

# Montaje de los archivos de entrada al modelo SS3, formatos requeridos y aprovechamiento de los utilizados en las entradas de a4a.

March, 30, 2023

## Contents

<b>1</b>	<b>Contexto</b>	<b>3</b>
<b>2</b>	<b>Tareas a desarrollar.</b>	<b>3</b>
2.1	Temporalización y dinámica de las actividades propuestas. . . . .	3
<b>3</b>	<b>Descripción general del modelo de merluza europea (<i>Merluccius merluccius</i>) de la GSA1 (mar de Alborán).</b>	<b>4</b>
3.1	Modelo conceptual . . . . .	4
3.2	Descripción general del modelo implementado en <b>a4a</b> . . . . .	4
<b>4</b>	<b>Revisión de los archivos requeridos para cada enfoque de modelación</b>	<b>5</b>
4.1	Archivos utilizado para enfoque de modelación <b>a4a</b> . . . . .	5
4.2	Archivos utilizado para enfoque de modelación <b>SS3</b> . . . . .	6
<b>5</b>	<b>Descripción del Formado de entrada de datos para cada enfoque de modelación</b>	<b>7</b>
5.1	Datos de entrada . . . . .	7
5.1.1	Información general del modelo Formato <b>a4a</b> . . . . .	7
5.1.2	Información general del modelo Formato <b>SS3</b> . . . . .	7
5.1.3	Capturas en formato <b>a4a</b> . . . . .	8
5.1.4	Capturas en formato <b>SS3</b> . . . . .	8
5.1.5	Indices de abundancia formato <b>a4a</b> . . . . .	11
5.1.6	Indices de abundancia formato <b>SS3</b> . . . . .	11
5.1.7	Descartes y tallas medias Formato <b>a4a</b> . . . . .	15
5.1.8	Descartes y tallas medias Formato <b>SS3</b> . . . . .	15
5.1.9	Composición de tallas Formato <b>a4a</b> . . . . .	16
5.1.10	Composición de tallas Formato <b>SS3</b> . . . . .	16
5.1.11	Composición de edad Formato <b>a4a</b> . . . . .	19
5.1.12	Composición de edad Formato <b>SS3</b> . . . . .	19
5.1.13	Otros datos . . . . .	23
5.1.14	Escribir archivo de datos modificado con la función <b>SS_write</b> para el enfoque de modelación <b>SS3</b> . . . . .	25
5.2	Archivo control.ss . . . . .	26
5.2.1	Especificaciones iniciales . . . . .	26
5.2.2	Datos de los archivos . . . . .	26
5.2.3	Especificaciones del crecimiento . . . . .	26
5.2.4	Distribución del reclutamiento . . . . .	26
5.2.5	Bloques . . . . .	27
5.2.6	Tiempo variable . . . . .	27
5.2.7	Parámetros biológicos . . . . .	28
5.2.8	Parámetros biológicos . . . . .	29
5.2.9	Relación stock recluta . . . . .	35

5.2.10	Desvíos de los reclutamientos . . . . .	36
5.2.11	Mortalidad por pesca . . . . .	37
5.2.12	Capturabilidad . . . . .	38
5.2.13	Selectividad . . . . .	38
5.2.14	Lambdas . . . . .	41
<b>6</b>	<b>GSA6 (Levante-Norte de España).</b>	<b>46</b>
6.1	Descripción del modelo . . . . .	46
6.2	Archivos utilizado para enfoque de modelación <b>a4a</b> . . . . .	47

# 1 Contexto

Se pretende desarrollar un modelado con Stock Synthesis de los stocks de merluza europea (*Merluccius merluccius*) de la GSA1 (mar de Alborán) y GSA6 (Levante-Norte de España). Ambos stocks se encuentran definidos en el GFCM en el Mediterráneo español y llevan varios años siendo evaluados con metodología analítica SCAA utilizando el modelo **a4a**, con evaluaciones validadas y aprobadas en el GFCM en los últimos años. Se trabajaría con la serie histórica actualizada en el último WGSAD de diciembre de 2022 (2002-2021).

Los dos objetivos principales serían por un lado Intentar testar el paso de un modelo **SCAA** a uno integrado en el entorno **GFCM** con stocks españoles, algo hasta el momento inédito, y por otro lado presentar los resultados al próximo Simposio de Modelado y Evaluación de Recursos Pesqueros (SIMERPE II) que se celebrará en Cádiz desde el 24 al 27 de octubre de 2023.

## 2 Tareas a desarrollar.

Dentro de la propuesta se han identificado las posibles tareas a desarrollar que serían por orden cronológico:

1. Montaje de los archivos de entrada al modelo SS3, formatos requeridos y aprovechamiento si fuera posible de los utilizados en las entradas de a4a.
2. Identificar tareas adicionales que requieran una recopilación de datos de entrada extra o con formatos diferentes a los usados en a4a.
3. Puesta a punto de ficheros de entrada SS3.
4. Primeras rodadas tentativas.
5. Análisis de resultados. Comparación con salidas XSA y a4a.
6. Presentación de los resultados en el SIMERPE II
7. Estudio de la viabilidad y/o idoneidad de estructurar la evaluación con estos stocks usando Stock Synthesis para los grupos de trabajo del GFCM el año que viene (WGSAD 2023).

### 2.1 Temporalización y dinámica de las actividades propuestas.

El flujo de trabajo tendría tres fases bien definidas, una inicial en la cual de forma telemática se completarían los puntos 1, 2 y 3 de la propuesta. Se propone completar esta fase antes del 31 de marzo de 2023.

En una segunda fase de desarrollo, se realizarían las primeras rodadas tentativas y se adquirirían las competencias necesarias para rodar el modelo de forma autónoma, al menos en sus requerimientos más básicos. Esta fase se realizaría en el C.O de Cádiz y Encarnación García y José Luis Pérez se desplazarían a este centro para tal fin para trabajar con María José Zúñiga bajo la supervisión de Margarita Rincón. Esta segunda fase tendría que estar completada el 30 de abril.

En una última fase se analizarían telemáticamente los resultados obtenidos y la viabilidad e idoneidad del paso de a4a a Stock Synthesis para los dos stocks evaluados. Esta fase tendría dos fechas diferenciadas, una a final de mayo para confeccionar y enviar los resultados al SIMERPE II y otra en noviembre para dejar lista la evaluación de ambos stocks para su posible presentación en el WGSAD-GFCM en diciembre de 2023.

### 3 Descripción general del modelo de merluza europea (*Merluccius merluccius*) de la GSA1 (mar de Alborán).

#### 3.1 Modelo conceptual

La conceptualización del modelo biológico de **GSA1** considera los siguientes componentes de la dinámica poblacional:

- Estructura geográfica
- Reproducción
- Reclutamiento
- Mortalidad natural
- Crecimiento
- otros...

#### 3.2 Descripción general del modelo implementado en **a4a**

El modelo de evaluación de stock de **GSA1** se basa en un análisis estadístico de la dinámica de estructuras de edad anual que incorpora información biológica y pesquera. La información que ingresa al modelo consiste en ...

- Años de evaluación de stock = 2003\_2021
- número de edades = 6 años ?

## 4 Revisión de los archivos requeridos para cada enfoque de modelación

### 4.1 Archivos utilizado para enfoque de modelación a4a

El modelo SCAA a4a está implementado en FLR ([https://flr-project.org/doc/Loading\\_your\\_data\\_into\\_FLR.html](https://flr-project.org/doc/Loading_your_data_into_FLR.html)), por lo tanto la arquitectura de sus ficheros input sigue se estructura en torno a la clase FLStock con los siguientes slots:

- Catch
- catch.n
- catch.wt
- discards
- discards.n
- discards.wt
- landings
- landings.n
- landings.wt
- stock
- stock.n
- stock.wt
- m
- mat
- harvest
- harvest.spwn
- m.spwn

No obstante, para facilitar la importación de los archivos input, usamos ficheros con formato .DAT que mediante el archivo índice “LOWIND.DAT” y la sentencia “stk <- readFLStock(“LOWIND.DAT”)”, nos lee nuestros ficheros creándonos el objeto FLStock.

A continuación describimos brevemente cada uno de los archivos .DAT que usamos como inputs:

- CATCH: Capturas anuales del stock (En nuestro caso GSA correspondiente). Toneladas.
- CATNUM: Matriz de número de individuos por edad/año de las capturas. (Miles de individuos).
- CATWT: Peso medio por edad y año de tu matriz de captura. (Kilos).
- STOCWT: medio por edad y año asumida para el stock (Normalmente = CATWT). (Kilos).
- TUNEFF: Fichero con los índices de abundancia para calibrar (CPUEs o Campañas, o ambas). Normalmente para índices de campaña empleamos número/KM2.
- NATMOR: Vector de mortalidad Natural por edad.
- PROPMAT: Ogiva de madurez por edad.
- PROPM: Factor de proporcionalidad de M antes de la puesta. Sería el slot m.spwn. Normalmente computamos 0.5 si la puesta es alrededor de Junio.
- PROP: Factor de proporcionalidad de F antes de la puesta. Sería el slot harvest.spwn. Normalmente computamos 0.5 si la puesta es alrededor de Junio.
- LOWIND: Fichero índice.
- INCHECK: Este no es un fichero input propiamente dicho. Se trata del fichero resumen tras realizar el slicing con el programa L2age. Se recogen aquí el set de parámetros de crecimiento utilizados.

```
dir_GSA1 <- here("hke-GSA1-a4a_format")
dir(dir_GSA1)
## [1] "CATCH.DAT"
## [2] "CATNUM.DAT"
## [3] "CATWT.DAT"
## [4] "Definiciones_Ficheros_imput_a4a.docx"
## [5] "INCHECK.TXT"
```

```
## [6] "LOWIND.DAT"
## [7] "NATMOR.DAT"
## [8] "PROPF.DAT"
## [9] "PROPM.DAT"
## [10] "PROPMAT.DAT"
## [11] "STOCWT.DAT"
## [12] "TUNEFF.DAT"
```

## 4.2 Archivos utilizado para enfoque de modelación SS3

1. Identificamos el directorio donde se encuentra el modelo base simple

```
dirname.base <- here("simple")
```

3. Creamos un nuevo directorio para la nueva versión del modelo modificado

```
dirname.simple_mod <- here("GSA6_SS3")
dir.create(path=dirname.simple_mod, showWarnings = TRUE, recursive = TRUE)
```

5. Copiamos los archivos para el modelo que vamos a modificar

```
copy_SS_inputs(dir.old = dirname.base,
               dir.new = dirname.simple_mod,
               copy_exe = TRUE,
               verbose = FALSE)
## [1] FALSE
```

## 5 Descripción del Formado de entrada de datos para cada enfoque de modelación

### 5.1 Datos de entrada

#### 5.1.1 Información general del modelo Formato a4a

Buscar esta información en archivos entregados!!!

#### 5.1.2 Información general del modelo Formato SS3

En la parte superior se especifica información general del modelo: los años del modelo, número de temporadas, número de sexos, edad máxima, número de áreas, número de flotas

[Consulte la Guía de usuario de SS3: Sección 7.5 “Model Dimensions”](#).

Revisamos los nombres de los componentes de la lista del archivo .dat

```
dat <- r4ss::SS_readdat(here(dirname.base,"data.ss")) #base
dat1<-dat # para modificar
#names(dat1) # muestra los objetos de la lista

#Especificaciones iniciales
dat1$styr <-2003 #_StartYr
dat1$endyr <-2021 #_EndYr
dat1$nseas <-1 #_Nseas
dat1$months_per_seas<-12 #_months/season
dat1$Nsubseasons <-2 #_Nsubseasons (even number, minimum is 2)
dat1$spawn_month <-6 #_spawn_month (puesta alrededor de junio)
dat1$Ngenders <-1 #_Ngenders: 1, 2, -1 (use -1 for 1 sex setup with SSB
# multiplied by female_frac parameter)

dat1$Nsexes <-1
dat1$Nages <-6 #_Nages=accumulator age, first age is always age 0
dat1$N_areas <-1 #_Nareas
dat1$Nfleets <-4 #_Nfleets (including surveys)
```

### 5.1.3 Capturas en formato a4a

Capturas anuales del stock (toneladas)

```
dir(dir_GSA1)[1]
## [1] "CATCH.DAT"
CATCH.DAT <- read.table(paste(dir_GSA1,dir(dir_GSA1)[1],sep="/"),
                        header=T,sep=" ",na="NA",fill=T,skip = 4)

CATCH.DAT
##      X5
## 1 353.4
## 2 464.5
## 3 322.9
## 4 333.0
## 5 310.2
## 6 293.4
## 7 570.7
## 8 530.8
## 9 652.7
## 10 447.7
## 11 337.7
## 12 248.2
## 13 175.1
## 14 170.5
## 15 295.7
## 16 425.4
## 17 274.5
## 18 170.4
## 19 287.5
```

### 5.1.4 Capturas en formato SS3

Primero ingresamos las especificaciones de los Datos de captura de la flota

[Consulte la Guía de usuario de SS3: Sección 7.9 “Catch”.](#)

