

**Manual de Uso de Software:
Evaluación del stock de pulpos en
Yucatán usando las bases de datos
de Avisos de Arribos, R, CatDyn y
ADMB**

Version 1.0.0

Rubén H. Roa-Ureta
Consultor Independiente
11 de abril de 2023

Índice

1. Introducción	2
2. Requerimientos y herramientas de software	3
3. Etapa 1: Procesamiento de datos de la base de datos Avisos de Arribo y datos biológicos	7
3.1. Avisos de Arribos	7
3.2. Datos biológicos	12
4. Etapa 2: modelos de depleción intra-anales	13
4.1. Temporada 2000 de <i>O. maya</i>	14
4.1.1. Objeto de datos para CatDyn	15
4.1.2. Semanas candidatas a entradas de abundancia: estadístico 'catch spike'	17
4.1.3. Semanas candidatas a entradas de abundancia: modelo de depleción pura	17
4.1.4. Ajuste de modelos de depleción con entradas de abundancia	28
4.1.5. Selección de la mejor variante	32
4.2. Temporada 2017 de <i>O. americanus</i>	37
4.2.1. Objeto de datos para CatDyn	37
4.2.2. Semanas candidatas a entradas de abundancia: estadístico 'catch spike'	38
4.2.3. Semanas candidatas a entradas de abundancia: modelo de depleción pura	38
4.2.4. Ajuste de modelos de depleción con entradas de abundancia	40
4.2.5. Seleccción de la mejor variante	44
4.3. Reuniendo todas las evaluaciones anuales	47
5. Etapa 2: Modelos de producción excedente	59
5.1. Optimización estadística en ADMB	61
5.2. Estimación de tasa de cosecha sostenible máxima en R	71
5.2.1. Selección de la mejor variante	71
5.2.2. Parámetros derivados y gráfica de la dinámica de poblaciones en biomasa	73

1. Introducción

Este documento es una extensión del curso Capacitación en Modelos de Depleción Generalizados y de Producción Excedente, patrocinado por COBI Comunidad y Biodiversidad, que se realizó online en cinco sesiones entre el 27 y el 31 de Marzo de 2023. El documento describe paso a paso la actualización del proceso de evaluación de stocks de pulpos (*Octopus maya* y *Octopus americanus*) en México a partir de los resultados previos ya obtenidos desde las temporadas de pesca 2000 hasta 2021. Por lo tanto este documento puede empezar a ser aplicado con datos nuevos de la temporada 2022.

Los estudiantes y aplicadores de la capacitación cuentan con 22 (2000 a 2022) bases de datos anuales de la base de datos Avisos de Arribos en formato csv (comma separated values). Esta base contiene información de capturas y esfuerzo de pesca por viaje de pesca para más de 800 recursos pesqueros del Pacífico y el Atlántico mexicanos. Por lo tanto el curso y este documento también pueden ser utilizados para evaluar otros recursos pesqueros mexicanos contenidos en la base de datos Avisos de Arribo, utilizando la metodología que combina modelos de depleción generalizados del tipo intra-anual, con modelos de producción excedente.

Los estudios científicos que respaldan los métodos empleados aquí son un artículo publicado en el volumen 78, número 6, año 2021 del [ICES Journal of Marine Science](#) y la sección 2.3 del reporte de la serie de Reportes Científicos de la organización europea de ciencias del mar ICES [ICES Scientific Reports 2:99](#), publicado en 2020. Se supone que el usuario conoce estos documentos y por lo tanto ya está familiarizado/a con las teorías y técnicas científicas que se aplican. Dado lo anterior, este documento describe paso a paso el uso del software desarrollado para completar la evaluación de stocks con la metodología descrita en el artículo y el reporte científico.

Esencialmente la metodología consiste en dos etapas: primero, para cada temporada de pesca por separado, ajustar modelos de depleción generalizados de tipo intra-anales a los datos semanales de esfuerzos y capturas y elegir el mejor modelo para los datos, y segundo, ajustar modelos de producción excedente de tipo Pella-Tomlinson a las biomásas anuales predichas por el mejor modelo de depleción de cada temporada de pesca. Los mejores modelos de depleción para las temporadas 2000 a 2020 ya están disponibles en trabajos previos de este autor, así que este manual sigue la actualización de la evaluación de stock con los nuevos datos de la temporada 2021. Esto sirve como ejemplo para el uso del software por parte de los usuarios para evaluar el stock en la temporada 2022.

2. Requerimientos y herramientas de software

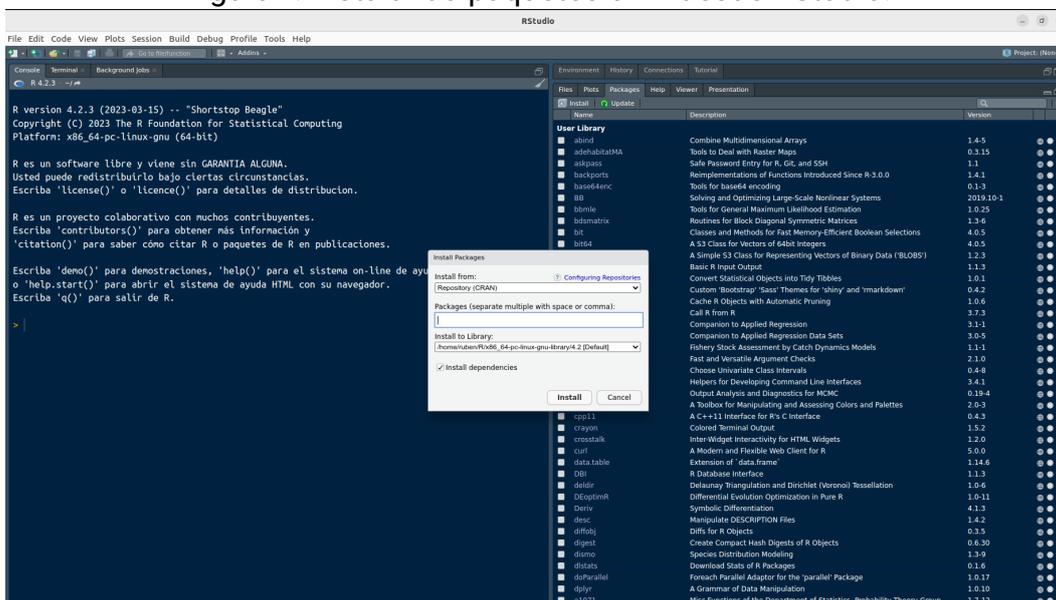
Usuarios de este manual deben tener un ordenador moderno, con un sistema operativo de 64 bits que puede ser Linux, MS Windows, o Apple, con al menos 16 GB de RAM. Si el sistema es MS Windows, el usuario debe cambiar la conducta por defecto de MS Windows 10 de ocultar las extensiones de los ficheros. El usuario debe abrir el Explorador de Windows, click en la pestaña 'Vista', click en 'Opciones', click en 'Cambiar opciones de carpeta y búsqueda'. En el menú emergente, click en la pestaña 'Ver', bajar por la lista de opciones, y quitar el tick en la opción 'Ocultar las extensiones para tipos de archivos conocidos'. Luego click en 'Aceptar'.

En el sistema el usuario debe tener instalados los siguientes programas: [lenguaje de programación R](#), [RStudio](#) y [AD Studio](#). Estos dos últimos son convenientes interfaces para el lenguaje R y el lenguaje ADMB, respectivamente. AD Studio viene con el lenguaje ADMB contenido en su instalación con programa instalador aunque éste sólo está disponible para MS Windows, así que usuarios de Linux y Apple deberán instalar ADStudio manualmente. En el caso de sistemas Linux-Ubuntu se requiere instalar ADMB usando el administrador de paquete [Debian](#). Este autor ha instalado ADMB exitosamente en Ubuntu 18.04 y Ubuntu 22.04 usando el administrador de paquetes Debian. Para todos los sistemas operativos, es recomendable primero instalar la última versión de R (v. 4.2.3 al momento de escribir este manual), luego instalar la versión libre y gratuita de RStudio (2023.03.0 Build 386 al momento de escribir este manual), y finalmente instalar ADStudio. Posteriormente, el usuario deberá instalar los siguientes paquetes de R desde el repositorio oficial [CRAN](#): [CatDyn](#), [Runuran](#), [lattice](#), [grid](#), [gridBase](#), [mice](#) y [Hmisc](#). Para instalar paquetes en R, una buena opción es usar el botón correspondiente de RStudio mientras existe una conexión a internet (Fig. 1). Este botón se encuentra en el panel de la mano derecha, en la pestaña 'packages'. El botón 'install' conduce a un menú en donde seleccionar el origen del paquete (por defecto es CRAN) y escribir el nombre del paquete.

Las herramientas de software específicas para este manual se detallan en la siguiente lista de ficheros, cuyo orden es el orden en que van apareciendo durante el aprendizaje:

- [Avisos de arriba 2021.csv](#): Datos de capturas y esfuerzo (todas los Estados y cerca de 900 especies) (138,9 MB).
- [AvisosDeArribos.Data.R](#): Guión de código R para extraer y configurar datos de capturas y esfuerzo (121,9 KB).
- [AvisosDeArribos.Data.RData](#): Binario que guarda los objetos creados con el guión del mismo nombre (9,3 MB).

Figura 1: Instalando paquetes en R desde RStudio.



- omaya.inapesca.bio.csv: Datos biológicos (longitudes y pesos en la captura) *O. maya* de INAPESCA (3,3 MB).
- oamer.inapesca.bio.csv: Datos biológicos (longitudes y pesos en la captura) *O. americanus* de INAPESCA (116,0 MB).
- Ivan Biologicos pulpo México_07.04.22_RR.csv: Datos biológicos (longitudes, pesos, fechas y localidades) *O. maya* y *O. americanus* de proyectos del Dr. J. Iván Velázquez-Abunader (31,4 KB).
- Yucoct.biodata.R: Guión de código R para procesar datos biológicos de ambas especies (6,3 KB).
- Yucoct.biodata.RData: Binario que guarda los objetos creados con el guión del mismo nombre (991,3 KB).
- Yucoct.Maya.CatDyn.IAGD.2000.R: Guión de código R que sirve de ejemplo del ajuste de MDIA (*O. maya*, 2000) (19,8 KB).
- Yucoct.Maya.CatDyn.IAGD.2000.RData: Binario que guarda los objetos creados con el guión del mismo nombre (75,4 KB).
- Yucoct.Amer.CatDyn.IAGD.2017.R: Guión de código R que sirve de ejemplo del ajuste de MDIA (*O. americanus*, 2017) (33,7 KB).

- Yucoct.Amer.CatDyn.IAGD.2017.RData: Binario que guarda los objetos creados con el guión del mismo nombre (143,6 KB).
- Yucoct.Maya.CatDyn.IAGD.R: Guión de código R para reunir todas las evaluaciones intra-anales (22, 2000 a 2021) usando *O. maya* como ejemplo (38,6 KB).
- Yucoct.Maya.CatDyn.IAGD.RData: Binario que guarda los objetos creados con el guión del mismo nombre (3,4 MB).
- marlikptoayuc1reg3p.dat: Datos para ADMB con hipótesis de estabilidad y $B_0 = K$ (0,935 KB).
- marlikptoayuc1reg3p.tpl: Guión ADMB con hipótesis de estabilidad y $B_0 = K$ (1,7 KB).
- marlikptoayuc1reg4p.dat: Datos para ADMB con hipótesis de estabilidad y $B_0 \neq K$ (0,935 KB).
- marlikptoayuc1reg4p.tpl: Guión ADMB con hipótesis de estabilidad y $B_0 \neq K$ (1,8 KB).
- marlikptoayuc2regr4p.dat: Datos para ADMB con hipótesis de cambio en r y $B_0 \neq K$ (0,963 KB).
- marlikptoayuc2regr4p.tpl: Guión ADMB con hipótesis de cambio en r y $B_0 \neq K$ (2,2 KB).
- marlikptoayuc2regrK5p.dat: Datos para ADMB con hipótesis de cambio en r y K y $B_0 \neq K$ (0,963 KB).
- marlikptoayuc2regrK5p.tpl: Guión ADMB con hipótesis de cambio en r y K y $B_0 \neq K$ (2,3 KB).
- marlikptoayuc2regrp5p.dat: Datos para ADMB con hipótesis de cambio en r y p y $B_0 \neq K$ (0,963 KB).
- marlikptoayuc2regrp5p.tpl: Guión ADMB con hipótesis de cambio en r y p y $B_0 \neq K$ (2,3 KB).
- marlikptoayuc2regrpK6p.dat: Datos para ADMB con hipótesis de cambio en r y p y K y $B_0 \neq K$ (0,963 KB).
- marlikptoayuc2regrpK6p.tpl: Guión ADMB con hipótesis de cambio en r y p y K y $B_0 \neq K$ (2,5 KB).

- Yucoct.Summary.R: Guión de código R para reunir todas las evaluaciones intra-anales (44, 2000 a 2021, 2 especies) y traer los resultados de ADMB a R y calcular los estimados finales de tasa de cosecha sostenible (97,8 KB).
- Yucoct.Summary.RData: Binario que guarda los objetos creados con el guión del mismo nombre (15,2 MB).

Estos 28 ficheros han sido entregados junto con este manual y deben estar ubicados todos en el mismo directorio de trabajo dentro del disco duro del ordenador.

Cada vez que se abra un guión de extensión 'R' o 'r' aparecerán tres paneles en el ambiente de trabajo de RStudio, dos a mano izquierda y uno a mano derecha. El panel izquierdo superior es el **guión** con instrucciones en lenguaje R, el panel izquierdo inferior es la **consola** de R, donde se ejecutan las instrucciones del guión, y el panel de la derecha es un panel **multi-propósito** donde caben gráficas, páginas de ayuda, listas de paquetes, listas de ficheros en el directorio de trabajo, y otros elementos.

Los tres paneles de RStudio pueden comunicarse entre ellos de la siguiente manera

- Guión a consola: ponga el cursor en una línea o marque un grupo de líneas del guión y presione el botón 'Run' arriba a la derecha del panel del guión. Las instrucciones se ejecutarán en la consola, abajo.
- Guión o consola a panel multi-propósito: algunas instrucciones tales como '?', 'plot' ejecutadas en la consola (directamente desde la consola o como instrucciones desde el guión) y muchas otras producen un resultado en el panel multi-propósito.

Es altamente recomendable que el ordenador en que se lleve a cabo este trabajo tenga dos pantallas y que la segunda pantalla no sea una copia, sino una extensión de la pantalla del ordenador. Si ese es el caso y la segunda pantalla está a su derecha, al abrir RStudio y maximizar su interface, sólo se maximizará dentro de una pantalla. Es entonces muy conveniente desplazar el borde derecho de la interface hacia la segunda pantalla y así permitir que el guión y la consola ocupen la pantalla del ordenador y que el panel multi-propósito ocupe la pantalla adicional. Así las gráficas producidas durante el trabajo se podrán revisar en detalle y las líneas de código más largas podrán ser visualizadas dentro de la misma línea del guión.

3. Etapa 1: Procesamiento de datos de la base de datos Avisos de Arribo y datos biológicos

El objetivo final del procesamiento de datos de la base de datos Avisos de Arribo y de bases de datos biológicas es, para cada año, tener una tabla con cuatro columnas: semana, capturas totales en kg por semana, esfuerzo de pesca total por semana, y peso medio en la captura por semana. Describimos primero el procesamiento de las bases de datos Avisos de Arribos y luego explicamos el procesamiento de los datos biológicos.

3.1. Avisos de Arribos

En su forma original, la base de datos Avisos de Arribos es una planilla de cálculo que comienza en 2000 y se extiende hasta el presente, donde los datos de cada año ocupan cada uno una página de la planilla de cálculo.

Los datos de 2000 a 2021 ya han sido guardados como csv en 22 ficheros separados disponibles para el usuario. Para procesar los datos de nuevos años de datos (desde 2022 en adelante), el usuario sólo debe ocuparse de guardar como csv cada nuevo año de datos. El fichero csv de nuevos datos debe tener las siguientes columnas con **exactamente** los mismos nombres de columna en **exactamente** el orden indicado aquí:

- RNP ACTIVO
- NOMBRE ACTIVO
- CLAVE SITIO DESEMBARQUE
- NOMBRE SITIO DESEMBARQUE
- RNPA UNIDAD ECONOMICA
- UNIDAD ECONOMICA
- **NOMBRE ESTADO**
- CLAVE OFICINA
- NOMBRE OFICINA
- TIPO AVISO
- FOLIO AVISO
- **FECHA AVISO**

- ORIGEN
- CLAVE LUGARCAPTURA
- NOMBRE LUGARCAPTURA
- NUMERO EMBARCACIONES
- MES CORTE
- AÑO CORTE
- PERIODO INICIO
- PERIODO FIN
- DURACION
- DIAS EFECTIVOS
- TIPO ZONA
- PRODUCCION ACUACULTURAL
- NUMERO PERMISO
- FECHA EXPEDICION
- FECHA VIGENCIA
- NOMBRE PRINCIPAL
- CLAVE ESPECIE
- NOMBRE ESPECIE
- PESO DESEMBARCADO
- PESO VIVO
- PRECIO
- VALOR
- LITORAL

Los nombres de columnas marcados con color rojo deben tener datos mientras los otros nombres de columnas pueden estar completamente vacías.

Si en un año determinado la base de datos Avisos de Arribos carece de alguna de las columnas indicadas el usuario deberá agregarla en su lugar correspondiente sin importar que no tenga datos (i.e. se agrega una columna vacía con el nombre correcto en la posición correcta).

Si en un año determinado la base de datos Avisos de Arribos tiene un nombre de columna diferente al que se indica arriba el usuario deberá renombrarla para que los nombres de columnas queden idénticos al listado indicado arriba.

Puede surgir otra complicación con una base de datos de un nuevo año. Por ejemplo, los datos mismos pueden incluir nombres con tildes o caracteres que fueron incluidos por error. Además de esos posibles problemas, otro problema probable es que debido al uso de MS Excel algunas filas (entre miles) queden mal estructuradas. Para probar este problema, el usuario deberá usar una planilla de cálculo (se recomienda usar Calc de LibreOffice) para abrir la base de datos en formato csv:

- al posicionarse en la última columna a la derecha (LITORAL, columna AI) o en la columna previa (VALOR, AH) y dirigirse a la última fila con ↓ Fin, si la celda en la que se detiene el traslado no es la celda de la última fila entonces el usuario debe revisar si la fila donde se ha detenido ha quedado corta (i.e. le faltan columnas);
- al posicionarse en la columna (AJ) a la derecha de la última columna (LITORAL, AI) y bajar al final con ↓ Fin, si el traslado se detiene en una fila anterior a la fila final entonces la fila ha quedado larga (i.e. tiene una o más celdas vacías aumentando el número de columnas).

Un caso como este se muestra en la Fig. 2, que corresponde a 2019, y que tiene la fila 218060 excediéndose en una columna hacia la derecha. La causa de este error se observa en la parte inferior de la figura: la columna FECHA EXPEDICION está vacía y su contenido está en la siguiente columna, causando que todas sus columnas a la derecha estén desplazadas.

En cualquiera de estos casos el usuario deberá corregir la base de datos manualmente en la planilla de cálculo antes de usar los guiones de R que se proporcionan aquí. La base de datos debe ser una matriz cuadrada perfecta, aunque puede tener celdas vacías.

Una vez que la nueva base de datos anual ha sido revisada y formateada para quedar exactamente como se indica arriba, el usuario puede usar el guión [AvisosDeArribos.Data.R](#), que comienza como se indica en la Fig. 3.

Figura 2: Error típico en base de datos Avisos de Arribo.

The figure consists of two screenshots of a spreadsheet application. The top screenshot shows a table with the following columns: 'PESO DESEMBARCADO', 'PESO VIVO', 'PRECIO VALOR', and 'LITORAL'. The data includes various fish species and their weights and prices. The bottom screenshot shows a table with the following columns: 'FECHA EXPEDICION', 'FECHA VIGENCIA', 'NOMBRE PRINCIPAL', 'CLAVE ESPECIE', 'NOMBRE ESPECIE', and 'PESO DESEMBARCADO'. This table lists specific fish arrivals with their dates, species names, and weights.

El usuario deberá definir su directorio de trabajo, traer librería necesarias, y definir el(los) estado(s) y la especie de interés. Para definir la especie, deberá usarse una secuencia de caracteres que se encuentre en todas las filas con datos de esa especie. Por ejemplo para las dos especies de pulpos, la secuencia 'PULPO' cubre cuatro nombres de especies que contienen datos de pulpos.

El guión [AvisosDeArribos.Data.R](#) realiza un primer procesamiento de datos que consiste en extraer las filas del Estado y la especie de interés, y transformar algunos formatos numéricos y de fechas, (Fig. 4). Este proceso debe repetirse para todos los años en el caso de usar esta metodología por primera vez o sólo para el nuevo año en el caso de una actualización de la evaluación de stocks.

El guión [AvisosDeArribos.Data.R](#) realiza un segundo procesamiento de datos que consiste en crear una nueva columna con la semana como número de semana, y luego agregar sumando los datos de capturas y esfuerzos de pesca por semana (Fig. 5). En algunos años la primera semana va a ser la semana 0 pero esto deberá corregirse sumando 1 para que las semanas comiencen en 1. Este cambio se realizará en la siguiente sección. Este proceso debe repetirse para todos los años en el caso de usar esta metodología por primera vez o sólo para el nuevo año en el caso de una actualización de la evaluación de stocks.

En el caso de los pulpos la medida de esfuerzo correcta es DIAS EFECTIVOS puesto que esta medida cuenta a los alijos por separado y de esa manera hace aditivos los esfuerzo de las flotas costeras que tienen uno o dos

Figura 3: Inicio de guión `AvisosDeArribos.Data.R` para procesar datos de la base de datos Avisos de Arribo.

