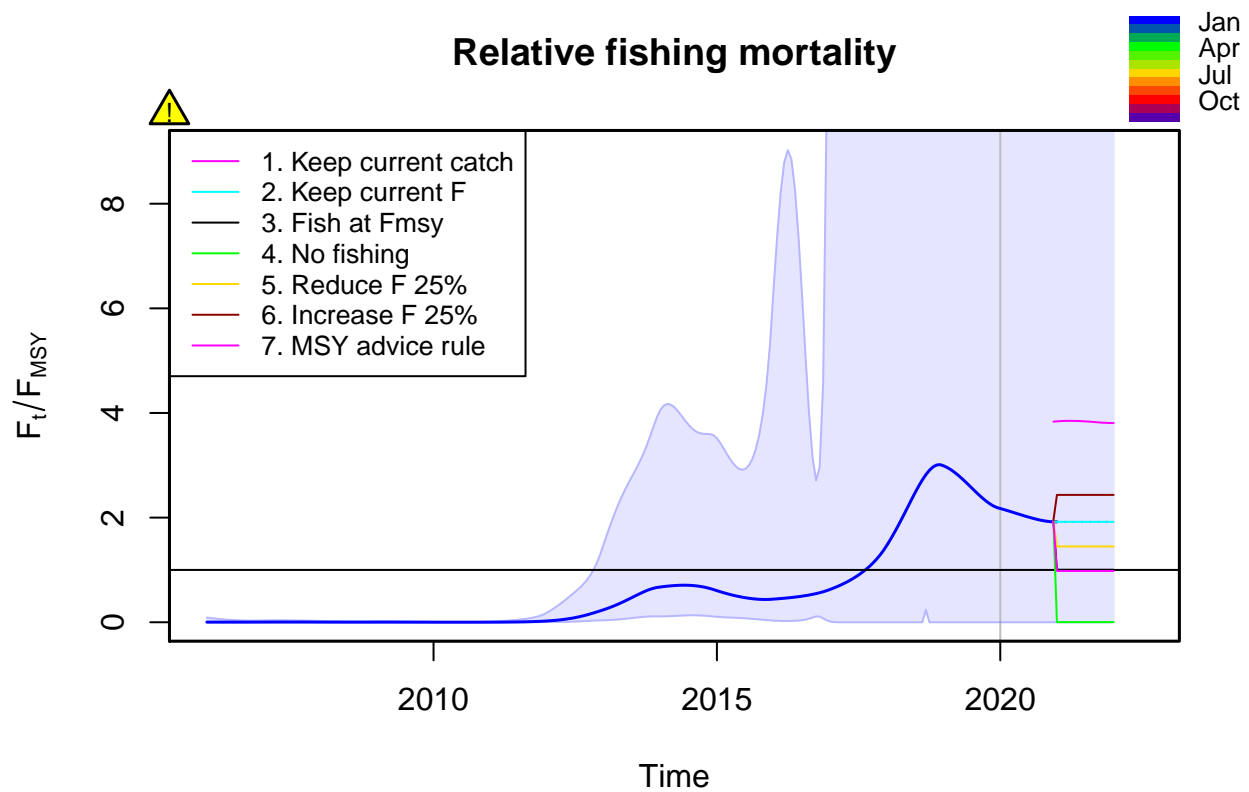
Ploteo de escenarios

```
plotspict.bbmsy(manesau)
```