#### 5.1.4.1 Especificaciones de datos de captura

```
#-----
#_fleet_type: 1=catch fleet; 2=bycatch only fleet; 3=survey; 4=ignore
#_sample_timing: -1 for fishing fleet to use season-long catch-at-age for
#observations, or 1 to use observation month; (always 1 for surveys)
#_fleet_area: area the fleet/survey operates in
#_units of catch: 1=bio; 2=num (ignored for surveys; their units read later)
#_catch_mult: 0=no; 1=yes
#_rows are fleets
#_fleet_type fishery_timing area catch_units need_catch_mult fleetname
#-----
# Arreglo de datos
fleetnames1<-c("FISHERY", "SURVEY1", "SURVEY2","SURVEY3")
type1<-c(1,3,3,3)
surveytiming1<-c(-1,1,1,1)
units_of_catch1<-c(1,2,2,2)
areas1<-c(1,1,1,1)
need_catch_mult1<-c(0,0,0,0)
#-----
```



```
# crear data.frame
fleetinfo1<-data.frame(type = type1,
                        surveytiming =surveytiming1,
                        area=areas1,
                        units=units_of_catch1,
                        need_catch_mult =need_catch_mult1,
                        fleetname=fleetnames1)

#-----
dat1$fleetinfo<-fleetinfo1
dat1$fleetinfo
##   type surveytiming area units need_catch_mult fleetname
## 1    1           -1    1      1              0  FISHERY
## 2    3            1    1      2              0  SURVEY1
## 3    3            1    1      2              0  SURVEY2
## 4    3            1    1      2              0  SURVEY3
#-----
```

#### 5.1.4.2 Datos de captura

```
#-----
#_Catch data: yr, seas, fleet, catch, catch_se
#_catch_se: standard error of log(catch)
#_NOTE: catch data is ignored for survey fleets
#-----
# Arreglo de Datos
year<-2003:2021
nyear<-length(year)
catch_year<-c(-999,year)
catch_seas<-rep(1,nyear+1)
catch_fleet<-rep(1,nyear+1)
catch_catch<-c(0,CATCH.DAT$X5)
catch_catch_se<-rep(0.01,nyear+1) # se asume cv = 0.01 Revisar!!!!
#-----
# crear data.frame
catch1<-data.frame(year=catch_year,
                   seas=catch_seas,
                   fleet=catch_fleet,
                   catch=catch_catch,
                   catch_se=catch_catch_se)

#-----
dat1$catch<-catch1
dat1$catch
##   year seas fleet catch catch_se
## 1  -999    1    1   0.0   0.01
## 2  2003    1    1 353.4   0.01
## 3  2004    1    1 464.5   0.01
## 4  2005    1    1 322.9   0.01
## 5  2006    1    1 333.0   0.01
## 6  2007    1    1 310.2   0.01
## 7  2008    1    1 293.4   0.01
## 8  2009    1    1 570.7   0.01
## 9  2010    1    1 530.8   0.01
## 10 2011    1    1 652.7   0.01
## 11 2012    1    1 447.7   0.01
```

```
## 12 2013      1      1 337.7      0.01
## 13 2014      1      1 248.2      0.01
## 14 2015      1      1 175.1      0.01
## 15 2016      1      1 170.5      0.01
## 16 2017      1      1 295.7      0.01
## 17 2018      1      1 425.4      0.01
## 18 2019      1      1 274.5      0.01
## 19 2020      1      1 170.4      0.01
## 20 2021      1      1 287.5      0.01
#-----
```

- La primera línea del fragmento de código anterior muestra los encabezados de columna para los datos de captura.
- Tenga en cuenta que toda la captura proviene de la pesquería. La línea `-999 1 1 0 0.01` especifica la captura de equilibrio para los años anteriores al inicio del modelo; en este caso, no hay captura de equilibrio porque la columna de captura es 0.

### 5.1.5 Índices de abundancia formato a4a

Índices de abundancia para calibrar (CPUEs o Campañas, o ambas). Normalmente para índices de campaña empleamos número/KM2.

```
dir(dir_GSA1)[12]
## [1] "TUNEFF.DAT"
TUNEFF.DAT<-read.table(paste(dir_GSA1,dir(dir_GSA1)[12],sep="/"),
                        sep=" ",na="NA",fill=T,skip = 6)

TUNEFF.DAT
##      V1      V2      V3      V4      V5      V6      V7
## 1  1 242.2  33.6  2.7 0.2 0.1 0.001
## 2  1 187.9  25.2  3.3 0.2 0.1 0.001
## 3  1 168.1  20.2  2.8 0.2 0.1 0.001
## 4  1 350.2  34.7  3.6 0.3 0.2 0.001
## 5  1 356.6  27.9  2.0 1.2 0.2 0.001
## 6  1 306.3  36.5  4.5 0.6 0.1 0.001
## 7  1 329.6  78.8  5.2 0.3 0.1 0.001
## 8  1 147.0 125.6 11.0 0.2 0.1 0.001
## 9  1 116.9  47.7  3.0 0.2 0.1 0.001
## 10 1  62.9  17.4  2.9 0.5 0.3 0.001
## 11 1  38.4  22.3  2.5 0.2 0.1 0.001
## 12 1 383.7  23.1  3.1 1.7 0.8 0.001
## 13 1 161.7  24.9  4.9 0.6 0.2 0.001
## 14 1 238.7  37.8  1.3 0.2 0.1 0.001
## 15 1  80.7  64.4  2.4 0.3 0.2 0.001
## 16 1 143.9  96.3  1.7 0.4 0.1 0.001
## 17 1  28.6  42.5  2.6 0.5 0.1 0.001
## 18 1  50.6  40.7  1.3 0.3 0.1 0.001
## 19 1 104.9  71.7  1.9 0.2 0.1 0.001
```

### 5.1.6 Índices de abundancia formato SS3

Luego viene la especificación de los índices de abundancia. Primero está la configuración para todas las flotas.

[Consulte la Guía de usuario de SS3: Sección 7.10 “Índices”.](#)

#### 5.1.6.1 Especificaciones de los índices de abundancia

```
#-----
#_CPUE_and_surveyabundance_observations
#_Units: 0=numbers; 1=biomass; 2=F; >=30 for special types
#_Errtype: -1=normal; 0=lognormal; >0=T
#_SD_Report: 0=no sdreport; 1=enable sdreport
#_Fleet Units Errtype SD_Report
#-----
# Arreglo de datos
CPUEinfo_Fleet<-c(1,2,3,4)
CPUEinfo_Units<-c(1,1,1,1) # unidades la dejamos en 1=biomass Revisar!!!
CPUEinfo_Errtype<-c(0,0,0,0) # en general se trabaja lognormal
CPUEinfo_SD_Report<-c(0,0,0,0) # esto se puede cambiar después si se necesita
# SD_Report, por ahora no es necesario...
CPUEinfo_names<-c("FISHERY", "SURVEY1", "SURVEY2", "SURVEY3")
#-----
# crear data.frame
CPUEinfo1<-data.frame(Fleet=CPUEinfo_Fleet,
```

```

Units=CPUEinfo_Units,
Errtype=CPUEinfo_Errtype,
SD_Report=CPUEinfo_SD_Report)

```

```

row.names(CPUEinfo1)<-CPUEinfo_names

```

```

#-----
dat1$CPUEinfo<-CPUEinfo1
dat1$CPUEinfo
##           Fleet Units Errtype SD_Report
## FISHERY      1      1      0      0
## SURVEY1      2      1      0      0
## SURVEY2      3      1      0      0
## SURVEY3      4      1      0      0
#-----

```

- Los encabezados de las columnas de esta sección están directamente encima de los números. Tenga en cuenta que aquí se definen todas las flotas (es decir, cada flota necesita una línea), incluida la pesquería, y se enumeran en el mismo orden que cuando se especificaron los tipos de flota.
- Lo más importante en esta sección es que se especifican las unidades y el tipo de error que se utilizará al leer los índices de abundancia.
- En este caso, la pesquería y las campañas tienen unidades de biomasa. Revisar que pasa si lo cambiamos a número. Se asume un error logarítmico normal para las 3 flotas.
- Inmediatamente después de su encabezado, se incluyen los datos de índices de abundancia:

### 5.1.6.2 Datos de índices de abundancia

```

#-----
# Arreglo de datos
yearf1<-2003:2021
yearf2<-2003:2021
yearf3<-2003:2021
CPUE_year<-c(yearf1,yearf2,yearf3)
#CPUE_seas = fecha e las campañas, se asume 1 (enero),
#corregir por el mes correspondiente
CPUE_seas<-rep(1,length(CPUE_year))
# Los números de "CPUE_index" son los mismos números de "Fleet"
# que se especifican en "CPUEinfo"
CPUE_index<-c(rep(2,length(yearf1)),
              rep(3,length(yearf2)),
              rep(4,length(yearf3)))
CPUE_obs<-c(TUNEFF.DAT$V2,
            TUNEFF.DAT$V3,
            TUNEFF.DAT$V4)
CPUE_se_log<-c(TUNEFF.DAT$V5,
              TUNEFF.DAT$V6,
              TUNEFF.DAT$V7)
#-----
# crear data.frame
CPUE1<-data.frame(year=CPUE_year,
                  seas=CPUE_seas,
                  index=CPUE_index,
                  obs=CPUE_obs,
                  se_log=CPUE_se_log)

```

```

#-----
dat1$CPUE<--CPUE1
dat1$CPUE
##      year seas index  obs se_log
## 1  2003     1     2 242.2  0.200
## 2  2004     1     2 187.9  0.200
## 3  2005     1     2 168.1  0.200
## 4  2006     1     2 350.2  0.300
## 5  2007     1     2 356.6  1.200
## 6  2008     1     2 306.3  0.600
## 7  2009     1     2 329.6  0.300
## 8  2010     1     2 147.0  0.200
## 9  2011     1     2 116.9  0.200
## 10 2012     1     2  62.9  0.500
## 11 2013     1     2  38.4  0.200
## 12 2014     1     2 383.7  1.700
## 13 2015     1     2 161.7  0.600
## 14 2016     1     2 238.7  0.200
## 15 2017     1     2  80.7  0.300
## 16 2018     1     2 143.9  0.400
## 17 2019     1     2  28.6  0.500
## 18 2020     1     2  50.6  0.300
## 19 2021     1     2 104.9  0.200
## 20 2003     1     3  33.6  0.100
## 21 2004     1     3  25.2  0.100
## 22 2005     1     3  20.2  0.100
## 23 2006     1     3  34.7  0.200
## 24 2007     1     3  27.9  0.200
## 25 2008     1     3  36.5  0.100
## 26 2009     1     3  78.8  0.100
## 27 2010     1     3 125.6  0.100
## 28 2011     1     3  47.7  0.100
## 29 2012     1     3  17.4  0.300
## 30 2013     1     3  22.3  0.100
## 31 2014     1     3  23.1  0.800
## 32 2015     1     3  24.9  0.200
## 33 2016     1     3  37.8  0.100
## 34 2017     1     3  64.4  0.200
## 35 2018     1     3  96.3  0.100
## 36 2019     1     3  42.5  0.100
## 37 2020     1     3  40.7  0.100
## 38 2021     1     3  71.7  0.100
## 39 2003     1     4   2.7  0.001
## 40 2004     1     4   3.3  0.001
## 41 2005     1     4   2.8  0.001
## 42 2006     1     4   3.6  0.001
## 43 2007     1     4   2.0  0.001
## 44 2008     1     4   4.5  0.001
## 45 2009     1     4   5.2  0.001
## 46 2010     1     4  11.0  0.001
## 47 2011     1     4   3.0  0.001
## 48 2012     1     4   2.9  0.001
## 49 2013     1     4   2.5  0.001

```

## 50 2014	1	4	3.1	0.001
## 51 2015	1	4	4.9	0.001
## 52 2016	1	4	1.3	0.001
## 53 2017	1	4	2.4	0.001
## 54 2018	1	4	1.7	0.001
## 55 2019	1	4	2.6	0.001
## 56 2020	1	4	1.3	0.001
## 57 2021	1	4	1.9	0.001
#-----				

### 5.1.7 Descartes y tallas medias Formato a4a

En este modelo no se ingresan datos de estructuras de tallas.

### 5.1.8 Descartes y tallas medias Formato SS3

A continuación, se podrían especificar los datos de descartes y tallas media.

[Consulte la Guía de usuario de SS3: Sección 7.11 “Discard”](#).

#### 5.1.8.1 Descarte

```
dat1$N_discard_fleets<-0 #_N_fleets_with_discard
#-----
#_discard_units (1=same_as_catchunits(bio/num);
#                2=fraction;
#                3=numbers)
#_discard_errtype: >0 for DF of T-dist(read CV below);
#                  0 for normal with CV;
#                  -1 for normal with se;
#                  -2 for lognormal;
#                  -3 for trunc normal with CV
# note: only enter units and errtype for fleets with discard
# note: discard data is the total for an entire season, so input of month
#       here must be to a month in that season
#_Fleet units errtype
#-----
# -9999 0 0 0.0 0.0 # terminator for discard data
```

#### 5.1.8.2 Tallas medias

```
dat1$use_meanbodywt<-0 #_use meanbodysize_data (0/1)
#-----
#_COND_0 #_DF_for_meanbodysize_T-distribution_like
# note: type=1 for mean length; type=2 for mean body weight
#_yr month fleet part type obs stderr
#-----
# -9999 0 0 0 0 0 0 # terminator for mean body size data
```

### 5.1.9 Composición de tallas Formato a4a

En este modelo no se ingresan datos de estructuras de tallas.