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
AvisosDeArribos.Data.R
Source on Save Run
1 #####
2 #
3 # DATA EXTRACTION FROM FISHERIES INPUT/OUTPUT AVISOS DE ARRIBOS #
4 # DATABASES OF THE MEXICAN PACIFIC AND CARIBBEAN COASTLINES #
5 # (1) FOR GENERAL USE IN R AND #
6 # (2) FOR BUILDING STOCK ASSESSMENT DATA FILES #
7 # FOR GENERALIZED DEPLETION MODELS IN CatDyn PACKAGE #
8 #
9 # Author: Ruben H. Raa-Ureta #
10 # March, 2023 #
11 #
12 #####
13 #
14 wdir <- "/home/ruben/Workplace/Consultancy/COBI/2023/Training/Sesston2" #
15 setwd(wdir) #
16 options(max.print=12000,scipen=4) #
17 library(Hmisc) # For plotting with standard error bars #
18 #
19 library(nice) # Needed for imputing a few weeks (week 41 in maya 2020, #
20 # eff 5.9 times mean eff, , 3.4 times max eff, week 39 in amer #
21 # 2021, eff 297.2 times mean eff, 166.2 times max eff) with #
22 # likely wrong effort data #
23 #
24 library(Runuran) # For re-sampling with truncated distributions for mean weekly #
25 # weight in the catch. #
26 #
27 ##### 1.- ESTADO/ESPECIE SWITCH #####
28 #
29 estado <- c("YUCATAN","CAMPECHE","QUINTANA ROO") #
30 stock <- "PULPO" #
31 #
32 ##### 2.- UPLOAD ANNUAL DATABASES AND SUBSET TO ESTADO AND SPECIES #####
33 #
34 # Las bases de datos deben entrar una cada año. #
35 # Cada base de datos anual debe tener las siguientes columnas, con exactamente #
36 # los mismos nombres (MAYUSCULAS) en exactamente el mismo orden. #
37 # Las nombres de columnas con *** deben tener datos. #
38 #
39 #
40 #
41 #
42 #
43 #
44 #
45 #
46 #
47 #
48 #
49 #
50 #
51 #
52 #
53 #
54 #
55 #
56 #
57 #
58 #
59 #
60 #
61 #
62 #
63 #
64 #
65 #
66 #
67 #
68 #
69 #
70 #
71 #
72 #
73 #
74 #
75 #
76 #
77 #
78 #
79 #
80 #
81 #
82 #
83 #
84 #
85 #
86 #
87 #
88 #
89 #
90 #
91 #
92 #
93 #
94 #
95 #
96 #
97 #
98 #
99 # 2000
100 arr2000all <- read.csv("Avisos dearribo 2000.csv",headers=TRUE,colClasses=c(rep("factor",11)))
101 arr2000 <- arr2000all[arr2000all$NOMBRE_ESTADO %in% estado,]
102 pulpo <- grep(stock, levels(arr2000$NOMBRE_ESPECIE))
103 arr2000 <- arr2000[arr2000$NOMBRE_ESPECIE==levels(arr2000$NOMBRE_ESPECIE)[pulpo[1]] |
104 arr2000$NOMBRE_ESPECIE==levels(arr2000$NOMBRE_ESPECIE)[pulpo[2]] |
105 arr2000$NOMBRE_ESPECIE==levels(arr2000$NOMBRE_ESPECIE)[pulpo[3]] |
106 arr2000$NOMBRE_ESPECIE==levels(arr2000$NOMBRE_ESPECIE)[pulpo[4]],]
107 arr2000 <- dropLevels(arr2000)
108 arr2000maya <- arr2000[arr2000$CLAVE_ESPECIE=="7291420H" | arr2000$CLAVE_ESPECIE=="7291628H",]
109 arr2000maya <- dropLevels(arr2000maya)
110 arr2000amer <- arr2000[arr2000$CLAVE_ESPECIE=="7281421H" | arr2000$CLAVE_ESPECIE=="7281629H",]
111 arr2000amer <- dropLevels(arr2000amer)
112 arr2000maya$FECHA_AVISO <- as.Date(arr2000maya$FECHA_AVISO,format="%d/%m/%Y")
113 arr2000maya$PERIODO_INICIO <- as.Date(arr2000maya$PERIODO_INICIO,format="%d/%m/%Y")
114 arr2000maya$PERIODO_FIN <- as.Date(arr2000maya$PERIODO_FIN,format="%d/%m/%Y")
115 arr2000maya$FECHA_EXPEDICION <- as.Date(arr2000maya$FECHA_EXPEDICION,format="%d/%m/%Y")
116 arr2000maya$FECHA_VIGENCIA <- as.Date(arr2000maya$FECHA_VIGENCIA,format="%d/%m/%Y")
117 arr2000maya$PESO_VIVO <- as.numeric(gsub(
118 # ONLY for strings containing numerics, comma, numerics
119 "^[0-9]+(,[0-9]+)?$",
120 # Substitute by the first part, dot, second part
121 "\\1\\.\\2",
122 arr2000maya$PESO_VIVO))
123 arr2000amer$FECHA_AVISO <- as.Date(arr2000amer$FECHA_AVISO,format="%d/%m/%Y")
124 arr2000amer$PERIODO_INICIO <- as.Date(arr2000amer$PERIODO_INICIO,format="%d/%m/%Y")
125 arr2000amer$PERIODO_FIN <- as.Date(arr2000amer$PERIODO_FIN,format="%d/%m/%Y")
126 arr2000amer$FECHA_EXPEDICION <- as.Date(arr2000amer$FECHA_EXPEDICION,format="%d/%m/%Y")
127 arr2000amer$FECHA_VIGENCIA <- as.Date(arr2000amer$FECHA_VIGENCIA,format="%d/%m/%Y")
128 arr2000amer$PESO_VIVO <- as.numeric(gsub(
129 "^[0-9]+(,[0-9]+)?$",
130 "\\1\\.\\2",
131 arr2000amer$PESO_VIVO))
132 rm(arr2000,arr2000all)
133 sapply(arr2000maya,class)
134
135 #####
136 #
137 #
138 #
139 #
140 #
141 #
142 #
143 #
144 #
145 #
146 #
147 #
148 #
149 #
150 #
151 #
152 #
153 #
154 #
155 #
156 #
157 #
158 #
159 #
160 #
161 #
162 #
163 #
164 #
165 #
166 #
167 #
168 #
169 #
170 #
171 #
172 #
173 #
174 #
175 #
176 #
177 #
178 #
179 #
180 #
181 #
182 #
183 #
184 #
185 #
186 #
187 #
188 #
189 #
190 #
191 #
192 #
193 #
194 #
195 #
196 #
197 #
198 #
199 #
200 #
201 #
202 #
203 #
204 #
205 #
206 #
207 #
208 #
209 #
210 #
211 #
212 #
213 #
214 #
215 #
216 #
217 #
218 #
219 #
220 #
221 #
222 #
223 #
224 #
225 #
226 #
227 #
228 #
229 #
230 #
231 #
232 #
233 #
234 #
235 #
236 #
237 #
238 #
239 #
240 #
241 #
242 #
243 #
244 #
245 #
246 #
247 #
248 #
249 #
250 #
251 #
252 #
253 #
254 #
255 #
256 #
257 #
258 #
259 #
260 #
261 #
262 #
263 #
264 #
265 #
266 #
267 #
268 #
269 #
270 #
271 #
272 #
273 #
274 #
275 #
276 #
277 #
278 #
279 #
280 #
281 #
282 #
283 #
284 #
285 #
286 #
287 #
288 #
289 #
290 #
291 #
292 #
293 #
294 #
295 #
296 #
297 #
298 #
299 #
300 #
301 #
302 #
303 #
304 #
305 #
306 #
307 #
308 #
309 #
310 #
311 #
312 #
313 #
314 #
315 #
316 #
317 #
318 #
319 #
320 #
321 #
322 #
323 #
324 #
325 #
326 #
327 #
328 #
329 #
330 #
331 #
332 #
333 #
334 #
335 #
336 #
337 #
338 #
339 #
340 #
341 #
342 #
343 #
344 #
345 #
346 #
347 #
348 #
349 #
350 #
351 #
352 #
353 #
354 #
355 #
356 #
357 #
358 #
359 #
360 #
361 #
362 #
363 #
364 #
365 #
366 #
367 #
368 #
369 #
370 #
371 #
372 #
373 #
374 #
375 #
376 #
377 #
378 #
379 #
380 #
381 #
382 #
383 #
384 #
385 #
386 #
387 #
388 #
389 #
390 #
391 #
392 #
393 #
394 #
395 #
396 #
397 #
398 #
399 #
400 #
401 #
402 #
403 #
404 #
405 #
406 #
407 #
408 #
409 #
410 #
411 #
412 #
413 #
414 #
415 #
416 #
417 #
418 #
419 #
420 #
421 #
422 #
423 #
424 #
425 #
426 #
427 #
428 #
429 #
430 #
431 #
432 #
433 #
434 #
435 #
436 #
437 #
438 #
439 #
440 #
441 #
442 #
443 #
444 #
445 #
446 #
447 #
448 #
449 #
450 #
451 #
452 #
453 #
454 #
455 #
456 #
457 #
458 #
459 #
460 #
461 #
462 #
463 #
464 #
465 #
466 #
467 #
468 #
469 #
470 #
471 #
472 #
473 #
474 #
475 #
476 #
477 #
478 #
479 #
480 #
481 #
482 #
483 #
484 #
485 #
486 #
487 #
488 #
489 #
490 #
491 #
492 #
493 #
494 #
495 #
496 #
497 #
498 #
499 #
500 #
501 #
502 #
503 #
504 #
505 #
506 #
507 #
508 #
509 #
510 #
511 #
512 #
513 #
514 #
515 #
516 #
517 #
518 #
519 #
520 #
521 #
522 #
523 #
524 #
525 #
526 #
527 #
528 #
529 #
530 #
531 #
532 #
533 #
534 #
535 #
536 #
537 #
538 #
539 #
540 #
541 #
542 #
543 #
544 #
545 #
546 #
547 #
548 #
549 #
550 #
551 #
552 #
553 #
554 #
555 #
556 #
557 #
558 #
559 #
560 #
561 #
562 #
563 #
564 #
565 #
566 #
567 #
568 #
569 #
570 #
571 #
572 #
573 #
574 #
575 #
576 #
577 #
578 #
579 #
580 #
581 #
582 #
583 #
584 #
585 #
586 #
587 #
588 #
589 #
590 #
591 #
592 #
593 #
594 #
595 #
596 #
597 #
598 #
599 #
600 #
601 #
602 #
603 #
604 #
605 #
606 #
607 #
608 #
609 #
610 #
611 #
612 #
613 #
614 #
615 #
616 #
617 #
618 #
619 #
620 #
621 #
622 #
623 #
624 #
625 #
626 #
627 #
628 #
629 #
630 #
631 #
632 #
633 #
634 #
635 #
636 #
637 #
638 #
639 #
640 #
641 #
642 #
643 #
644 #
645 #
646 #
647 #
648 #
649 #
650 #
651 #
652 #
653 #
654 #
655 #
656 #
657 #
658 #
659 #
660 #
661 #
662 #
663 #
664 #
665 #
666 #
667 #
668 #
669 #
670 #
671 #
672 #
673 #
674 #
675 #
676 #
677 #
678 #
679 #
680 #
681 #
682 #
683 #
684 #
685 #
686 #
687 #
688 #
689 #
690 #
691 #
692 #
693 #
694 #
695 #
696 #
697 #
698 #
699 #
700 #
701 #
702 #
703 #
704 #
705 #
706 #
707 #
708 #
709 #
710 #
711 #
712 #
713 #
714 #
715 #
716 #
717 #
718 #
719 #
720 #
721 #
722 #
723 #
724 #
725 #
726 #
727 #
728 #
729 #
730 #
731 #
732 #
733 #
734 #
735 #
736 #
737 #
738 #
739 #
740 #
741 #
742 #
743 #
744 #
745 #
746 #
747 #
748 #
749 #
750 #
751 #
752 #
753 #
754 #
755 #
756 #
757 #
758 #
759 #
760 #
761 #
762 #
763 #
764 #
765 #
766 #
767 #
768 #
769 #
770 #
771 #
772 #
773 #
774 #
775 #
776 #
777 #
778 #
779 #
780 #
781 #
782 #
783 #
784 #
785 #
786 #
787 #
788 #
789 #
790 #
791 #
792 #
793 #
794 #
795 #
796 #
797 #
798 #
799 #
800 #
801 #
802 #
803 #
804 #
805 #
806 #
807 #
808 #
809 #
810 #
811 #
812 #
813 #
814 #
815 #
816 #
817 #
818 #
819 #
820 #
821 #
822 #
823 #
824 #
825 #
826 #
827 #
828 #
829 #
830 #
831 #
832 #
833 #
834 #
835 #
836 #
837 #
838 #
839 #
840 #
841 #
842 #
843 #
844 #
845 #
846 #
847 #
848 #
849 #
850 #
851 #
852 #
853 #
854 #
855 #
856 #
857 #
858 #
859 #
860 #
861 #
862 #
863 #
864 #
865 #
866 #
867 #
868 #
869 #
870 #
871 #
872 #
873 #
874 #
875 #
876 #
877 #
878 #
879 #
880 #
881 #
882 #
883 #
884 #
885 #
886 #
887 #
888 #
889 #
890 #
891 #
892 #
893 #
894 #
895 #
896 #
897 #
898 #
899 #
900 #
901 #
902 #
903 #
904 #
905 #
906 #
907 #
908 #
909 #
910 #
911 #
912 #
913 #
914 #
915 #
916 #
917 #
918 #
919 #
920 #
921 #
922 #
923 #
924 #
925 #
926 #
927 #
928 #
929 #
930 #
931 #
932 #
933 #
934 #
935 #
936 #
937 #
938 #
939 #
940 #
941 #
942 #
943 #
944 #
945 #
946 #
947 #
948 #
949 #
950 #
951 #
952 #
953 #
954 #
955 #
956 #
957 #
958 #
959 #
960 #
961 #
962 #
963 #
964 #
965 #
966 #
967 #
968 #
969 #
970 #
971 #
972 #
973 #
974 #
975 #
976 #
977 #
978 #
979 #
980 #
981 #
982 #
983 #
984 #
985 #
986 #
987 #
988 #
989 #
990 #
991 #
992 #
993 #
994 #
995 #
996 #
997 #
998 #
999 #
1000 #

```

Figura 4: Procesamiento inicial de los datos de la base de datos Avisos de Arribo.

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
AvisosDeArribos.Data.R
Source on Save Run
98 #
99 # 2000
100 arr2000all <- read.csv("Avisos dearribo 2000.csv",headers=TRUE,colClasses=c(rep("factor",11)))
101 arr2000 <- arr2000all[arr2000all$NOMBRE_ESTADO %in% estado,]
102 pulpo <- grep(stock, levels(arr2000$NOMBRE_ESPECIE))
103 arr2000 <- arr2000[arr2000$NOMBRE_ESPECIE==levels(arr2000$NOMBRE_ESPECIE)[pulpo[1]] |
104 arr2000$NOMBRE_ESPECIE==levels(arr2000$NOMBRE_ESPECIE)[pulpo[2]] |
105 arr2000$NOMBRE_ESPECIE==levels(arr2000$NOMBRE_ESPECIE)[pulpo[3]] |
106 arr2000$NOMBRE_ESPECIE==levels(arr2000$NOMBRE_ESPECIE)[pulpo[4]],]
107 arr2000 <- dropLevels(arr2000)
108 arr2000maya <- arr2000[arr2000$CLAVE_ESPECIE=="7291420H" | arr2000$CLAVE_ESPECIE=="7291628H",]
109 arr2000maya <- dropLevels(arr2000maya)
110 arr2000amer <- arr2000[arr2000$CLAVE_ESPECIE=="7281421H" | arr2000$CLAVE_ESPECIE=="7281629H",]
111 arr2000amer <- dropLevels(arr2000amer)
112 arr2000maya$FECHA_AVISO <- as.Date(arr2000maya$FECHA_AVISO,format="%d/%m/%Y")
113 arr2000maya$PERIODO_INICIO <- as.Date(arr2000maya$PERIODO_INICIO,format="%d/%m/%Y")
114 arr2000maya$PERIODO_FIN <- as.Date(arr2000maya$PERIODO_FIN,format="%d/%m/%Y")
115 arr2000maya$FECHA_EXPEDICION <- as.Date(arr2000maya$FECHA_EXPEDICION,format="%d/%m/%Y")
116 arr2000maya$FECHA_VIGENCIA <- as.Date(arr2000maya$FECHA_VIGENCIA,format="%d/%m/%Y")
117 arr2000maya$PESO_VIVO <- as.numeric(gsub(
118 # ONLY for strings containing numerics, comma, numerics
119 "^[0-9]+(,[0-9]+)?$",
120 # Substitute by the first part, dot, second part
121 "\\1\\.\\2",
122 arr2000maya$PESO_VIVO))
123 arr2000amer$FECHA_AVISO <- as.Date(arr2000amer$FECHA_AVISO,format="%d/%m/%Y")
124 arr2000amer$PERIODO_INICIO <- as.Date(arr2000amer$PERIODO_INICIO,format="%d/%m/%Y")
125 arr2000amer$PERIODO_FIN <- as.Date(arr2000amer$PERIODO_FIN,format="%d/%m/%Y")
126 arr2000amer$FECHA_EXPEDICION <- as.Date(arr2000amer$FECHA_EXPEDICION,format="%d/%m/%Y")
127 arr2000amer$FECHA_VIGENCIA <- as.Date(arr2000amer$FECHA_VIGENCIA,format="%d/%m/%Y")
128 arr2000amer$PESO_VIVO <- as.numeric(gsub(
129 "^[0-9]+(,[0-9]+)?$",
130 "\\1\\.\\2",
131 arr2000amer$PESO_VIVO))
132 rm(arr2000,arr2000all)
133 sapply(arr2000maya,class)
134
135 #####
136 #
137 #
138 #
139 #
140 #
141 #
142 #
143 #
144 #
145 #
146 #
147 #
148 #
149 #
150 #
151 #
152 #
153 #
154 #
155 #
156 #
157 #
158 #
159 #
160 #
161 #
162 #
163 #
164 #
165 #
166 #
167 #
168 #
169 #
170 #
171 #
172 #
173 #
174 #
175 #
176 #
177 #
178 #
179 #
180 #
181 #
182 #
183 #
184 #
185 #
186 #
187 #
188 #
189 #
190 #
191 #
192 #
193 #
194 #
195 #
196 #
197 #
198 #
199 #
200 #
201 #
202 #
203 #
204 #
205 #
206 #
207 #
208 #
209 #
210 #
211 #
212 #
213 #
214 #
215 #
216 #
217 #
218 #
219 #
220 #
221 #
222 #
223 #
224 #
225 #
226 #
227 #
228 #
229 #
230 #
231 #
232 #
233 #
234 #
235 #
236 #
237 #
238 #
239 #
240 #
241 #
242 #
243 #
244 #
245 #
246 #
247 #
248 #
249 #
250 #
251 #
252 #
253 #
254 #
255 #
256 #
257 #
258 #
259 #
260 #
261 #
262 #
263 #
264 #
265 #
266 #
267 #
268 #
269 #
270 #
271 #
272 #
273 #
274 #
275 #
276 #
277 #
278 #
279 #
280 #
281 #
282 #
283 #
284 #
285 #
286 #
287 #
288 #
289 #
290 #
291 #
292 #
293 #
294 #
295 #
296 #
297 #
298 #
299 #
300 #
301 #
302 #
303 #
304 #
305 #
306 #
307 #
308 #
309 #
310 #
311 #
312 #
313 #
314 #
315 #
316 #
317 #
318 #
319 #
320 #
321 #
322 #
323 #
324 #
325 #
326 #
327 #
328 #
329 #
330 #
331 #
332 #
333 #
334 #
335 #
336 #
337 #
338 #
339 #
340 #
341 #
342 #
343 #
344 #
345 #
346 #
347 #
348 #
349 #
350 #
351 #
352 #
353 #
354 #
355 #
356 #
357 #
358 #
359 #
360 #
361 #
362 #
363 #
364 #
365 #
366 #
367 #
368 #
369 #
370 #
371 #
372 #
373 #
374 #
375 #
376 #
377 #
378 #
379 #
380 #
381 #
382 #
383 #
384 #
385 #
386 #
387 #
388 #
389 #
390 #
391 #
392 #
393 #
394 #
395 #
396 #
397 #
398 #
399 #
400 #
401 #
402 #
403 #
404 #
405 #
406 #
407 #
408 #
409 #
410 #
411 #
412 #
413 #
414 #
415 #
416 #
417 #
418 #
419 #
420 #
421 #
422 #
423 #
424 #
425 #
426 #
427 #
428 #
429 #
430 #
431 #
432 #
433 #
434 #
435 #
436 #
437 #
438 #
439 #
440 #
441 #
442 #
443 #
444 #
445 #
446 #
447 #
448 #
449 #
450 #
451 #
452 #
453 #
454 #
455 #
456 #
457 #
458 #
459 #
460 #
461 #
462 #
463 #
464 #
465 #
466 #
467 #
468 #
469 #
470 #
471 #
472 #
473 #
474 #
475 #
476 #
477 #
478 #
479 #
480 #
481 #
482 #
483 #
484 #
485 #
486 #
487 #
488 #
489 #
490 #
491 #
492 #
493 #
494 #
495 #
496 #
497 #
498 #
499 #
500 #
501 #
502 #
503 #
504 #
505 #
506 #
507 #
508 #
509 #
510 #
511 #
512 #
513 #
514 #
515 #
516 #
517 #
518 #
519 #
520 #
521 #
522 #
523 #
524 #
525 #
526 #
527 #
528 #
529 #
530 #
531 #
532 #
533 #
534 #
535 #
536 #
537 #
538 #
539 #
540 #
541 #
542 #
543 #
544 #
545 #
546 #
547 #
548 #
549 #
550 #
551 #
552 #
553 #
554 #
555 #
556 #
557 #
558 #
559 #
560 #
561 #
562 #
563 #
564 #
565 #
566 #
567 #
568 #
569 #
570 #
571 #
572 #
573 #
574 #
575 #
576 #
577 #
578 #
579 #
580 #
581 #
582 #
583 #
584 #
585 #
586 #
587 #
588 #
589 #
590 #
591 #
592 #
593 #
594 #
595 #
596 #
597 #
598 #
599 #
600 #
601 #
602 #
603 #
604 #
605 #
606 #
607 #
608 #
609 #
610 #
611 #
612 #
613 #
614 #
615 #
616 #
617 #
618 #
619 #
620 #
621 #
622 #
623 #
624 #
625 #
626 #
627 #
628 #
629 #
630 #
631 #
632 #
633 #
634 #
635 #
636 #
637 #
638 #
639 #
640 #
641 #
642 #
643 #
644 #
645 #
646 #
647 #
648 #
649 #
650 #
651 #
652 #
653 #
654 #
655 #
656 #
657 #
658 #
659 #
660 #
661 #
662 #
663 #
664 #
665 #
666 #
667 #
668 #
669 #
670 #
671 #
672 #
673 #
674 #
675 #
676 #
677 #
678 #
679 #
680 #
681 #
682 #
683 #
684 #
685 #
686 #
687 #
688 #
689 #
690 #
691 #
692 #
693 #
694 #
695 #
696 #
697 #
698 #
699 #
700 #
701 #
702 #
703 #
704 #
705 #
706 #
707 #
708 #
709 #
710 #
711 #
712 #
713 #
714 #
715 #
716 #
717 #
718 #
719 #
720 #
721 #
722 #
723 #
724 #
725 #
726 #
727 #
728 #
729 #
730 #
731 #
732 #
733 #
734 #
735 #
736 #
737 #
738 #
739 #
740 #
741 #
742 #
743 #
744 #
745 #
746 #
747 #
748 #
749 #
750 #
751 #
752 #
753 #
754 #
755 #
756 #
757 #
758 #
759 #
760 #
761 #
762 #
763 #
764 #
765 #
766 #
767 #
768 #
769 #
770 #
771 #
772 #
773 #
774 #
775 #
776 #
777 #
778 #
779 #
780 #
781 #
782 #
783 #
784 #
785 #
786 #
787 #
788 #
789 #
790 #
791 #
792 #
793 #
794 #
795 #
796 #
797 #
798 #
799 #
800 #
801 #
802 #
803 #
804 #
805 #
806 #
807 #
808 #
809 #
810 #
811 #
812 #
813 #
814 #
815 #
816 #
817 #
818 #
819 #
820 #
821 #
822 #
823 #
824 #
825 #
826 #
827 #
828 #
829 #
830 #
831 #
832 #
833 #
834 #
835 #
836 #
837 #
838 #
839 #
840 #
841 #
842 #
843 #
844 #
845 #
846 #
847 #
848 #
849 #
850 #
851 #
852 #
853 #
854 #
855 #
856 #
857 #
858 #
859 #
860 #
861 #
862 #
863 #
864 #
865 #
866 #
867 #
868 #
869 #
870 #
871 #
872 #
873 #
874 #
875 #
876 #
877 #
878 #
879 #
880 #
881 #
882 #
883 #
884 #
885 #
886 #
887 #
888 #
889 #
890 #
891 #
892 #
893 #
894 #
895 #
896 #
897 #
898 #
899 #
900 #
901 #
902 #
903 #
904 #
905 #
906 #
907 #
908 #
909 #
910 #
911 #
912 #
913 #
914 #
915 #
916 #
917 #
918 #
919 #
920 #
921 #
922 #
923 #
924 #
925 #
926 #
927 #
928 #
929 #
930 #
931 #
932 #
933 #
934 #
935 #
936 #
937 #
938 #
939 #
940 #
941 #
942 #
943 #
944 #
945 #
946 #
947 #
948 #
949 #
950 #
951 #
952 #
953 #
954 #
955 #
956 #
957 #
958 #
959 #
960 #
961 #
962 #
963 #
964 #
965 #
966 #
967 #
968 #
969 #
970 #
971 #
972 #
973 #
974 #
975 #
976 #
977 #
978 #
979 #
980 #
981 #
982 #
983 #
984 #
985 #
986 #
987 #
988 #
989 #
990 #
991 #
992 #
993 #
994 #
995 #
996 #
997 #
998 #
999 #
1000 #

```

Figura 5: Procesamiento final de los datos de la base de datos Avisos de Arribo.

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
AvisosDeArribos Data R
Source on Save Run
929- ##### 3.- AGGREGATE EFFORT AND CATCH AND WEIGHT BY WEEK #####
930- #
931- # O. maya
932- # 2000
933- arr2000maya$Week <- as.numeric(format(arr2000maya$FECHA.AVISO, "%W"))
934- arr2000maya$DURACION <- as.numeric(as.character(arr2000maya$DURACION))
935- arr2000maya$DIAS.EFECTIVOS <- as.numeric(as.character(arr2000maya$DIAS.EFECTIVOS))
936- cateff2000maya <- data.frame(aggregate(arr2000maya$PESO.VIVO,list(arr2000maya$Week),sum),
937- aggregate(arr2000maya$DIAS.EFECTIVOS,list(arr2000maya$Week),sum)$x)
938- names(cateff2000maya) <- c("Week", "Catch.kg", "Eff.days")
939- plot(cateff2000maya$Eff.days, cateff2000maya$Catch.kg)

```

Figura 6: Bases de datos biológicas de pulpos de INAPESCA. Lado izquierdo: *O. americanus*; lado derecho: *O. maya*.

FECHA	YEAR	LM,mm	PESO.TOTAL.g	PESO.EVICERADO.g	SEXO
2336	2016-09-02	2016	167	1326	M
2337	2016-09-02	2016	147	1081	M
2338	2016-09-02	2016	172	1931	M
2339	2016-09-02	2016	120	979	M
2340	2016-09-02	2016	113	880	M
2341	2016-09-02	2016	157	1285	M
2342	2016-09-02	2016	115	771	H
2343	2016-09-02	2016	120	788	M
2344	2016-09-02	2016	132	942	M
2345	2016-09-02	2016	100	398	M
2346	2016-09-02	2016	83	353	M
2347	2016-09-02	2016	118	1164	M
2348	2016-09-02	2016	120	1030	M
2349	2016-09-02	2016	142	1192	H
2350	2016-09-02	2016	137	1081	H
2351	2016-09-02	2016	120	619	M
2352	2016-09-02	2016	155	1902	H
2353	2016-09-02	2016	128	1010	M
2354	2016-09-02	2016	143	1079	H
2355	2016-09-02	2016	128	1193	H
2356	2016-09-02	2016	140	1346	M
2357	2016-09-02	2016	162	1932	M
2358	2016-09-02	2016	107	879	M
2359	2016-09-02	2016	138	1812	M
2360	2016-09-02	2016	112	1090	M
2361	2016-09-02	2016	147	1396	H
2362	2016-09-02	2016	100	1105	M
2363	2016-09-02	2016	124	1187	H
2364	2016-09-02	2016	112	1091	M
2365	2016-09-02	2016	142	1442	M

alijos y las flota de naves nodriza que pueden llevar muchos más alijos. En otros casos en que la flota sea homogénea la mejor medida de esfuerzo puede ser DURACION.

El guión `AvisosDeArribos.Data.R` finaliza reuniendo la información biológica (pesos medios por semana) con la información pesquera (captura y esfuerzo por semana) en una tabla final que ya está lista para usar en la evaluación de stocks.

3.2. Datos biológicos

El procesamiento de las bases de datos biológicas va a depender de cómo se hayan guardado los datos biológicos así que no es posible presentar un procedimiento estandarizado para todas las especies presentes en la base

de datos Avisos de Arribo. Por lo tanto lo que se describe a continuación en el guión [Yucoct.biodata.R](#) es aplicable a cualquier especie cuyos datos biológicos estén estructurados de manera similar a los datos de pulpos que colecta INAPESCA.

El fin último del guión [Yucoct.biodata.R](#) es producir una tabla con la semana, el peso medio esperado, y la desviación estándar del peso medio esperado de cada una de las dos especies de pulpos.

Los datos biológicos de INAPESCA se muestran en la Fig. 6. Estos datos consisten en fecha de muestreo, longitud del manto, peso total, peso eviscerado y sexo. Lo que importa obtener de aquí es el peso medio por semana para cada una de las semanas de todas las temporadas de pesca desde 2000 a 2021. Sin embargo, no todos los años ni todas las semanas de cada temporada anual contienen datos de peso no obstante que este dato es necesario para todas las semanas de todas las temporadas de pesca. Por este motivo el guión [Yucoct.biodata.R](#) se ocupa de procesar estos datos agregando los pesos medios por semana para todos los años disponible y con estos datos ajusta un modelo *spline* a la relación entre peso medio y semana para predecir los pesos medios esperados en todas las semanas de todas las temporadas de pesca.

Finalmente, al final del guión [AvisosDeArribos.Data.R](#), el fichero con la semana, el peso medio esperado, y la desviación estándar del peso medio esperado de cada una de las dos especies de pulpos es traído al espacio de trabajo. Aquí es usado para generar vectores de peso medio re-muestreados desde una distribución normal estandar truncada (librería [Runuran](#)) para adicionar como columna nueva a cada una de las tablas de capturas y esfuerzo agregados a la semana para cada especie de pulpo y cada temporada de pesca.

El resultado final de cada año se guarda en el disco con el nombre `catfeffXXXXyyyy` donde `XXXX` es el año y `yyyy` es la especie (maya o amer).

4. Etapa 2: modelos de depleción intra-anales

Los modelos de depleción intra-anales en [CatDyn](#) analizan la *dinámica de capturas* diaria o semanal para producir estimados de máxima verosimilitud de la abundancia inicial del stock (N_0), la mortalidad natural promedio en la temporada de pesca (M), entradas y salidas episódicas de abundancia durante la temporada de pesca (R para entradas y S para salidas), el coeficiente de capturabilidad generalizado (k), un modulador de potencia del efecto del esfuerzo de pesca sobre la captura (la respuesta al esfuerzo, α) y un modulador de potencia del efecto de la abundancia del stock sobre la captura (la respuesta a la abundancia, β). Puesto que la pesca de los pulpos

en Yucatán y alrededores ocurre en una temporada que deja fuera a la temporada reproductiva, parece no haber mecanismos de salidas de abundancia que deban ser incluidos en los modelos de depleción en estas aplicaciones. Con estas definiciones los modelos intra-anales están definidos por:

$$\begin{aligned}
 C_t &= kE_t^\alpha N_t^\beta = kE_t^\alpha m f_t(M, N_0, C_{i < t}, R, S) \\
 &= kE_t^\alpha m \left(N_0 e^{-Mt} - m \left[\sum_{i=1}^{i=t-1} C_{i,j} e^{-M(t-i-1)} \right] + \sum_{j=1}^{j=u} I_j R_j e^{-M(t-\tau_j)} \right)^\beta \quad (1) \\
 t > 0, C_t \geq 0, E_t \geq 0, k > 0, N_0 > 0, \alpha > 0, \beta > 0, M > 0, R_j > 0
 \end{aligned}$$

donde C_t es la verdadera captura en la semana t , que no es observada y en cuyo lugar se observa una captura con errores, que es una variable aleatoria; E_t es el esfuerzo observado de pesca y que es conocido exactamente como una variable determinista; $m = \exp(-M/2)$ es un conector de tiempo continuo a tiempo discreto que hace que toda la captura ocurra instantáneamente a mitad de semana; N es la abundancia del stock y que es una variable latente (no observada exactamente ni medida con error); el índice j corre sobre un número u de entradas de abundancia que ocurren en semanas específicas de la temporada de pesca τ_j ; y finalmente la variable indicadora I_j es 0 antes de τ_j y es 1 en la semana τ_j y después. En estos modelos el conjunto de parámetros a ser estimados simultáneamente por máxima verosimilitud es $\theta_{IAGD} = [k M N_0 \{R_j\} \alpha \beta]$. En [CatDyn](#) todos los parámetros son estimados simultáneamente y sin restricciones desde valores iniciales proporcionados por el usuario.

El guión [Yucoct.Maya.CatDyn.IAGD.2000.R](#) contiene una evaluación de un caso relativamente simple, con 2 entradas de abundancia durante la temporada de pesca. En cambio el guión [Yucoct.Amer.CatDyn.IAGD.2017.R](#) contiene un caso más complejo donde hay que explorar mucho más para obtener una variante de modelos con tres entradas de abundancia.

4.1. Temporada 2000 de O. maya

Abra el guión [Yucoct.Maya.CatDyn.IAGD.2000.R](#), cambiando la dirección del disco duro, trayendo el resultado del guión [AvisosDeArribos.Data.R](#) que es su tabla de datos de captura, esfuerzo y pesos medios, y cargando la librería [CatDyn](#).

```
library(CatDyn)
```

4.1.1. Objeto de datos para **CatDyn**

Re-ordene los datos con la función `as.CatDynData`, sólo incluyendo las semanas de la temporada de pesca, posterior a la semana 31:

```
1 maya2000 <- as.CatDynData(x=cateff2000maya[cateff2000maya$Week  
2 >31,],  
3 step="week",  
4 fleet.name="alijos",  
5 coleff=3,  
6 colcat=2,  
7 colmbw=7,  
8 unitseff="effdays",  
9 unitscat="kg",  
10 unitsmbw="g",  
11 nmult="thou",  
season.dates=c("2000-08-01", "2000-12-05")  
))
```

Los números de columnas son del esfuerzo (col. 3), capturas (col. 2, y pesos medios (col. 7), lo que puede diferir en el caso del cada usuario puesto que siguiendo el guión `AvisosDeArribos.Data.R` la columna de los pesos pedios debe ser la 4. Siempre se puede obtener documentación de qué significa cada argumento de la función ingresando un signo de interrogación seguido por el nombre de la función en la consola de R.

Examine el nuevo objeto de datos de nombre 'maya2000'. Para lograr esto, simplemente en el guión marque el nombre del objeto con doble-click y luego pulse 'Run'. Verá un objeto con la forma que se muestra en la Fig. 7. El primer componente, 'Properties', tiene tres elementos. El primero son las unidades: paso de tiempo semanal, captura en kg, pesos de los pulpos en g, y multiplicador de número en miles ('thou' es una abreviación de 'thousands'). El uso del multiplicador de números es necesario debido a que las capturas en números son muy grandes, así que para la posterior optimización numérica, resulta mucho más conveniente trabajar con las capturas en miles de pulpos. El segundo elemento son los nombres de las flotas y las unidades del esfuerzo de pesca. En este caso sólo hay una flota. El tercer componente son las fechas. La función `as.CatDynData` tiene un contador de semanas que usa las fechas entregadas y si este número es diferente al número de filas de los datos de captura, esfuerzo y pesos medios, genera un error. Para que el conteo de semanas a partir de las fechas y a partir del número de filas de los datos sea coincidente, hay que manipular el día de la fecha de inicio o de la fecha de fin de la temporada hasta que ya no se genere el mensaje de error.

El segundo componente de este objeto son los datos propiamente tales.

Figura 7: Datos de capturas, esfuerzos y pesos medios organizados con la función `as.CatDynData` de `CatDyn`, para el ajuste de modelos de depleción.