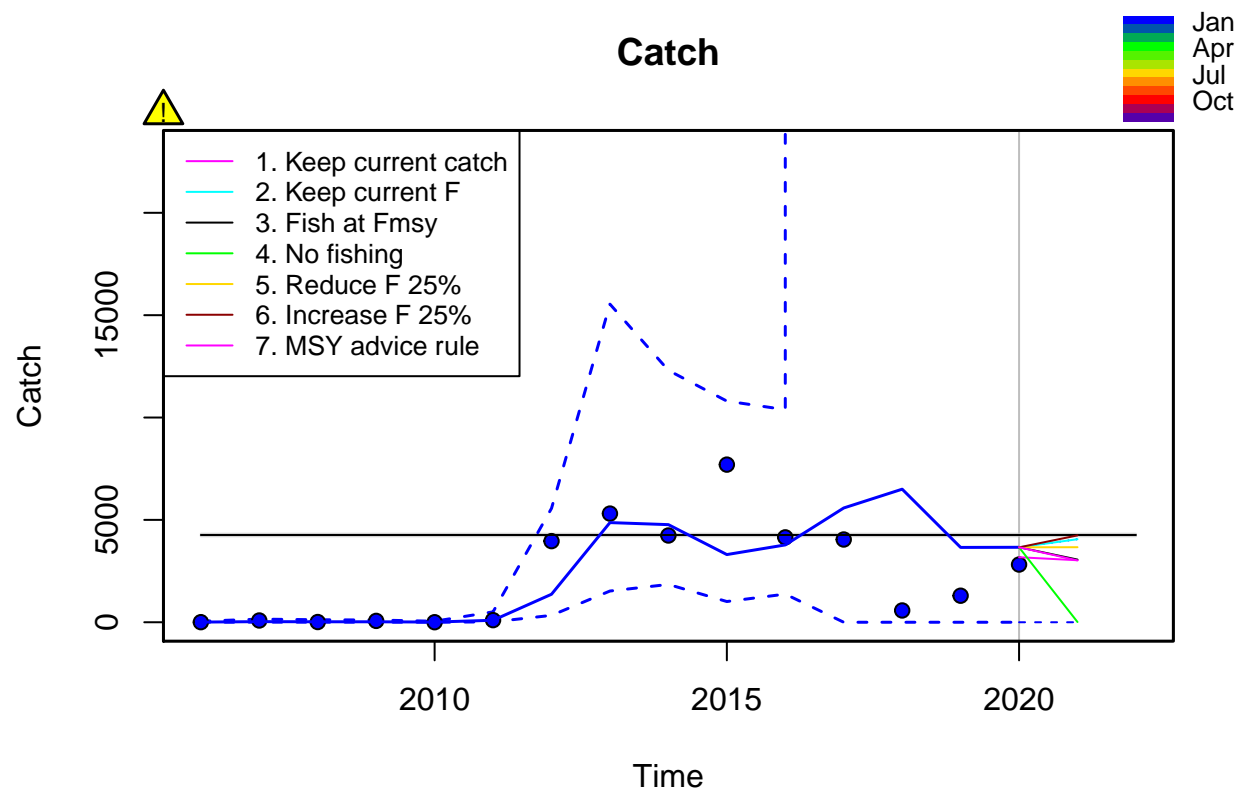
spict_v1.2.8@d9ece0a31623f1a26d3cb4328499f16136822d14

```
plotspict.ffmsy(manesau)
```