### 5.1.10 Composición de tallas Formato SS3

La siguiente sección configura los intervalos de talla (`length bin`) de la población.

Esto debe especificarse ya sea que se utilicen o no datos de composiciones de tallas (aunque podría generar los intervalos de longitud de la población a partir de los intervalos de datos de composiciones de tallas).

[Consulte la Guía de usuario de SS3: Sección 7.14 “Length Composition Data Structure”.](#)

#### 5.1.10.1 Bins tallas

```
# set up population length bin structure (note - irrelevant if not using size
#data and using empirical wtatage
dat1$lbmethod<-2      # length bin method:1=use databins;
                        # 2=generate from binwidth,min,max below;
                        # 3=read vector
dat1$binwidth<-2      # binwidth for population size comp
dat1$minimum_size<-10 # minimum size in the population
                        # (lower edge of first bin and size at age 0.00)
dat1$maximum_size<-70 # maximum size in the population (lower edge of last bin)
                        # Valor asumido, Revisar!!! aunque no se usan datos de tallas
                        # en este recurso.
dat1$use_lencomp<-0   # use length composition data (0/1)
```

Después de los intervalos de tallas de la población está la especificación para la composición de tallas (asumiendo 1 línea por flota):

#### 5.1.10.2 Especificación composición de tallas

```
#-----
#_mintailcomp: upper and lower distribution for females and males separately are
#               accumulated until exceeding this level.
#_addtocomp:   after accumulation of tails; this value added to all bins
#_combM+F:     males and females treated as combined gender below this bin number
#_compressbins: accumulate upper tail by this number of bins; acts simultaneous with
#               mintailcomp; set=0 for no forced accumulation
#_Comp_Error:  0=multinomial,
#               1=dirichlet using Theta*n,
#               2=dirichlet using beta,
#               3=MV_Tweedie
#_ParmSelect:  consecutive index for dirichlet or MV_Tweedie
#_minsamplesize: minimum sample size; set to 1 to match 3.24, minimum value is 0.001
#
#_mintailcomp addtocomp combM+F CompressBins CompError ParmSelect minsamplesize
#-----
# Arreglo de datos
len_info_mintailcomp<-rep(-1,4)
len_info_addtocomp<-rep(0.001,4)
len_info_combine_M_F<-rep(0,4)
len_info_CompressBins<-rep(0,4)
len_info_CompError<-rep(0,4)
len_info_ParmSelect<-rep(0,4)
len_info_minsamplesize<-rep(1,4)
```



```

#-----
# crear data.frame
len_info1<-data.frame(mintailcomp=len_info_mintailcomp,
                      addtocomp=len_info_addtocomp,
                      combine_M_F=len_info_combine_M_F,
                      CompressBins=len_info_CompressBins,
                      CompError=len_info_CompError,
                      ParmSelect=len_info_ParmSelect,
                      minsamplesize=len_info_minsamplesize)

row.names(len_info1)<-c("FISHERY", "SURVEY1", "SURVEY2","SURVEY3")
#-----
dat1$len_info<-len_info1 #data.frame
dat1$len_info
##          mintailcomp addtocomp combine_M_F CompressBins CompError ParmSelect
## FISHERY           -1    0.001           0           0           0           0
## SURVEY1           -1    0.001           0           0           0           0
## SURVEY2           -1    0.001           0           0           0           0
## SURVEY3           -1    0.001           0           0           0           0
##          minsamplesize
## FISHERY              1
## SURVEY1              1
## SURVEY2              1
## SURVEY3              1
#-----

```

### 5.1.10.3 Especificación del vector de tallas

```

dat1$N_lbins<-26
dat1$lbins_vector<-seq(20,70,2)

```

### 5.1.10.4 Datos de composición de tallas

```

#-----
# sex codes:  0=combined;
#             1=use female only;
#             2=use male only;
#             3=use both as joint sexlength distribution
# partition codes: (0=combined;
#                  1=discard;
#                  2=retained
#-----
# Arreglo de datos
new_lencomp <- data.frame(Yr = "# -9999",
                          Seas = 0,
                          FltSvy = 0,
                          Gender = 0,
                          Part = 0,
                          Nsamp = 0)

dat_rows_names <- paste("L",seq(20,70,2),sep="")
dat_rows <- as.data.frame(matrix(data = 0,
                                nrow = nrow(new_lencomp),
                                ncol = length(dat_rows_names)))

```

```

names(dat_rows)<-dat_rows_names
#-----
# crear data.frame
new_lencomp1<-cbind(new_lencomp, dat_rows)
#-----
dat1$lencomp<-new_lencomp1
dat1$lencomp
##      Yr Seas FltSvy Gender Part Nsamp L20 L22 L24 L26 L28 L30 L32 L34 L36 L38
## 1 # -9999    0      0      0    0    0    0    0    0    0    0    0    0    0    0    0
##   L40 L42 L44 L46 L48 L50 L52 L54 L56 L58 L60 L62 L64 L66 L68 L70
## 1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
#-----

```

### 5.1.11 Composición de edad Formato a4a

Matriz de número de individuos por edad/año de las capturas (Miles de individuos)

```
dir(dir_GSA1)[2]
## [1] "CATNUM.DAT"
CATNUM.DAT<-read.table(paste(dir_GSA1,dir(dir_GSA1)[2],sep="/"),
                        sep=" ",na="NA",fill=T,skip = 5)

CATNUM.DAT
##      V1      V2      V3      V4      V5      V6
## 1  746.7 1502.0 216.0 44.0 2.1 0.5
## 2 1425.9 2837.4 231.9 20.5 1.7 0.5
## 3  212.8 1132.5 277.0 36.6 1.2 0.7
## 4 2848.6  861.6 256.2 30.9 1.2 1.4
## 5  421.0 1409.5 163.7 25.2 4.6 1.2
## 6  306.3  986.0 208.9 28.1 4.7 1.2
## 7  650.6 2902.1 291.0 37.6 2.1 0.8
## 8  197.6 1962.0 404.5 18.4 2.2 0.3
## 9  278.9 3410.3 324.9 15.8 1.1 0.1
## 10 133.8 2464.1 198.4 11.7 0.7 0.1
## 11  36.6 1360.9 223.6 19.4 0.3 0.1
## 12  48.2 1086.1 136.4 13.3 0.3 0.2
## 13  61.7  791.1 106.1  7.7 0.7 0.3
## 14  79.5  989.3  78.1  2.7 0.1 0.1
## 15 809.8 1750.2 126.9  4.6 0.2 0.1
## 16 754.5 2320.0 189.8 13.7 0.3 0.1
## 17 107.4 1171.8 197.6  5.5 0.2 0.1
## 18 123.8  996.9  78.3  3.0 0.3 0.1
## 19 731.8 2299.1  87.5  4.6 0.1 0.1
```

### 5.1.12 Composición de edad Formato SS3

A continuación se presentan los datos de composición por edad. En primer lugar, se establecen las categorías de edad y las definiciones de error de edad.

[Consulte la Guía de usuario de SS3: Sección 7.16 “Age Composition Option”.](#)

#### 5.1.12.1 Bins de edad

```
dat1$N_agebins<-6
dat1$N_agebins
## [1] 6
dat1$agebin_vector<-seq(0,5,1)
dat1$agebin_vector
## [1] 0 1 2 3 4 5
```

#### 5.1.12.2 Datos de error edad

```
# Arreglo de datos
matrix_ageerror<-rbind(rep(-1,26),rep(0.001,26))
#-----
dat1$N_ageerror_definitions<-1 #_N_ageerror_definitions
dat1$N_ageerror_definitions
## [1] 1
dat1$ageerror<-matrix_ageerror #_ageerror_definitions
dat1$ageerror
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,] -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000
## [2,]  0.001  0.001  0.001  0.001  0.001  0.001  0.001  0.001  0.001  0.001
##      [,11] [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20]
## [1,] -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000
## [2,]  0.001  0.001  0.001  0.001  0.001  0.001  0.001  0.001  0.001  0.001
##      [,21] [,22] [,23] [,24] [,25] [,26]
## [1,] -1.000 -1.000 -1.000 -1.000 -1.000 -1.000
## [2,]  0.001  0.001  0.001  0.001  0.001  0.001
```

### 5.1.12.3 Especificaciones de los datos de composicion de edad

```
#-----
#_mintailcomp: upper and lower distribution for females and males separately are
#               accumulated until exceeding this level.
#_addtocomp:   after accumulation of tails; this value added to all bins
#_combM+F:    males and females treated as combined gender below this bin number
#_compressbins: accumulate upper tail by this number of bins; acts simultaneous
#               with mintailcomp; set=0 for no forced accumulation
#_Comp_Error:  0=multinomial,
#               1=dirichlet using Theta*n,
#               2=dirichlet using beta,
#               3=MV_Tweedie
#_ParmSelect:  consecutive index for dirichlet or MV_Tweedie
#_minsamplesize: minimum sample size; set to 1 to match 3.24, minimum value is 0.001
#
#_mintailcomp addtocomp combM+F CompressBins CompError ParmSelect minsamplesize
#-----
# crear data.frame
age_info1<-data.frame(mintailcomp=rep(-1,4),
                      addtocomp=rep(0.001,4),
                      combine_M_F=rep(0,4),
                      CompressBins=rep(0,4),
                      CompError=rep(0,4),
                      ParmSelect=rep(0,4),
                      minsamplesize=rep(1,4))

row.names(age_info1)<-c("FISHERY", "SURVEY1", "SURVEY2", "SURVEY3")
#-----
dat1$age_info<-age_info1
dat1$age_info
##      mintailcomp addtocomp combine_M_F CompressBins CompError ParmSelect
## FISHERY         -1      0.001          0           0           0           0
## SURVEY1         -1      0.001          0           0           0           0
## SURVEY2         -1      0.001          0           0           0           0
## SURVEY3         -1      0.001          0           0           0           0
##      minsamplesize
## FISHERY           1
## SURVEY1           1
## SURVEY2           1
## SURVEY3           1
#-----
```

#### 5.1.12.4 Datos de composicion de edad

```
#-----
dat1$Lbin_method <- 1 #_Lbin_method_for_Age_Data: 1=poplenbins; 2=datalenbins; 3=lengths
#-----
# sex codes: 0=combined;
#             1=use female only;
#             2=use male only;
#             3=use both as joint sexlength distribution
# partition codes: (0=combined;
#                  1=discard;
#                  2=retained
#_yr month fleet sex part ageerr Lbin_lo Lbin_hi Nsamp datavector(female-male)
#-----
# Arreglo de datos
new_agecomp<-data.frame(Yr=2003:2021,
                        Seas=7,
                        FltSvy=-1,
                        Gender=3,
                        Part=0,
                        Ageerr=2,
                        Lbin_lo=-1,
                        Lbin_hi=-1,
                        Nsamp=75)

dat_rows_names<-paste("E",seq(0,5,1),sep="")

#dat_rows<-as.data.frame(matrix(data = CATNUM.DAT,
#                               nrow = nrow(new_agecomp),
#                               ncol = length(dat_rows_names)))

names(CATNUM.DAT)<-dat_rows_names

new_agecomp<-cbind(new_agecomp, CATNUM.DAT)
#-----
dat1$agecomp<-new_agecomp
dat1$agecomp
##      Yr Seas FltSvy Gender Part Ageerr Lbin_lo Lbin_hi Nsamp      E0      E1
## 1  2003    7    -1     3     0      2     -1     -1    75  746.7 1502.0
## 2  2004    7    -1     3     0      2     -1     -1    75 1425.9 2837.4
## 3  2005    7    -1     3     0      2     -1     -1    75  212.8 1132.5
## 4  2006    7    -1     3     0      2     -1     -1    75 2848.6  861.6
## 5  2007    7    -1     3     0      2     -1     -1    75  421.0 1409.5
## 6  2008    7    -1     3     0      2     -1     -1    75  306.3  936.0
## 7  2009    7    -1     3     0      2     -1     -1    75  650.6 2902.1
## 8  2010    7    -1     3     0      2     -1     -1    75  197.6 1962.0
## 9  2011    7    -1     3     0      2     -1     -1    75  278.9 3410.3
## 10 2012    7    -1     3     0      2     -1     -1    75  133.8 2464.1
## 11 2013    7    -1     3     0      2     -1     -1    75   36.6 1360.9
## 12 2014    7    -1     3     0      2     -1     -1    75   48.2 1086.1
## 13 2015    7    -1     3     0      2     -1     -1    75   61.7  791.1
## 14 2016    7    -1     3     0      2     -1     -1    75   79.5  989.3
## 15 2017    7    -1     3     0      2     -1     -1    75  809.8 1750.2
```

```

## 16 2018      7      -1      3      0      2      -1      -1      75  754.5 2320.0
## 17 2019      7      -1      3      0      2      -1      -1      75  107.4 1171.8
## 18 2020      7      -1      3      0      2      -1      -1      75  123.8  996.9
## 19 2021      7      -1      3      0      2      -1      -1      75  731.8 2299.1
##          E2    E3   E4   E5
## 1  216.0 44.0 2.1 0.5
## 2  231.9 20.5 1.7 0.5
## 3  277.0 36.6 1.2 0.7
## 4  256.2 30.9 1.2 1.4
## 5  163.7 25.2 4.6 1.2
## 6  208.9 28.1 4.7 1.2
## 7  291.0 37.6 2.1 0.8
## 8  404.5 18.4 2.2 0.3
## 9  324.9 15.8 1.1 0.1
## 10 198.4 11.7 0.7 0.1
## 11 223.6 19.4 0.3 0.1
## 12 136.4 13.3 0.3 0.2
## 13 106.1  7.7 0.7 0.3
## 14  78.1  2.7 0.1 0.1
## 15 126.9  4.6 0.2 0.1
## 16 189.8 13.7 0.3 0.1
## 17 197.6  5.5 0.2 0.1
## 18  78.3  3.0 0.3 0.1
## 19  87.5  4.6 0.1 0.1
#-----