```

File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console Terminal Background Jobs
R 4.2.3 ~/Workplace/Consultancy/COBI/2023/Training/Session3/
> #
> maya2000
$Properties
$Properties$Units
      Time Step      Catch      Bodymass NumbersMultiplier
      "week"      "kg"      "g"      "thou"

$Properties$Fleets
  Fleet Units
1 alijos effdays

$Properties$Dates
  StartDate  EndDate
"2000-08-01" "2000-12-05"

$Data
$Data$alijos
  time.step obseff.effdays obscat.kg obsmbw.g obscat.thou  spikecat
1         31             80 254668.2 623.1706  408.66531 -0.70192021
2         32             128 576924.7 625.6906  922.06067  1.15074868
3         33             168 742608.3 629.5982 1179.49556  1.25000000
4         34             104 355273.7 630.6013  563.38879 -0.64014347
5         35              66 250196.6 638.5652  391.81059 -0.11565119
6         36             162 639858.5 646.0008  990.49180 -0.03991184
7         37             160 604898.6 652.8135  926.60248 -0.47741165
8         38             163 567678.6 660.9573  858.87330 -1.20788322
9         39             28 125110.8 672.5556  186.02299  0.11880688
10        40             103 358192.8 679.5597  527.09544 -0.89576248
11        41             192 804190.7 690.0714 1165.37317 -0.11973250
12        42             152 679957.0 701.8903  968.75111  0.29659958
13        43             104 529399.8 719.6238  735.66184  0.82042202
14        44              70 314366.5 728.0692  431.78107  0.01489324
15        45             125 385552.3 745.2563  517.34189 -2.12428832
16        46             89 171926.5 762.3659  225.51703 -2.72343802
17        47              39  78364.1 778.3061  100.68546 -1.17761850
18        48              60 271003.7 802.0666  337.88179 -0.26037039
19        49              5 12158.3 818.9109   14.84691 -0.13454155

attr(,"class")
[1] "CatDynData"
>

```

El nombre de la flota ('alijos') se usa como nombre de la tabla de datos. Esta tabla contiene el paso de tiempo (contando consecutivamente desde el número de la semana inicial), el esfuerzo (observado) de pesca, la captura (observada) en peso, el peso medio (observado), la captura (observada) en número, y un estadístico no-paramétrico ('catch spike') que sirve para identificar semanas de reclutamiento y de emigración desovante, aunque lo último no ocurre en las pesquerías de pulpos de Yucatán porque la temporada de pesca no incluye el período de emparejamiento y desove. Finalmente el objeto tiene una clase, específica de CatDyn, y que se llama 'CatDynData'. Los objetos con clase pueden ser manipulados de manera específica por funciones genéricas, tales como `plot`.

4.1.2. Semanas candidatas a entradas de abundancia: estadístico 'catch spike'

Utilice la función general de R `plot` con el objeto de clase 'CatDynData' recién creado:

```
1 plot(x=maya2000, mark=TRUE, offset=c(9,10), hem="N")
```

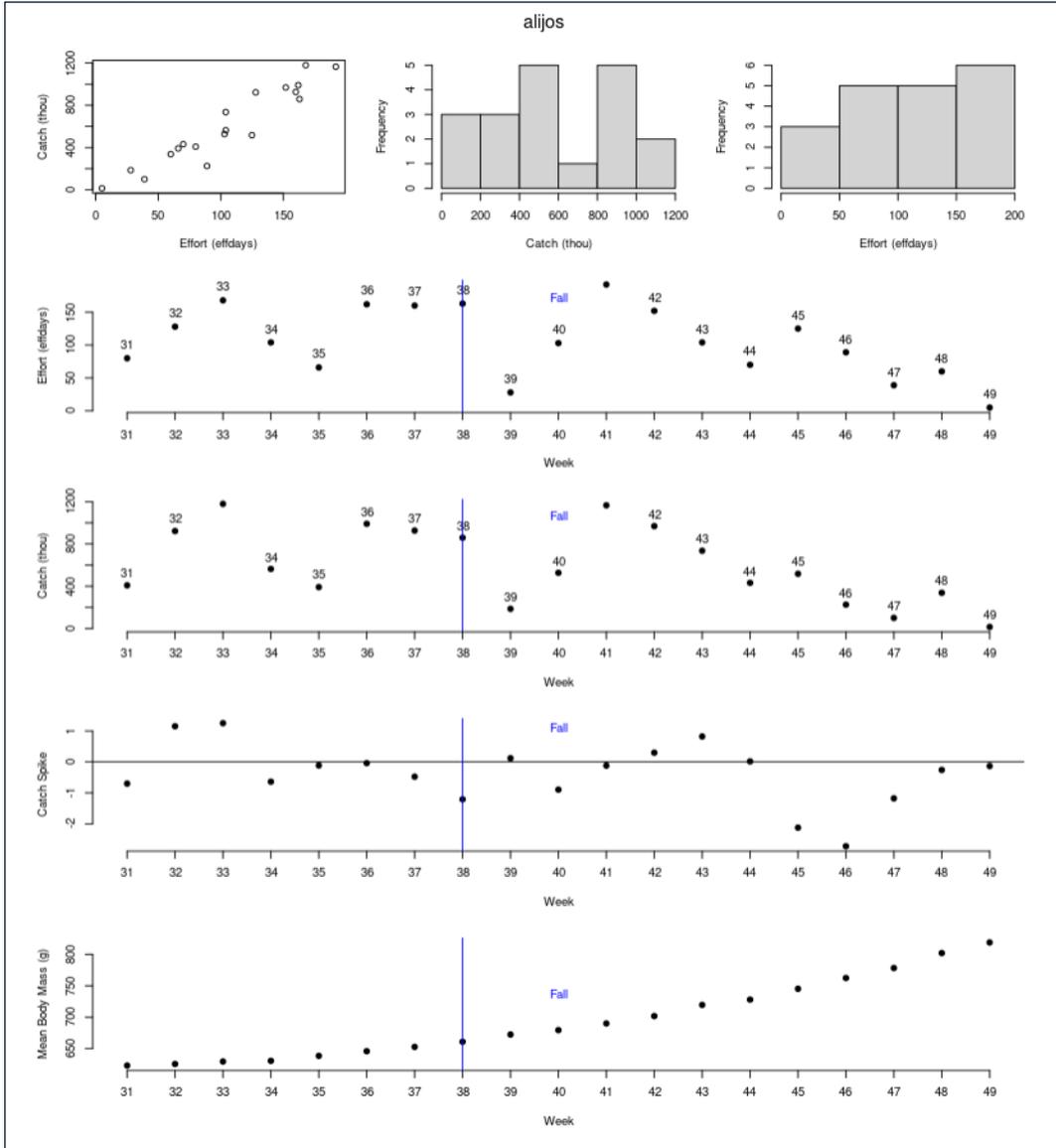
La gráfica que aparece en el panel multi-propósito tiene 7 paneles (Fig. 7). El primer panel superior muestra la conexión entre esfuerzo (la causa) y capturas en miles de pulpos (el resultado), que debe idealmente ser como es en estos datos, es decir incremental. Los siguientes dos paneles superiores son las distribuciones de frecuencia de la captura y el esfuerzo. Los cuatro paneles inferiores son las series de tiempo de esfuerzo, captura, el estadístico no-paramétrico 'catch spike', y el peso medio.

Cuando el estadístico no-paramétrico *catch spike* es más positivo señala a pasos de tiempo en los cuales hubo un incremento de capturas que no puede ser explicado por un aumento puntual del esfuerzo de pesca, implicando entonces un aumento de abundancia, es decir una entrada de abundancia que re-inicia el proceso de depleción. En esta figura vemos que las semanas con valores más positivos del 'catch spike' son las semanas 32, 33 y 43, y tomamos nota de esas semanas para la modelización de los modelos de depleción.

4.1.3. Semanas candidatas a entradas de abundancia: modelo de depleción pura

Determinar y probar pasos de tiempo en que hubo entradas de abundancia es muy importante en los modelos de depleción generalizados. Estos

Figura 8: Datos crudos para los modelos de depleción de la temporada 2000 de *O. maya*.



pasos de tiempo en que ocurren estos pulsos de entrada (y pulsos de salida en algunos casos) son parámetros a estimar pero no son parámetros diferenciables pues sólo pueden tomar valores discretos, pasos de tiempo específicos. Así que no se pueden estimar por máxima verosimilitud con métodos basados en derivadas como se hace con los otros parámetros del modelo, los parámetros diferenciables. Por supuesto que es aún posible obtener estimados de máxima verosimilitud de los pasos de tiempo en que ocurren entradas (y salidas) de abundancia, pero la maximización se logra ejecutando la optimización estadística para diversos valores de esos pasos de tiempo y seleccionando la variante que maximiza la verosimilitud de los datos o que minimiza el Akaike Information Criterion (AIC).

Además del *catch spike* es prudente utilizar otro método para la determinación de buenas semanas candidatas en que ocurre una entrada (o salida) de abundancia. Este método consiste en ajustar estadísticamente modelos de depleción sin entradas (o salidas) de abundancia, es decir una pesca hipotética donde sólo ocurre depleción a lo largo de toda la temporada de pesca, lo que requiere una variante de modelo con **depleción pura**. Examinando los resultados del ajuste del modelo de depleción pura podemos identificar a las semanas con los residuos más positivos (y los más negativos) como buenas semanas candidatas de entradas (y salidas) de abundancia.

Esta prueba nos permitirá además comenzar con el uso de funciones de exploración de variantes de modelos y su ajuste estadístico con métodos numéricos de optimización. Comenzamos por definir buenos valores iniciales para los parámetros a estimar del modelo de depleción pura. El valor inicial del parámetro de mortalidad natural puede obtenerse desde la longevidad de la especie de interés y la relación empírica entre longevidad y mortalidad natural publicada por John M. Hoenig en 1983, y que ha sido incorporada en [CatDyn](#) para utilizarla en la generación de valores iniciales de la tasa de mortalidad natural M :

```

1 # Pure depletion - Null model to detect timings
2 max.age      <- 2          # in years
3 time.step    <- "week"    # time step of these depletion models
4 M.Hoenig(max.age=max.age, time.step=time.step)
5 #   M.pred.week M.pred.SE.week
6 #1   0.04031779  0.005108589

```

Un valor inicial en los alrededores de 0.04 es adecuado según la relación empírica entre longevidad y mortalidad natural.

El valor inicial de N_0 puede obtenerse sumando las capturas en número (la columna *obs.cat.thou* en la Fig. 7) y multiplicando el resultado por algún

factor entre 5 y 10:

```
1 sum(maya2000$Data$alijos$obscat.thou)
2 #[1] 11452.35
```

El valor inicial de k debe estar entre 10^{-4} y 10^{-8} pues esta constante corresponde a la fracción del stock que es extraída por una unidad de esfuerzo y una unidad de abundancia. Los valores de α y β siempre deben estar bien cerca de 1, y finalmente los valores de la dispersión en las funciones de verosimilitud exactas se obtienen de la varianza muestral en los datos de captura en número. En **CatDyn** todos los parámetros son estimados en la escala logarítmica (natural) y los valores iniciales deben ser entregados en la escala logarítmica. Todo lo anterior se representa en el código como:

```
1 M.ini      <- 0.09 #1/Time step 1.5/53
2 N0.ini     <- 150000
3 k.ini      <- 8e-5
4 alpha.ini  <- 1.5
5 beta.ini   <- 0.8
6 psi.n.ini  <- sd(maya2000$Data$alijos$obscat.thou)^2
7 psi.ln.ini <- sd(log(maya2000$Data$alijos$obscat.thou))^2
8 pars.ini.OP <- log(c(M.ini ,N0.ini ,k.ini ,alpha.ini ,beta.ini))
```

Los modelos de depleción en **CatDyn** también deben recibir un vector de fechas que contenga la semana inicial, las semanas en las que ocurre una entrada (salida) de abundancia, y la semana final. En los modelos de depleción pura este vector sólo tiene la semana inicial y la final:

```
1 dates.OP <- c(head(maya2000$Data$alijos$time.step,1) ,
2               tail(maya2000$Data$alijos$time.step,1))
```

Enseguida podemos visualizar qué tan buenos son los valores iniciales creando un modelo lúdico. Este es el modelo que se obtiene desde los valores iniciales de los parámetros y cuyo único uso es obtener buenos valores iniciales para la posterior optimización. Estos modelos lúdicos se crean con la función **catdynexp** y luego haciendo un plot con el objeto resultante y principalmente verificando que la predicción de capturas del modelo lúdico cae sobre los datos de capturas:

```

1 maya2000.OP.apn.ini <- catdynexp(x=maya2000,
2                               p=0,
3                               par=pars.ini.OP,
4                               dates=dates.OP,
5                               distr="apnormal")
6 plot(x=maya2000.OP.apn.ini,
7      leg.pos="topleft",
8      Biom.tstep=1,
9      Biom.xpos=0.75,
10     Biom.ypos=1.25,
11     Cat.tstep=dim(maya2000$Data$alijos)[1],
12     Cat.xpos=0.75,
13     Cat.ypos=1.15)

```

Posteriormente, utilizaremos la función **CatDynFit** para generar un modelo estadístico propiamente tal, producido a través de un proceso de optimización numérica. Una analogía útil para entender el uso y la importancia del modelo lúdico es la siguiente: en un partido de fútbol lo que se busca es marcar goles. Un gol es equivalente a un modelo generado por optimización numérica. Sin embargo antes de poder marcar un gol, es decir antes de meter la pelota en el arco, es necesario llevar la pelota al campo de juego. El acto de llevar la pelota al campo de juego es equivalente a construir un modelo lúdico, sirve de punto de partida para el proceso serio, que sería marcar un gol o equivalentemente ajustar el modelo mediante optimización numérica.

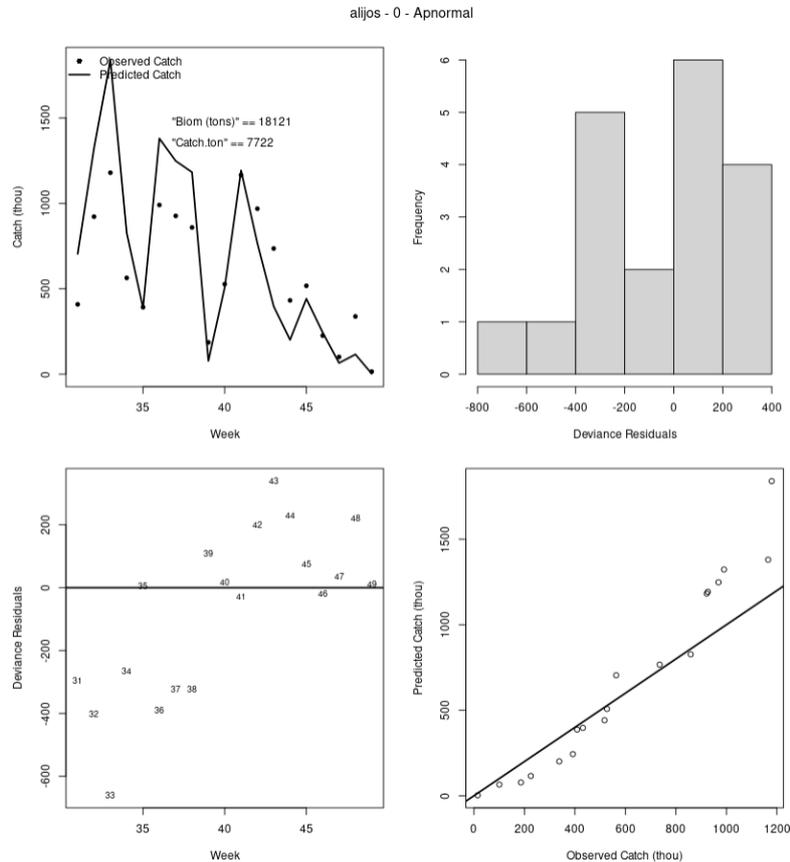
El modelo de depleción pura tiene la siguiente forma matemática:

$$C_t = kE_t^\alpha N_t^\beta = kE_t^\alpha e^{M/2} \left(N_0 e^{-Mt} - e^{M/2} \left[\sum_{i=1}^{j=t-1} C_i e^{-M(t-i-1)} \right] \right)^\beta \quad (2)$$

donde C_t es la captura en número por semana t y E_t es el esfuerzo de pesca por semana, y cuyos parámetros a estimar son la mortalidad natural M , la abundancia inicial N_0 , el escalamiento k , el modulador de la respuesta al esfuerzo α y el modulador de la respuesta a la abundancia β .

Al hacer una gráfica con el resultado de la función **catdynexp** obtenemos cuatro paneles (Fig. 9): el superior izquierdo muestra los datos (puntos) y el modelo lúdico (línea), el panel superior derecho muestra la distribución de frecuencia de los residuos del modelo, el panel inferior izquierdo muestra la nube de puntos residuales y el panel inferior derecho muestra el gráfico q-q con los cuantiles observados y predichos por el modelo. Se observa que el modelo lúdico ya tiene buenos valores iniciales para los parámetros puesto que el modelo (la línea) cae sobre los datos, la distribución de frecuencia

Figura 9: Modelo lúdico de depleción pura para los datos de la temporada 2000 de pulpo maya con valores iniciales para los parámetros determinados por el usuario.



de los residuos es algo simétrica, la nube residual parece estar equilibrada sobre la línea del 0 y en el gráfico q-q casi todos los puntos caen sobre la diagonal de 45 grados. Por lo tanto con estos valores iniciales, en el vector 'pars.ini.OP', ya son útiles para la optimización estadística.

Cuando los valores iniciales de los parámetros son muy inapropiados, la función `catdynexp` genera un mensaje de error, que puede decir explícitamente que la mortalidad natural o la abundancia inicial deben ser cambiadas para aumentar la abundancia o puede emitir un mensaje de error más críptico que se refiera a problemas en resolver la función `uniroot`. Cuando esto ocurre, el problema es que el usuario ha definido una mortalidad natural muy alta y/o una abundancia muy baja y por lo tanto la captura obtenida no puede ser explicada por el modelo. El problema se resuelve aumentando el valor inicial de N_0 o disminuyendo el valor inicial de M hasta que `catdynexp()` ya no produzca el mensaje de error.

El ajuste estadístico del modelo de depleción pura (y de todos los modelos con entradas y salidas de abundancia) se ejecuta utilizando la función `CatDynFit`, lo que puede tardar desde segundos a minutos (mientras se ejecuta la optimización, un botón rojo que dice STOP aparece en la esquina superior derecha de la consola; si se presiona ese botón la optimización es abortada):

```
1 maya2000.OP.apn.fit <- CatDynFit(x=maya2000,  
2                               p=0,  
3                               par=pars.ini.OP,  
4                               dates=dates.OP,  
5                               distr="apnormal",  
6                               method=c("spg", "CG"),  
7                               itnmax=50000)
```

El argumento $p = 0$ determina que este es un modelo de depleción pura, sin entradas (ni salidas) de abundancia durante la temporada de pesca. En `CatDyn` hay dos formas de determinar distribución normal de los datos: 'apnormal' es una aproximación que elimina en parámetro de dispersión y 'normal' es la distribución normal exacta, que incluye el parámetro de dispersión y debe ser estimado junto con los parámetros del modelo. Como en el uso de la función `CatDynFit` estamos realizando una optimización numérica para obtener un modelo estadístico, es necesario entregar los métodos numéricos a ser utilizados (argumento 'method') y el número máximo de iteraciones (argumento 'itnmax') para evaluar la función de optimización (que es el negativo del logaritmo de la función de verosimilitud). Siempre deben usarse los métodos 'spg' y 'CG' puesto que éstos han demostrado ser los más robustos para obtener convergencia numérica exitosa en los modelos de depleción generalizados de `CatDyn`. Otros métodos numéricos que pueden ayudar cuando 'spg' y 'CG' fallen son 'Nelder-Mead' y 'BFGS'.

Examinando el objeto que contiene los resultados del ajuste estadístico:

```
1 maya2000.OP.apn.fit
```

vemos que este objeto es una lista de tres componentes. Primero, Los datos, que incluye las propiedades y los datos originales enviados al ajuste estadístico. Segundo, los valores iniciales para los parámetros enviados al ajuste estadístico. Tercero, el modelo con sus resultados. El tercer componente, se divide en tantos elementos como métodos de optimización numérica hayan sido entregados. En este caso, dos, correspondientes a los métodos 'spg' y

'CG'. Examinando el componente 'spg',

```
1 maya2000.OP.apn.fit $Model$spg '
```

vemos que los resultados se dividen en

- 'Type' que es el tipo de modelo ajustado y que en este caso de depleción pura es 0 pues hay 0 entradas (y 0 salidas) de abundancia;
- 'Dates' que es el vector de fechas;
- 'Distr' que es la distribución de probabilidad de los datos y por lo tanto la función de verosimilitud;
- 'converg' que es un reporte del método numérico y que vale 0 cuando todo está bien y puede valer números positivos para distintos incidentes;
- 'kkt' (Karush–Kuhn–Tucker conditions) son reportes del método de optimización respecto de la calidad de la optimización en el primer y en el segundo orden tal que cuando ambos son TRUE se puede concluir que la optimización tuvo un resultados perfectamente adecuado;
- 'AIC' es el Akaike Information Criterion;
- 'bt.par' es el vector de estimados de máxima verosimilitud retro-transformados desde la escala logarítmica en que fueron optimizados;
- 'bt.num.grads' es el vector de gradiente numéricos al final de la optimización;
- 'bt.stdev' es el vector de errores estándar de los estimados de máxima verosimilitud; y
- 'Cor' es la matriz de correlaciones pareadas entre los estimados de máxima verosimilitud.

Para obtener predicciones del modelo estadístico utilizamos la función **CatDynPred** y luego generamos una gráfica con las predicciones:

```

1 maya2000.OP.apn.pred.spg <- CatDynPred(x=maya2000.OP.apn.fit ,
    method="spg") ### 32 43
2 plot(x=maya2000.OP.apn.pred.spg ,
3     leg.pos="topleft" ,
4     Biom.tstep=1 ,
5     Biom.xpos=0.75 ,
6     Biom.ypos=0.8 ,
7     Cat.tstep=dim(maya2000$Data$alijos)[1] ,
8     Cat.xpos=0.75 ,
9     Cat.ypos=0.7)

```

La función `CatDynPred` sólo tiene dos argumentos, el primero es el nombre del objeto que contiene los resultados estadísticos, producidos anteriormente con la función `CatDynFit`, y el segundo es el método numérico particular de entre aquellos que fueron usados en el llamado a la función `CatDynFit`. Es decir que con dos métodos en el llamado a la función `CatDynFit`, 'spg', 'CG', es posible generar dos objetos de predicciones con la función `CatDynPred`.

En el guión `Yucoct.Maya.CatDyn.IAGD.2000.R` se genera el objeto de predicciones y su gráfica primero para el método 'spg' y luego para el método 'CG'.

El objetivo de ajustar estadísticamente modelos de depleción pura es obtener nuevas pistas sobre buenas semanas candidatas de ser semanas de entradas (o salidas) de abundancia y/o confirmar los resultados obtenidos con el 'catch spike'. En la Fig. 10, panel inferior central, se observa que las semanas 39 y 43 aparecen como residuos positivos altos, y tomamos nota de este resultado.

No tenemos certeza que los datos de captura en número son generados por un proceso aleatorio con distribución normal. Podría ser otro tipo de proceso, por ejemplo un proceso que genere datos de distribución lognormal. En `CatDyn` están implementadas dos versiones que fijan una distribución lognormal de los datos, 'apnormal' y 'lognormal', en que la primera opción excluye el parámetro de dispersión mientras que la segunda opción lo incluye. En esta aplicación también hemos ajustado el modelo de depleción pura con la distribución lognormal de los datos y al crear el objeto de predicciones y graficarlos vemos que nuevamente las semanas 39 y 43 aparecen como candidatas a entradas de abundancia durante la temporada de pesca.

Este análisis preliminar y exploratorio en la secciones 4.1 y 4.2 de los datos con el estadístico no-paramétrico 'catch spike' y con modelos estadísticos de depleción pura nos ha entregado cuatro opciones para semana del reclutamiento (32, 33, 39, y 43) y los dos enfoques coinciden en identificar a la semana 43 como candidata. Aparte de esto, los modelos de depleción pura no tienen utilidad en la inmensa mayoría de los casos.

Figura 10: Modelo estadístico de depleción pura con distribución normal de los datos de la temporada 2000 de pulpo maya. La biomasa indicada es la biomasa en la última semana de la temporada de pesca y la captura indicada es la captura total durante la temporada de pesca.

Fleet = alijos, Perturbations = 0, Distribution = Aplnormal, Numerical algorithm = spg

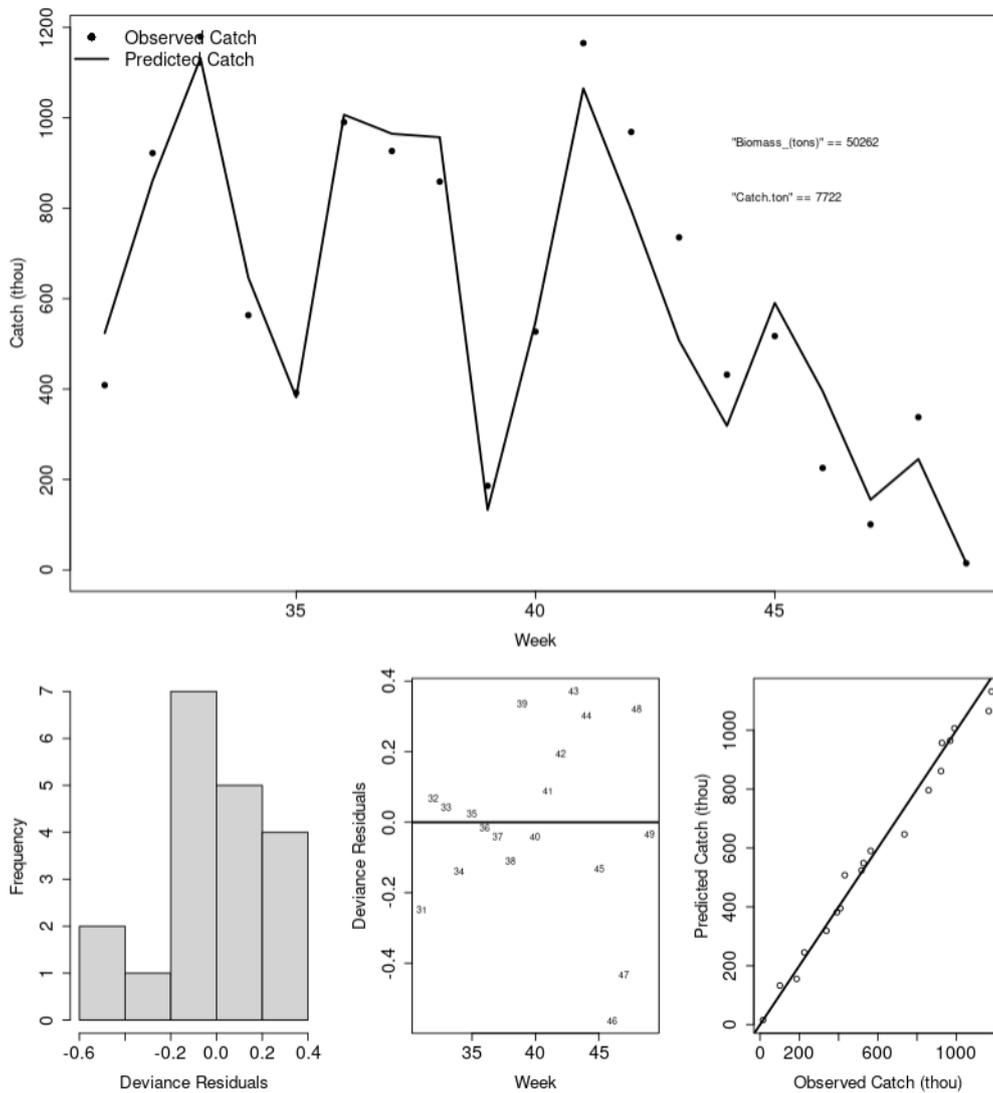
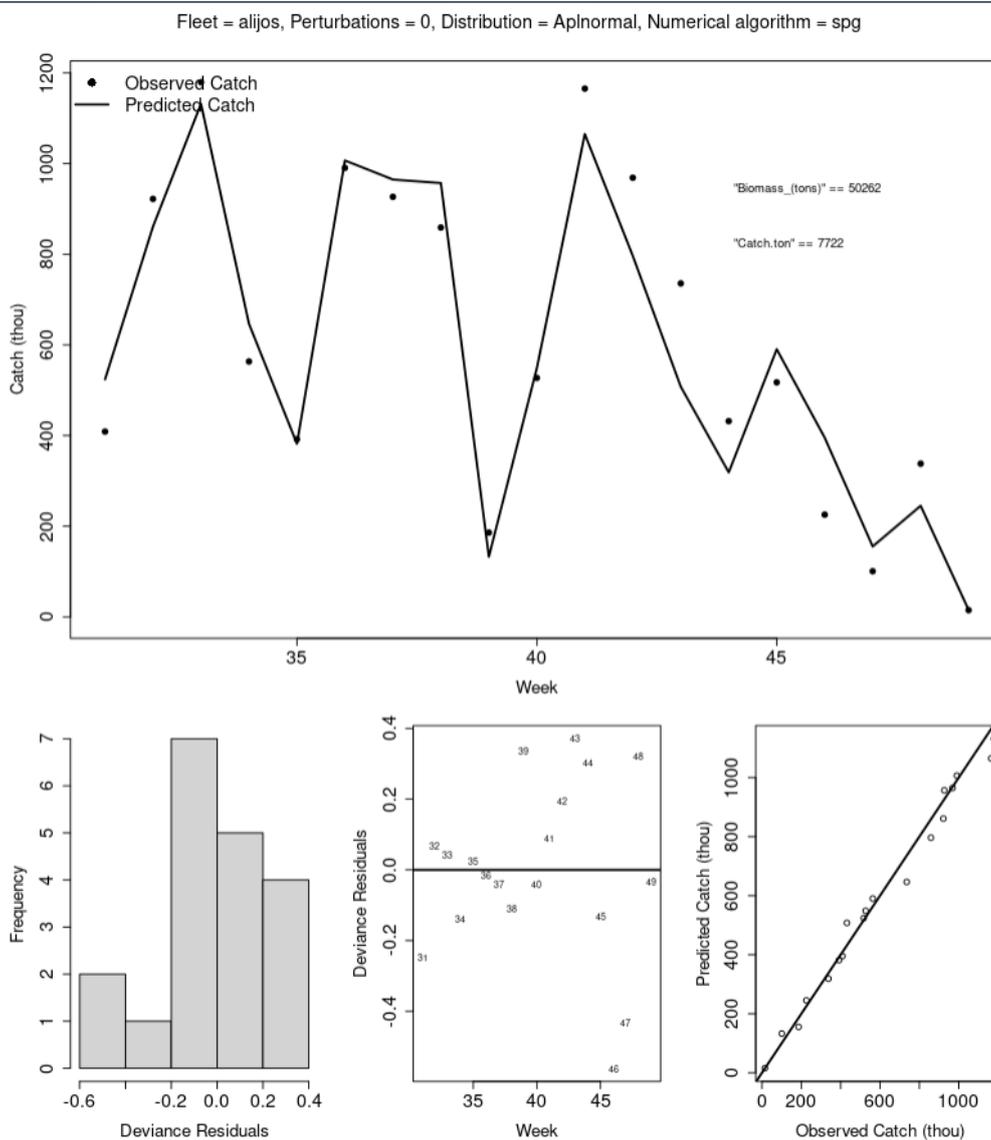


Figura 11: Modelo estadístico de depleción pura con distribución lognormal de los datos de la temporada 2000 de pulpo maya. La biomasa indicada es la biomasa en la última semana de la temporada de pesca y la captura indicada es la captura total durante la temporada de pesca.



4.1.4. Ajuste de modelos de depleción con entradas de abundancia

Estos son modelos que ya pueden tener utilidad científica y en la gestión de los recursos pesqueros. La estrategia va a ser comenzar con modelos simples, que tienen una entrada de abundancia, y seguir con modelos más complejos que pueden tener, dos o más entradas de abundancia en semanas diferentes de la temporada. En el caso de los pulpos de México hemos llegado a obtener modelos con hasta cinco entradas de abundancia en una temporada. Esta estrategia se aplica ajustando primero un modelo con una entrada de abundancia utilizando la semana candidata que ocurre primero a través de la serie de tiempo semanal. Otra estrategia factible es comenzar con la semana que más resalta en el 'catch spike' y/o en el modelo de depleción pura. Enseguida se van agregando las otra semanas candidatas hasta que los residuos del último modelo ajustado ya no tengan ninguna semana que resalte con un valor muy positivo.

En la temporada que estamos analizando la primera semana que se destaca es la semana 32 y luego la semana 43 alcanza los más altos valores positivos del 'catch spike' y de residuos del modelo de depleción pura, así que primero ajustaremos un modelo de una entrada de abundancia en la semana 32 y luego un segundo modelo con una entrada de abundancia en la semana 32 y otra entrada de abundancia en la semana 43. Utilizaremos cuatro modelos de verosimilitud (apnormal, aplnormal, normal and lognormal) y dos métodos numéricos.

