

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

May, 14, 2023

Contents

1	Contexto	3
2	Tareas a desarrollar.	3
2.1	Temporalización y dinámica de las actividades propuestas.	3
3	Descripción general del modelo de merluza europea (<i>Merluccius merluccius</i>) de la GSA1 (mar de Alborán).	4
3.1	Modelo conceptual	4
3.2	Descripción general del modelo implementado en a4a	4
4	Revisión de los archivos requeridos para cada enfoque de modelación	5
4.1	Archivos utilizado para enfoque de modelación a4a	5
4.2	Archivos utilizado para enfoque de modelación SS3	6
5	Descripción del Formado de entrada de datos para cada enfoque de modelación	7
5.1	Archivo data	7
5.1.1	Información general del modelo Formato a4a	7
5.1.2	Información general del modelo Formato SS3	7
5.1.3	Capturas en formato a4a	9
5.1.4	Capturas en formato SS3	10
5.1.5	Indices de abundancia formato a4a	13
5.1.6	Indices de abundancia formato SS3	14
5.1.7	Descartes y tallas medias Formato a4a	17
5.1.8	Descartes y tallas medias Formato SS3	17
5.1.9	Composición de tallas Formato a4a	18
5.1.10	Composición de tallas Formato SS3	22
5.1.11	Composición de edad Formato a4a	25
5.1.12	Composición de edad Formato SS3	27
5.1.13	Otros datos	32
5.1.14	Escribir archivo de datos modificado con la función SS_write para el enfoque de modelación SS3	34
5.2	Archivo control.ss	35
5.2.1	Especificaciones iniciales	36
5.2.2	Especificaciones del crecimiento	36
5.3	Distribución del reclutamiento	36
5.3.1	Bloques	37
5.3.2	Parámetros que varían en el tiempo	37
5.3.3	Parámetros biológicos	39
5.3.4	Mortalidad natural	41
5.3.5	Crecimiento	42
5.3.6	Madurez-fecundidad	42

5.3.7	Hermafroditismo	42
5.3.8	Relación stock recluta	49
5.3.9	Desvíos de los reclutamientos	52
5.3.10	Mortalidad por pesca	54
5.3.11	Capturabilidad	55
5.3.12	Opciones de capturabilidad por flota (CPUE o CAMPANAS)	55
5.3.13	Selectividad	56
5.3.14	tipos de selectividad a la talla	56
5.3.15	tipos de selectividad a la edad	56
5.3.16	otros parámetros	61
5.3.17	Escribir archivo de control modificado con la función <code>SS_write</code> para el enfoque de modelación <code>SS3</code>	61
5.4	Archivo <code>starter.ss</code>	62
5.4.1	Escribir archivo de <code>starter</code> modificado con la función <code>SS_write</code> para el enfoque de modelación <code>SS3</code>	62
5.5	Archivo <code>forecast.ss</code>	63
5.5.1	Escribir archivo de <code>forecast</code> modificado con la función <code>SS_write</code> para el enfoque de modelación <code>SS3</code>	64
5.6	Archivo <code>watatege</code>	65
5.6.1	Formato <code>a4a</code>	65
5.6.2	Formato <code>SS3</code>	69
5.6.3	Escribir archivo de <code>wtatage</code> modificado con la función <code>SS_write</code> para el enfoque de modelación <code>SS3</code>	69
6	comprobar si el modelo corre al modificar este archivo	71

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), 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.
- PROPF: 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("empirical_wtatage_and_age_selex")
dir(here("empirical_wtatage_and_age_selex"))
## [1] "control.ss"      "data.ss"         "forecast.ss"     "ss_linux"
## [5] "ss_summary.sso" "ss_win.exe"      "starter.ss"      "wtatage.ss"
```

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
```

Comprobar que modelo simple base se ejecuta sin problema antes de modificarlo

```
exe_path <- here("Ejecutables_SS3", "3.30.18_release")
ss_exe_mac <- paste(exe_path, "ss_osx", sep= "/")

r4ss::run(
  dir = dirname.simple_mod,
  exe = ss_exe_mac,
  extras = "",
  skipfinished = FALSE,
  show_in_console = TRUE,
  verbose = TRUE
)
```