```

### 5.1.13 Otros datos

#### 5.1.13.1 Tallas medias a la edad

```
#-----
dat1$use_MeanSize_at_Age_obs <- 0 #_Use_MeanSize-at-Age_obs (0/1)
dat1$use_MeanSize_at_Age_obs
## [1] 0
#-----
# sex codes: 0=combined;
#             1=use female only;
#             2=use male only;
#             3=use both as joint sexlength distribution
# partition codes: (0=combined;
#                   1=discard;
#                   2=retained
# ageerr codes: positive means mean length-at-age; negative means mean bodywt_at_age
#_yr month fleet sex part ageerr ignore datavector(female-male)
#                                     samplesize(female-male)
#-----
# Arreglo de datos
MeanSize_at_Age_obs_esp<-data.frame(Yr="# -9999",
                                     Seas=0,
                                     FltSvy=0,
                                     Gender=0,
                                     Part=0,
                                     AgeErr=0,
                                     Ignore=0)

dat_rows_names <- c(paste("L",seq(0,5,1),sep=""),paste("nmL",seq(0,5,1),sep="")) # los datos de esta ma
dat_rows <- as.data.frame(matrix(data = 0,
                                nrow = nrow(MeanSize_at_Age_obs_esp),
                                ncol = length(dat_rows_names)))
names(dat_rows)<-dat_rows_names
#-----
# crear data.frame
MeanSize_at_Age_obs1 <-cbind(MeanSize_at_Age_obs_esp, dat_rows)
#-----
dat1$MeanSize_at_Age_obs<-MeanSize_at_Age_obs1
dat1$MeanSize_at_Age_obs
##           Yr Seas FltSvy Gender Part AgeErr Ignore L0 L1 L2 L3 L4 L5 nmL0 nmL1
## 1 # -9999    0      0      0    0      0      0  0  0  0  0  0  0  0  0
##   nmL2 nmL3 nmL4 nmL5
## 1    0    0    0    0
#-----
```

#### 5.1.13.2 Datos ambientales

(tarea = buscar ejemplo donde ingresen estos datos)

```
dat1$N_environ_variables<-0 #_N_environ_variables
dat1$N_environ_variables
## [1] 0
# -2 in yr will subtract mean for that env_var;
# -1 will subtract mean and divide by stddev (e.g. Z-score)
#Yr Variable Value
```

```
#
```

### 5.1.13.3 Sizefreq data

(tarea = buscar ejemplo donde ingresen estos datos)

```
# Sizefreq data. Defined by method because a fleet can use multiple methods
dat1$N_sizefreq_methods<-0 # N sizefreq methods to read (or -1 for expanded options)
dat1$N_sizefreq_methods
## [1] 0
```

### 5.1.13.4 Datos de tags

(tarea = buscar ejemplo donde ingresen estos datos)

```
dat1$do_tags<-0 # do tags (0/1/2); where 2 allows entry of TG_min_recap
dat1$do_tags
## [1] 0
```

### 5.1.13.5 Datos de morfos

(tarea = buscar ejemplo donde ingresen estos datos)

```
dat1$morphcomp_data<-0 # morphcomp data(0/1)
dat1$morphcomp_data
## [1] 0
# Nobs, Nmorphs, mincomp
# yr, seas, type, partition, Nsamp, datavector_by_Nmorphs
#
```

#### 5.1.13.5.1 Priors de selectividad

(tarea = buscar ejemplo donde ingresen estos datos)

```
dat1$use_selectivity_priors<-0 # Do dataread for selectivity priors(0/1)
dat1$use_selectivity_priors
## [1] 0
# Yr, Seas, Fleet, Age/Size, Bin, selex_prior, prior_sd
# feature not yet implemented
#
```

### 5.1.13.6 Final de archivo data.ss

imprime línea final

```
dat1$eof<-TRUE
dat1$eof
## [1] TRUE
```



#### 5.1.14 Escribir archivo de datos modificado con la función SS\_write para el enfoque de modelación SS3

```
r4ss::SS_writedat(dat1,outfile=here(dirname.simple_mod,"data.ss"),overwrite = TRUE)
```

## 5.2 Archivo control.ss

Revisamos los nombres de los componentes de la lista del archivo .dat

```
ctl <- r4ss::SS_readctl(here(dirname.base, "control.ss"),
                        verbose = FALSE,
                        use_datlist = TRUE,
                        datlist = dat) #base
ctl1<-ctl # para modificar
#names(ctl1) # muestra los objetos de la lista
```

### 5.2.1 Especificaciones iniciales

```
ctl1$nseas<-1
ctl1$N_areas<-1
ctl1$Nages<-40
ctl1$Nsexes<-2
ctl1$Npopbins<-43
ctl1$Nfleets<-3
ctl1$Do_AgeKey<-0
```

### 5.2.2 Datos de los archivos

```
ctl1$fleetnames <- c("FISHERY", "SURVEY1", "SURVEY2")
ctl1$eof<-TRUE
```

### 5.2.3 Especificaciones del crecimiento

```
ctl1$EmpiricalWAA<-0 # 0 means do not read wtatage.ss;
                      # 1 means read and use wtatage.ss and also read and use growth parameters
ctl1$N_GP<-1          #_N_Growth_Patterns (Growth Patterns, Morphs, Bio Patterns, GP are terms used int
ctl1$N_platoon<-1      #_N_platoons_Within_GrowthPattern
```

### 5.2.4 Distribución del reclutamiento

```
ctl1$recr_dist_method # recr_dist_method for parameters: 2=main effects for GP, Area, Settle timing; 3=
## [1] 2
ctl1$recr_global_area # not yet implemented; Future usage: Spawner-Recruitment: 1=global; 2=by area
## [1] 1
ctl1$recr_dist_read # number of recruitment settlement assignments
## [1] 1
ctl1$recr_dist_inx # unused option
## [1] 0
ctl1$recr_dist_pattern #GPattern month area age (for each settlement assignment)
##          GPattern month area age
## recr_dist_pattern1      1      1      1      0
#-----
#
#_Cond 0 # N_movement_definitions goes here if Nareas > 1
#_Cond 1.0 # first age that moves (real age at begin of season, not integer) also cond on do_migration>
#_Cond 1 1 1 2 4 10 # example move definition for seas=1, morph=1, source=1 dest=2, age1=4, age2=10
#
```

```
#-----
```

### 5.2.5 Bloques

```
ctl$N_Block_Designs #_Nblock_Patterns
## [1] 1
ctl$blocks_per_pattern #_blocks_per_pattern
## blocks_per_pattern_1
## 1
ctl$Block_Design # begin and end years of blocks
## [[1]]
## [1] 1970 1970
```

### 5.2.6 Tiempo variable

```
#-----
#
# controls for all timevary parameters
ctl$time_vary_adjust_method #_time-vary parm bound check (
## [1] 1
# 1=warn relative to base parm bounds;
# 3=no bound check);
# Also see env (3) and dev (5) options to constrain with base bounds
#-----
#
# AUTOGEN
ctl$time_vary_auto_generation # autogen: 1st element for biology,
## time_vary_auto_generation_1 time_vary_auto_generation_2
## 0 0
## time_vary_auto_generation_3 time_vary_auto_generation_4
## 0 0
## time_vary_auto_generation_5
## 0
# 2nd for SR,
# 3rd for Q,
# 4th reserved,
# 5th for selez
# where: 0 = autogen time-varying parms of this category;
# 1 = read each time-varying parm line;
# 2 = read then autogen if parm min== -12345
#
#-----
#_Available timevary codes
#_Block types:
# 0: P_block=P_base*exp(TVP);
# 1: P_block=P_base+TVP;
# 2: P_block=TVP;
# 3: P_block=P_block(-1) + TVP

#_Block_trends:
# -1: trend bounded by base parm min-max and parms in transformed units (beware);
# -2: endtrend and infl_year direct values;
# -3: end and infl as fraction of base range
```

```

#_EnvLinks:
# 1:  $P(y) = P_{\text{base}} \cdot \exp(TVP \cdot \text{env}(y))$ ;
# 2:  $P(y) = P_{\text{base}} + TVP \cdot \text{env}(y)$ ;
# 3:  $P(y) = f(TVP, \text{env\_Zscore})$  w/ logit to stay in min-max;
# 4:  $P(y) = 2.0 / (1.0 + \exp(-TVP1 \cdot \text{env}(y) - TVP2))$ 

#_DevLinks:
# 1:  $P(y) \cdot \exp(\text{dev}(y) \cdot \text{dev\_se})$ ;
# 2:  $P(y) + \text{dev}(y) \cdot \text{dev\_se}$ ;
# 3: random walk;
# 4: zero-reverting random walk with  $\rho$ ;
# 5: like 4 with logit transform to stay in base min-max

#_DevLinks(more):
# 21-25 keep last dev for rest of years
#
#_Prior_codes:
# 0=none;
# 6=normal;
# 1=symmetric beta;
# 2=CASAL's beta;
# 3=lognormal;
# 4=lognormal with biascorr;
# 5=gamma
#

```

## 5.2.7 Parámetros biológicos

```

# setup for M, growth, wt-len, maturity, fecundity, (hermaphro), recr_distr, cohort_grow, (movement), (
#_NATMORT
ctl$natM_type #_natM_type:_0=1Parm;
## [1] 0

#
# _1=N_breakpoints;
# _2=Lorenzen;
# _3=agespecific;
# _4=agespec_withseasinterpolate;
# _5=BETA:_Maunder_link_to_maturity;
# _6=Lorenzen_range
#_no additional input for selected M option; read 1P per morph
#
ctl$GrowthModel # GrowthModel: 1=vonBert with L1&L2; 2=Richards with L1&L2; 3=age_specific_K_incr; 4=ag
## [1] 1

ctl$Growth_Age_for_L1 #_Age(post-settlement)_for_L1;linear growth below this
## [1] 0
ctl$Growth_Age_for_L2 #_Growth_Age_for_L2 (999 to use as Linf)
## [1] 25
ctl$Exp_Decay #_exponential decay for growth above maxage (value should approx initial Z; -999 replicat
## [1] -999

ctl$Growth_Placeholder #_placeholder for future growth feature
## [1] 0
ctl$SD_add_to_LAA #_SD_add_to_LAA (set to 0.1 for SS2 V1.x compatibility)

```

```
## [1] 0
ctl$CV_Growth_Pattern #_CV_Growth_Pattern: 0 CV=f(LAA); 1 CV=F(A); 2 SD=F(LAA); 3 SD=F(A); 4 logSD=F(A)
## [1] 0
#
ctl$maturity_option #_maturity_option: 1=length logistic; 2=age logistic; 3=read age-maturity matrix b
## [1] 1

ctl$First_Mature_Age #_First_Mature_Age
## [1] 1
ctl$fecundity_option #_fecundity_at_length option: (1)eggs=Wt*(a+b*Wt); (2)eggs=a*L^b; (3)eggs=a*Wt^b; (4)
## [1] 1

ctl$hermaphroditism_option #_hermaphroditism option: 0=none; 1=female-to-male age-specific fxn; -1=mal
## [1] 0

ctl$parameter_offset_approach #_parameter_offset_approach for M, G, CV_G: 1- direct, no offset**; 2- m
## [1] 1

#** in option 1, any male parameter with value = 0.0 and phase <0 is set equal to female parameter
#
```