```
1 P1 <- 32 # 32 39 43
2 dates.maya2000.P1.1 <- c(head(maya2000$Data[[1]]$time.step,1),
3                             P1,
4                             tail(maya2000$Data[[1]]$time.step,1))
5 #
6 M.ini <- 1.041223e-01
7 N0.ini <- 1e5
8 P1.ini <- 6e4
9 k.ini <- 9e-05
10 alpha.ini <- 1.5
11 beta.ini <- 0.8
12 pars.maya2000.P1.1.ini <- log(c(M.ini ,
13                                N0.ini ,
14                                P1.ini ,
15                                k.ini ,
16                                alpha.ini ,
17                                beta.ini))
18 #
19 maya2000.P1.1.apn.ini <- catdynexp(x=maya2000,
20                                   p=1,
21                                   par=pars.maya2000.P1.1.ini ,
```

```

22                                     dates=dates . maya2000 . P1 . 1 ,
23                                     distr="apnormal")
24 plot(x=maya2000 . P1 . 1 . apn . ini ,
25       leg . pos="topleft" ,
26       Biom . tstep =1 ,
27       Biom . xpos =0.75 ,
28       Biom . ypos =0.75 ,
29       Cat . tstep=dim(maya2000$Data$alijos) [1] ,
30       Cat . xpos =0.75 ,
31       Cat . ypos =0.65)

```

Ya hemos notado que debido a que el emparejamiento y desove en ambas especies ocurre antes de la temporada de pesca, durante el primer semestre del año según los artículos científicos que abordan este tópico, pues entonces no existen mecanismos aparentes para suponer que hay salidas de abundancia desde los stocks que están siendom explotados. Respecto de la entradas de abundancia, estas pueden deberse en general a dos mecanismos. Primero, hay una cohorte de pulpos recién nacidos antes de la temporada de pesca, que crecen hasta el tamaño buscado y retenido por los pescadores. Esta es una entrada de abundancia por reclutamiento por crecimiento. También puede haber entradas debido a la conducta de las flotas. Cuando las flotas comienzan a pescar en un área que en la semanas anteriores de la temporada no habían operado, esto para el modelo de depleción es una entrada de abundancia pues se agrega un nueva parte del stock a la explotación que antes no estaba presente en la serie de tiempo de capturas. Las entradas de abundancia por reclutamiento por crecimiento debieran ocurrir siempre alrededor de la misma epoca del año en todos los años de la serie de tiempo, pues es un proceso estacional inherente al stock, mientras que las entradas por movimientos de las flotas pueden ser mucho más variables en su ubicación a lo largo de la serie de tiempo de capturas.

Puesto que los valores iniciales ya ofrecen un buen punto de partida para la optimización, ajustamos esta variante haciendo una llamada a la función **CatDynFit**, que después de unos instantes termina de optimizar y procedemos a crear los dos objetos de predicciones con **CatDynPred**, uno para los resultados del método 'spg' y otro para los resultados del método 'CG':

```

1 maya2000 . P1 . 1 . apn . fit      <- CatDynFit(x=maya2000 ,
2                                     p=1 ,
3                                     par=pars . maya2000 . P1 . 1 . ini ,
4                                     dates=dates . maya2000 . P1 . 1 ,
5                                     distr="apnormal" ,
6                                     method=c("spg" , "CG") ,
7                                     itnmax=50000)

```

```

8 #
9 maya2000.P1.1.apn.pred.spg <- CatDynPred(x=maya2000.P1.1.apn.fit ,
    method="spg") ###01 43
10 plot(x=maya2000.P1.1.apn.pred.spg ,
11     leg.pos="topleft" ,
12     Biom.tstep=1 ,
13     Biom.xpos=0.75 ,
14     Biom.ypos=0.8 ,
15     Cat.tstep=dim(maya2000$Data$alijos)[1] ,
16     Cat.xpos=0.75 ,
17     Cat.ypos=0.7)
18 #
19 maya2000.P1.1.apn.pred.CG <- CatDynPred(x=maya2000.P1.1.apn.fit ,
    method="CG") ###02 43
20 plot(x=maya2000.P1.1.apn.pred.CG ,
21     leg.pos="topleft" ,
22     Biom.tstep=1 ,
23     Biom.xpos=0.75 ,
24     Biom.ypos=0.8 ,
25     Cat.tstep=dim(maya2000$Data$alijos)[1] ,
26     Cat.xpos=0.75 ,
27     Cat.ypos=0.7)

```

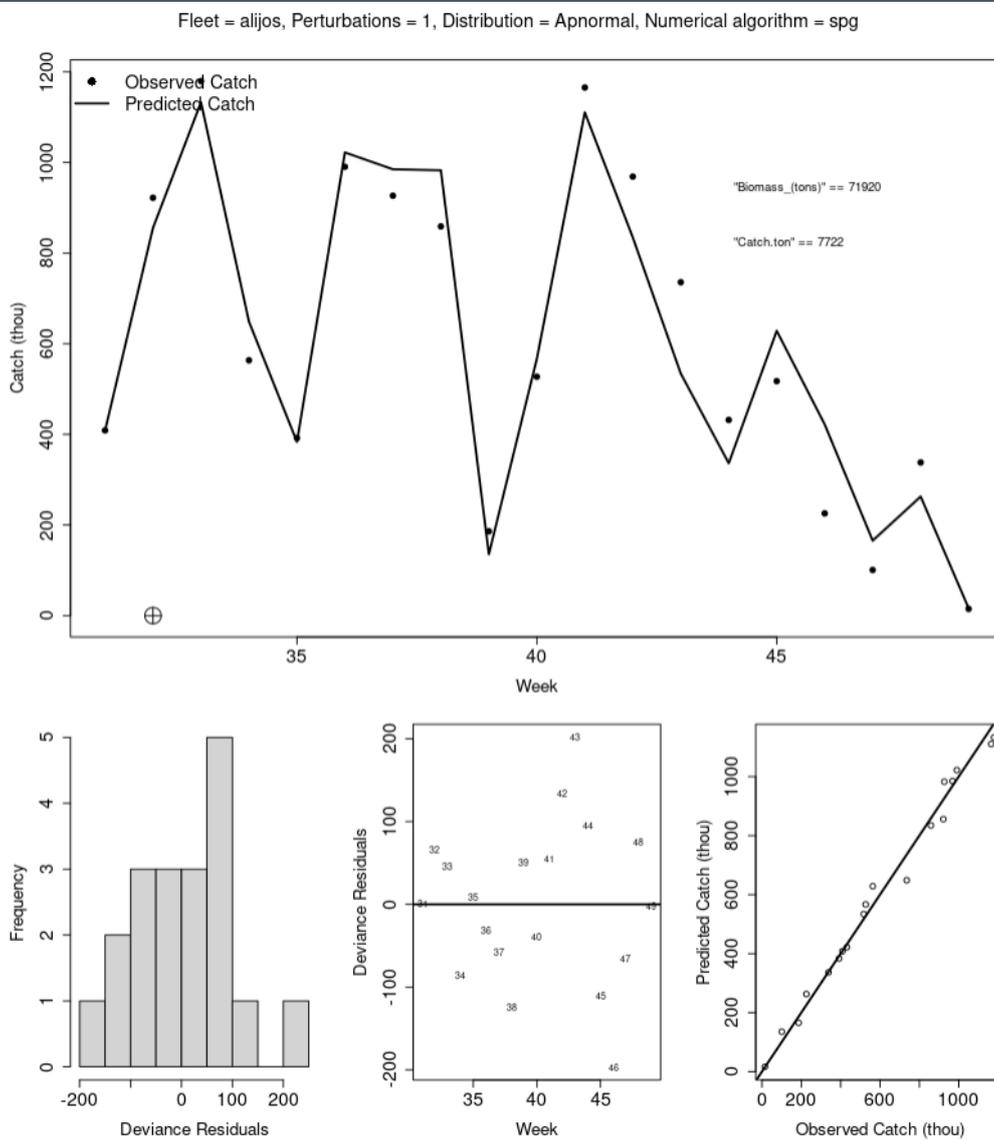
Asimismo ponemos una marca junto a cada objeto de predicción puesto que luego vamos a reunir a todos las variantes para elegir la mejor mediante técnicas de selección de modelos. Estos dos resultados son las variantes ###01 y ###02. Además marcamos la(s) semana(s) que aún muestre(n) residuos positivos altos, que es la semana 43 (Fig. 12) ya marcada desde previas exploraciones.

Como el valor para el argumento 'Biom.tstep' es 1 en el llamado a la función genérica `plot` sobre un objeto generado por la función `CatDynPred`, entonces la biomasa que aparece indicada en el plot es la biomasa en la última semana de la temporada de pesca, es decir la biomasa de escape. Esta biomasa predicha por esta variante va a ir bajando a medida que agreguemos más entradas de abundancia debido al carácter recursivo del modelo de depleción. En el caso del valor de captura que aparece en el plot, puesto que el valor del argumento 'Cat.tstep' es

```
1 dim(maya2000$Data$alijos)[1]
```

que corresponde a la extensión completa de la temporada (19 semanas) entonces el valor de captura que se muestra es la suma de las capturas en peso de toda la temporada.

Figura 12: Modelo estadístico de depleción con una entrada de abundancia en la semana 32 y con distribución lognormal de los datos de la temporada 2000 de pulpo maya. La biomasa indicada es la biomasa en la última semana de la temporada de pesca y la captura indicada es la captura total durante la temporada de pesca.



En el guión `Yucoct.Maya.CatDyn.IAGD.2000.R` el ajuste estadístico con `CatDynFit` se repite con 3 nuevas fórmulas para la función de verosimilitud, los mismos dos métodos numéricos de optimización, y dos hipótesis para el número de entradas de abundancia, y todos los casos producen convergencia exitosa, dando lugar a 16 variantes para la selección de modelos. Ocho de estas variantes son modelos con una entrada de abundancia (semana 32), y ocho son modelos con dos entradas de abundancia (semanas 32 y 43).

4.1.5. Selección de la mejor variante

Aplicamos criterios biológicos, estadísticos y numéricos para seleccionar la mejor variante. Usualmente esta combinación rinde resultados inambiguos aunque en casos extremos puede ser necesario seleccionar la variante menos mala.

El criterio numérico es que los gradientes numéricos sean todos menores que 1 en valor absoluto. Este criterio suele reducir el conjunto de variantes. Otro criterio numérico es la existencia de errores estándar para todos los parámetros. Anomalías de la curva de log-verosimilitud alrededor del máximo pueden causar que algunos errores estándar sean 'NaN', 'not a number'. Una variante con menos 'NaN' o sin ellos es mejor que una variante equivalente en otros sentidos que tiene 'NaN' entre sus errores estándar.

El criterio estadístico es que las correlaciones entre los estimados sean más cercanas a cero y los estimados tengan el menor coeficiente de variación. El primer criterio busca elegir variantes en que todos los parámetros jueguen un rol necesario y el segundo criterio busca elegir variantes en que la precisión estadística sea la mejor, con más bajos coeficientes de variación.

El criterio biológico es que la mortalidad natural esté cercana a la predicción de la compilación de Hoenig y que la biomasa de escape no sea un valor excesivo, por ejemplo 100 veces mayor que la captura total de la temporada. En algunos casos no es posible obtener una variante con una estimación de mortalidad natural en el orden de magnitud de la predicción de la compilación de Hoenig. En esos casos siempre es demasiado baja y su coeficiente de variación es muy alto, de varios cientos. Puede ser necesario elegir una variante así si otras variantes son aún peores.

Lo primero es crear una lista con todas las variantes retenidas para examen:

```
1 x <- list(maya2000.P1.1.apn.fit ,      ### 01
2          maya2000.P1.1.apn.fit ,      ### 02
3          maya2000.P1.1.apln.fit ,     ### 03
4          maya2000.P1.1.apln.fit ,     ### 04
5          maya2000.P1.1.n.fit ,        ### 05
```

```

6      maya2000.P1.1.n.fit ,      ### 06
7      maya2000.P1.1.ln.fit ,    ### 07
8      maya2000.P1.1.ln.fit ,    ### 08
9      maya2000.P2.1.apn.fit ,   ### 09
10     maya2000.P2.1.apn.fit ,   ### 10
11     maya2000.P2.1.apln.fit ,  ### 11
12     maya2000.P2.1.apln.fit ,  ### 12
13     maya2000.P2.1.n.fit ,     ### 13
14     maya2000.P2.1.n.fit ,     ### 14
15     maya2000.P2.1.ln.fit ,    ### 15
16     maya2000.P2.1.ln.fit )    ### 16
17 method <- rep(c("spg", "CG"), 8)
18 maya2000.sum.0 <- CatDynSum(x=x, season="2000", method=method)
19 maya2000.sum.0[maya2000.sum.0$Distribution=="apnormal",]
20 maya2000.sum.0[maya2000.sum.0$Distribution=="aplnormal",]
21 maya2000.sum.0[maya2000.sum.0$Distribution=="normal",]
22 maya2000.sum.0[maya2000.sum.0$Distribution=="lognormal",]
23 maya2000.sum.1 <- maya2000.sum.0[maya2000.sum.0$Max.Abs.Grads. <
  1,]

```

Vamos a usar la función `CatDynSum` para crear una tabla con casi todos los resultados de las variantes, los 16 objetos que resultan de la aplicación de `CatDynFit`. Lo primero es crear una lista, 'x', con todos los nombres de las variantes, repitiendo cada nombre tantas veces como métodos numéricos de optimización hayan sido usados, dos en este ejemplo. Otro vector, 'method', debe contener los nombres de los métodos numéricos. La llamada a la función `CatDynSum` con esta información resulta en la tabla de la Fig. 13

Esta tabla tiene una fila por cada variante retenida y 25 columnas. Las primeras 6 columnas identifican a la variante, las siguientes 18 columnas tiene resultados, y la última columna esta reservada para marcar la variante de mayor calidad, la seleccionada como mejor modelo de la presente temporada.

El primer criterio de selección es eliminar todas las variantes en que el gradiente numérico máximo (en valor absoluto) es mayor que 1. Esto sólo elimina a la variante `###09`. Respecto de la mortalidad natural, todas las variantes estiman un valor razonable dada la longevidad de los pulpos así que este criterio no sirve para reducir la lista.

Examinando los AIC de las variantes comparables, aquellas que fueron ajustadas con las mismas fórmulas de verosimilitud:

```

1 maya2000.sum.0[maya2000.sum.0$Distribution=="apnormal",]
2 maya2000.sum.0[maya2000.sum.0$Distribution=="aplnormal",]
3 maya2000.sum.0[maya2000.sum.0$Distribution=="normal",]
4 maya2000.sum.0[maya2000.sum.0$Distribution=="lognormal",]

```

Figura 13: Tabla comparativa de resultados de variantes obtenida con la función **CatDynSum**.

Fleet	Effort	Season	Model	Distribution	Method	AIC	Max.Abs.Grads.	M	SE.M	N0	SE.N0	P1.alijos	SE.P1.alijos	
1	alijos	effdays	2000	1P	apnormal	spg	216.66432	0.010511370	0.01764766	0.007834949	104477.3	3458.349	30646.03	15312.199
2	alijos	effdays	2000	1P	apnormal	CG	216.69970	0.146767279	0.02099963	0.008795269	158761.4	7699.977	56361.56	25830.618
3	alijos	effdays	2000	1P	aplnormal	spg	12.61107	0.005196910	0.03178402	0.008847824	115082.5	NaN	48521.09	22231.923
4	alijos	effdays	2000	1P	aplnormal	CG	12.61408	0.005782528	0.03244080	0.009089547	135924.5	NaN	56979.00	25288.142
5	alijos	effdays	2000	1P	normal	spg	240.71670	0.011949816	0.01746969	NaN	102388.4	NaN	29980.36	NaN
6	alijos	effdays	2000	1P	normal	CG	240.75784	0.147838770	0.02118018	NaN	158037.4	396606.648	56443.35	NaN
7	alijos	effdays	2000	1P	lognormal	spg	243.85949	0.004133388	0.03173324	0.138589827	113403	234507.278	47865.02	392057.897
8	alijos	effdays	2000	1P	lognormal	CG	243.86323	0.009983281	0.03248185	0.147886034	135975.2	585849.013	57169.22	568137.047
9	alijos	effdays	2000	2P	apnormal	spg	216.32237	1.640179761	0.01937594	0.008371522	41361.8	1180.876	11819.11	6428.795
10	alijos	effdays	2000	2P	apnormal	CG	216.51992	0.192789449	0.03274717	NaN	105768.8	NaN	22952.69	NaN
11	alijos	effdays	2000	2P	aplnormal	spg	14.43178	0.005228302	0.03707624	0.017682989	75016.93	2470.415	37015.58	15220.677
12	alijos	effdays	2000	2P	aplnormal	CG	14.44617	0.006736425	0.03869839	0.017788013	96072.6	2724.206	47043.04	18988.405
13	alijos	effdays	2000	2P	normal	spg	240.11675	0.115971083	0.02086011	NaN	44617.53	47620.533	12864.47	NaN
14	alijos	effdays	2000	2P	normal	CG	240.32542	0.263706074	0.03303645	NaN	105471.1	154689.782	22988.31	NaN
15	alijos	effdays	2000	2P	lognormal	spg	245.66489	0.010462631	0.03793933	0.169492681	86078.07	186400.123	42258.37	341222.579
16	alijos	effdays	2000	2P	lognormal	CG	245.66886	0.007512081	0.03870019	0.178239141	96098.81	259372.001	47075.46	414490.434

vemos que en este caso el AIC no es decisivo, pues debe haber una distancia mayor que 2 entre variantes para seleccionar una, lo que no ocurre dentro de ninguno de los cuatro grupos.

Revisamos las correlaciones entre los estimados utilizando la función **CatDynCor**, que produce un histograma de frecuencias de las correlaciones del triángulo superior, diagonal excluida, de la matriz de correlación:

```

1 CatDynCor(x=x[as.numeric(row.names(maya2000.sum.1))], #
2   ##11 ##12
3   ttl=paste(maya2000.sum.1$Model,
4             substring(maya2000.sum.1$Distribution, 1,4),
5             maya2000.sum.1$Method,
6             as.numeric(row.names(maya2000.sum.1))),
7   method=method[as.numeric(row.names(maya2000.sum.1))],
   arr=c(3,5))

```

Las variantes que estamos considerando tienen entre 6 y 7 parámetros, así que pueden generar $(6 * 6 - 6)/2 = 15$ correlaciones o $(7 * 7 - 7)/2 = 21$ correlaciones pareadas. Sus histogramas se observan en la Fig. 14. Claramente las variantes ###11 y ###12 destacan como las mejores correlaciones. Entonces es posible reducir el conjunto de variantes a sólo 2, las variantes ###11 y ###12:

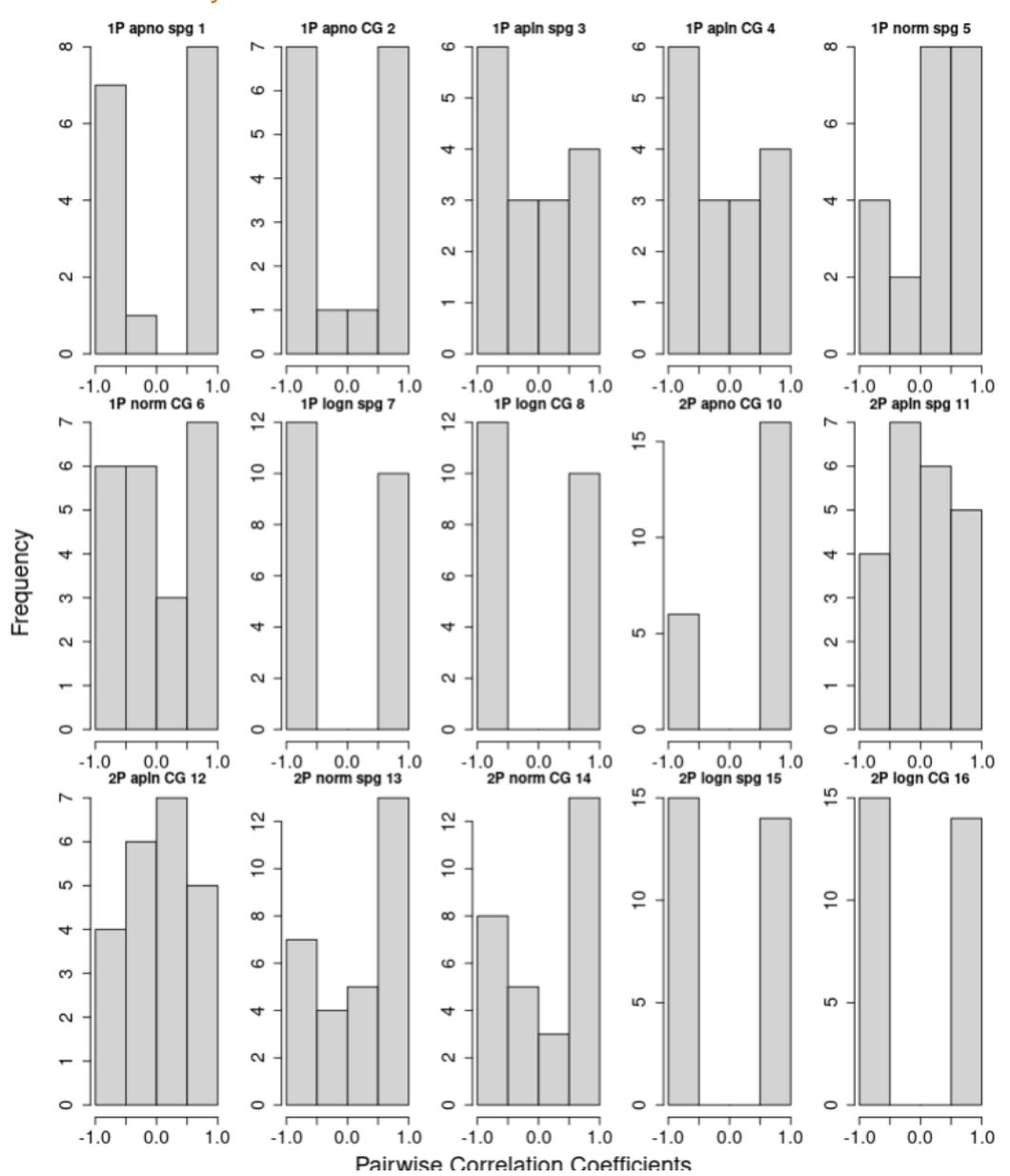
```
1 maya2000.sum.2 <- maya2000.sum.0[row.names(maya2000.sum.0) %in% c(
  "11", "12") ,]
```

Para decidir entre estas dos variantes, utilizamos la función `CatDynPar`, que va a producir una tabla con los estimados y sus coeficientes de variación. Las variantes deseables tienen todos sus coeficientes de variación (es decir, todos los errores estándar pudieron ser calculados al final de la optimización numérica, ninguno de ellos fue reportado como 'NaN') y son los más bajos. La tabla producida por `CatDynPar` también está diseñada para ser usada casi tal cual es en reportes, y se consigue con sólo dos argumentos, el objeto producido por `CatDynFit` y el método numérico:

```
1 CatDynPar(x=maya2000.P2.1.apln.fit, method="spg") ###11
2 #           Parameter           Timing    Estimates CVpCent
3 #           M.1/week                0.03613    48.2
4 #           N0.thou                  65802.00588  3.6
5 #Recruitment.thou.Wave1 2000-08-06 2000-08-12 32648.24302  41.4
6 #Recruitment.thou.Wave2 2000-10-22 2000-10-28 7010.02055  192.2
7 #           k.1/effdays             0.00016    4.4
8 #           alpha                    1.07597    6.9
9 #           beta                     0.90905    4.2
10 CatDynPar(x=maya2000.P2.1.apln.fit, method="CG") ###12
11 #           Parameter           Timing    Estimates CVpCent
12 #           M.1/week                0.03865    46.1
13 #           N0.thou                  96279.97261  2.9
14 #Recruitment.thou.Wave1 2000-08-06 2000-08-12 47138.93296  40.4
15 #Recruitment.thou.Wave2 2000-10-22 2000-10-28 10002.23967  193.7
16 #           k.1/effdays             0.00010    4.1
17 #           alpha                    1.07585    6.9
18 #           beta                     0.91867    4.1
```

Las dos variantes rinden resultados muy similares de estimados y coeficientes de variación. La variante ###12 es marginalmente mejor. Es importante que ambas tienen bajos coeficiente de variación de la mortalidad natural y de dos de los tres parámetros de abundancia. Sólo la segunda entrada

Figura 14: Histograma de correlaciones entre estimados de las variantes con la función *CatDynCor*.



de abundancia, que es bastante menor en tamaño, tiene un coeficiente de variación mayor que 100. Examinando gráficamente las predicciones:

```
1 plot(x=maya2000.P2.1.apln.pred.spg,      ###11
2     leg.pos="topright",
3     Biom.tstep=1,
4     Cat.tstep=dim(maya2000$Data$alijos)[1],
5     Biom.xpos=0.9,
6     Biom.ypos=0.8,
7     Cat.xpos=0.9,
8     Cat.ypos=0.75)
9 plot(x=maya2000.P2.1.apln.pred.CG,      ###12
10    leg.pos="topright",
11    Biom.tstep=1,
12    Cat.tstep=dim(maya2000$Data$alijos)[1],
13    Biom.xpos=0.9,
14    Biom.ypos=0.8,
15    Cat.xpos=0.9,
16    Cat.ypos=0.75)
```

observamos que ambas variantes estiman biomazas de escape realistas, pero la variante ###12 presenta ligeramente mejor histograma de residuos. Tomando esto en cuenta y también que la variante ###12 tiene ligeramente mejores CV, la decisión es seleccionar la variante ###12 como la mejor para los datos de esta temporada:

```
1 maya2000.sum.0$Sel.Model[12] <- 1
```

4.2. Temporada 2017 de *O. americanus*

Abra el guión [Yucoct.Amer.CatDyn.IAGD.2017.R](#), cambiando la dirección del disco duro, trayendo el resultado del guión [AvisosDeArribos.Data.R](#) que es su tabla de datos de captura, esfuerzo y pesos medios, y cargando la librería [CatDyn](#). Esta temporada contrasta con la temporada 2000 de *O. maya* de la sección 4.1 en que vamos a tener que ajustar más variantes debido a una más intensa dinámica de entradas de abundancia.

4.2.1. Objeto de datos para [CatDyn](#)

Examinando el objeto de datos 'cateff2017aner' vemos que aunque hubo un reporte de capturas en la semana 5 ya no hay más capturas hasta la

semana 30, que es cuando comienza la temporada oficial, así que para crear el objeto de datos para ajustar modelos de depleción en **CatDyn** hacemos:

```
1 amer2017 <- as.CatDynData(x=cateff2017amer[cateff2017amer$Week
  >29,],
2                               step="week",
3                               fleet.name="alijos",
4                               coleff=3,
5                               colcat=2,
6                               colmbw=7,
7                               unitseff="effdays",
8                               unitscat="kg",
9                               unitsmbw="g",
10                              nmult="thou",
11                              season.dates=c("2017-07-27", "2017-12-29"
                                             ))
```

4.2.2. Semanas candidatas a entradas de abundancia: estadístico 'catch spike'

Graficando los datos con la función genérica **plot** (note que en comparación con el caso anterior en la sección 4.1.2, aquí hemos agregado un tercer valor para el argumento 'offset' para incluir el número de la semana arriba del punto en el gráfico del 'catch spike'):

```
1 plot(x=amer2017, mark=TRUE, offset=c(9,10,5), hem="N") # 36 40 43 47
```

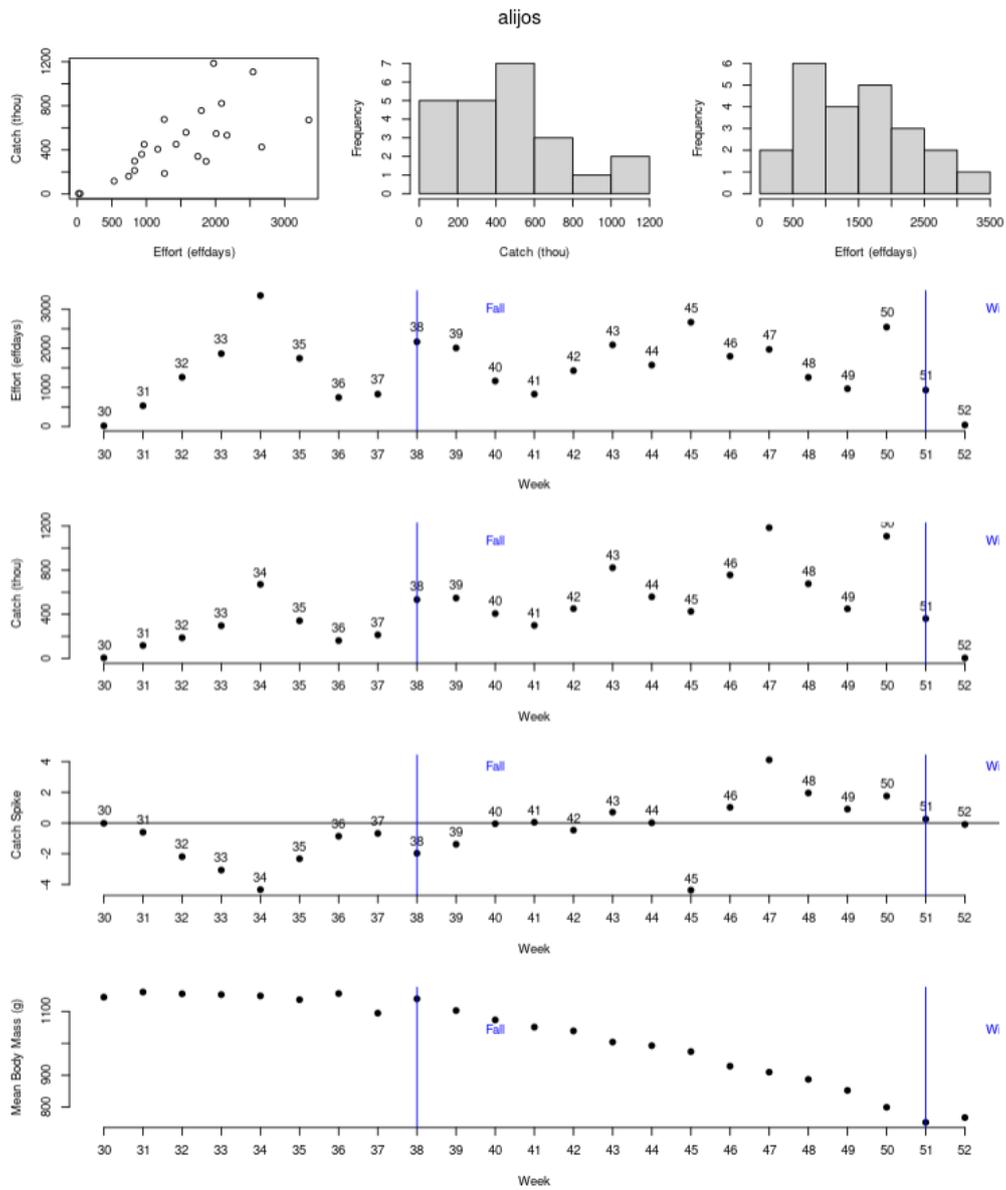
observamos que las semanas 36 y 40 están al final de un período de incrementos semanales del 'catch spike' y que las semanas 43 y 47 tienen valores positivos altos del 'catch spike' por sí solas (Fig. 15).

Otra observación relevante es que en el primer panel de la izquierda y arriba (Fig. 15) la relación entre esfuerzos de pesca semanales como causa y capturas semanales como efecto, es mucho más dispersa que en el caso de la temporada 2000 de *O. maya* (Fig. 8). Este es uno de los elementos que hacen más laborioso el análisis de los datos de esta temporada.

4.2.3. Semanas candidatas a entradas de abundancia: modelo de depleción pura

El segundo método para determinar semanas candidatas a ser semanas de entradas de abundancia es la aplicación de modelos de depleción pura.

Figura 15: Datos crudos para los modelos de depleción de la temporada 2017 de *O. americanus*.



En el guión [Yucoct.Amer.CatDyn.IAGD.2017.R](#) cuatro variantes (dos fórmulas de verosimilitud y dos métodos numéricos de optimización) convergen exitosamente e identifican a las semanas 43 y 47 (tal como ocurrió con el 'catch spike') como candidatas a semanas de entradas de abundancia debido a residuos positivos más altos.

4.2.4. Ajuste de modelos de depleción con entradas de abundancia

En el guión [Yucoct.Amer.CatDyn.IAGD.2017.R](#) la búsqueda de variantes que mejor explican los datos consideras a cuatro semanas como potenciales semanas de entradas de abundancia, y estas son las semanas 36, 40, 43 y 47. La primera hipótesis es que hay una entrada de abundancia en la semana 43, elegida porque aunque hay dos semanas candidatas anteriores (36 y 40) identificadas por el 'catch spike', la semana 43 es la más temprana cuyo residuo sobresale en los modelos de depleción pura (Fig. 16).

En esta gráfica también podemos apreciar una de las características de los modelos de depleción generalizados que están sub-parametrizados, es decir que tienen menos parámetros de las que deben tener. Esta gráfica corresponde a un modelo de depleción pura y por lo tanto está sub-parametrizado por carece de entradas de abundancia durante la temporada de pesca y la característica que hay que resaltar es que presenta una estimación de la biomasa de escape que es claramente sobre-estimada ($> 10^6$ toneladas). A medida que se vayan agregando entradas de abundancia, que es lo mismo que decir que el modelo va adquiriendo su real dotación de parámetros, la biomasa de escape va a ir bajando hacia valores realistas.

Una segunda hipótesis con entradas de abundancia es que hay dos entradas de abundancia, en las semanas 43 y 47, que son nuevamente elegidas porque de entre aquellas indicadas por el 'catch spike', son aquellas cuyos residuos sobresalen más en los modelos de depleción pura (Fig. 16).

Tanto las variantes con una entrada como las variantes con dos entradas son ajustadas con dos fórmulas de verosimilitud ('apnormal' y 'aplnormal') y dos métodos numéricos ('spg' y 'CG'), y todas convergen exitosamente, dando lugar a ocho variantes para la etapa de selección de modelos.

Una tercera hipótesis plantea la existencia de tres entradas de abundancia, en las semanas 38, 43 y 47. La semana 36, que estaba en la lista de candidatas debido al 'catch spike', no aparece como un residuo positivo en las variantes con una o dos entradas mientras que la semana 38 si lo hace y por eso reemplaza a la semana 36 como candidata a primera semana de entrada de abundancia en la serie semanal de capturas. Este tipo de consideraciones muestra que además de determinar semanas candidatas a ser semanas de entradas de abundancia por anticipado con el 'catch spike' y los

modelos de depleción pura, es adecuado ir actualizando la lista de semanas candidatas observando los residuos de los modelos con menos entradas de abundancia:

```
1 P1 <- 38
2 P2 <- 43
3 P3 <- 47
4 dates. amer2017. P3.1 <- c(head(amer2017$Data [[1]] $time. step ,1) ,
5                             P1 ,
6                             P2 ,
7                             P3 ,
8                             tail (amer2017$Data [[1]] $time. step ,1) )
```

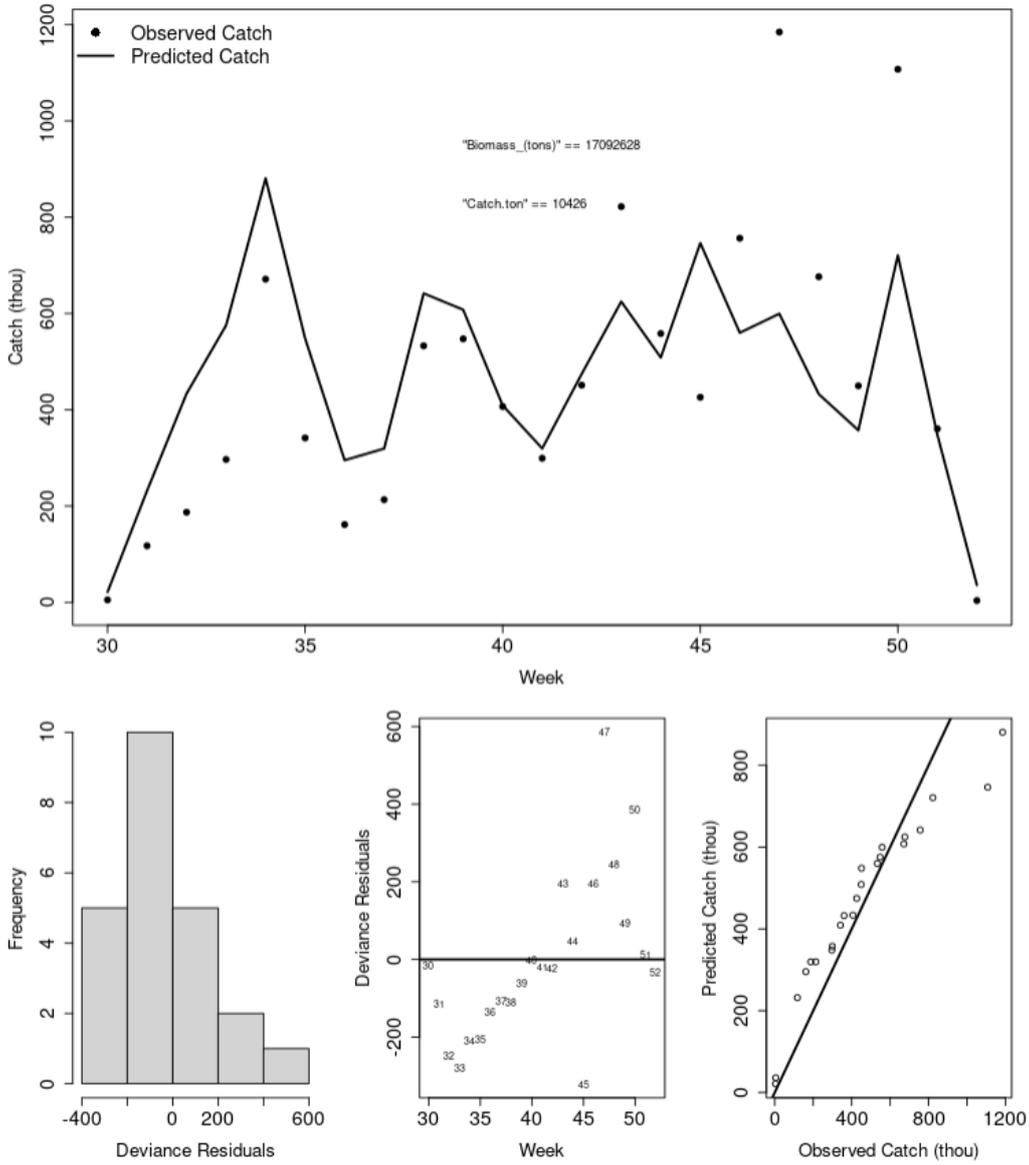
A medida que vamos obteniendo más variantes que convergen exitosamente, vamos obteniendo estimado de parametros que sirven como valores iniciales en variantes más complejas. Por ejemplo el ajuste de la variante con dos entradas estima k , α y β en valores aproximados de 0.0009, 0.8 y 0.6 respectivamente. Asimismo, cuando agregamos una entrada de abundancia para ajustar una variante más compleja, es útil quitar una parte de la abundancia total que se obtiene de la suma de N_0 y las entradas de abundancia de la variante anterior, para hacer lugar para la nueva entrada. Así que para el ajuste de la variante con tres entradas de abundancia usamos los valores iniciales:

```
1 M. ini <- 0.03
2 N0. ini <- 3e4
3 P1. ini <- 1e4
4 P2. ini <- 2e4
5 P3. ini <- 1e4
6 k. ini <- 1e-3
7 alpha. ini <- 0.8
8 beta. ini <- 0.7
9 pars. amer2017. P3.1. ini <- log(c(M. ini ,
10                                N0. ini ,
11                                P1. ini ,
12                                P2. ini ,
13                                P3. ini ,
14                                k. ini ,
15                                alpha. ini ,
16                                beta. ini))
```

Existiendo múltiples semanas candidatas y ajustando variantes de tres entradas de abundancia, es conveniente probar variaciones de conjuntos de tres semanas para tener diversas variantes de tres entradas. En el script

Figura 16: Modelo de depleción pura aplicado a los datos de 2017 de *O. americanus*.

Fleet = alijos, Perturbations = 0, Distribution = Apsnormal, Numerical algorithm = spg



`Yucoct.Amer.CatDyn.IAGD.2017.R` tenemos tres de estas variantes, con la primera teniendo entradas de abundancia en las semanas 38, 43 y 47, la segunda en las semanas 43, 47 y 50, y la tercera en las semanas 40, 43 y 47. La tercera hipótesis con tres entradas fue ajustada con un tercer método numérico, además de 'spg' y 'CG', que también está entre los recomendados para los modelos de depleción generalizados, el método 'Nelder-Mead', aunque este método suele sobre-estimar la abundancia del stock. En este caso sin embargo, fue incluido porque los métodos estándar reportaron biomasa de escape bastante bajas. El llamado a un tercer método de optimización numérica es muy simple, gracias a la dependencia de `CatDyn` respecto de la librería `optimx` que permite enviar múltiples métodos de optimización para atacar un único problema de optimización:

```

1 amer2017.P3.3.apn.fit <- CatDynFit(x=amer2017,
2                                 p=3,
3                                 par=pars.amer2017.P3.3.ini,
4                                 dates=dates.amer2017.P3.3,
5                                 distr="apnormal",
6                                 method=c("spg", "CG", "Nelder
7                                 -Mead"),
                                 itnmax=50000)

```

Puesto que la mayor parte de las variantes de una a tres entradas de abundancia (21 variantes) estimaron biomasa de escape muy altas ($> 10^5$) y mortalidad natural muy baja ($< 10^{-3}$) y que las biomasa estimadas bajan a medida que se agregan entradas, fue necesario explorar variantes con cuatro entradas. Estas variantes postularon que las semanas de entradas de abundancia fueron las 40, 43, 47 y 50, y 38, 43, 47 y 50. En el caso de la segunda hipótesis de cuatro entradas la variante con verosimilitud 'apnormal' y métodos de optimización 'spg' y 'CG' no convergieron:

```

1 amer2017.P4.2.apn.fit <- CatDynFit(x=amer2017,
2                                 p=4,
3                                 par=pars.amer2017.P4.2.ini
4                                 ,
5                                 dates=dates.amer2017.P4.2,
6                                 distr="apnormal",
7                                 method=c("spg", "CG"),
8                                 itnmax=50000)
9 Error in qr.default(if (d[1L] < d[2L]) t(z) else z) :
  NA/NaN/Inf en llamada a una función externa (arg 1)

```

Este mensaje de error viene del código de los métodos numéricos y puede provenir de 'spg', de 'CG' o de ambos. Para resolver si sólo es uno de los métodos el que colapsa se debe enviar cada uno por separado:

```

1 amer2017.P4.2.apn.fit      <- CatDynFit(x=amer2017,
2                               p=4,
3                               par=pars.amer2017.P4.2.ini ,
4                               dates=dates.amer2017.P4.2,
5                               distr="apnormal" ,
6                               method="spg" ,#c("spg" ,"CG")
7                               ,
8                               itnmax=50000)
9 "Error in qr.default(if (d[1L] < d[2L]) t(z) else z) :
10 NA/NaN/Inf en llamada a una funci\on externa (arg 1)"
11 amer2017.P4.2.apn.fit      <- CatDynFit(x=amer2017,
12                               p=4,
13                               par=pars.amer2017.P4.2.ini ,
14                               dates=dates.amer2017.P4.2,
15                               distr="apnormal" ,
16                               method="CG" ,#c("spg" ,"CG") ,
17                               itnmax=50000)
18 "Error in qr.default(if (d[1L] < d[2L]) t(z) else z) :
19 NA/NaN/Inf en llamada a una funci\on externa (arg 1)"

```

Ambos métodos colapsan así que realizamos la optimización con el método numérico alternativo 'Nelder-Mead', que sí logró converger, resolviendo el problema de optimización con esta variante.

4.2.5. Selección de la mejor variante

Creamos una lista con todas las variantes (28) y un vector de nombres de métodos numéricos que se alinea exactamente con los nombres de las variantes:

```

1 x <- list(amer2017.P1.1.apn.fit ,      ###01
2           amer2017.P1.1.apn.fit ,      ###02
3           amer2017.P1.1.apln.fit ,     ###03
4           amer2017.P1.1.apln.fit ,     ###04
5           amer2017.P2.1.apn.fit ,     ###05
6           amer2017.P2.1.apn.fit ,     ###06
7           amer2017.P2.1.apln.fit ,     ###07
8           amer2017.P2.1.apln.fit ,     ###08
9           amer2017.P3.1.apn.fit ,     ###09
10          amer2017.P3.1.apn.fit ,     ###10
11          amer2017.P3.1.apln.fit ,     ###11

```

```

12     amer2017.P3.1.apln.fit ,      ###12
13     amer2017.P3.2.apn.fit ,      ###13
14     amer2017.P3.2.apn.fit ,      ###14
15     amer2017.P3.2.apln.fit ,     ###15
16     amer2017.P3.2.apln.fit ,     ###16
17     amer2017.P3.3.apn.fit ,      ###17
18     amer2017.P3.3.apn.fit ,      ###18
19     amer2017.P3.3.apn.fit ,      ###19
20     amer2017.P3.3.apln.fit ,     ###20
21     amer2017.P3.3.apln.fit ,     ###21
22     amer2017.P4.1.apn.fit ,      ###22
23     amer2017.P4.1.apn.fit ,      ###23
24     amer2017.P4.1.apln.fit ,     ###24
25     amer2017.P4.1.apln.fit ,     ###25
26     amer2017.P4.2.apn.fit ,      ###26
27     amer2017.P4.2.apln.fit ,     ###27
28     amer2017.P4.2.apln.fit )     ###28
29 method <- c(rep(c("spg", "CG"),9), "Nelder-Mead", rep(c("spg", "CG")
30              ,3), "Nelder-Mead", "spg", "CG")
amer2017.sum.0 <- CatDynSum(x=x, season="2017", method=method)

```

La función **CatDynSum** creará una tabla con todos los resultados de las 28 variantes a ser consideradas en la selección de modelos.

La primera selección elimina a la única variante (###26) que tiene un gradiente mayor que 1. La segunda comparación involucra al AIC y sólo es válida entre variantes que usaron la misma fórmula de verosimilitud. Vemos que tanto dentro de los modelos normales como lognormales, variantes con tres entradas de abundancia son superiores a las variantes con 1, 2 y 4 entradas de abundancia:

```

1 amer2017.sum.1 <- amer2017.sum.0[amer2017.sum.0$Max.Abs.Grads. <
  1,]
2 amer2017.sum.1[amer2017.sum.1$Distribution=="apnormal",]
3 amer2017.sum.1[amer2017.sum.1$Distribution=="aplnormal",]

```

El examen de las correlaciones entre estimados se realiza con con la función **CatDynCor**:

```

1 CatDynCor(x=x[as.numeric(row.names(amer2017.sum.1))],
2           ttl=paste(amer2017.sum.1$Model,
3                     substring(amer2017.sum.1$Distribution, 1,6),
4                     amer2017.sum.1$Method,
5                     as.numeric(row.names(amer2017.sum.1))),
6           method=method[as.numeric(row.names(amer2017.sum.1))],

```

```
7 | arr=c(6,5)
```

y también nos muestra que las mejores variantes son las que tienen tres entradas de abundancia (específicamente, ###17, ###18 y ###19) aunque hay otras variantes con una cierta concentración de correlaciones cercanas a 0. En este punto creamos una nueva tabla resumen que contiene un subconjunto de líneas de la tabla resumen de la primera selección y usamos la nueva tabla resumen para reproducir el gráfico de correlaciones:

```
1 | amer2017.sum.2 <- amer2017.sum.0[row.names(amer2017.sum.0) %in% c(
  | "5", "6", "9", "10", "17", "18", "19", "22", "23"),]
2 | CatDynCor(x=x[as.numeric(row.names(amer2017.sum.2))],
  | ttl=paste(amer2017.sum.2$Model,
  |          substring(amer2017.sum.2$Distribution, 1,4),
  |          amer2017.sum.2$Method,
  |          as.numeric(row.names(amer2017.sum.2))),
  | method=method[as.numeric(row.names(amer2017.sum.2))],
  | arr=c(3,3))
```

Las variantes de esta lista corta presentan correlaciones aceptables (Fig. 17).

Examinamos los coeficientes de variación de los estimados utilizando la función **CatDynPar**:

```
1 | CatDynPar(x=amer2017.P2.1.apn.fit, method="spg")          ###05
2 | CatDynPar(x=amer2017.P2.1.apn.fit, method="CG")        ###06
3 | CatDynPar(x=amer2017.P3.1.apn.fit, method="spg")        ###09
4 | CatDynPar(x=amer2017.P3.1.apn.fit, method="CG")        ###10
5 | CatDynPar(x=amer2017.P3.3.apn.fit, method="CG")        ###17
6 | CatDynPar(x=amer2017.P3.3.apn.fit, method="CG")        ###18
7 | CatDynPar(x=amer2017.P3.3.apn.fit, method="Nelder-Mead") ###19 **
8 | CatDynPar(x=amer2017.P4.1.apn.fit, method="CG")        ###22
9 | CatDynPar(x=amer2017.P4.1.apn.fit, method="CG")        ###23
```

con lo que se observa que la variante ###19 tiene los coeficientes de variación mas bajos y sólo ha perdido uno de ocho errores estándar:

```
1 | CatDynPar(x=amer2017.P3.3.apn.fit, method="Nelder-Mead") ###19 **
2 | "           Parameter           Timing      Estimates
3 | CVpCent           M.1/week           3.3208e-02  121.2
```

4							
5							
6							
7							
8							
9							
10							

Reproducimos el gráfico de las predicciones de la variante ###19:

```

1 plot(x=amer2017.P3.3.apn.pred.NM, ###19
2     leg.pos="topright",
3     Biom.tstep=1,
4     Cat.tstep=dim(amer2017$Data$alijos)[1],
5     Biom.xpos=0.75,
6     Biom.ypos=0.8,
7     Cat.xpos=0.75,
8     Cat.ypos=0.75)

```

lo que nos permite confirmar que la estimación de biomasa de escape es realista y que la variante tiene buenas propiedades estadística en sus residuos (Fig. 18).

En definitiva, la variante ###19, ajustada con tres entradas de abundancia, la fórmula normal ('apnormal') de verosimilitud, y el método numérico 'Nelder-Mead' tiene gradientes cercanos a 0, supera en AIC a las variantes con menos y con más entradas de abundancia, tiene las mejores correlaciones y coeficientes de variación, y estima abundancias y mortalidad natural realistas. De hecho es la única variante que entrega una estimación realista de la mortalidad natural pues todas las otras variantes estiman esta tasa en el orden de magnitud de 10^{-3} que es muy bajo para la historia de vida de los pulpos. Seleccionamos esta variante haciendo:

```

1 amer2017.sum.0$Sel.Model[19] <- 1

```

4.3. Reuniendo todas las evaluaciones anuales

Habiendo seleccionado una variante de los modelos de depleción que supera a las restantes en criterios de calidad numérica, estadística y biológica, para las temporadas 2000 a 2021 de cada una de las dos especies, la tarea es

Figura 17: Correlaciones entre estimados de nueva variantes de modelos de depleción para los datos de la temporada de 2017 de *O. americanus*.

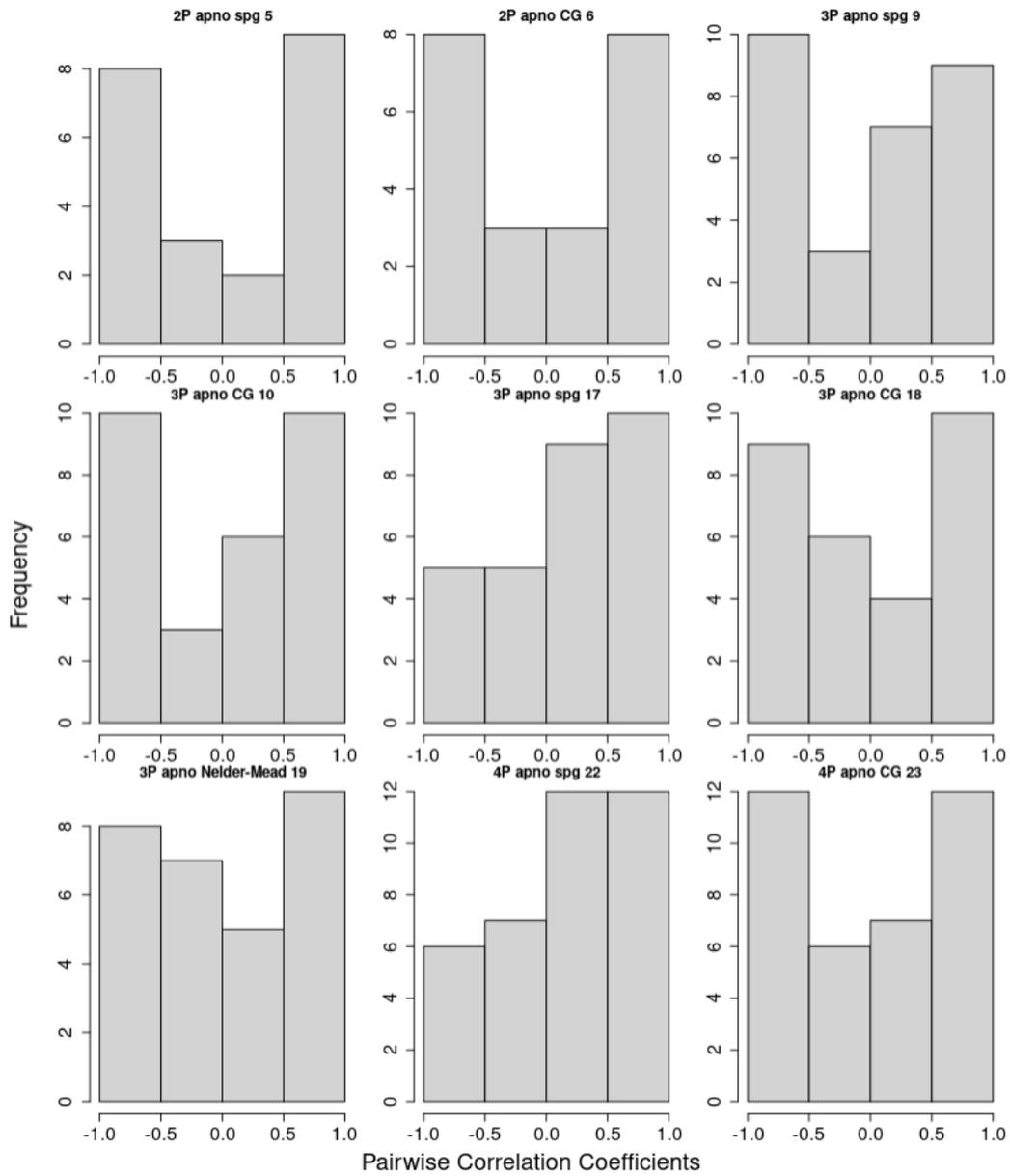


Figura 18: Variante seleccionada de modelo de depleción para los datos de la temporada de 2017 de *O. americanus*.

Fleet = alijos, Perturbations = 3, Distribution = Aponential, Numerical algorithm = Nelder-Mead

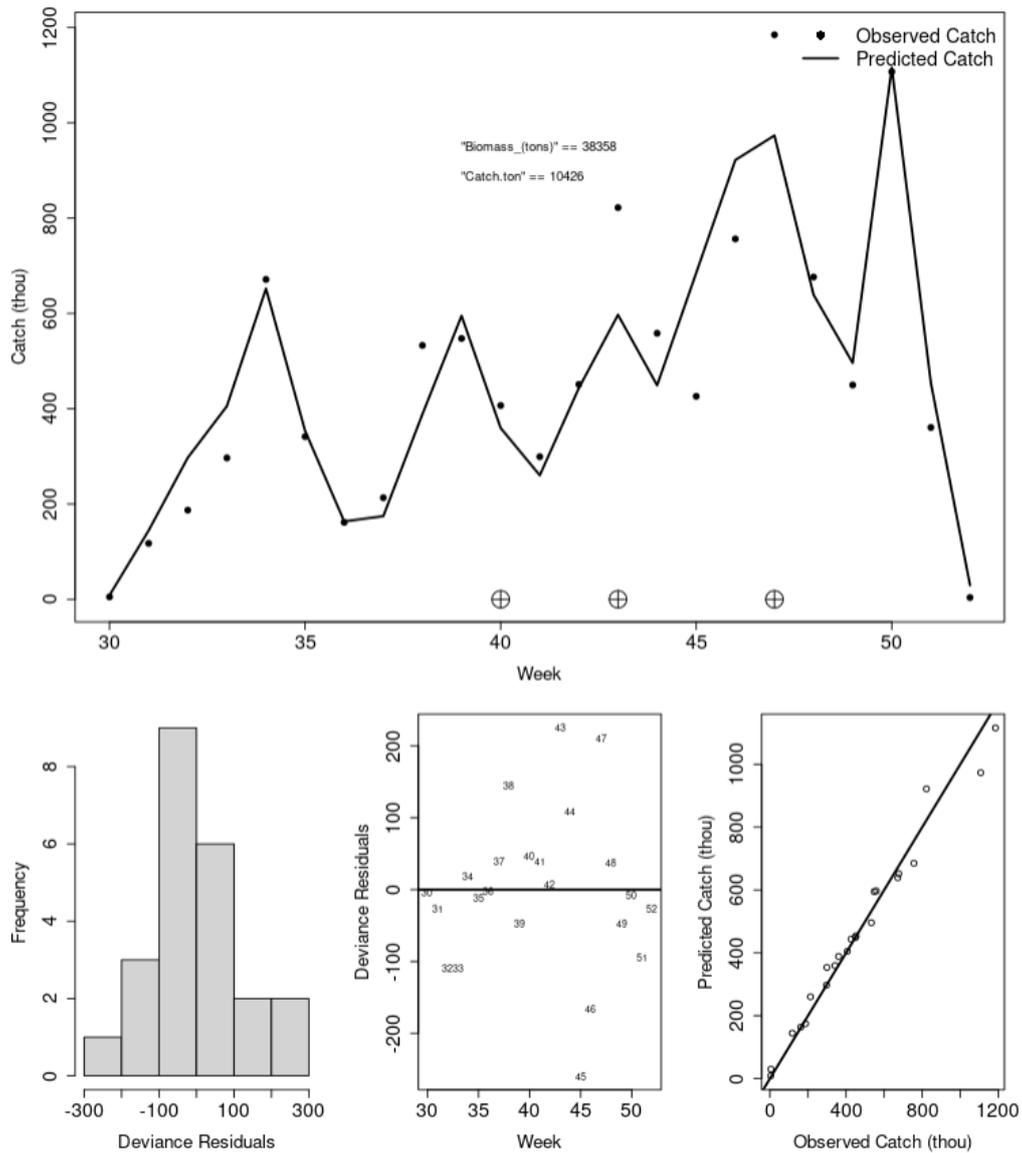


Figura 19: Tabla resumen de trabajo con **CatDyn** en la evaluación de las 22 temporadas de *O. maya*, 2000 a 2021.

Year	NumberFits	ConvFits	SelMod.name	SelMod.meth	P1.timing	P2.timing	P3.timing	P4.timing	P5.timing
1 2000	20	16	maya2000.P2.1.apln.fit	CG	32	43			
2 2001	28	24	maya2001.P3.1.apn.fit	CG	35	39	45		
3 2002	24	20	maya2002.P1.1.apn.fit	CG	35				
4 2003	20	16	maya2003.P2.1.apn.fit	CG	35	41			
5 2004	16	12	maya2004.P2.1.apn.ftt	CG	38	46			
6 2005	30	28	maya2005.P2.1.n.ftt	spg	38	38			
7 2006	20	16	maya2006.P1.1.apn.fit	CG	41				
8 2007	24	16	maya2007.P3.1.apn.fit	CG	34	37	44		
9 2008	20	16	maya2008.P1.1.apn.fit	CG	32				
10 2009	24	20	maya2009.P1.2.apn.fit	CG	39				
11 2010	16	12	maya2010.P1.1.apn.fit	CG	41				
12 2011	32	21	maya2011.P4.1.apn.fit	CG	31	38	40	45	
13 2012	36	22	maya2012.P1.1.apn.fit	spg	35				
14 2013	16	12	maya2013.P3.1.apn.fit	CG	33	40	41		
15 2014	18	10	maya2014.P3.1.apn.fit	spg	34	40	42		
16 2015	20	16	maya2015.P2.1.apn.fit	CG	35	39			
17 2016	16	6	maya2016.P2.1.apn.fit	CG	36	41			
18 2017	18	12	maya2017.P3.1.apn.fit	CG	32	49	45		
19 2018	18	14	maya2018.P4.1.apn.fit	spg	33	35	37	45	
20 2019	86	78	maya2019.P5.2.apn.fit	CG	37	39	41	45	48
21 2020	20	17	maya2020.P3.1.apn.fit	CG	35	38	48		
22 2021	44	40	maya2021.P5.1.apn.ftt	CG	30	32	37	39	42

ahora reunir esas evaluaciones para producir series de tiempo anuales con resultados colectivos que sirvan a la gestión pesquera y que sean usados para ajustar modelos de dinámica de poblaciones con biomasa agregada.

En este manual vamos a usar el caso de *O. maya*, sabiendo que la misma metodología puede implementarse a *O. americanus* o, en principio, a cualquier otra especie de la base de datos Avisos de Arribos.

En esta parte trabajaremos con el guión **Yucoct.Maya.CatDyn.IAGD.2017.R**. Abra el guión y cambie la variable 'wdir' para adaptarla a su directorio de trabajo. Cargue las tres librerías que se indican.

Cada temporada ha sido analizada separadamente con modelos de depleción y se han guardado ficheros binarios con los objetos creados durante cada evaluación, incluyendo las variantes seleccionadas como mejores para cada temporada. Comenzamos trayendo todos esos ficheros con **load**.

La función **data.frame** aprovecha las patrones de nombres de los objetos creados por **CatDynFit** y las dimensiones de los objetos creados con **CatDynSum** además de fijar otras variables, para construir una tabla que muestra las principales características del trabajo con **CatDyn** (Fig. 19).

Enseguida creamos una lista con todos los objetos producidos por **CatDynFit** con las variantes seleccionadas para cada temporada.

```

1 Maya.lagdfits <- list(maya2000.P2.1.apln.fit ,
2                       maya2001.P3.1.apn.fit ,
3                       maya2002.P1.1.apn.fit ,
4                       maya2003.P2.1.apn.fit ,

```

```

5         maya2004.P2.1.apn.fit ,
6         maya2005.P2.1.n.fit ,
7         maya2006.P1.1.apn.fit ,
8         maya2007.P3.1.apn.fit ,
9         maya2008.P1.1.apn.fit ,
10        maya2009.P1.2.apn.fit ,
11        maya2010.P1.1.apn.fit ,
12        maya2011.P4.1.apn.fit ,
13        maya2012.P1.1.apn.fit ,
14        maya2013.P3.1.apn.fit ,
15        maya2014.P3.1.apn.fit ,
16        maya2015.P2.1.apn.fit ,
17        maya2016.P2.1.apn.fit ,
18        maya2017.P3.1.apn.fit ,
19        maya2018.P4.1.apn.fit ,
20        maya2019.P5.2.apn.fit ,
21        maya2020.P3.1.apn.fit ,
22        maya2021.P5.1.apn.fit )
23 names(Maya.lagdfits) <- 2000:2021

```

Creamos otra lista con los objetos producidos por **CatDynPred**:

```

1 Maya.lagdprede <- list(maya2000.P2.1.apln.pred.CG,
2                       maya2001.P3.1.apn.pred.CG,
3                       maya2002.P2.1.n.pred.spg,
4                       maya2003.P2.1.apn.pred.CG,
5                       maya2004.P2.1.apn.pred.CG,
6                       maya2005.P2.1.apn.pred.spg,
7                       maya2006.P1.1.apn.pred.CG,
8                       maya2007.P3.1.apn.pred.CG,
9                       maya2008.P1.1.apn.pred.CG,
10                      maya2009.P1.2.apn.pred.CG,
11                      maya2010.P1.1.apn.pred.CG,
12                      maya2011.P4.1.apn.pred.CG,
13                      maya2012.P1.1.apn.pred.spg,
14                      maya2013.P3.1.apn.pred.CG,
15                      maya2014.P3.1.apn.pred.spg,
16                      maya2015.P2.1.apn.pred.CG,
17                      maya2016.P2.1.apn.pred.CG,
18                      maya2017.P3.1.apn.pred.CG,
19                      maya2018.P4.1.apn.pred.CG,
20                      maya2019.P5.2.apn.pred.CG,
21                      maya2020.P3.1.apn.pred.CG,
22                      maya2021.P5.1.apn.pred.CG)
23 names(Maya.lagdprede) <- 2000:2021

```

Figura 20: Tabla resumen con todas las variantes de modelos de depleción seleccionadas para cada temporada en la evaluación de las 22 temporadas de *O. maya*, 2000 a 2021.

The screenshot shows an R console window with a terminal window open. The terminal displays a summary table for 22 seasons (2000-2021) and a large matrix of 0.000 values. The table columns include: Fleet, Effort, Season, Model, Distribution, Method, AIC, Max. Abs. Grads., M, SE.M, N0, SE.N0, P1.alljos, SE.P1.alljos, ts.P1.alljos, P2.alljos, SE.P2.alljos, ts.P2.alljos, P3.alljos, and SE.P3.alljos. The matrix below the table shows the results of model selection for each season and model combination, with most values being 0.000.

El propósito de estas listas que reúnen todas las variantes seleccionadas de cada una de las evaluaciones anuales separadas es tener un objeto único para realizar gráficos de utilidad directa para la dinámica anual y la gestión.

En primer lugar se exportan los gráficos de ajustes de las variantes seleccionadas (por ejemplo, ver la Fig. 18) al formato de imagen 'tiff'. Este formato produce gráficas de gran resolución pero genera ficheros muy pesados. Por eso luego de generar los 'tiff' es conveniente transformarlos a formato 'jpg' que va a ser mucho menos pesado preservando la resolución de la imagen.

A continuación utilizamos la función **CatDynSum**, que antes habíamos usado para comparar variantes de modelos de una temporada, para generar una tabla de resultados con todos las variantes seleccionadas para cada temporada. La segunda línea del código abajo es necesaria porque cuando **CatDyn** extrae los resultados desde los métodos numéricos y organizados por **optimx** omite remover la clase 'list' del estimado de N_0 . Este detalle y muchos otros que han sido listados por el autor y usuarios va a ser subsanado en la próxima versión de **CatDyn** que está programada para ser publicada en el repositorio oficial de R CRAN durante este año:

```

1 maya.sum.2000.2021 <- CatDynSum(x=Maya.lagdfits , season="
  2000.2021" ,method=Maya.SelMods$SelMod.meth)
2 maya.sum.2000.2021$N0 <- as.numeric(maya.sum.2000.2021$N0)
3 maya.sum.2000.2021$Season <- 2000:2021
4 maya.sum.2000.2021$Sel.Model <- 1

```

El resultado se muestra en la Fig. 20. Como se puede apreciar, varias de las variantes seleccionadas carecen de algunos errores estándar de sus parámetros debido a anomalías en la forma de la función de verosimilitud alrededor del máximo respecto de algunos parámetros, lo que conduce a generar un 'NaN'. Esta deficiencia se puede solucionar utilizando técnicas de imputación de datos. Estas son metodologías cuantitativas y objetivas que usan la mejor información para imputar datos faltantes, o en este caso, errores estándar faltantes. Con este fin utilizaremos la librería *mice*.

En la tabla (Fig. 20) observamos que existen errores estándar faltantes en seis parámetros: M , N_0 , la primera entrada de abundancia, k , α y β . Este código:

```

1 imp1 <- mice(maya.sum.2000.2021[,9:10], method="pmm",m=1,maxit
  =1000)
2 maya.sum.2000.2021[,9:10] <- complete(imp1)
3 imp1 <- mice(maya.sum.2000.2021[,11:12], method="pmm",m=1,maxit
  =1000)
4 maya.sum.2000.2021[,11:12] <- complete(imp1)
5 imp1 <- mice(maya.sum.2000.2021[,13:14], method="pmm",m=1,maxit
  =1000)
6 maya.sum.2000.2021[,13:14] <- complete(imp1)
7 imp1 <- mice(maya.sum.2000.2021[,28:29], method="pmm",m=1,maxit
  =1000)
8 maya.sum.2000.2021[,28:29] <- complete(imp1)
9 imp1 <- mice(maya.sum.2000.2021[,30:31], method="pmm",m=1,maxit
  =1000)
10 maya.sum.2000.2021[,30:31] <- complete(imp1)
11 imp1 <- mice(maya.sum.2000.2021[,32:33], method="pmm",m=1,maxit
  =1000)
12 maya.sum.2000.2021[,32:33] <- complete(imp1)
13 rm(imp1)

```

va a reemplazar exclusivamente los errores estándar faltantes dentro de cada parámetro utilizando los estimados del parámetro y los errores estándar presentes para este parámetro a través de las temporadas. De esta manera los errores estándar faltantes y reemplazados por la imputación van a preservar el nivel promedio de incertidumbre estadística que existe a través de las

evaluaciones de stock para ese parámetro en particular. El orden en que se ejecutan las imputaciones es el mismo que se indica arriba (M , N_0 , la primera entrada de abundancia, k , α y β).

Habiendo resuelto la falta de algunos errores estándar de la manera más objetiva posible, podemos realizar gráficas con los estimados de los modelos y barras de sus errores estándar a través de todas las temporadas de pesca. La Fig. 21 muestra estas series anuales de parámetros. La mortalidad natural (M) sube hasta 2005 y luego baja hasta 2021, con una fuerte caída desde 2016 a 2017. En general los estimados tienen buena precisión estadística. La abundancia inicial (N_0) tiende a subir pero fluctúa mucho a partir de 2011. Los estimados son en general muy precisos. La primera entrada de abundancia es relativamente menos varíanle y la precisión es la menor de todos los parámetros. Los tres parámetros de la operatividad pesquera (k , α y β) son estimados con mucha precisión. Se observan una caída y subsecuente recuperación de la respuesta al esfuerzo (α) y una dinámica hiper-estable de la respuesta a la abundancia (β). Las tendencias de la respuesta al esfuerzo y la respuesta a la abundancia sugieren que los pescadores han logrado pescar más efectivamente en los últimos años.

El siguiente gráfico es de utilidad para determinar el estatus de explotación del stock a partir de los resultados de los modelo de depleción puramente, sin llegar a modelizar aún la dinámica de poblaciones y la capacidad productiva del stock. La tasa de explotación instantánea está definida como:

$$\epsilon = \frac{F}{M + F} \quad (3)$$

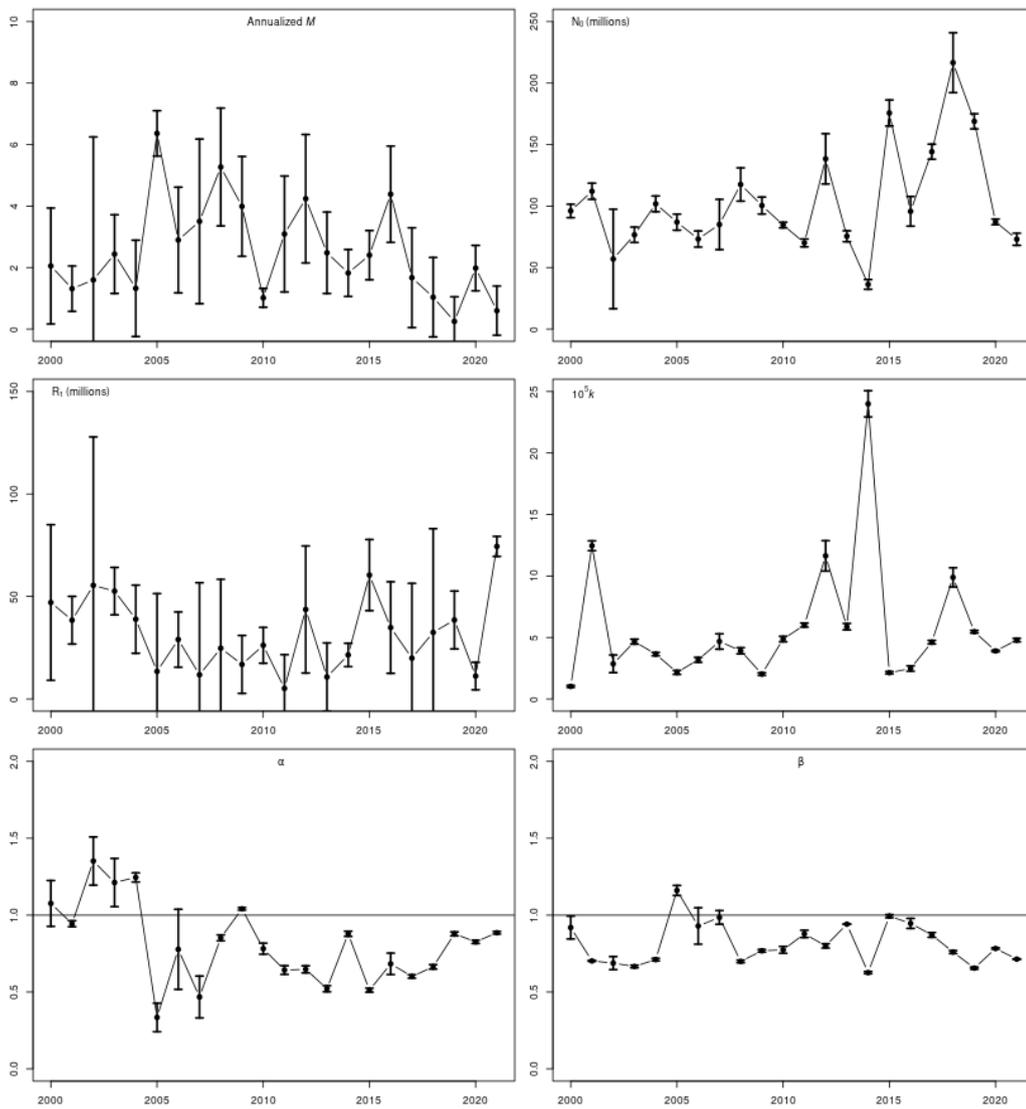
donde F es la mortalidad por pesca. Existe un punto biológico de referencia que establece que una tasa de explotación instantánea límite es 0,4, es decir que la mortalidad por pesca no debe ser más que un 40 % de la mortalidad total. Los modelos de depleción no contienen a F como uno de sus parámetros pero F puede ser obtenida después del ajuste de los modelos desde la ecuación de captura de Baranov:

$$C_t = N_t \frac{F_t}{M_t + F_t} (1 - e^{-M_t - F_t}) \quad (4)$$

La función **CatDynPred** tiene la captura (C_t), la abundancia (N_t), y la mortalidad natural (M) en cada paso de tiempo semanal, así que usa la función **uniroot** para resolver F a cada paso de tiempo. Por lo tanto es posible conocer la tasa de explotación instantánea ϵ a cada paso de tiempo dentro de cada temporada y verificar si cumple con evaluar en menos de 0,4.

Como se aprecia en la Fig. 22, sólo en las temporadas de pesca de 2014, 2018, 2019 y 2021 el stock ha sido explotado por arriba de la tasa de explotación instantánea de 0,4 mientras que en el resto de los años la explotación

Figura 21: Parámetros de los modelos de depleción estimados por todas las variantes seleccionadas para cada temporada en la evaluación de las 22 temporadas de *O. maya*, 2000 a 2021.



ha ocurrido en general muy por debajo del umbral, excepto en 2020 en que ocurrió por debajo pero cerca del umbral.

Además de obtener estas series de tiempo de interés en la gestión del recurso, la colección de evaluaciones anuales debe ser usada para generar series de tiempo para el estudio de la dinámica de poblaciones, en particular las series de biomasa anual y sus errores estándar para ajustar modelos de producción excedente.

Las biomasa ya están disponible en los objetos de predicción generados por `CatDynPred` pero estas series de biomasa tienen dos limitaciones que impiden su uso directo para generar observaciones para el ajuste de modelo de producción excedente: (1) necesitamos la biomasa al inicio de la temporada de pesca pero la que tenemos en los objetos de predicción generados `CatDynPred` no incluye la parte que se hace manifiesta más tarde durante la temporada, las entradas de abundancia, y (2) las series de biomasa generados por `CatDynPred` no tienen su error estándar como medida de incertidumbre estadística. Para generar series de biomasa completas y sus errores estándar `CatDyn` tiene la función `CatDynBSD`, que utiliza intensivamente el método delta y el modelo de decaimiento exponencial de las cohortes hacia atrás en el tiempo para generar estimados de la biomasa total inicial y su error estándar, incluyendo las entradas de abundancia que ocurren durante la temporada.

Para utilizar la función `CatDynBSD` con modelos intra-anales debemos tener

- una lista con todos los objetos generados por `CatDynFit` que corresponden a las variantes seleccionadas para cada temporada,
- un vector con los métodos numéricos que corresponden a las variantes seleccionadas para cada temporada, y
- los pesos medios en la captura al inicio de cada temporada.