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

5.1 Archivo data

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
## [1] "sourcefile" "type"
## [3] "ReadVersion" "Comments"
## [5] "styr" "endyr"
## [7] "nseas" "months_per_seas"
## [9] "Nsubseasons" "spawn_month"
## [11] "Nsexes" "Nages"
## [13] "N_areas" "Nfleets"
## [15] "fleetinfo" "fleetnames"
## [17] "surveytiming" "units_of_catch"
## [19] "areas" "catch"
## [21] "CPUEinfo" "CPUE"
## [23] "N_discard_fleets" "use_meanbodywt"
## [25] "lbin_method" "binwidth"
## [27] "minimum_size" "maximum_size"
## [29] "use_lencomp" "N_agebins"
## [31] "agebin_vector" "N_ageerror_definitions"
## [33] "ageerror" "age_info"
## [35] "Lbin_method" "agecomp"
## [37] "use_MeanSize_at_Age_obs" "N_envirom_variables"
## [39] "N_sizefreq_methods_rd" "N_sizefreq_methods"
## [41] "do_tags" "morphcomp_data"
## [43] "use_selectivity_priors" "eof"
## [45] "spawn_seas" "Nfleet"
## [47] "Nsurveys" "fleetinfo1"
## [49] "fleetinfo2" "N_meanbodywt"
## [51] "N_lencomp" "N_MeanSize_at_Age_obs"
## [53] "N_lbinspop" "lbin_vector_pop"
```

#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     <-2    #_Nfleets (including surveys)

```


5.1.3 Capturas en formato a4a

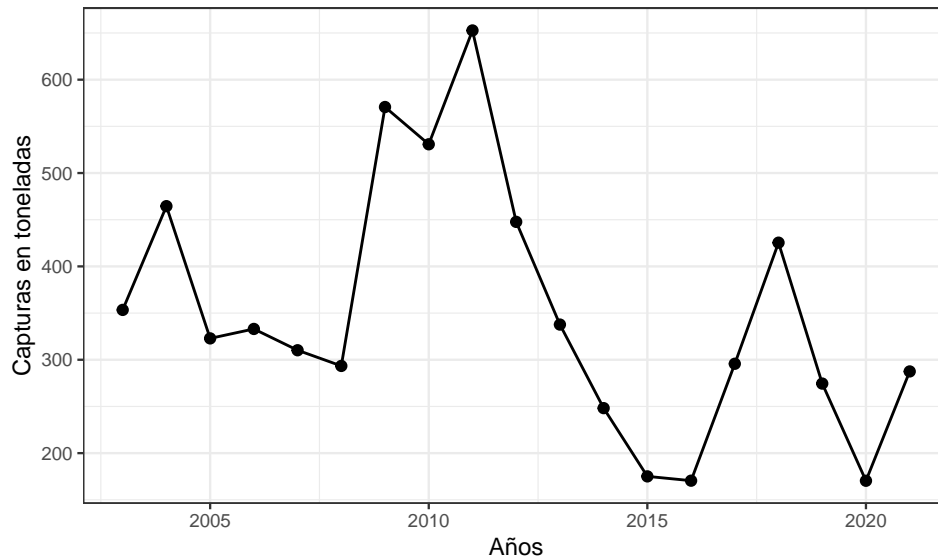
Capturas anuales del stock (toneladas)

```
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
```

```
catchdat<-as.data.frame(CATCH.DAT) %>%
  mutate(Yr=2003:2021)

ggplot(catchdat,aes(x=Yr,y=X5))+
  geom_line()+
  geom_point()+
  labs(x="Años", y="Capturas en toneladas") +
  theme_bw(base_size=9)
```



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

fleetinfo es una matriz donde se ingresan las especificaciones de los datos de captura y debe contener la siguiente información:

- type: 1=catch fleet; 2=bycatch only fleet; 3=survey; 4=ignore
- surveytiming: -1 for fishing fleet to use season-long catch-at-age for observations, or 1 to use observation month; (always 1 for surveys)
- area: area the fleet/survey operates in
- units: 1=bio; 2=num (ignored for surveys; their units read later)
- need_catch_mult: 0=no; 1=yes
- fleetname:

```
fleetnames1<-c("FISHERY", "SURVEY1")
type1<-c(1,3)
surveytiming1<-c(-1,1)
units_of_catch1<-c(1,2)
areas1<-c(1,1)
need_catch_mult1<-c(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
#-----
```

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.