## 5.2.8 Parámetros biológicos

```
#_growth_parms
#_ LO HI INIT PRIOR PR_SD PR_type PHASE env_var&link dev_link dev_minyr dev_maxyr dev_PH Block Block_Fx
# Sex: 1 BioPattern: 1 NatMort
ctl$MG_parms
##          LO          HI          INIT          PRIOR PR_SD PR_type
## NatM_p_1_Fem_GP_1 5e-02 0.150000 0.10000000 0.10000000 0.8 0
## L_at_Amin_Fem_GP_1 -1e+01 45.000000 21.65350000 36.00000000 10.0 6
## L_at_Amax_Fem_GP_1 4e+01 90.000000 71.64930000 70.00000000 10.0 6
## VonBert_K_Fem_GP_1 5e-02 0.250000 0.14729700 0.15000000 0.8 6
## CV_young_Fem_GP_1 5e-02 0.250000 0.10000000 0.10000000 0.8 0
## CV_old_Fem_GP_1 5e-02 0.250000 0.10000000 0.10000000 0.8 0
## Wtlen_1_Fem_GP_1 -3e+00 3.000000 0.00000244 0.00000244 0.8 0
## Wtlen_2_Fem_GP_1 -3e+00 4.000000 3.34694000 3.34694000 0.8 0
## Mat50%_Fem_GP_1 5e+01 60.000000 55.00000000 55.00000000 0.8 0
## Mat_slope_Fem_GP_1 -3e+00 3.000000 -0.25000000 -0.25000000 0.8 0
## Eggs_alpha_Fem_GP_1 -3e+00 3.000000 1.00000000 1.00000000 0.8 0
## Eggs_beta_Fem_GP_1 -3e+00 3.000000 0.00000000 0.00000000 0.8 0
## NatM_p_1_Mal_GP_1 5e-02 0.150000 0.10000000 0.10000000 0.8 0
## L_at_Amin_Mal_GP_1 0e+00 45.000000 0.00000000 36.00000000 10.0 0
## L_at_Amax_Mal_GP_1 4e+01 90.000000 69.53620000 70.00000000 10.0 6
## VonBert_K_Mal_GP_1 5e-02 0.250000 0.16353300 0.15000000 0.8 6
## CV_young_Mal_GP_1 5e-02 0.250000 0.10000000 0.10000000 0.8 0
## CV_old_Mal_GP_1 5e-02 0.250000 0.10000000 0.10000000 0.8 0
## Wtlen_1_Mal_GP_1 -3e+00 3.000000 0.00000244 0.00000244 0.8 0
## Wtlen_2_Mal_GP_1 -3e+00 4.000000 3.34694000 3.34694000 0.8 0
## RecrDist_GP_1 0e+00 0.000000 0.00000000 0.00000000 0.0 0
## RecrDist_Area_1 0e+00 0.000000 0.00000000 0.00000000 0.0 0
## RecrDist_month_1 0e+00 0.000000 0.00000000 0.00000000 0.0 0
## CohortGrowDev 1e-01 10.000000 1.00000000 1.00000000 1.0 0
## FracFemale_GP_1 1e-06 0.999999 0.50000000 0.50000000 0.5 0
```

##	PHASE	env_var	link	dev_link	dev_minyr	dev_maxyr	dev_PH
## NatM_p_1_Fem_GP_1	-3	0	0	0	0	0	0
## L_at_Amin_Fem_GP_1	2	0	0	0	0	0	0
## L_at_Amax_Fem_GP_1	4	0	0	0	0	0	0
## VonBert_K_Fem_GP_1	4	0	0	0	0	0	0
## CV_young_Fem_GP_1	-3	0	0	0	0	0	0
## CV_old_Fem_GP_1	-3	0	0	0	0	0	0
## Wtlen_1_Fem_GP_1	-3	0	0	0	0	0	0
## Wtlen_2_Fem_GP_1	-3	0	0	0	0	0	0
## Mat50%_Fem_GP_1	-3	0	0	0	0	0	0
## Mat_slope_Fem_GP_1	-3	0	0	0	0	0	0
## Eggs_alpha_Fem_GP_1	-3	0	0	0	0	0	0
## Eggs_beta_Fem_GP_1	-3	0	0	0	0	0	0
## NatM_p_1_Mal_GP_1	-3	0	0	0	0	0	0
## L_at_Amin_Mal_GP_1	-3	0	0	0	0	0	0
## L_at_Amax_Mal_GP_1	4	0	0	0	0	0	0
## VonBert_K_Mal_GP_1	4	0	0	0	0	0	0
## CV_young_Mal_GP_1	-3	0	0	0	0	0	0
## CV_old_Mal_GP_1	-3	0	0	0	0	0	0
## Wtlen_1_Mal_GP_1	-3	0	0	0	0	0	0
## Wtlen_2_Mal_GP_1	-3	0	0	0	0	0	0
## RecrDist_GP_1	-4	0	0	0	0	0	0
## RecrDist_Area_1	-4	0	0	0	0	0	0
## RecrDist_month_1	-4	0	0	0	0	0	0
## CohortGrowDev	-1	0	0	0	0	0	0
## FracFemale_GP_1	-99	0	0	0	0	0	0
##	Block	Block_Fxn					
## NatM_p_1_Fem_GP_1	0	0					
## L_at_Amin_Fem_GP_1	0	0					
## L_at_Amax_Fem_GP_1	0	0					
## VonBert_K_Fem_GP_1	0	0					
## CV_young_Fem_GP_1	0	0					
## CV_old_Fem_GP_1	0	0					
## Wtlen_1_Fem_GP_1	0	0					
## Wtlen_2_Fem_GP_1	0	0					
## Mat50%_Fem_GP_1	0	0					
## Mat_slope_Fem_GP_1	0	0					
## Eggs_alpha_Fem_GP_1	0	0					
## Eggs_beta_Fem_GP_1	0	0					
## NatM_p_1_Mal_GP_1	0	0					
## L_at_Amin_Mal_GP_1	0	0					
## L_at_Amax_Mal_GP_1	0	0					
## VonBert_K_Mal_GP_1	0	0					
## CV_young_Mal_GP_1	0	0					
## CV_old_Mal_GP_1	0	0					
## Wtlen_1_Mal_GP_1	0	0					
## Wtlen_2_Mal_GP_1	0	0					
## RecrDist_GP_1	0	0					
## RecrDist_Area_1	0	0					
## RecrDist_month_1	0	0					
## CohortGrowDev	0	0					
## FracFemale_GP_1	0	0					

```

row.names(ctl$MG_parms)
## [1] "NatM_p_1_Fem_GP_1" "L_at_Amin_Fem_GP_1" "L_at_Amax_Fem_GP_1"
## [4] "VonBert_K_Fem_GP_1" "CV_young_Fem_GP_1" "CV_old_Fem_GP_1"
## [7] "Wtlen_1_Fem_GP_1" "Wtlen_2_Fem_GP_1" "Mat50%_Fem_GP_1"
## [10] "Mat_slope_Fem_GP_1" "Eggs_alpha_Fem_GP_1" "Eggs_beta_Fem_GP_1"
## [13] "NatM_p_1_Mal_GP_1" "L_at_Amin_Mal_GP_1" "L_at_Amax_Mal_GP_1"
## [16] "VonBert_K_Mal_GP_1" "CV_young_Mal_GP_1" "CV_old_Mal_GP_1"
## [19] "Wtlen_1_Mal_GP_1" "Wtlen_2_Mal_GP_1" "RecrDist_GP_1"
## [22] "RecrDist_Area_1" "RecrDist_month_1" "CohortGrowDev"
## [25] "FracFemale_GP_1"

names(ctl$MG_parms)
## [1] "LO" "HI" "INIT" "PRIOR" "PR_SD"
## [6] "PR_type" "PHASE" "env_var&link" "dev_link" "dev_minyr"
## [11] "dev_maxyr" "dev_PH" "Block" "Block_Fxn"

```

### 5.2.8.1 Parámetros hembras

```

ctl$MG_parms[1,] # NatM_uniform_Fem_GP_1
##           LO      HI  INIT PRIOR PR_SD PR_type PHASE env_var&link
## NatM_p_1_Fem_GP_1 0.05 0.15  0.1  0.1  0.8      0    -3          0
##           dev_link dev_minyr dev_maxyr dev_PH Block Block_Fxn
## NatM_p_1_Fem_GP_1      0          0      0      0      0      0

```

#### 5.2.8.1.1 Mortalidad natural Fem GP\_1

```

# Sex: 1 BioPattern: 1 Growth
ctl$MG_parms[2,] # L_at_Amin_Fem_GP_1
##           LO HI      INIT PRIOR PR_SD PR_type PHASE env_var&link
## L_at_Amin_Fem_GP_1 -10 45 21.6535 36 10      6      2          0
##           dev_link dev_minyr dev_maxyr dev_PH Block Block_Fxn
## L_at_Amin_Fem_GP_1      0          0      0      0      0      0
ctl$MG_parms[3,] # L_at_Amax_Fem_GP_1
##           LO HI      INIT PRIOR PR_SD PR_type PHASE env_var&link
## L_at_Amax_Fem_GP_1 40 90 71.6493 70 10      6      4          0
##           dev_link dev_minyr dev_maxyr dev_PH Block Block_Fxn
## L_at_Amax_Fem_GP_1      0          0      0      0      0      0

```

#### 5.2.8.1.2 longitud a la edad Fem GP\_1

```

ctl$MG_parms[4,] # VonBert_K_Fem_GP_1
##           LO      HI      INIT PRIOR PR_SD PR_type PHASE env_var&link
## VonBert_K_Fem_GP_1 0.05 0.25 0.147297 0.15 0.8      6      4          0
##           dev_link dev_minyr dev_maxyr dev_PH Block Block_Fxn
## VonBert_K_Fem_GP_1      0          0      0      0      0      0

```

#### 5.2.8.1.3 Tasa de crecimiento Fem GP\_1

```

ctl$MG_parms[5,] # CV_young_Fem_GP_1
##           LO      HI  INIT PRIOR PR_SD PR_type PHASE env_var&link

```

```
## CV_young_Fem_GP_1 0.05 0.25 0.1 0.1 0.8 0 -3 0
## dev_link dev_minyr dev_maxyr dev_PH Block Block_Fxn
## CV_young_Fem_GP_1 0 0 0 0 0 0 0
ctl$MG_parms[6,] # CV_old_Fem_GP_1
## LO HI INIT PRIOR PR_SD PR_type PHASE env_var@link dev_link
## CV_old_Fem_GP_1 0.05 0.25 0.1 0.1 0.8 0 -3 0 0
## dev_minyr dev_maxyr dev_PH Block Block_Fxn
## CV_old_Fem_GP_1 0 0 0 0 0 0
```

#### 5.2.8.1.4 CV crecimiento Fem GP\_1

```
# Sex: 1 BioPattern: 1 WtLen
ctl$MG_parms[7,] # Wtlen_1_Fem_GP_1
## LO HI INIT PRIOR PR_SD PR_type PHASE env_var@link
## Wtlen_1_Fem_GP_1 -3 3 2.44e-06 2.44e-06 0.8 0 -3 0
## dev_link dev_minyr dev_maxyr dev_PH Block Block_Fxn
## Wtlen_1_Fem_GP_1 0 0 0 0 0 0
ctl$MG_parms[8,] # Wtlen_2_Fem_GP_1
## LO HI INIT PRIOR PR_SD PR_type PHASE env_var@link
## Wtlen_2_Fem_GP_1 -3 4 3.34694 3.34694 0.8 0 -3 0
## dev_link dev_minyr dev_maxyr dev_PH Block Block_Fxn
## Wtlen_2_Fem_GP_1 0 0 0 0 0 0
```

#### 5.2.8.1.5 Relación longitud-peso Fem GP\_1

```
# Sex: 1 BioPattern: 1 Maturity@Fecundity
ctl$MG_parms[9,] # Mat50%_Fem_GP_1
## LO HI INIT PRIOR PR_SD PR_type PHASE env_var@link dev_link
## Mat50%_Fem_GP_1 50 60 55 55 0.8 0 -3 0 0
## dev_minyr dev_maxyr dev_PH Block Block_Fxn
## Mat50%_Fem_GP_1 0 0 0 0 0
ctl$MG_parms[10,] # Mat_slope_Fem_GP_1
## LO HI INIT PRIOR PR_SD PR_type PHASE env_var@link dev_link
## Mat_slope_Fem_GP_1 -3 3 -0.25 -0.25 0.8 0 -3 0 0
## dev_minyr dev_maxyr dev_PH Block Block_Fxn
## Mat_slope_Fem_GP_1 0 0 0 0 0
```

#### 5.2.8.1.6 Relación Madurez Fem GP\_1

```
ctl$MG_parms[11,] # Eggs/kg_inter_Fem_GP_1
## LO HI INIT PRIOR PR_SD PR_type PHASE env_var@link dev_link
## Eggs_alpha_Fem_GP_1 -3 3 1 1 0.8 0 -3 0 0
## dev_minyr dev_maxyr dev_PH Block Block_Fxn
## Eggs_alpha_Fem_GP_1 0 0 0 0 0
ctl$MG_parms[12,] # Eggs/kg_slope_wt_Fem_GP_1
## LO HI INIT PRIOR PR_SD PR_type PHASE env_var@link dev_link
## Eggs_beta_Fem_GP_1 -3 3 0 0 0.8 0 -3 0 0
## dev_minyr dev_maxyr dev_PH Block Block_Fxn
## Eggs_beta_Fem_GP_1 0 0 0 0 0
```