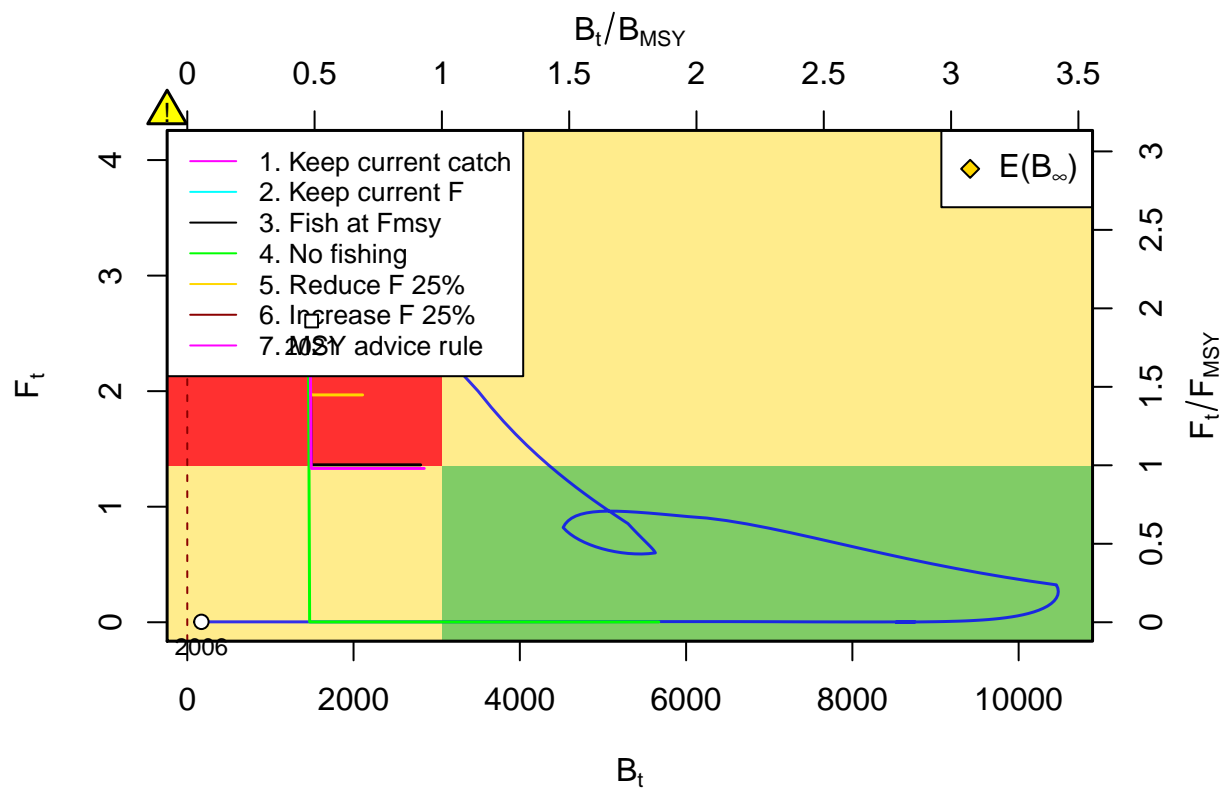
spict_v1.2.8@d9ece0a31623f1a26d3cb4328499f16136822d14

`plotspict.catch(manesau)`



spict_v1.2.8@d9ece0a31623f1a26d3cb4328499f16136822d14

plotspict.fb(manesau)