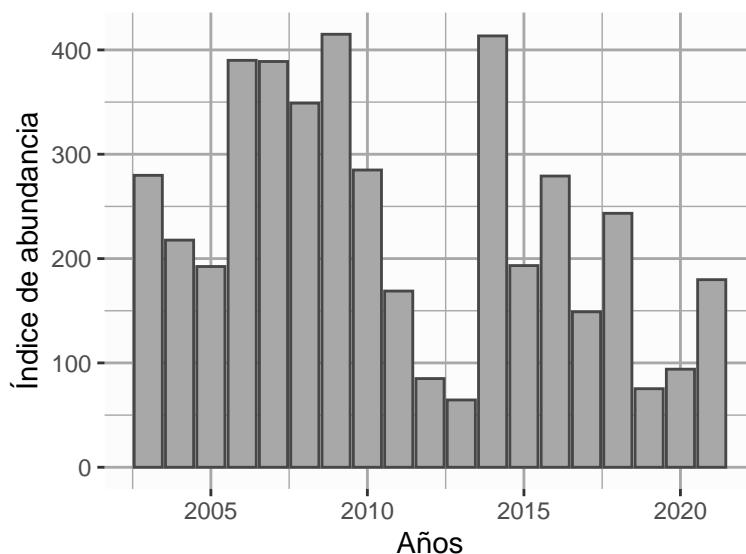
5.1.5 Índices de abundancia formato a4a

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
```

```
names_ind<-c('index_1','Yr')
indeexdat<-data.frame(Yr=2003:2021,suma=rowSums(TUNEFF.DAT))

ggplot() +
  geom_bar(data=indeexdat, aes(x=Yr, y =suma),
           stat="identity", fill='gray66', color = 'gray28') +
  labs(x="Años", y="Índice de abundancia") +
  theme(panel.background = element_rect(fill ="gray99")) +
  theme(panel.grid=element_line(color="gray66"))
```



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)
CPUEinfo_Units<-c(1,1) # unidades la dejamos en 1=biomass Revisar!!!
CPUEinfo_Errtype<-c(0,0) # en general se trabaja lognormal
CPUEinfo_SD_Report<-c(0,0) # esto se puede cambiar después si se necesita
                        # SD_Report, por ahora no es necesario...

CPUEinfo_names<-c("FISHERY", "SURVEY1")
#-----

# 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
```

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

- 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
CPUE_year<-c(yearf1)
#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)))
CPUE_obs<-c(TUNEFF.DAT$V2)
CPUE_se_log<-c(TUNEFF.DAT$V5)
#-----
# 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   0.2
## 2  2004    1    2 187.9   0.2
## 3  2005    1    2 168.1   0.2
## 4  2006    1    2 350.2   0.3
## 5  2007    1    2 356.6   1.2
## 6  2008    1    2 306.3   0.6
## 7  2009    1    2 329.6   0.3
## 8  2010    1    2 147.0   0.2
## 9  2011    1    2 116.9   0.2
## 10 2012    1    2  62.9   0.5
## 11 2013    1    2  38.4   0.2
## 12 2014    1    2 383.7   1.7
## 13 2015    1    2 161.7   0.6
## 14 2016    1    2 238.7   0.2
## 15 2017    1    2  80.7   0.3
## 16 2018    1    2 143.9   0.4
## 17 2019    1    2  28.6   0.5
```

##	18	2020	1	2	50.6	0.3
##	19	2021	1	2	104.9	0.2
#	-----					

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
```