#### 5.2.8.1.7 Eggs/kg Fem GP\_1



### 5.2.8.2 Parámetros Machos

```
# Sex: 2 BioPattern: 1 NatMort
ctl$MG_parms[13,] # NatM_uniform_Mal_GP_1
##          LO    HI INIT PRIOR PR_SD PR_type PHASE env_var@link
## NatM_p_1_Mal_GP_1 0.05 0.15 0.1 0.1 0.8 0 -3 0
##          dev_link dev_minyr dev_maxyr dev_PH Block Block_Fxn
## NatM_p_1_Mal_GP_1 0 0 0 0 0 0 0
```

#### 5.2.8.2.1 Mortalidad natural Mal GP\_1

```
# Sex: 2 BioPattern: 1 Growth
ctl$MG_parms[14,] # L_at_Amin_Mal_GP_1
##          LO HI INIT PRIOR PR_SD PR_type PHASE env_var@link dev_link
## L_at_Amin_Mal_GP_1 0 45 0 36 10 0 -3 0 0
##          dev_minyr dev_maxyr dev_PH Block Block_Fxn
## L_at_Amin_Mal_GP_1 0 0 0 0 0 0
ctl$MG_parms[15,] # L_at_Amax_Mal_GP_1
##          LO HI INIT PRIOR PR_SD PR_type PHASE env_var@link
## L_at_Amax_Mal_GP_1 40 90 69.5362 70 10 6 4 0
##          dev_link dev_minyr dev_maxyr dev_PH Block Block_Fxn
## L_at_Amax_Mal_GP_1 0 0 0 0 0 0 0
```

#### 5.2.8.2.2 longitud a la edad Mal GP\_1

```
ctl$MG_parms[16,] # VonBert_K_Mal_GP_1
##          LO    HI INIT PRIOR PR_SD PR_type PHASE env_var@link
## VonBert_K_Mal_GP_1 0.05 0.25 0.163533 0.15 0.8 6 4 0
##          dev_link dev_minyr dev_maxyr dev_PH Block Block_Fxn
## VonBert_K_Mal_GP_1 0 0 0 0 0 0 0
```

#### 5.2.8.2.3 Tasa de crecimiento Mal GP\_1

```
ctl$MG_parms[17,] # CV_young_Mal_GP_1
##          LO    HI INIT PRIOR PR_SD PR_type PHASE env_var@link
## CV_young_Mal_GP_1 0.05 0.25 0.1 0.1 0.8 0 -3 0
##          dev_link dev_minyr dev_maxyr dev_PH Block Block_Fxn
## CV_young_Mal_GP_1 0 0 0 0 0 0 0
ctl$MG_parms[18,] # CV_old_Mal_GP_1
##          LO    HI INIT PRIOR PR_SD PR_type PHASE env_var@link dev_link
## CV_old_Mal_GP_1 0.05 0.25 0.1 0.1 0.8 0 -3 0 0
##          dev_minyr dev_maxyr dev_PH Block Block_Fxn
## CV_old_Mal_GP_1 0 0 0 0 0 0
```

#### 5.2.8.2.4 CV crecimiento Mal GP\_1

```
# Sex: 2 BioPattern: 1 WtLen
ctl$MG_parms[19,] # Wtlen_1_Mal_GP_1
##          LO HI INIT PRIOR PR_SD PR_type PHASE env_var@link
## Wtlen_1_Mal_GP_1 -3 3 2.44e-06 2.44e-06 0.8 0 -3 0
```

```
##          dev_link dev_minyr dev_maxyr dev_PH Block Block_Fxn
## Wtlen_1_Mal_GP_1      0      0      0      0      0      0
ctl$MG_parms[20,] # Wtlen_2_Mal_GP_1
##          LO HI  INIT  PRIOR PR_SD PR_type PHASE env_var&link
## Wtlen_2_Mal_GP_1 -3  4 3.34694 3.34694 0.8      0    -3      0
##          dev_link dev_minyr dev_maxyr dev_PH Block Block_Fxn
## Wtlen_2_Mal_GP_1      0      0      0      0      0      0
```

#### 5.2.8.2.5 Relación longitud-peso Mal GP\_1

#### 5.2.8.2.6 Hermaphroditism Revisar como se ingresa esta información, en ejemplo simple no se utiliza

```
# Hermaphroditism
```

```
# Recruitment Distribution
ctl$MG_parms[21,] # RecrDist_GP_1
##          LO HI  INIT  PRIOR PR_SD PR_type PHASE env_var&link dev_link
## RecrDist_GP_1  0  0    0    0    0      0    -4      0      0
##          dev_minyr dev_maxyr dev_PH Block Block_Fxn
## RecrDist_GP_1      0      0      0      0      0
ctl$MG_parms[22,] # RecrDist_Area_1
##          LO HI  INIT  PRIOR PR_SD PR_type PHASE env_var&link dev_link
## RecrDist_Area_1  0  0    0    0    0      0    -4      0      0
##          dev_minyr dev_maxyr dev_PH Block Block_Fxn
## RecrDist_Area_1      0      0      0      0      0
ctl$MG_parms[23,] # RecrDist_month_1
##          LO HI  INIT  PRIOR PR_SD PR_type PHASE env_var&link dev_link
## RecrDist_month_1  0  0    0    0    0      0    -4      0      0
##          dev_minyr dev_maxyr dev_PH Block Block_Fxn
## RecrDist_month_1      0      0      0      0      0
```

#### 5.2.8.2.7 Distribución del reclutamiento

```
# Cohort growth dev base
ctl$MG_parms[24,] # CohortGrowDev
##          LO HI  INIT  PRIOR PR_SD PR_type PHASE env_var&link dev_link
## CohortGrowDev 0.1 10    1    1    1      0    -1      0      0
##          dev_minyr dev_maxyr dev_PH Block Block_Fxn
## CohortGrowDev      0      0      0      0      0
```

#### 5.2.8.2.8 Cohort growth dev base Revisar como se ingresa esta información, en ejemplo simple no se utiliza

```
# Movement
# Age Error from parameters
# catch multiplier
```

```
# fraction female, by GP
ctl$MG_parms[25,] # FracFemale_GP_1
##          LO          HI  INIT  PRIOR PR_SD PR_type PHASE env_var&link
## FracFemale_GP_1 1e-06 0.999999 0.5  0.5  0.5      0   -99      0
```

```
##          dev_link dev_minyr dev_maxyr dev_PH Block Block_Fxn
## FracFemale_GP_1      0      0      0      0      0      0
```

### 5.2.8.2.9 fraction female, by GP

**5.2.8.2.10 M2 parameter for each predator fleet** Revisar como se ingresa esta información, en ejemplo simple no se utiliza

```
# M2 parameter for each predator fleet
#
```

#####\_no timevary MG parameters Revisar como se ingresa esta información, en ejemplo simple no se utiliza

```
#_no timevary MG parameters
#
```

#####\_seasonal\_effects\_on\_biology\_parms

```
ctl$MGparm_seas_effects #_femwtlen1,femwtlen2,mat1,mat2,fec1,fec2,Malewtlen1,malewtlen2,L1,K
## MGparm_seas_effects_1 MGparm_seas_effects_2 MGparm_seas_effects_3
##      0      0      0
## MGparm_seas_effects_4 MGparm_seas_effects_5 MGparm_seas_effects_6
##      0      0      0
## MGparm_seas_effects_7 MGparm_seas_effects_8 MGparm_seas_effects_9
##      0      0      0
## MGparm_seas_effects_10
##      0
#_ LO HI INIT PRIOR PR_SD PR_type PHASE
#_Cond -2 2 0 0 -1 99 -2 #_placeholder when no seasonal MG parameters
#
```

## 5.2.9 Relación stock recluta

```
ctl$SR_function #_Spawner-Recruitment; Options:
## [1] 3

# 1=NA;
# 2=Ricker;
# 3=std_B-H;
# 4=SCAA;
# 5=Hockey;
# 6=B-H_flattop;
# 7=survival_3Parm;
# 8=Shepherd_3Parm;
# 9=RickerPower_3parm

ctl$Use_steep_init_equi # # 0/1 to use steepness in initial equ recruitment calculation
## [1] 0
ctl$Sigma_R_FofCurvature # future feature: 0/1 to make realized sigmaR a function of SR curvature
## [1] 0
#-----
#_          LO          HI          INIT          PRIOR          PR_SD          PR_type          PHASE          env-
names(ctl$SR_parms)
## [1] "LO"          "HI"          "INIT"          "PRIOR"          "PR_SD"
## [6] "PR_type"      "PHASE"          "env_var$link" "dev_link"          "dev_minyr"
## [11] "dev_maxyr"    "dev_PH"          "Block"          "Block_Fxn"
```

```

row.names(ctl$SR_parms)
## [1] "SR_LN(R0)" "SR_BH_steep" "SR_sigmaR" "SR_regime" "SR_autocorr"
ctl$SR_parms[1,]
##      LO HI      INIT PRIOR PR_SD PR_type PHASE env_var link dev_link
## SR_LN(R0) 3 31 8.81505 10.3 10 0 1 0 0
##      dev_minyr dev_maxyr dev_PH Block Block_Fxn
## SR_LN(R0) 0 0 0 0 0
ctl$SR_parms[2,]
##      LO HI      INIT PRIOR PR_SD PR_type PHASE env_var link dev_link
## SR_BH_steep 0.2 1 0.614248 0.7 0.05 1 4 0 0
##      dev_minyr dev_maxyr dev_PH Block Block_Fxn
## SR_BH_steep 0 0 0 0 0
ctl$SR_parms[3,]
##      LO HI INIT PRIOR PR_SD PR_type PHASE env_var link dev_link dev_minyr
## SR_sigmaR 0 2 0.6 0.8 0.8 0 -4 0 0 0
##      dev_maxyr dev_PH Block Block_Fxn
## SR_sigmaR 0 0 0 0
ctl$SR_parms[4,]
##      LO HI INIT PRIOR PR_SD PR_type PHASE env_var link dev_link dev_minyr
## SR_regime -5 5 0 0 1 0 -4 0 0 0
##      dev_maxyr dev_PH Block Block_Fxn
## SR_regime 0 0 0 0
ctl$SR_parms[5,]
##      LO HI INIT PRIOR PR_SD PR_type PHASE env_var link dev_link
## SR_autocorr 0 0 0 0 0 0 -99 0 0
##      dev_minyr dev_maxyr dev_PH Block Block_Fxn
## SR_autocorr 0 0 0 0 0
#-----

```

## 5.2.10 Desvíos de los reclutamientos

```

#_no timevary SR parameters
ctl$do_recdev #do_recdev: 0=none; 1=devvector (R=F(SSB)+dev); 2=deviations (R=F(SSB)+dev); 3=deviation
## [1] 1

ctl$MainRdevYrFirst # first year of main recr_devs; early devs can precede this era
## [1] 1971
ctl$MainRdevYrLast # last year of main recr_devs; forecast devs start in following year
## [1] 2001
ctl$recdev_phase #_recdev phase
## [1] 2
ctl$recdev_adv # (0/1) to read 13 advanced options
## [1] 1
ctl$recdev_early_start #_recdev_early_start (0=none; neg value makes relative to recdev_start)
## [1] 0
ctl$recdev_early_phase #_recdev_early_phase
## [1] -4
ctl$Fcast_rec_rphase #_forecast_recruitment phase (incl. late recr) (0 value resets to maxphase+1)
## [1] 0

ctl$lambda4Fcast_rec_rlike #_lambda for Fcast_rec_rlike occurring before endyr+1
## [1] 1

```

```

ctl$last_early_yr_nobias_adj #_last_yr_nobias_adj_in_MPD; begin of ramp
## [1] 1900
ctl$first_yr_fullbias_adj #_first_yr_fullbias_adj_in_MPD; begin of plateau
## [1] 1900
ctl$last_yr_fullbias_adj #_last_yr_fullbias_adj_in_MPD
## [1] 2001
ctl$first_recent_yr_nobias_adj #_end_yr_for_ramp_in_MPD (can be in forecast to shape ramp, but SS3 sets
## [1] 2002

ctl$max_bias_adj #_max_bias_adj_in_MPD (typical ~0.8; -3 sets all years to 0.0; -2 sets all non-foreca
## [1] 1

```

#### 5.2.10.1 Sesgo

```

ctl$period_of_cycles_in_recr #_period of cycles in recruitment (N parms read below)
## [1] 0
ctl$min_rec_dev #min_rec_dev
## [1] -5
ctl$max_rec_dev #max_rec_dev
## [1] 5
ctl$N_Read_recdevs #_read_recdevs
## [1] 0
#_end of advanced SR options
#
#_placeholder for full parameter lines for recruitment cycles
# read specified recr devs
#_Yr Input_value
#
# all recruitment deviations
# 1971R 1972R 1973R 1974R 1975R 1976R 1977R 1978R 1979R 1980R 1981R 1982R 1983R 1984R 1985R 1986R 1987
# 0.1268 -0.0629442 0.0998014 -0.174095 0.0306484 0.714818 -0.0228752 0.00379775 0.261267 0.173626 0.0
#