spict_v1.2.8@d9ece0a31623f1a26d3cb4328499f16136822d14

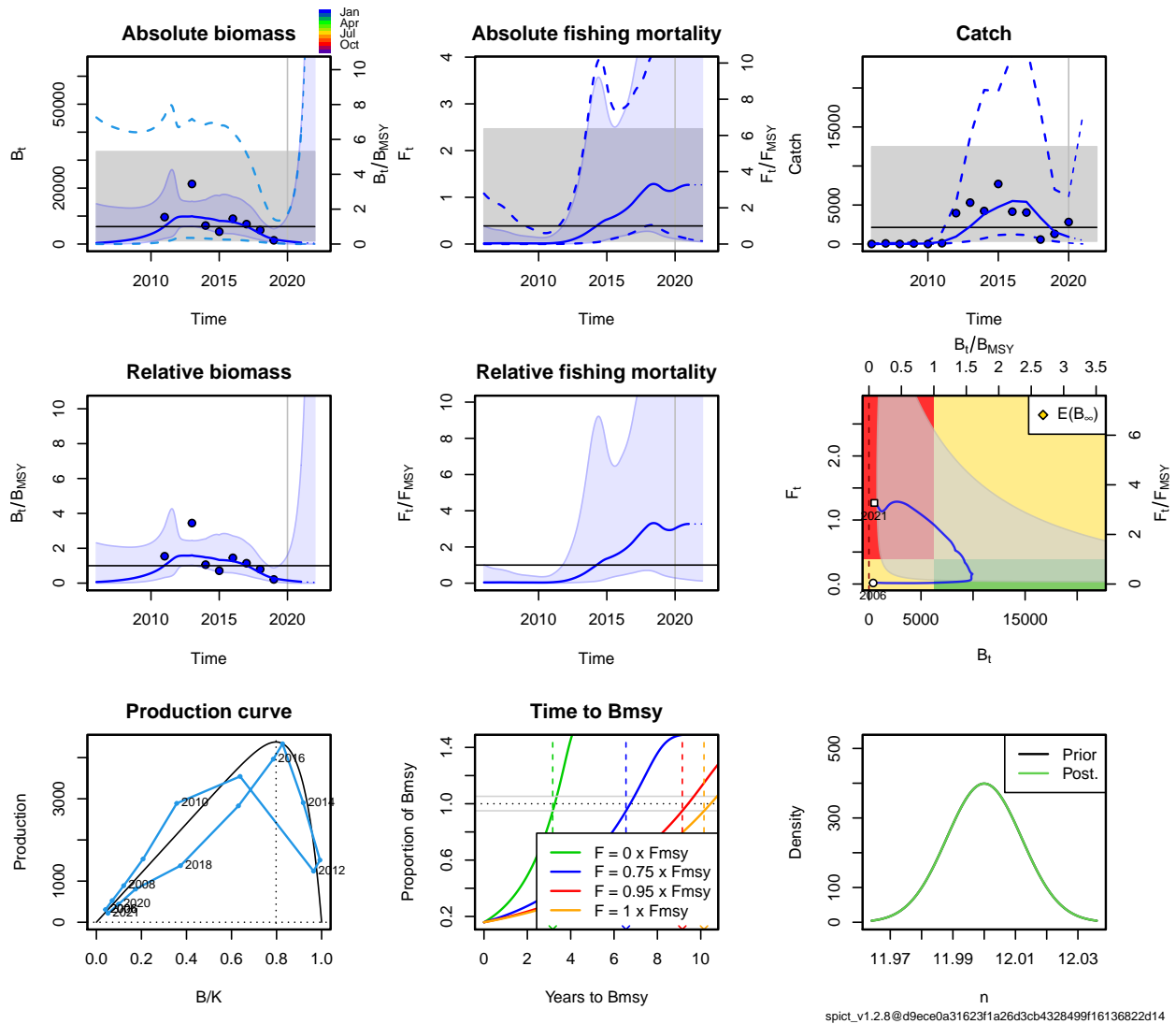
Fijando parametros

Los parámetros del modelo se pueden fijar usando fases como se describió anteriormente. Esta técnica, sin embargo, solo se utiliza para fijar parámetros del modelo y, por lo tanto, no cantidades derivadas como $\log\alpha$, $\log r$ (que se deriva de $\log K$, $\log m$ y $\log n$). La fijación de un parámetro puede considerarse como una imposición de un carácter altamente informativo antes del parámetro.

Identifico los parametros que se pueden setear. Tambien es posible fijar fases de estimación.

```
list.possible.priors()
[1] "logn"      "logalpha"  "logbeta"   "logr"      "logK"      "logm"
[7] "logq"      "logqf"     "logbkfrac" "logB"      "logF"      "logBBmsy"
[13] "logFFmsy"  "logsdb"    "logsdf"    "logsdi"    "logsde"    "logsdc"
[19] "logsdm"    "logpsi"    "mu"        "BmsyB0"    "logngamma"
```

```
inp <- sauxi
inp$priors$logn <- c(log(12), 1e-3)
inp$priors$logalpha <- c(log(2), 1e-3)
inp$priors$logbeta <- c(log(2), 1e-3)
resaufix <- fit.spict(inp)
  Removing zero, negative, and NAs in I series 1
plot(resaufix)
```



Brechas analíticas

- Como se estima F , m ?
- Como se condicionan las priors?
- predicciones. Identificar condiciones sobre las cuales se generan, por ejemplo, capturas, F , B , etc.
- Identificar claramente los escenarios por default
- Simular datos para testear
- Usar mas de un indice, incluso se puede usar el esfuerzo como indice
- Datos estacionales
-