```
1 omaya.wbw.trend <- read.csv("omaya.wbw.trend.csv", header=TRUE)
2 mbw.sd <- data.frame(Year=2000:2021,
3                       Mean.weekly.weight.g=omaya.wbw.trend$Mbw.g[
4                       omaya.wbw.trend$Week==30],
5                       SD.Mean.weekly.weight=omaya.wbw.trend$Sd.Mbw.
6                       g[omaya.wbw.trend$Week==30])
7 maya.biom.2000.2021 <- CatDynBSD(x=Maya.lagdfits,
8                                method=Maya.SelMods$SelMod.meth,
9                                multi=FALSE,
10                               mbw.sd=mbw.sd)
11 maya.biom.2000.2021$B0Tot.ton.CV <- 100*maya.biom.2000.2021$B0Tot.
12 ton.SE/maya.biom.2000.2021$B0Tot.ton
```

Figura 22: Tasas de explotación instantánea estimada por los modelos de depleción de las variantes seleccionadas para cada temporada en la evaluación de las 22 temporadas de *O. maya*, 2000 a 2021.

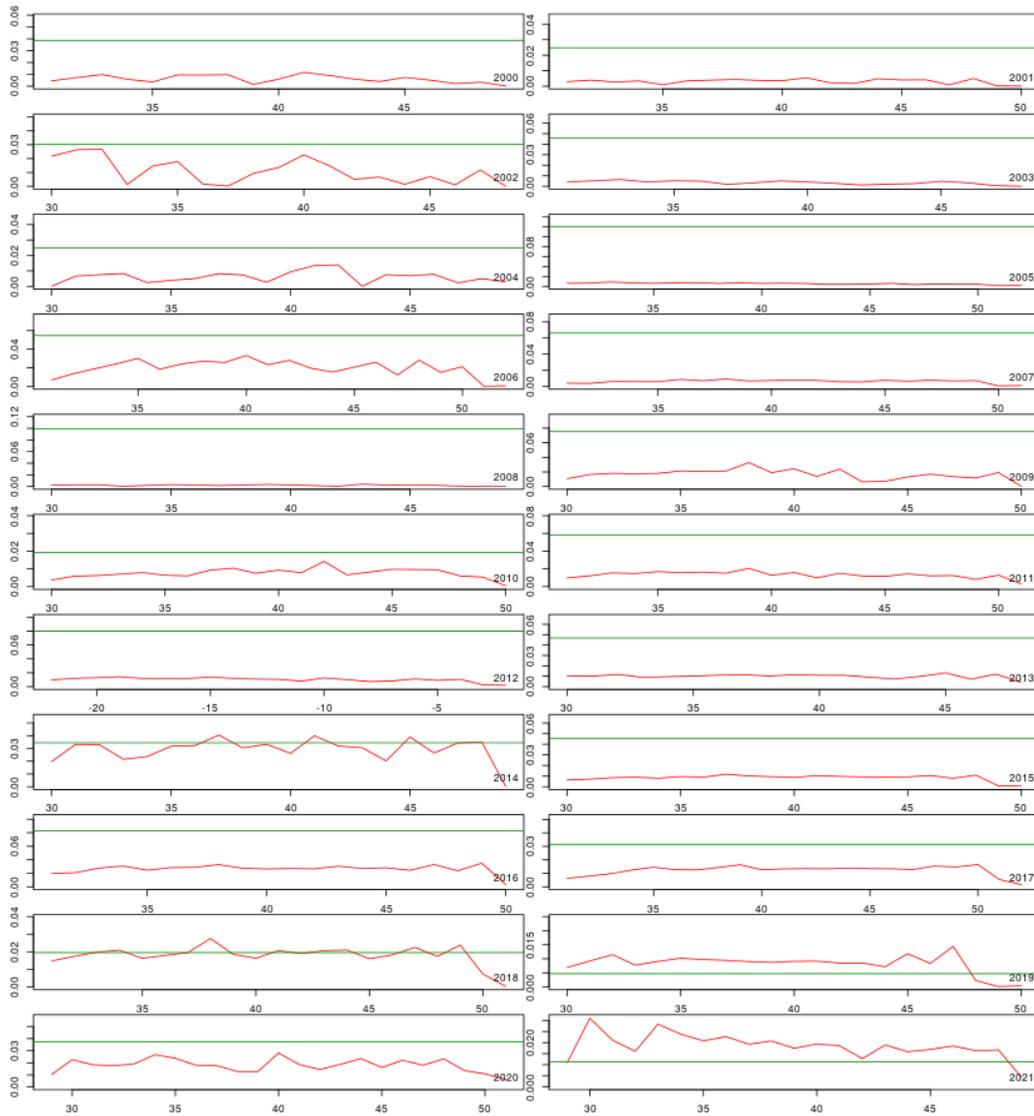


Figura 23: Tabla generada por la función **CatDynBSD** para calcular los errores estándar de las biomazas iniciales de cada temporada de pesca de *O. maya*, 2000 a 2021

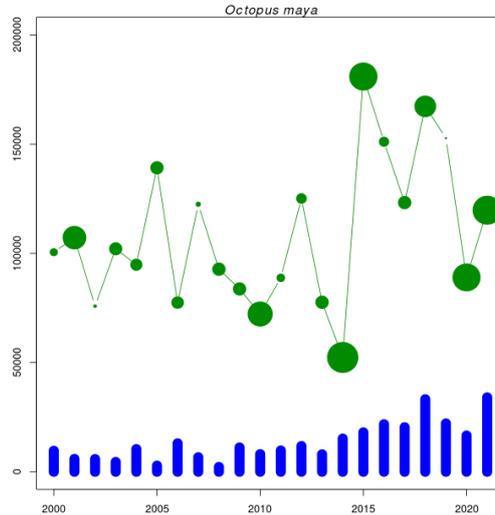
Year	Mw.kg	SDmw.kg	N0Tot.thou	N0Tot.thou.SE	B0Tot.ton	B0Tot.ton.SE	B0Tot.ton.CV
1 2000	614.5772	2.438525	163543.69	42586.482	100510219	26175718	26.042843
2 2001	614.5772	2.438525	174390.18	14639.432	107176223	9007105	8.404014
3 2002	614.5772	2.438525	123380.53	81010.741	75826861	49788263	65.660456
4 2003	614.5772	2.438525	166100.84	25668.861	102081790	15780695	15.458874
5 2004	614.5772	2.438525	154232.11	25870.215	94787536	15903692	16.778252
6 2005	614.5772	2.438525	226428.61	34570.903	139157856	21253662	15.273059
7 2006	614.5772	2.438525	126044.39	20515.500	77464005	12612104	16.281244
8 2007	614.5772	2.438525	199259.96	84390.096	122460626	51866504	42.353616
9 2008	614.5772	2.438525	150908.45	22825.631	92744890	14032938	15.130686
10 2009	614.5772	2.438525	136203.39	20735.479	83707496	12747880	15.229078
11 2010	614.5772	2.438525	117527.28	9241.579	72229585	5686890	7.873352
12 2011	614.5772	2.438525	144463.92	35418.935	88784232	21770520	24.520706
13 2012	614.5772	2.438525	203552.19	39251.948	125098534	24128458	19.287562
14 2013	614.5772	2.438525	126279.41	19113.790	77608442	11750935	15.141310
15 2014	614.5772	2.438525	85159.73	5382.712	52337229	3314603	6.333166
16 2015	614.5772	2.438525	294384.61	20583.901	180922064	12670748	7.003429
17 2016	614.5772	2.438525	245819.60	49576.368	151075120	30474401	20.171687
18 2017	614.5772	2.438525	200485.58	30479.905	123213862	18738633	15.208218
19 2018	614.5772	2.438525	272173.75	24881.904	167271776	15306247	9.150526
20 2019	614.5772	2.438525	248488.37	322982.075	152715281	198498340	129.979357
21 2020	614.5772	2.438525	144777.21	10190.974	88976768	6273082	7.050247
22 2021	614.5772	2.438525	194824.06	13206.487	119734421	8130298	6.790276

El uso de 'multi=FALSE' significa que los modelos son intra-anales, no multi-anales. La tabla resultante se muestra en la Fig. 23.

La función **CatDynBSD** genera también abundancias iniciales (en la semana anterior al inicio de la temporada) en número (miles) y sus errores estándar. En general los errores estándar de las biomazas son bien precisos en este caso, casi todos generando coeficientes de variación menores que 100 %, y frecuentemente menores que 50 %.

Esta tabla es el fin de los trabajos con los modelos de depleción generalizados en **CatDyn**. Las estimaciones de biomazas anuales y sus errores estándar van a ser entregadas a código en AD Model Builder (ADMB) para estimar los modelos de producción excedente y así determinar la dinámica de poblaciones y la productividad en biomasa de los stocks.

Figura 24: Capturas (barras azules) y biomazas (puntos verdes, tamaño proporcional a precisión estadística) de *O. maya* en la Península de Yucatán y alrededores.



5. Etapa 2: Modelos de producción excedente

Los modelos de producción excedente analizan la dinámica de poblaciones desde el punto de vista de la biomasa agregada del stock. En su formulación estándar, el modelo de producción excedente de Pella y Tomlinson tiene la forma:

$$B_y = B_{y-1} + rB_{y-1} \left(1 - \left(\frac{B_{y-1}}{K} \right)^{p-1} \right) - C_{y-1}, 2000 \leq y \leq 2021 \quad (5)$$

$$r > 0, K > 0, p > 1$$

con parámetros r (tasa intrínseca de crecimiento poblacional), p (simetría de la función de producción de biomasa), K (capacidad de carga del ambiente), y B_0 (la biomasa justo antes del primer año de la serie de tiempo de capturas, i.e. 1999). Cuando la biomasa inicial B_0 se supone igual a K se genera el modelo de producción excedente de tres parámetros. El modelo de producción excedente de cuatro parámetros no presupone que $B_0 = K$ sino que B_0 es un parámetro a estimar con los datos.

Las fuentes de información para el modelo de producción excedente son las series de capturas anuales y las biomazas estimadas por los modelos de

depleción. Esstas observaciones son desplegadas en la Fig. 24. Esta figura muestras un incremento de las capturas y la biomasa a partir de 2015, lo que es muy improbable a menos que haya ocurrido un cambio en el tamaño del stock por razones externas a la pesca. Por este motivo ajustaremos modelos de producción excedente con parámetros constantes a través de la serie de tiempo (eq. 5) y con parámetros variables:

$$\begin{aligned}
 B_y &= B_{y-1} + r_1 B_{y-1} \left(1 - \left(\frac{B_{y-1}}{K_1} \right)^{p_1-1} \right) - C_{y-1}, & 2000 \leq y < y_x \\
 B_y &= B_{y-1} + r_2 B_{y-1} \left(1 - \left(\frac{B_{y-1}}{K_2} \right)^{p_2-1} \right) - C_{y-1}, & y_x \leq y < 2021
 \end{aligned} \tag{6}$$

$$r_1 > 0, K_1 > 0, p_1 > 1, r_2 > 0, K_2 > 0, p_2 > 1$$

en que el año y_x es un año en que cambia uno o más de los parámetros del modelo de producción excedente.

El guión [Yucoct.Maya.CatDynIAGD.R](#) continúa con un modelo lúdico de producción excedente para obtener buenos parámetros iniciales para la optimización en ADMB. Encontramos primero un modelo con parámetros constantes y luego uno con posibles cambios en los parámetros que ocurren en 2015 (en *O. americanus* este cambio también se observa pero en 2011). Tal como en el caso de los modelos lúdicos de depleción, el objetivo aquí es obtener buenos valores iniciales y para conseguir esto tenemos que hacer que las curvas de biomasa predichas por los modelos lúdicos caigan sobre las biomazas estimadas por los modelos de depleción (Fig. 25).

El guión [Yucoct.Maya.CatDynIAGD.R](#) concluye

- imprimiendo en pantalla un vector horizontal con las series anuales de capturas que se puede copiar y pegar en el fichero de datos para el ajuste en ADMB,
- imprimiendo en pantalla una tabla con el año, las biomazas anuales estimadas por los modelos de depleción, y los errores estándar de esas biomazas, y
- imprimiendo en pantalla los valores iniciales conseguidos con los modelos lúdicos de parámetros constantes y parámetros que cambian en y_x .

Las primeras dos líneas del código abajo son las observaciones para el modelo de producción excedente y las siguientes líneas simplemente transforman los valores conseguidos con el modelo lúdico a logaritmo:

```

1 t(Maya.InniBiom.ton$Landings.ton)
2 Maya.InniBiom.ton[,c(1,3,4)]
3 (log.B0.ini <- log(B0.ini))
4 #[1] 11.51293
5 (log.K.ini <- log(K.ini))
6 #[1] 11.73607
7 (log.p.ini <- log(p.ini))
8 #[1]0.6931472
9 (log.r.ini <- log(r.ini))
10 #[1] 0.9162907
11 (log.K1.ini <- log(K1.ini))
12 #[1] 11.44197
13 (log.K2.ini <- log(K2.ini))
14 #[1] 11.66511
15 (log.p1.ini <- log(p1.ini))
16 #[1] 0.6881346
17 (log.p2.ini <- log(p2.ini))
18 #[1] 0.6881346
19 (log.r1.ini <- log(r1.ini))
20 #[1] 0.9162907
21 (log.r2.ini <- log(r2.ini))
22 #[1] 1.252763

```

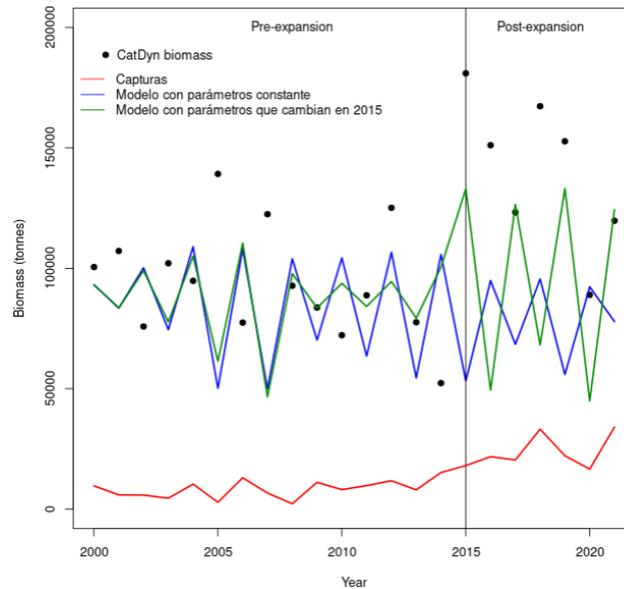
5.1. Optimización estadística en ADMB

Vamos a cubrir esta parte de los trabajos de evaluación de stock volviendo al caso de *O. americanus*, pues este presenta una dinámica de poblaciones que nos permitirá ilustrar un punto clave en la gestión científica de los stocks de vida corta, como los pulpos.

Para ajustar un modelo estadístico en ADMB hay que proporcionar al menos dos ficheros, uno con el código y otro con los datos, y los dos ficheros deben tener el mismo nombre y diferir sólo en la extensión. En nuestras aplicaciones tenemos los siguientes ficheros ADMB:

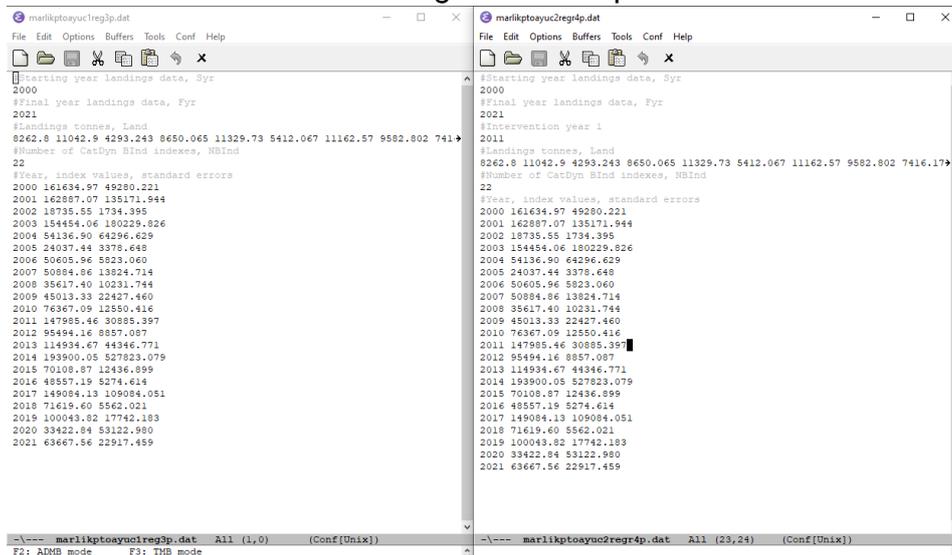
- marlikptoayuc1reg3p.dat: datos para modelos con parámetros constantes y $B_0 = K$
- marlikptoayuc1reg3p.tpl: código para modelos con parámetros constantes y $B_0 = K$
- marlikptoayuc1reg4p.dat: datos para modelos con parámetros constantes y $B_0 \neq K$

Figura 25: Observaciones de biomasa anual, capturas y predicciones de modelos de producción excedente con parámetros constantes y con parámetros que cambian en 2015 para el stock de *O. maya*, temporadas 2000 a 2021.



- marlikptoayuc1reg4p.tpl: código para modelos con parámetros constantes y $B_0 \neq K$
- marlikptoayuc2regr4p.dat: datos para modelos con r que cambia en 2015 y $B_0 = K$
- marlikptoayuc2regr4p.tpl: código para modelos con r que cambia en 2015 y $B_0 = K$
- marlikptoayuc2regrK5p.dat: datos para modelos con r y K que cambia en 2015 y $B_0 = K$
- marlikptoayuc2regrK5p.tpl: código para modelos con r y K que cambian en 2015 y $B_0 = K$
- marlikptoayuc2regrp5p.dat: datos para modelos con r y p que cambian en 2015 y $B_0 = K$
- marlikptoayuc2regrp5p.tpl: código para modelos con r y p que cambian en 2015 y $B_0 = K$
- marlikptoayuc2regrpK6p.dat: datos para modelos con r , p y K que cambian en 2015 y $B_0 = K$

Figura 26: Archivos de datos para el ajuste del modelo de producción excedente con ADMB. A la izquierda el archivo para modelos sin cambios en los parámetros, a la derecha el archivo para modelos con cambios en los parámetros en 2015. La interfaz gráfica es la presentación de ADStudio.



- marlikptoayuc2reg4pK6p.tpl: código para modelos con r , p y K que cambian en 2015 y $B_0 = K$

Los nombres de los archivos se derivan de la siguiente contracción: **m**arginal **l**ikelihood **p**ella **t**onlinson **o**ctopus **a**mericanus **y**ucatan followed by 1reg (1 régimen) or 2reg (2 regímenes) dependiendo de si es un modelo sin o con cambios en los parámetros, seguido por los nombres de los parámetros que cambian (ninguno, r , r y K , r y p , o r y p y K , y finalmente terminando con el número de parámetros del modelo (3p, 4p, 5p, o 6p).

Los archivos de datos, que tienen extensión .dat, contienen el año de inicio de las capturas anuales, el año final de las capturas anuales, el año del cambio en el valor de los parámetros en el caso de los modelos con cambios en los parámetros, el vector horizontal de capturas, el número de evaluaciones de la biomasa, y una matriz con tres columnas: año, biomasa, y error estándar de la biomasa (Fig. 26).

Para ilustrar el código ADMB vamos a considerar el modelo con un cambio en r y K (marlikptoayuc2reg4pK5p.tpl, .tpl viene de 'template') porque entre los cinco modelos listados arriba este es seleccionado en base a criterios estadísticos, numéricos y biológicos. Aunque en general los archivos .tpl pueden contener menos o más secciones, los archivos .tpl que han sido programados para ajustar modelos de producción excedente con esta metodología contienen 5 secciones (Fig. 27): DATA_SECTION, INITIALIZATION_SECTION, PARA-

Figura 27: Código ADMB completo para el ajuste del modelo de producción excedente con cambio en r y K en 2015. El fichero está cortado por la mitad y las dos mitades son desplegadas una al lado de la otra. La interface gráfica es la presentación de ADStudio.

```

DATA_SECTION
init_int Syz;
init_int Fyz;
init_int Iyrl;
init_vector Land(Syz, Fyz);
init_int NBiom;
init_matrix CatDynBiom(1, NBiom, 1, 3);
vector YrBiom(1, NBiom);
vector MLBiom(1, NBiom);
vector SDBiom(1, NBiom);

INITIALIZATION_SECTION
logK1 10.60;
logK2 11.29;
logp 0.693;
logr1 0.916;
logr2 1.099;

PARAMETER_SECTION
init_bounded_number logK1(10.0, 11.0, 1);
//init_number logK1;
init_bounded_number logK2(11.0, 12.0, 1);
//init_number logK2;
init_bounded_number logp(0.5, 0.8, 3);
//init_number logp(3);
init_bounded_number logr1(0.5, 1.0, 2);
//init_number logr1(2);
init_bounded_number logr2(1.0, 1.6, 2);
//init_number logr2(2);
vector Biom(Syz, Fyz+1);
vector FredBiom(1, NBiom);
vector Squiff(1, NBiom);
number Prod;
number B0;
sreport_number R1;
sreport_number R2;
sreport_number p;
sreport_number r1;
sreport_number r2;
sreport_vector Biom_sd(Syz, Fyz+1);
objective_function value ff;

PRELIMINARY_CALCS_SECTION

```

```

YrBiom=column(CatDynBiom, 1);
MLBiom=column(CatDynBiom, 2);
SDBiom=column(CatDynBiom, 3);

PROCEDURE_SECTION
R1=mfexp(logK1);
R2=mfexp(logK2);
p=mfexp(logp);
r1=mfexp(logr1);
r2=mfexp(logr2);
B0=R1;
int yr;
int yrBiom;
Biom(Syz)=B0;
for (yr=Syz+1; yr<=Iyrl-1; yr++)
{
  Prod=Biom(yr-1)+r1*Biom(yr-1)*(1-pow(Biom(yr-1)/K1, p-1));
  if (Prod<Land(yr-1)) Prod=Land(yr-1)+1000;
  Biom(yr)=Prod-Land(yr-1);
}
for (yr=Iyrl; yr<=Fyz+1; yr++)
{
  Prod=Biom(yr-1)+r2*Biom(yr-1)*(1-pow(Biom(yr-1)/K2, p-1));
  if (Prod<Land(yr-1)) Prod=Land(yr-1)+1000;
  Biom(yr)=Prod-Land(yr-1);
}
variable loglikBiom;
loglikBiom=0;
yrBiom=1;
for (yr=Syz; yr<=Fyz; yr++)
{
  if (yr==YrBiom(yrBiom))
  {
    FredBiom(yrBiom)=(Biom(yr)+Biom(yr+1))/2.;
    Squiff(yrBiom)=square(MLBiom(yrBiom)-FredBiom(yrBiom));
    loglikBiom = loglikBiom - 0.5*(log(2*3.1416*square(SDBiom(yrBiom)))+Squiff(yrBiom));
    yrBiom=yrBiom+1;
  }
}
ff = -(loglikBiom);
Biom_sd=Biom;

```

METER_SECTION, PRELIMINARY_CALCS_SECTION, y PROCEDURE_SECTION. Nada de este código, ni de los restantes códigos ADMB para otros tipos de modelos de producción excedente que son entregados junto con este manual (marlikptoayuc1reg3p.tpl, marlikptoayuc1reg4p.tpl, marlikptoayuc2regr4p.tpl, marlikptoayuc2regrp5p.tpl, marlikptoayuc2regrpK6p.tpl) debe ser modificado por el usuario excepto las primeras líneas de PARAMETER_SECTION,

```

1  init_bounded_number logK1(10.0, 11.0, 1);
2  //init_number logK1;
3  init_bounded_number logK2(11.0, 12.0, 1);
4  //init_number logK2;
5  init_bounded_number logp(0.5, 0.8, 3);
6  //init_number logp(3);
7  init_bounded_number logr1(0.5, 1.0, 2);
8  //init_number logr1(2);
9  init_bounded_number logr2(1.0, 1.6, 2);
10 //init_number logr2(2);

```

De estas diez líneas (y las correspondientes en los restantes códigos ADMB para otros tipos de modelos de producción excedente) sólo cinco deben ser ejecutadas en cada optimización pues hay dos opciones para cada uno de los cinco parámetro. Estas dos opciones para cada parámetro son

optimizar respecto de ese parámetro dentro de bordes:

```
1  init_bounded_number logK1(10.0,11.0,1);
```

o de manera completamente libre:

```
1  init_number logK1;
```

El uso de `'//'` desactiva una de las dos opciones así que en cada optimización debe haber cinco de estas líneas desactivadas de tal manera que para cada uno de los cinco parámetros una de las opciones esté activa.

Si el usuario está actualizando la evaluación de stock de *O. maya* u *O. americanus* es altamente recomendable no manipular ninguna otra parte del código a menos que el usuario sepa muy bien lo que hace. Sin embargo, es posible aunque improbable que sea conveniente en algún caso re-ajustar los valores iniciales de los parámetros, y en ese caso el usuario modificará:

```
1  logK1 10.60;  
2  logK2 11.29;  
3  logp 0.693;  
4  logr1 0.916;  
5  logr2 1.099;
```

cambiando los valores numéricos por otro (siempre en escala logarítmica).

Es posible que al acumular más años de estimaciones de biomasa desde modelos de depleción la estimación del modelo de producción excedente en ADMB pueda realizarse exitosamente con todos los parámetros libres, pero es más probable que sea necesario poner bordes a todos o algunos de los parámetros para facilitar la optimización durante muchos años más. En la definición de los parámetros como `'init_bounded_number'`:

```
1  init_bounded_number logK1(10.0,11.0,1);  
2  //init_number logK1;
```

el primer número es el borde inferior, el segundo número es el borde superior, y el tercer número es la fase en la cual el parámetro entra en la optimización.

Es importante que los parámetros no se estrellen contra uno de los bordes al finalizar la optimización. Si esto ocurre el usuario debe extender el borde

para darle más espacio al parámetro para que encuentre su valor optimizado y volver a correr la optimización. En unos pocos casos es necesario aceptar que un parámetro esté muy cerca de un borde, pero esto sólo es aceptable si se cumplen dos condiciones:

- que al extender el borde incluso por décimas la optimización no sea exitosa
- que el error estándar del parámetro que está cerca de su borde sea lo bastante grande, más del 1% del valor del estimado

La fase de la optimización (el único número de la declaración `init_number` o el tercer número de la declaración `init_bounded_number`) se puede aplicar a parámetros que son optimizados entre bordes y a parámetros que son optimizados con completa libertad. Esta fase es una muy útil opción de ADMB (no está disponible en R) y significa lo siguiente. Cuando la fase no se declara entonces todos los parámetros son optimizados simultáneamente de una sola vez. Cuando se declaran dos fases, la 1 y la 2, entonces los parámetros que entran en la fase 1 son optimizados manteniendo los parámetros de la fase 2 fijos en su valor inicial entregado por el usuario, y luego en la fase 2 los valores optimizados de los parámetros que fueron declarados en la fase 1 reemplazan a los valores iniciales entregados por el usuario y se realiza una optimización nueva con todos los parámetros simultáneamente. Esto significa que los parámetros más difíciles de estimar, aquellos para los cuales es más difícil obtener buenos valores iniciales por parte del usuario, debieran ser declarados para optimizar en la fase 1. En el código del modelo con cambio en r y K de la Fig. 27, K_1 y K_2 entran en la primera fase y son optimizados manteniendo fijos a p , r_1 y r_2 en los valores iniciales (0,693, 0,916 y 1,099, respectivamente), luego entran r_1 y r_2 que son optimizados con los valores de optimización de la fase 1 para K_1 y K_2 manteniendo fijo a p en el valor entregado por el usuario (0,693), y finalmente en la fase 3 los cinco parámetros son optimizados utilizando los valores previamente optimizados de K_1 , K_2 , r_1 y r_2 como valores iniciales para estos parámetros mientras que p se optimiza desde el valor entregado por el usuario. Esto es así porque p tiene un rango de variación factible (aproximadamente 1.2 a 3) bien conocido así que puede entrar más tarde en la optimización general.

La optimización comienza por presionar el botón 'Translate' (primer botón a su derecha de la barra del menú de íconos de la interfase ADStudio). El ícono es un enchufe acercándose a un fuente de poder (Fig. 28). El resultado se informa en el panel a la derecha del usuario. Debe terminar con un mensaje que contenga la palabra 'finished' en color verde.

En el siguiente paso se presiona el botón 'Build' (segundo botón a su derecha de la barra del menú de íconos). El ícono es un enchufe conectado a

Figura 28: Inicio del proceso de optimización estadística en ADStudio con el botón 'Translate' para el modelo de producción excedente de tres parámetros.

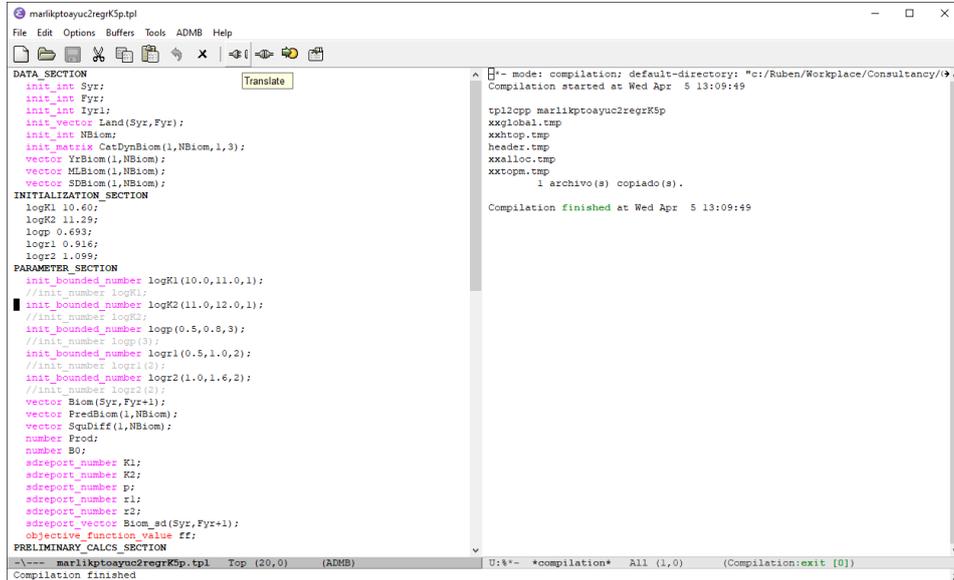
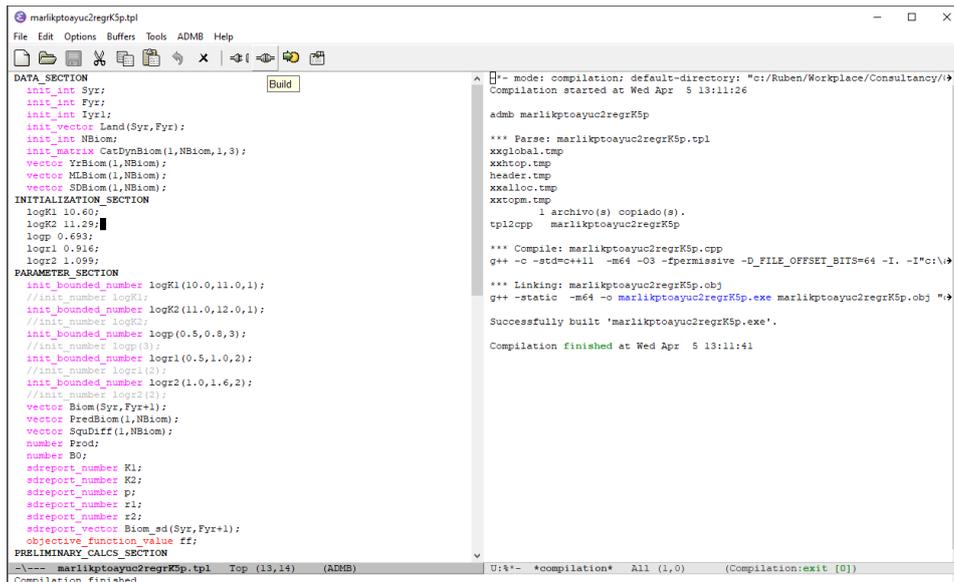


Figura 29: Continuación del proceso de optimización estadística en ADS-tudio con el botón 'Build' para el modelo de producción excedente de tres parámetros.



un fuente de poder (Fig. 29). El resultado se informa en el panel a la derecha del usuario. Debe terminar con un mensaje que contenga la palabra 'finished' en color verde.

En el último paso se presiona el botón 'Run' (tercer botón a su derecha de la barra del menú de íconos). El ícono es una flecha sobre un botón amarillo (Fig. 30). El resultado se informa en el panel a la derecha del usuario. Es en este punto en que el resultado deseable es que el proceso termine con el mensaje 'Process marlikptoayuc2regrK5p finished', que es el caso que se muestra en la Fig. 30.

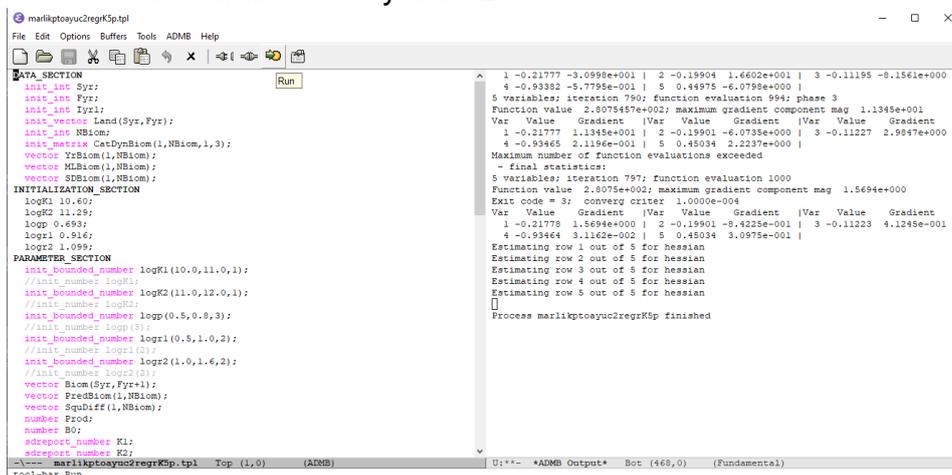
El usuario debe confirmar que la optimización fue exitosa examinando tres condiciones:

- que ninguno de los parámetros estimados dentro de bordes se haya estrellado contra una borde.
- que los gradientes de todos los parámetros son menores que 1: en el panel derecho de la Fig. 30 sobre las líneas repetidas 'Estimating row 1 ...' hay un reporte de 'final statistics' que tiene una tabla con tres columnas para cada parámetro (Var, Value y Gradient) y se debe verificar que todos los gradientes son menores que 1.
- que la matriz Hessiana sea positiva y definida: (a) ADMB no reporta nada y entonces la matriz es positiva y definida y se ha pasado este criterio, o (b) reporta que esta matriz no es positiva y definida y la optimización ha fallado, o (c) reporta que la matriz puede no ser positiva y definida y eso debe ser interpretado como una advertencia.

Para las tres condiciones hay excepciones. Ya se ha indicado arriba en qué casos es posible aceptar un resultado aunque uno o más de sus parámetros esté muy cerca de un borde. También es razonable relajar la demanda que todos los gradientes sean menores que 1 en aquellos casos en que sólo uno de los gradientes es mayor que 1 y no mucho mayor que 1 si no hay otras variantes de modelos que sean mejores en todos los sentidos. Por último hay matrices Hessianas que están en el borde de ser positivas y definidas y ADMB en esos casos envía una advertencia que dice que la matriz Hessiana *podría* no ser definida y positiva, y sin embargo el examen de la matriz de correlación revela que una gran parte de ella está bien condicionada, por lo que esta variante puede seguir siendo considerada.

En el caso del modelo de producción excedente con cambio en r y K vemos en la Fig. 30 que uno de los gradientes (el primero, correspondiente a K_1 tiene un gradiente un poco mayor que 1 y que los restantes gradientes son menores que 1. También apreciamos que la matriz Hessiana quedó completamente

Figura 30: Finalización del proceso de optimización estadística en AD Studio con el botón 'Run' para el modelo de producción excedente de cinco parámetros con cambios en r y K en 2015.



bien condicionada pues no hay reportes de error ni advertencias de ADMB respecto de esta matriz.

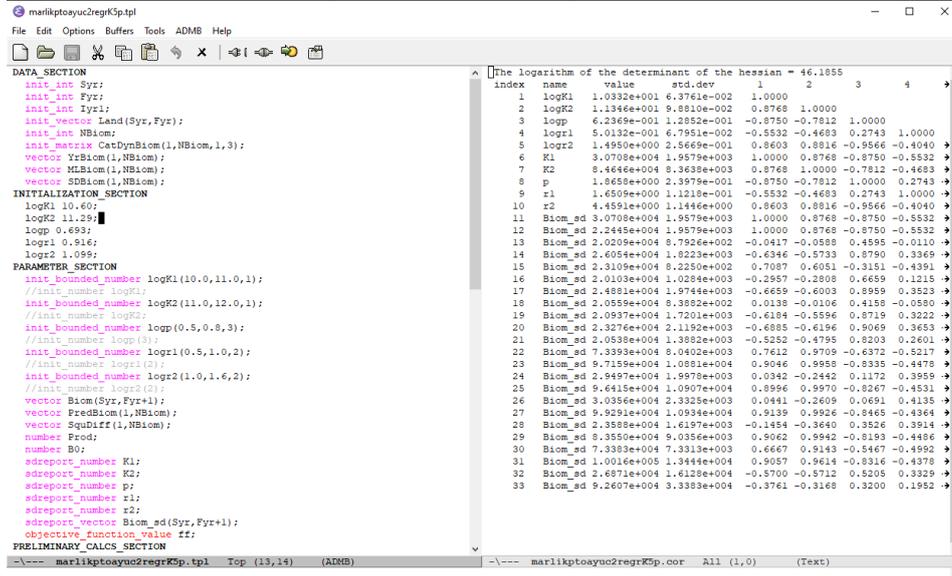
Al ejecutar el código ADMB como se indica en las Figs. 28, 29 y 30 el directorio de trabajo se va poblando con ficheros que tienen el mismo nombre que los ficheros .tpl y .dat pero que difieren en sus extensiones. Los resultados que nos interesan están en algunos de esos ficheros, en particular aquellos con extensión .par, .cor y .std. Para ver el fichero .cor en la interface gráfica de ADStudio se puede ir al menú y desplegar las opciones bajo 'ADMB'. Entre ellas está 'View Estimates'. Ejecutando esa opción se trae el fichero .cor al panel derecho de ADStudio (Fig. 31).

Se reportan en la primera columna los cinco parámetros originales ('logK1', 'logK2', 'logp', 'logr1' y 'logr2') y luego vienen los mismos parámetros pero en sus unidades originales y los estimados de biomasa desde el año 2000 hasta el año 2022, es decir hasta un año después del último año de observaciones. Junto a los parámetros estimados, en la segunda columna aparecen los errores estándar. Más a la derecha está la matriz de correlación.

En este despliegue, es importante observar si alguno de los parámetros se ha estrellado contra el borde comparando los valores estimado en logaritmo en el panel de la derecha con los bordes de cada parámetro en el panel de la izquierda. En este caso vemos que el único parámetro que está cerca de un borde es 'logr1' pero cualquier ampliación de los bordes en la línea