```

#### 5.2.10.2 Desvíos

#### 5.2.11 Mortalidad por pesca

```

#Fishing Mortality info
ctl$F_ballpark # F ballpark value in units of annual_F
## [1] 0.3
ctl$F_ballpark_year # F ballpark year (neg value to disable)
## [1] -2001

ctl$F_Method # F_Method: 1=Pope midseason rate; 2=F as parameter; 3=F as hybrid; 4=fleet-specific parm
## [1] 3

ctl$maxF # max F (methods 2-4) or harvest fraction (method 1)
## [1] 2.95

ctl$F_iter # N iterations for tuning in hybrid mode; recommend 3 (faster) to 5 (more precise if many fl
## [1] 4
#
#_initial_F_parms; for each fleet x season that has init_catch; nest season in fleet; count = 0

```

### 5.2.12 Capturabilidad

#### 5.2.12.1 Opciones de capturabilidad

### 5.2.12.2 Parámetros

```
#_size_selex_patterns
#Pattern: 0; parm=0; selex=1.0 for all sizes
#Pattern: 1; parm=2; logistic; with 95% width specification
#Pattern: 5; parm=2; mirror another size selex; PARMS pick the min-max bin to mirror
#Pattern: 11; parm=2; selex=1.0 for specified min-max population length bin range
#Pattern: 15; parm=0; mirror another age or length selex
#Pattern: 6; parm=2+special; non-parm len selex
#Pattern: 43; parm=2+special+2; like 6, with 2 additional param for scaling (average over bin range)
```

```

#Pattern:_8; parm=8; double_logistic with smooth transitions and constant above Linf option
#Pattern:_9; parm=6; simple 4-param double logistic with starting length; parm 5 is first length; parm
#Pattern:_21; parm=2+special; non-parm len sele $\alpha$ , read as pairs of size, then sele $\alpha$ 
#Pattern:_22; parm=4; double_normal as in CASAL
#Pattern:_23; parm=6; double_normal where final value is directly equal to sp(6) so can be >1.0
#Pattern:_24; parm=6; double_normal with sel(minL) and sel(maxL), using joiners
#Pattern:_2; parm=6; double_normal with sel(minL) and sel(maxL), using joiners, back compatible versi
#Pattern:_25; parm=3; exponential-logistic in length
#Pattern:_27; parm=special+3; cubic spline in length; parm1==1 resets knots; parm1==2 resets all
#Pattern:_42; parm=special+3+2; cubic spline; like 27, with 2 additional param for scaling (average over
#_discard_options:_0=none;_1=define_retention;_2=retention&mortality;_3=all_discarded_dead;_4=define_dor
#_Pattern Discard Male Special
ctl$size_selex_types
##          Pattern Discard Male Special
## FISHERY      1          0          0          0
## SURVEY1      1          0          0          0
## SURVEY2      0          0          0          0

```

### 5.2.13.1 tipos de patrones de selectividad a la talla

```

#
#_age_selex_patterns
#Pattern:_0; parm=0; sele $\alpha$ =1.0 for ages 0 to maxage
#Pattern:_10; parm=0; sele $\alpha$ =1.0 for ages 1 to maxage
#Pattern:_11; parm=2; sele $\alpha$ =1.0 for specified min-max age
#Pattern:_12; parm=2; age logistic
#Pattern:_13; parm=8; age double logistic. Recommend using pattern 18 instead.
#Pattern:_14; parm=nages+1; age empirical
#Pattern:_15; parm=0; mirror another age or length sele $\alpha$ 
#Pattern:_16; parm=2; Colera $\alpha$ ne - Gaussian
#Pattern:_17; parm=nages+1; empirical as random walk N parameters to read can be overridden by setting
#Pattern:_41; parm=2+nages+1; // like 17, with 2 additional param for scaling (average over bin range)
#Pattern:_18; parm=8; double logistic - smooth transition
#Pattern:_19; parm=6; simple 4-param double logistic with starting age
#Pattern:_20; parm=6; double_normal,using joiners
#Pattern:_26; parm=3; exponential-logistic in age
#Pattern:_27; parm=3+special; cubic spline in age; parm1==1 resets knots; parm1==2 resets all
#Pattern:_42; parm=2+special+3; // cubic spline; with 2 additional param for scaling (average over bin
#Age patterns entered with value >100 create Min_selage from first digit and pattern from remainder
#_Pattern Discard Male Special
ctl$age_selex_types
##          Pattern Discard Male Special
## FISHERY     11          0          0          0
## SURVEY1     11          0          0          0
## SURVEY2     11          0          0          0

```

### 5.2.13.2 tipos de patrones de selectividad a la edad

```

#
#_          LO          HI          INIT          PRIOR          PR_SD          PR_type          PHASE          env-
# 1 FISHERY LenSelex
ctl$size_selex_parms

```

```
##                               LO HI      INIT PRIOR PR_SD PR_type PHASE env_var&link
## SizeSel_P_1_FISHERY(1) 19.00 80 53.64110    50 0.01      1      2      0
## SizeSel_P_2_FISHERY(1)  0.01 60 18.92320    15 0.01      1      3      0
## SizeSel_P_1_SURVEY1(2) 19.00 70 36.65300    30 0.01      1      2      0
## SizeSel_P_2_SURVEY1(2)  0.01 60  6.59179    10 0.01      1      3      0
##                               dev_link dev_minyr dev_maxyr dev_PH Block Block_Fxn
## SizeSel_P_1_FISHERY(1)      0      0      0      0      0      0
## SizeSel_P_2_FISHERY(1)      0      0      0      0      0      0
## SizeSel_P_1_SURVEY1(2)      0      0      0      0      0      0
## SizeSel_P_2_SURVEY1(2)      0      0      0      0      0      0
```

### 5.2.13.3 parametros de selectividad a la talla

```
ctl$age_selex_parms
##                               LO HI INIT PRIOR PR_SD PR_type PHASE env_var&link
## AgeSel_P_1_FISHERY(1)  0 40      0      5    99      0  -99      0
## AgeSel_P_2_FISHERY(1)  0 40    40      6    99      0  -99      0
## AgeSel_P_1_SURVEY1(2)  0 40      0      5    99      0  -99      0
## AgeSel_P_2_SURVEY1(2)  0 40    40      6    99      0  -99      0
## AgeSel_P_1_SURVEY2(3)  0 40      0      5    99      0  -99      0
## AgeSel_P_2_SURVEY2(3)  0 40      0      6    99      0  -99      0
##                               dev_link dev_minyr dev_maxyr dev_PH Block Block_Fxn
## AgeSel_P_1_FISHERY(1)      0      0      0      0      0      0
## AgeSel_P_2_FISHERY(1)      0      0      0      0      0      0
## AgeSel_P_1_SURVEY1(2)      0      0      0      0      0      0
## AgeSel_P_2_SURVEY1(2)      0      0      0      0      0      0
## AgeSel_P_1_SURVEY2(3)      0      0      0      0      0      0
## AgeSel_P_2_SURVEY2(3)      0      0      0      0      0      0
```

### 5.2.13.4 parametros de selectividad a la edad

```
#_No_Dirichlet parameters
#_no timevary selex parameters
#
ctl$Use_2D_AR1_selectivity # use 2D_AR1 selectivity(0/1)
## [1] 0
#_no 2D_AR1 selex offset used
#
#-----
# Tag loss and Tag reporting parameters go next
ctl$TG_custom # TG_custom: 0=no read and autogen if tag data exist; 1=read
## [1] 0
#_Cond -6 6 1 1 2 0.01 -4 0 0 0 0 0 0 0 #_placeholder if no parameters
#
# no timevary parameters
#
#
# Input variance adjustments factors:
#_1=add_to_survey_CV
#_2=add_to_discard_stddev
#_3=add_to_bodywt_CV
#_4=mult_by_lencomp_N
```



```

#_5=mult_by_agecomp_N
#_6=mult_by_size-at-age_N
#_7=mult_by_generalized_sizecomp
#_Factor Fleet Value
ctl$DoVar_adjust
## [1] 0
ctl$maxlambdaphase #_maxlambdaphase
## [1] 4
ctl$sd_offset #_sd_offset; must be 1 if any growthCV, sigmaR, or survey extraSD is an estimated parameter
## [1] 1

```

### 5.2.13.5 otros parámetros ??? REVISAR

### 5.2.14 Lambdas

```

ctl$N_lambdas # read 3 changes to default Lambdas (default value is 1.0)
## [1] 3
# Like_comp codes: 1=surv; 2=disc; 3=mmwt; 4=length; 5=age; 6=SizeFreq; 7=sizeage; 8=catch; 9=init_equ; 10=recrdev; 11=parm_prior; 12=parm_dev; 13=CrashPen; 14=Morphcomp; 15=Tag-comp; 16=Tag-negbin; 17=F_b
#like_comp fleet phase value sizefreq_method
ctl$lambdas
##
##           like_comp fleet phase value
## Surv_SURVEY1_Ph2                1      2      2      1
## length_SURVEY1_sizefreq_method_1_Ph2                4      2      2      1
## length_SURVEY1_sizefreq_method_1_Ph23                4      2      3      1
##
##           sizefreq_method
## Surv_SURVEY1_Ph2                1
## length_SURVEY1_sizefreq_method_1_Ph2                1
## length_SURVEY1_sizefreq_method_1_Ph23                1

```

### REVISAR ESTA PARTE ### more stddev reporting

```

ctl$more_stddev_reporting # (0/1/2) read specs for more stddev reporting: 0 = skip, 1 = read specs for
## [1] 1

```

```

ctl$stddev_reporting_specs # Selectivity: (1) 0 to skip or fleet, (2) 1=len/2=age/3=combined, (3) year
## stddev_reporting_specs_1 stddev_reporting_specs_2 stddev_reporting_specs_3
##           1                1                -1
## stddev_reporting_specs_4 stddev_reporting_specs_5 stddev_reporting_specs_6
##           5                1                5
## stddev_reporting_specs_7 stddev_reporting_specs_8 stddev_reporting_specs_9
##           1                -1                5

```

### 5.2.14.1 specs ?

```

ctl$stddev_reporting_growth # Growth: (1) 0 to skip or growth pattern, (2) growth ages; NOTE: does each
## stddev_reporting_growth_1 stddev_reporting_growth_2 stddev_reporting_growth_3
##           1                2                14
## stddev_reporting_growth_4 stddev_reporting_growth_5
##           26                40

```

### 5.2.14.2 Crecimiento

```

ctl$stddev_reporting_N_at_A # Numbers-at-age: (1) 0 or area(-1 for all), (2) year, (3) N ages; NOTE: s
## stddev_reporting_N_at_A_1 stddev_reporting_N_at_A_2 stddev_reporting_N_at_A_3
## 1 2 14
## stddev_reporting_N_at_A_4 stddev_reporting_N_at_A_5
## 26 40

```

#### 5.2.14.3 Abundancia a la edad

```

ctl$stddev_reporting_selex # vector with selex std bins (-1 in first bin to self-generate)
## stddev_reporting_selex_1 stddev_reporting_selex_2 stddev_reporting_selex_3
## 5 15 25
## stddev_reporting_selex_4 stddev_reporting_selex_5
## 35 43

```

#### 5.2.14.4 Selectividad

#### 5.2.14.5 Peso medio por edad y año de tu matriz de captura (Kilos).

```
dir(dir_GSA1)[3]
## [1] "CATWT.DAT"
CATWT.DAT <- read.table(paste(dir_GSA1,dir(dir_GSA1)[3],sep="/"),
                        sep=" ",na="NA",fill=T,skip = 5)
CATWT.DAT
##      V1      V2      V3      V4      V5      V6
## 1 0.032 0.119 0.517 1.178 2.086 3.529
## 2 0.030 0.109 0.471 1.243 1.964 3.529
## 3 0.031 0.125 0.521 1.127 2.024 3.388
## 4 0.023 0.136 0.497 1.100 1.999 2.775
## 5 0.021 0.139 0.511 1.106 2.037 2.775
## 6 0.028 0.145 0.515 1.203 2.147 3.083
## 7 0.025 0.133 0.520 1.086 2.046 2.637
## 8 0.024 0.170 0.474 1.116 1.907 3.045
## 9 0.028 0.147 0.486 1.174 2.059 2.875
## 10 0.029 0.146 0.465 1.106 2.087 2.875
## 11 0.021 0.164 0.481 1.152 2.057 3.257
## 12 0.026 0.162 0.488 1.149 1.873 3.529
## 13 0.025 0.152 0.481 1.157 1.896 3.460
## 14 0.022 0.139 0.479 1.123 1.843 2.875
## 15 0.024 0.129 0.481 1.152 1.851 2.875
## 16 0.022 0.140 0.485 1.132 1.801 2.875
## 17 0.029 0.153 0.504 1.073 1.820 2.754
## 18 0.019 0.136 0.487 1.122 1.747 2.754
## 19 0.023 0.115 0.477 1.107 1.781 2.754
```

#### 5.2.14.6 Medio por edad y año asumida para el stock (Normalmente = CATWT) (Kilos).

```
dir(dir_GSA1)[11]
## [1] "STOCWT.DAT"
STOCWT.DAT <- read.table(paste(dir_GSA1,dir(dir_GSA1)[11],sep="/"),
                          sep=" ",na="NA",fill=T,skip = 5)
STOCWT.DAT
```

##	V1	V2	V3	V4	V5	V6
## 1	0.032	0.119	0.517	1.178	2.086	3.529
## 2	0.030	0.109	0.471	1.243	1.964	3.529
## 3	0.031	0.125	0.521	1.127	2.024	3.388
## 4	0.023	0.136	0.497	1.100	1.999	2.775
## 5	0.021	0.139	0.511	1.106	2.037	2.775
## 6	0.028	0.145	0.515	1.203	2.147	3.083
## 7	0.025	0.133	0.520	1.086	2.046	2.637
## 8	0.024	0.170	0.474	1.116	1.907	3.045
## 9	0.028	0.147	0.486	1.174	2.059	2.875
## 10	0.029	0.146	0.465	1.106	2.087	2.875
## 11	0.021	0.164	0.481	1.152	2.057	3.257
## 12	0.026	0.162	0.488	1.149	1.873	3.529
## 13	0.025	0.152	0.481	1.157	1.896	3.460
## 14	0.022	0.139	0.479	1.123	1.843	2.875
## 15	0.024	0.129	0.481	1.152	1.851	2.875
## 16	0.022	0.140	0.485	1.132	1.801	2.875
## 17	0.029	0.153	0.504	1.073	1.820	2.754
## 18	0.019	0.136	0.487	1.122	1.747	2.754
## 19	0.023	0.115	0.477	1.107	1.781	2.754

#### 5.2.14.7 Vector de mortalidad Natural por edad.

```
dir(dir_GSA1)[7]
## [1] "NATMOR.DAT"
NATMOR.DAT <- read.table(paste(dir_GSA1,dir(dir_GSA1)[7],sep="/"),
                        sep=" ",na="NA",fill=T,skip = 5)
NATMOR.DAT
##      V1 V2 V3 V4 V5 V6
## 1 1.9 0.7 0.39 0.29 0.23 0.2
```

#### 5.2.14.8 Ogiva de madurez por edad.

```
dir(dir_GSA1)[10]
## [1] "PROPMAT.DAT"
PROPMAT.DAT <- read.table(paste(dir_GSA1,dir(dir_GSA1)[10],sep="/"),
                        sep=" ",na="NA",fill=T,skip = 5)
PROPMAT.DAT
##      V1      V2      V3 V4 V5 V6
## 1  0 0.2965 0.9855 0.99  1  1
```

## 6 GSA6 (Levante-Norte de España).

La conceptualización del modelo biológico de **GSA6** considera los siguientes componentes de la dinámica poblacional:

- Estructura geográfica
- Reproducción
- Reclutamiento
- Mortalidad natural
- Crecimiento
- otros...

### 6.1 Descripción del modelo

El modelo de evaluación de stock de **GSA6** se basa en un análisis estadístico de la dinámica de estructuras de edad anual que incorpora información biológica y pesquera. La información que ingresa al modelo consiste en ...

- Años de evaluación de stock = 2002\_2021
- número de edades = 6 años ?

## 6.2 Archivos utilizado para enfoque de modelación a4a

```
dir_GSA6 <- here("hke_GSA6_a4a_format")
dir(dir_GSA6)
## [1] "CATCH.DAT"
## [2] "CATNUM.DAT"
## [3] "CATWT.DAT"
## [4] "Definiciones_Ficheros_imput_a4a.docx"
## [5] "INCHECK.TXT"
## [6] "LOWIND.DAT"
## [7] "NATMOR.DAT"
## [8] "PROPF.DAT"
## [9] "PROPM.DAT"
## [10] "PROPMAT.DAT"
## [11] "STOCWT.DAT"
## [12] "TUNEFF.DAT"
```

### 6.2.0.1 Capturas anuales del stock (toneladas)

```
dir(dir_GSA6)[1]
## [1] "CATCH.DAT"
CATCH.DAT <- read.table(paste(dir_GSA6,dir(dir_GSA6)[1],sep="/"),
                        header=T,sep=" ",na="NA",fill=T,skip = 4)
CATCH.DAT
##      X5
## 1 2834.9
## 2 4632.7
## 3 3150.5
## 4 3473.4
## 5 3627.2
## 6 2539.8
## 7 3340.9
## 8 3847.2
## 9 2821.6
## 10 3181.7
## 11 2641.4
## 12 2949.9
## 13 2489.3
## 14 1726.1
## 15 1809.8
## 16 1728.2
## 17 2442.5
## 18 1628.1
## 19 1099.3
## 20 1531.0
```

**6.2.0.2** índices de abundancia para calibrar (CPUEs o Campañas, o ambas). Normalmente para índices de campaña empleamos número/KM2.

```
dir(dir_GSA6)[12]
## [1] "TUNEFF.DAT"
TUNEFF.DAT <- read.table(paste(dir_GSA6,dir(dir_GSA6)[12],sep="/"),
                        sep=" ",na="NA",fill=T,skip = 6)
TUNEFF.DAT
```

```
##      V1      V2      V3      V4      V5      V6      V7
## 1  1 2119.2 101.1  7.4  1.7  NA  NA
## 2  1 2051.7 132.6 10.8  1.0  0.1  0.2
## 3  1 4189.4  90.6 23.5  1.8  NA  NA
## 4  1 3454.8  70.4  5.9  0.5  NA  NA
## 5  1 6054.1 168.2 27.2  0.7  0.7  NA
## 6  1 1565.0  56.4  6.3  1.5  0.1  NA
## 7  1 1799.5  74.8  2.7  1.7  0.2  NA
## 8  1 2188.2 114.0  7.6  1.0  0.4  NA
## 9  1 2824.6  62.9  5.7  1.0  NA  NA
## 10 1  917.2 132.3  5.9  0.3  NA  NA
## 11 1 1333.2  55.1  4.3  0.9  0.2  NA
## 12 1 1263.2 137.6  8.7  0.3  NA  NA
## 13 1  701.9  73.3  9.7  0.9  0.1  0.2
## 14 1  935.1  31.3  7.7  0.8  0.2  NA
## 15 1 1242.1  52.0  5.6  0.7  0.2  NA
## 16 1  931.3  72.7 10.1  0.5  0.3  NA
## 17 1  602.6  89.0  6.1  NA  0.2  NA
## 18 1  530.9  42.2 13.3  0.3  0.1  0.2
## 19 1  623.8  56.8  8.8  0.3  0.1  0.1
## 20 1 1875.8  79.5  2.8  0.7  0.3  NA
```

### 6.2.0.3 Matriz de número de individuos por edad/año de las capturas (Miles de individuos)

```
dir(dir_GSA6)[2]
## [1] "CATNUM.DAT"
CATNUM.DAT <- read.table(paste(dir_GSA6,dir(dir_GSA6)[2],sep="/"),
                          sep=" ",na="NA",fill=T,skip = 5)
CATNUM.DAT
##      V1      V2      V3      V4      V5      V6
## 1 109196.9 6665.5 1248.3 277.6 132.8 33.2
## 2  72755.2 29484.8 1855.7 179.8  58.3  7.7
## 3  61686.9 14773.6 1219.5 173.2  27.0 17.9
## 4  39457.7 16612.0 2048.1 349.0  37.5  3.0
## 5  53949.5 16097.8 2050.5 380.7  32.4  6.4
## 6  34983.3  8333.9 1804.8 310.0  54.4 12.8
## 7  61690.5 17293.3 1430.4 171.1  62.2  3.9
## 8  68385.7 18710.1 1619.0 264.6  60.0  1.7
## 9  11268.7 14311.1 2250.0 215.1  46.2  2.1
## 10  7536.9 18339.0 2469.5 227.8  16.7  5.1
## 11  9210.2 18153.2 1641.6 158.1  23.9  1.3
## 12  6636.3 19209.9 1886.4 152.7   9.5  2.1
## 13  7376.4 11887.3 2232.3 137.7   9.5  2.0
## 14  4831.9  9390.1 1350.5 105.8  13.2  0.1
## 15  7948.0 12312.3  989.8  91.4  13.4  0.6
## 16  2752.8 10938.7 1219.6 115.8  16.7  2.0
## 17  3727.2 16840.1 1518.9 156.6   9.5  0.5
## 18   593.7  7251.4 1513.8 186.3  12.3  2.6
## 19 1894.6  7109.8  770.7  67.8   4.5  0.6
## 20   875.9 10995.6  842.9  81.4  27.2  3.9
```

### 6.2.0.4 Peso medio por edad y año de tu matriz de captura (Kilos).



```

dir(dir_GSA6)[3]
## [1] "CATWT.DAT"
CATWT.DAT <- read.table(paste(dir_GSA6,dir(dir_GSA6)[3],sep="/"),
                        sep=" ",na="NA",fill=T,skip = 5)

CATWT.DAT
##      V1      V2      V3      V4      V5      V6
## 1 0.011 0.111 0.436 0.996 1.623 2.391
## 2 0.016 0.094 0.407 0.967 1.627 2.646
## 3 0.018 0.103 0.400 0.929 1.630 4.050
## 4 0.019 0.104 0.398 0.928 1.656 2.291
## 5 0.018 0.101 0.411 0.926 1.619 3.827
## 6 0.019 0.104 0.415 0.967 1.631 3.046
## 7 0.017 0.097 0.391 0.980 1.656 2.496
## 8 0.022 0.084 0.449 0.983 1.488 5.082
## 9 0.018 0.109 0.414 0.925 1.712 2.424
## 10 0.026 0.105 0.405 0.903 1.747 2.646
## 11 0.026 0.095 0.402 0.902 1.672 2.596
## 12 0.028 0.106 0.387 0.933 1.598 2.664
## 13 0.022 0.120 0.401 0.911 1.622 2.634
## 14 0.024 0.112 0.400 0.908 1.579 2.188
## 15 0.026 0.098 0.401 0.959 1.611 2.309
## 16 0.030 0.103 0.391 0.945 1.585 2.652
## 17 0.030 0.101 0.402 0.925 1.618 2.335
## 18 0.031 0.123 0.399 0.915 1.705 2.578
## 19 0.029 0.103 0.398 0.888 1.639 2.624
## 20 0.034 0.102 0.389 0.975 1.673 2.545

```

6.2.0.5 Medio por edad y año asumida para el stock (Normalmente = CATWT) (Kilos).

```

dir(dir_GSA6)[11]
## [1] "STOCWT.DAT"
STOCWT.DAT <- read.table(paste(dir_GSA6,dir(dir_GSA6)[11],sep="/"),
                        sep=" ",na="NA",fill=T,skip = 5)

STOCWT.DAT
##      V1      V2      V3      V4      V5      V6
## 1 0.011 0.111 0.436 0.996 1.623 2.391
## 2 0.016 0.094 0.407 0.967 1.627 2.646
## 3 0.018 0.103 0.400 0.929 1.630 4.050
## 4 0.019 0.104 0.398 0.928 1.656 2.291
## 5 0.018 0.101 0.411 0.926 1.619 3.827
## 6 0.019 0.104 0.415 0.967 1.631 3.046
## 7 0.017 0.097 0.391 0.980 1.656 2.496
## 8 0.022 0.084 0.449 0.983 1.488 5.082
## 9 0.018 0.109 0.414 0.925 1.712 2.424
## 10 0.026 0.105 0.405 0.903 1.747 2.646
## 11 0.026 0.095 0.402 0.902 1.672 2.596
## 12 0.028 0.106 0.387 0.933 1.598 2.664
## 13 0.022 0.120 0.401 0.911 1.622 2.634
## 14 0.024 0.112 0.400 0.908 1.579 2.188
## 15 0.026 0.098 0.401 0.959 1.611 2.309
## 16 0.030 0.103 0.391 0.945 1.585 2.652
## 17 0.030 0.101 0.402 0.925 1.618 2.335
## 18 0.031 0.123 0.399 0.915 1.705 2.578

```

```
## 19 0.029 0.103 0.398 0.888 1.639 2.624
## 20 0.034 0.102 0.389 0.975 1.673 2.545
```

#### 6.2.0.6 Vector de mortalidad Natural por edad.

```
dir(dir_GSA6)[7]
## [1] "NATMOR.DAT"
NATMOR.DAT <- read.table(paste(dir_GSA6,dir(dir_GSA6)[7],sep="/"),
                        sep=" ",na="NA",fill=T,skip = 5)
NATMOR.DAT
##      V1      V2      V3      V4      V5      V6
## 1 1.8 0.72 0.41 0.3 0.24 0.18
```

#### 6.2.0.7 Ogiva de madurez por edad.

```
dir(dir_GSA6)[10]
## [1] "PROPMAT.DAT"
PROPMAT.DAT <- read.table(paste(dir_GSA6,dir(dir_GSA6)[10],sep="/"),
                        sep=" ",na="NA",fill=T,skip = 5)
PROPMAT.DAT
##      V1      V2      V3      V4      V5      V6
## 1 0 0.2965 0.9855 0.99 1 1
```