```
1 init_bounded_number logr1(0.5,1.0,2);
```

Figura 31: Despliegue de resultados del proceso exitoso de optimización en ADMB para el MPE de tres parámetros en el panel de AD Studio a su derecha.



haciendo que el borde inferior sea menor, produce una falla completa en la optimización, produciendo un reporte que dice:

```

1 Warning - Hessian does not appear to be positive definite
2 Hessian does not appear to be positive definite
3
4 Process marlikptoayuc2regrK5p exited abnormally with code 1

```

y además todos los gradientes pasan a ser mayor que 1.

Por otro lado, una clave definitiva que un parámetro se ha estrellado contra un borde, además de estar cerca del borde, es que su error estándar es sumamente pequeño mientras que en este caso, como se observa en la Fig. 31, el error estándar de 'logr1' (y de su transformada a unidades originales r_1) es realista, sólo un orden de magnitud menor que el estimado (alrededor de 8% de coeficiente de variación). Por lo tanto en este caso no debemos rechazar esta variante.

Los restantes cuatro modelos de producción excedente, uno con parámetros constantes y tres con hipótesis diferentes acerca de qué parámetros cambian en 2011, son ajustados de la misma manera y rinden resultados que deben ahora ser llevado a R para realizar la selección del mejor modelo y calcular, desde el mejor modelo, la tasa de cosecha sostenible dada la dinámica de poblaciones y productividad derivadas de los parámetros estimados.

5.2. Estimación de tasa de cosecha sostenible máxima en R

La última tarea de esta evaluación de stocks que combina modelos de depleción generalizados y modelos de producción excedente se realiza en R y para esto utilizamos el guión `Yucoct.Summary.R`. Este guión tiene los resultados de ADMB de ambas especies pero en este manual sólo vamos a describir los pasos para estimar la tasa de cosecha sostenible de *O. americanus*. En esta especie se observa un cambio similar al que experimenta *O. maya* en 2015 (Fig. 25) pero el aparente cambio ocurre antes, en 2011. Como se puede apreciar en el guión los procedimientos son idénticos para terminar la evaluación de stocks de *O. maya* cambiando los nombres de los objetos.

5.2.1. Selección de la mejor variante

La selección de la mejor variante de entre las cinco ajustadas en ADMB sigue los mismos criterios que la selección de variantes de modelos de depleción. Consideramos los gradientes, las correlaciones entre estimados, el AIC y el realismo biológico. El código para extraer los gradientes y la información necesaria para calcular los AIC se muestra aquí para las variantes con parámetros constantes y con cambios en r y K en 2011 como ejemplos:

```
1 # Fit 1
2 maxgrad.admb.pt.oayuc.1reg3p <- read.table("
3   marlikptoayuc1reg3p.par",comment.char="",nrows=1)$V16
4 # [1] 4.481148e-06
5 lik.admb.pt.oayuc.1reg3p <- read.table("
6   marlikptoayuc1reg3p.par",comment.char="",nrows=1)$V11
7 AIC.admb.pt.oayuc.1reg3p <- 2*3 - 2*lik.admb.pt.oayuc.1
8   reg3p
9 # [1] -764.0168
10 # Fit 3
11 maxgrad.admb.pt.oayuc.2regrK5p <- read.table("
12   marlikptoayuc2regrK5p.par",comment.char="",nrows=1)$V16
13 # [1] 1.569415
14 lik.admb.pt.oayuc.2regrK5p <- read.table("
15   marlikptoayuc2regrK5p.par",comment.char="",nrows=1)$V11
16 AIC.admb.pt.oayuc.2regrK5p <- 2*5 - 2*lik.admb.pt.oayuc.2
17   regrK5p
18 # [1] -551.5091
```

Creamos una tabla que reúne todos estos resultados de gradientes y AIC para las cinco variantes (Fig. 32). Sólo las variantes con parámetros constantes y el modelo con cambios en r y K en 2011 producen gradientes aceptables y la variante con parámetros constantes tiene un mucho mejor AIC.

Figura 32: Gradientes y AIC de las cinco hipótesis para la dinámica de poblaciones en biomasa de *O. americanus* en Yucatán y alrededores.

```
> pt.scenarios.oayuc.admb
```

	Scenario	Npars	MaxAbsGrad	AIC
1	No change	3	4.481148e-06	-764.0168
2	Change in r 2010-2011	4	1.709593e+04	-675.6222
3	Change in r and K 2010-2011	5	1.569415e+00	-551.5091
4	Change in r and p 2010-2011	5	5.633743e+05	-687.8398
5	Change in r, K and p 2010-2011	6	8.848894e+02	-539.1017

Figura 33: Correlaciones entre estimados de los modelos de producción excedente de la dinámica de poblaciones en biomasa de *O. americanus* en Yucatán y alrededores, variante con parámetros constantes versus variante con cambios en r y K en 2011.

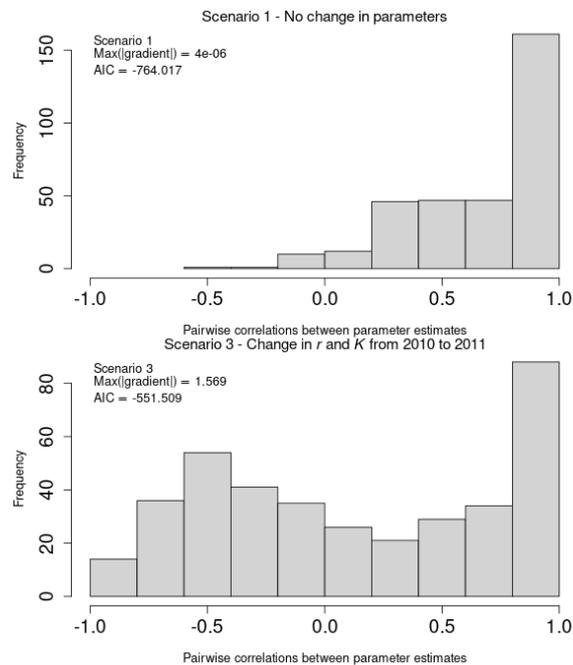


Figura 34: Estimados de los parámetros de las cinco variantes de modelos de producción excedente de la dinámica de poblaciones en biomasa de *O. americanus* en Yucatán y alrededores

```
> pt.pars.admb.oayuc
```

Scenario	Description	Par	MLE	SE	CV.pc	Pre.sel	Fin.sel
1	1	Null, no change	K 3.5269e+04	8.5392e-02	0.0	0	0
2	1	Null, no change	p 7.2872e+00	1.3942e+03	19132.2	0	0
3	1	Null, no change	r 3.6849e+00	1.3134e+00	35.6	0	0
4	2	Change in r 2010 to 2011	K 4.5398e+04	2.7880e+00	0.0	0	0
5	2	Change in r 2010 to 2011	p 1.4227e+00	1.8520e-04	0.0	0	0
6	2	Change in r 2010 to 2011	r1 1.0834e+00	2.0978e-04	0.0	0	0
7	2	Change in r 2010 to 2011	r2 1.1826e+01	2.1614e-02	0.2	0	0
8	3	Change in r and K 2010 to 2011	K1 3.0708e+04	1.9579e+03	6.4	1	1
9	3	Change in r and K 2010 to 2011	K2 8.4646e+04	8.3638e+03	9.9	1	1
10	3	Change in r and K 2010 to 2011	p 1.8658e+00	2.3979e-01	12.9	1	1
11	3	Change in r and K 2010 to 2011	r1 1.6509e+00	1.1218e-01	6.8	1	1
12	3	Change in r and K 2010 to 2011	r2 4.4591e+00	1.1446e+00	25.7	1	1
13	4	Change in r and p 2010 to 2011	K 4.1166e+04	3.1425e+01	0.1	0	0
14	4	Change in r and p 2010 to 2011	p1 2.4372e+00	2.4612e-03	0.1	0	0
15	4	Change in r and p 2010 to 2011	p2 4.1238e+00	4.7181e-04	0.0	0	0
16	4	Change in r and p 2010 to 2011	r1 5.9567e-01	9.4573e-05	0.0	0	0
17	4	Change in r and p 2010 to 2011	r2 2.2369e+00	2.8323e-04	0.0	0	0
18	5	Change in r, K and p 2010 to 2011	K1 3.1685e+04	1.1373e+03	3.6	0	0
19	5	Change in r, K and p 2010 to 2011	K2 7.8406e+04	2.8977e+03	3.7	0	0
20	5	Change in r, K and p 2010 to 2011	p1 2.0825e+00	2.2469e-02	1.1	0	0
21	5	Change in r, K and p 2010 to 2011	p2 1.9155e+00	5.6501e-05	0.0	0	0
22	5	Change in r, K and p 2010 to 2011	r1 1.3587e+00	8.7930e-02	6.5	0	0
23	5	Change in r, K and p 2010 to 2011	r2 4.3189e+00	1.1690e-04	0.0	0	0

Habiendo estrechado el cerco a dos variantes, examinamos sus correlaciones entre estimados. En este criterio la variante con cambios en r y K en 2011 es superior (Fig. 33). Enseguida creamos una tabla que reúne todos los estimados de los parámetros y sus errores estándar para juzgar el realismo biológico de las dos variantes que quedan en la lista corta (Fig. 34). La variante con parámetros constantes carece de realismo biológico porque el estimado de la simetría de la curva de producción es enorme, mayor que 7, y tiene un error estándar del orden de 10^3 . En esta tabla también se aprecia que la única variante que tiene todos sus coeficientes de variación con valores realistas es la variante con cambios en r y K en 2011. Por lo tanto la selección de modelos concluye seleccionando a la variante con cambios en r y K en 2011 como el mejor modelo de producción excedente para la dinámica de poblaciones en biomasa de *O. americanus* en Yucatán y alrededores.

5.2.2. Parámetros derivados y gráfica de la dinámica de poblaciones en biomasa

Entre los parámetros derivados están el máximo rendimiento sostenible,

$$MSY = rK(p - 1)p^{\frac{-p}{p-1}} \quad (7)$$

la biomasa que produce el máximo rendimiento sostenible,

$$B_{MSY} = Kp^{\frac{1}{1-p}} \quad (8)$$

y la productividad latente total,

$$TLP_y = \gamma MSY \frac{B_y}{K} \left(1 - \left(\frac{B_y}{K} \right)^{p-1} \right) + C_y, \quad \gamma = \frac{p^{p/(p-1)}}{p-1} \quad (9)$$

Como el modelo de dinámica de poblaciones en biomasa seleccionado es un modelo con parámetros que cambian en 2011, los parámetros derivados MSY , B_{MSY} y TLP deben ser calculados para cada uno de los dos períodos de la dinámica, que llamaremos el período pre-expansión y el período post-expansión, por razones se harán aparentes más adelante. Además de estimar estos parámetros derivados debemos también calcular sus errores estándar, lo que haremos con el método delta programado en **CatDyn**:

```

1 # MSY
2 MSY.oayuc.pre <- round(r1.oayuc*K1.oayuc*(p.oayuc-1)/(p.oayuc^(p.
  oayuc/(p.oayuc-1))))
3 #[1] 11447
4 MSY.SE.oayuc.pre <- round(deltamethod(g=list(~x3*x1*(x2-1)/(x2^(x2
  /(x2-1))))),
5                                     mean=c(K1.oayuc,p.oayuc,r1.
6                                     oayuc),
7                                     cov=pt.cov.oayuc.pre,
8                                     ses=TRUE))
9 #[1] 1945
10 MSY.oayuc.post <- round(r2.oayuc*K2.oayuc*(p.oayuc-1)/(p.oayuc^(p.
  oayuc/(p.oayuc-1))))
11 #[1] 85223
12 MSY.SE.oayuc.post <- round(deltamethod(g=list(~x3*x1*(x2-1)/(x2^(
  x2/(x2-1))))),
13                                     mean=c(K2.oayuc,p.oayuc,r2.
14                                     oayuc),
15                                     cov=pt.cov.oayuc.post,
16                                     ses=TRUE))
17 #[1] 15101
18 #
19 # B.MSY
20 B.MSY.oayuc.pre <- round(K1.oayuc*(p.oayuc^(1/(1-p.oayuc))))
21 #[1] 14942
22 B.MSY.SE.oayuc.pre <- round(deltamethod(g=list(~x1*x2^(1/(1-x2))),
23                                     mean=c(K1.oayuc,p.oayuc),
24                                     cov=pt.cov.oayuc.pre
25                                     [1:2,1:2],
26                                     ses=TRUE))

```

```

24 # [1] 467
25 #
26 B.MSY.oayuc.post <- round(K2.oayuc*(p.oayuc^(1/(1-p.oayuc))))
27 # [1] 41187
28 B.MSY.SE.oayuc.post <- round(deltamethod(g=list(~x1*x2^(1/(1-x2)))
29                                     ,
30                                     mean=c(K2.oayuc , p.oayuc) ,
31                                     cov=pt.cov.oayuc.pre
32                                     [1:2 , 1:2] ,
                                     ses=TRUE) )

```

Observamos que el estimado de *MSY* del período pre-expansión es muy razonable (11447 toneladas con un error estándar de 1945 toneladas) mientras que el estimado de *MSY* del período post-expansión es indudablemente excesivo (85223 toneladas con un error estándar de 15101 toneladas). La misma desconexión con el realismo se produce con el estimado de biomasa que produce el *MSY*, B_{MSY} . En el período pre-expansión el estimado es de 14942 toneladas con un error estándar de 467 toneladas mientras que para el período post-expansión el estimado de B_{MSY} es de 41187 toneladas con un error estándar de 1351 toneladas. Es decir que para el período de post-expansión, si creemos en la validez del *MSY* debemos creer que 41187 toneladas de pulpo producen un excedente de biomasa que puede ser extraído sosteniblemente de 85223 toneladas!

La causa de estos estimados carentes de credibilidad para el período post-expansión se aclara una vez que observamos la dinámica de poblaciones en biomasa del stock. El guión [Yucoct.Summary.R](#) contiene el código para crear un gráfico informativo de esta dinámica (Fig. 35). Entre 2000 y 2010 el stock tuvo una dinámica estable y dada esa dinámica el *MSY* es un estimado válido de la tasa de cosecha sostenible máxima. A partir de 2011 (marcado por una flecha) el stock crece pero al mismo tiempo entra en un régimen de fluctuaciones en el cual el *MSY* ya no es un estimado válido de la tasa de cosecha sostenible máxima puesto que el *MSY* no existe, matemática y conceptualmente, cuando los stocks carecen de un punto de equilibrio y en lugar de eso tienen un ciclo de equilibrio.

Por este motivo es necesario estimar un concepto más general de producción excedente que el *MSY* para generar una tasa de cosecha sostenible máxima para el período post-expansión. Este concepto es la productividad latente total *TLP*. La *TLP* es el caso general al cual el *MSY* pertenece como caso particular. Este caso particular es cuando el stock tiene una dinámica suave, en que pasa gradualmente de un punto de equilibrio a otro. En este caso el *MSY*, que requiere continuidad, es igual a la productividad latente

total promediada a través de muchos ciclos. Pero en los casos en que los stocks fluctúan fuertemente entre altas y bajas periódicas sólo la productividad latentes total promediada a través de muchos ciclos sigue siendo válida como tasa sostenible de cosecha, mientras que el *MSY* deja de existir.

En el guión [Yucoct.Summary.R](#) la productividad latente total y su promedio a través de los años son calculados con el siguientes código:

```

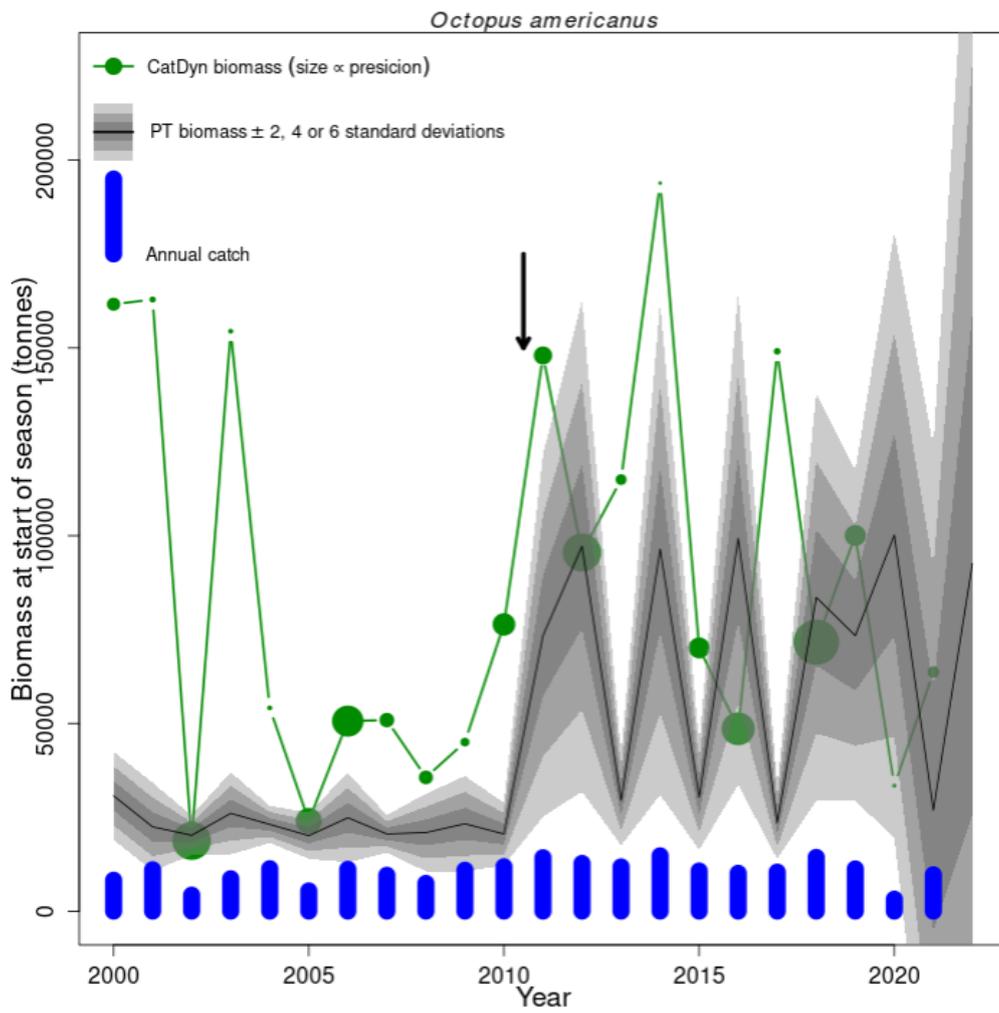
1 B.amer.pre.all <- Stock.admb.pt.oayuc.2regrK5p$
  Biomass.ton[Stock.admb.pt.oayuc.2regrK5p$EnvCycle %in% "PreExp"
  ]
2 B.amer.pre.all.sd <- Stock.admb.pt.oayuc.2regrK5p$SD
  .Biomass.ton[Stock.admb.pt.oayuc.2regrK5p$EnvCycle %in% "PreExp"
  ]
3 Lat.Prod.B.amer.pre.all <- r1.oayuc*B.amer.pre.all*(1-(B.
  amer.pre.all/K1.oayuc)^(p.oayuc-1))
4 Lat.Prod.B.amer.pre.all.Total <- Lat.Prod.B.amer.pre.all + oayuc
  .popdyn$Catch.ton[Stock.admb.pt.oayuc.2regrK5p$EnvCycle %in% "
  PreExp" ]
5 mean(Lat.Prod.B.amer.pre.all.Total)
6 #[1] 17075.63
7 B.amer.post.all <- Stock.admb.pt.oayuc.2regrK5p$
  Biomass.ton[Stock.admb.pt.oayuc.2regrK5p$EnvCycle %in% "PostExp"
  ]
8 B.amer.post.all.sd <- Stock.admb.pt.oayuc.2regrK5p$
  SD.Biomass.ton[Stock.admb.pt.oayuc.2regrK5p$EnvCycle %in% "
  PostExp" ]
9 Lat.Prod.B.amer.post.all <- r2.oayuc*B.amer.post.all*(1-(B
  .amer.post.all/K2.oayuc)^(p.oayuc-1))
10 Lat.Prod.B.amer.post.all.Total <- Lat.Prod.B.amer.post.all [1:11]
  + oayuc.popdyn$Catch.ton[Stock.admb.pt.oayuc.2regrK5p$EnvCycle
  %in% "PostExp" ][1:11]
11 mean(Lat.Prod.B.amer.post.all.Total)
12 #[1] 24179.41

```

Enseguida el guión [Yucoct.Summary.R](#) calcula los errores estándar de las productividades latentes anuales con un uso extensivo del método delta, lo que requiere tener cargado a [CatDyn](#).

Finalmente, creamos una tabla de resultados finales de tasas de cosecha. Incluimos el estimado de *MSY* del período pre-expansión porque es un estimado válido. Por lo tanto para la período pre-expansión hay dos estimados disponibles, aunque ya es un poco tarde para tomar acciones con estos resultados. Para el período post-expansión contamos con el estimado de productividad latente total promediada a través de los años. Cabe hacer notar que este estimado tiene un error estándar bastante alto, que da un coeficiente de variación de 80%. Por este motivo es muy recomendable rebajar el

Figura 35: Dinámica de poblaciones en biomasa de *O. americanus* en Yucatán y alrededores



estimado de tasa de cosecha sostenible máxima según un porcentaje del error estándar:

```
1 harvest.rate.oayuc <- data.frame(MSY.pre=MSY.oayuc.pre ,
2                                 MSY.pre.SE=MSY.SE.oayuc.pre ,
3                                 TLP.pre=mean(Lat.Prod.B.amer.pre.
4                                               all.Total) ,
5                                 TLP.pre.SE=mean(Lat.Prod.B.amer.
6                                               pre.all.SE) ,
7                                 TLP.post=mean(Lat.Prod.B.amer.
8                                               post.all.Total) ,
9                                 TLP.post.SE=mean(Lat.Prod.B.amer.
10                                                  post.all.SE[1:11] , na.rm=TRUE))
11 #   MSY.pre MSY.pre.SE TLP.pre TLP.pre.SE TLP.post TLP.post.SE
12 # 1   11447      1945 17075.63  1102.273 24179.41  19378.14
```

El error estándar de la productividad latente total promediada a través de los años del período post-expansión va a ir decreciendo a medida que se vayan añadiendo años a las series de tiempo de datos.

Con esto terminan los trabajos de evaluación de stocks de pulpos o de cualquier otra especie contenida en las bases de datos Avisos de Arribos de México.