5.1.8.2 Tallas medias

```
dat1$use_meanbodywt<-0
```

5.1.9 Composición de tallas Formato a4a

```
excel<-here("Indice_abundancia_tallas_meditis_GSA1_hake.xlsx")

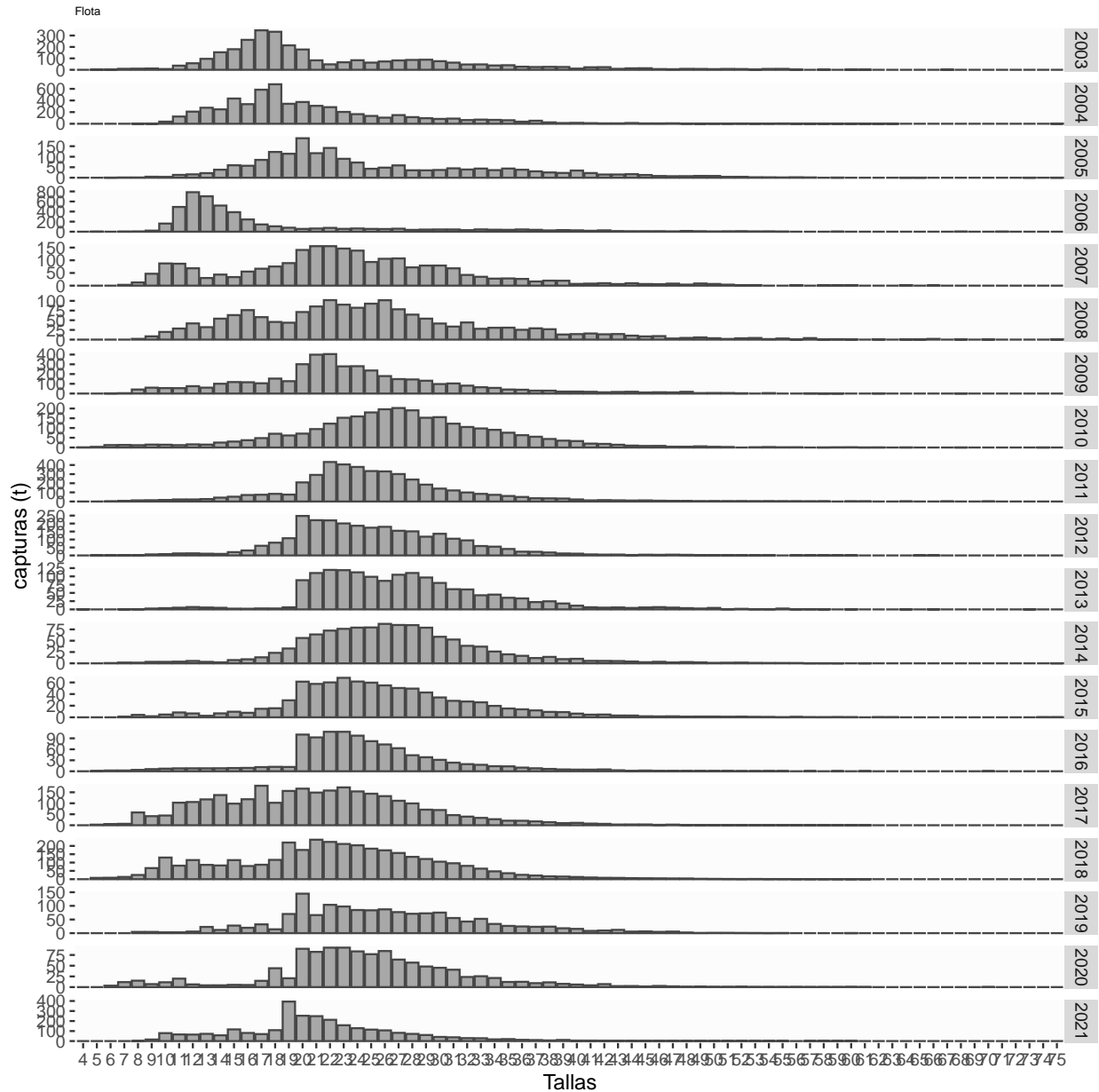
comptallas<-read_excel(excel, sheet = "catch",col_names= TRUE,col_types=NULL,na="",skip= 0)

yearcatch<-seq(2003,2021,1)
tallas<-seq(4,75,1)
comptallas<-comptallas[2:73,2:20]

CompTallas<-data.frame(row.names = NULL,year=yearcatch,t(comptallas))
colnames(CompTallas)<-c("year",tallas)

plot_tallas<-CompTallas %>% melt(id.vars=c("year"))

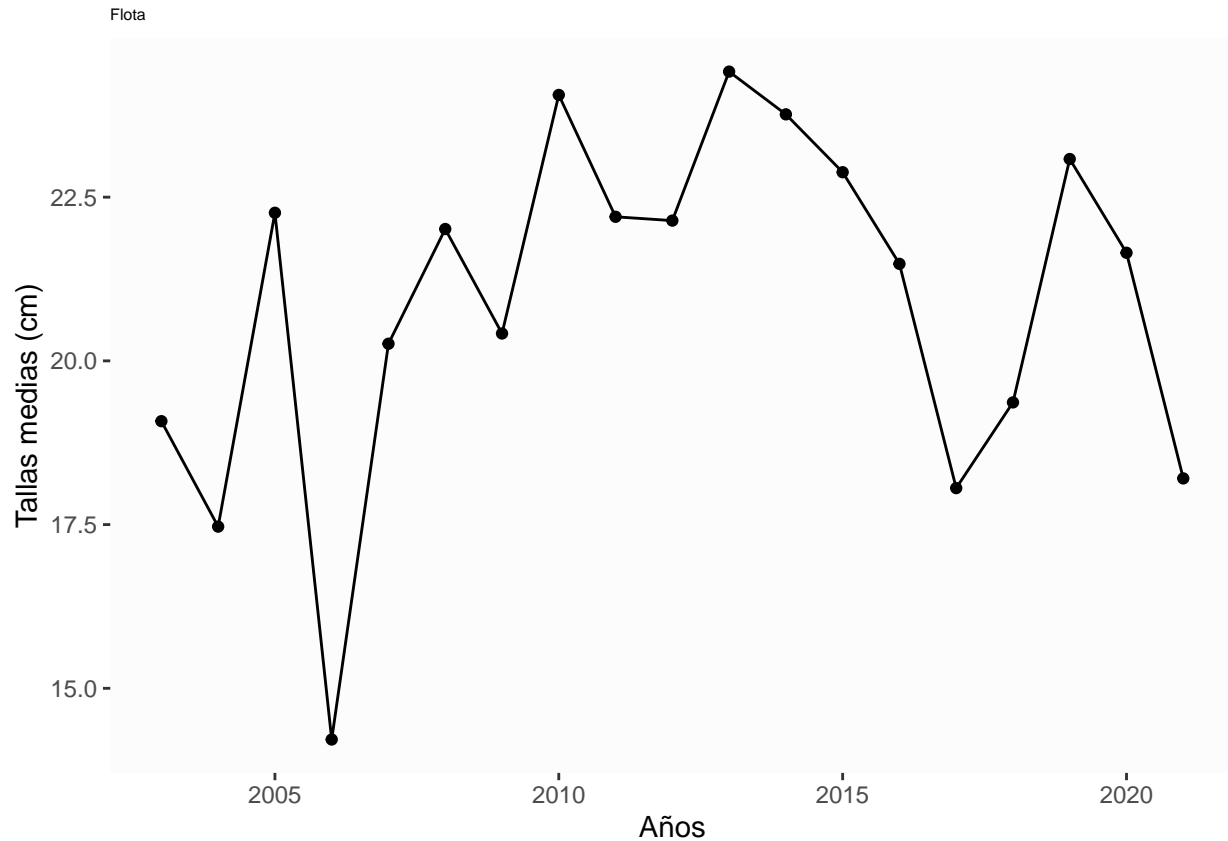
#-----
ggplot(plot_tallas) +
  geom_bar(aes(x = variable, y = value), stat="identity", fill='gray66', color = 'gray28') +
  facet_grid(rows = vars(year), as.table = TRUE,scales = "free") +
  labs(x = 'Tallas', y = 'capturas (t)') +
  theme(panel.background = element_rect(fill ="gray99")) +
  theme(panel.grid=element_line(color=NA)) +
  ggtitle('Flota')+
  theme(plot.title = element_text(size = 6))
```



```
mediaTalla_flota<-as.numeric(plot_tallas$variable)*as.numeric(plot_tallas$value)

TallasMedias_flota<-plot_tallas %>% mutate(value2=mediaTalla_flota) %>% group_by(year) %>%
  summarize(sum1 = sum(value),sum2 = sum(value2)) %>% mutate(Lmed=sum2/sum1)

ggplot(TallasMedias_flota,aes(x=year,y=Lmed))+ geom_point()+geom_line()+
  labs(x = 'Años', y = 'Tallas medias (cm)') +
  theme(panel.background = element_rect(fill = "gray99")) +
  theme(panel.grid=element_line(color=NA)) +
  ggtitle('Flota')+
  theme(plot.title = element_text(size = 6))
```



```

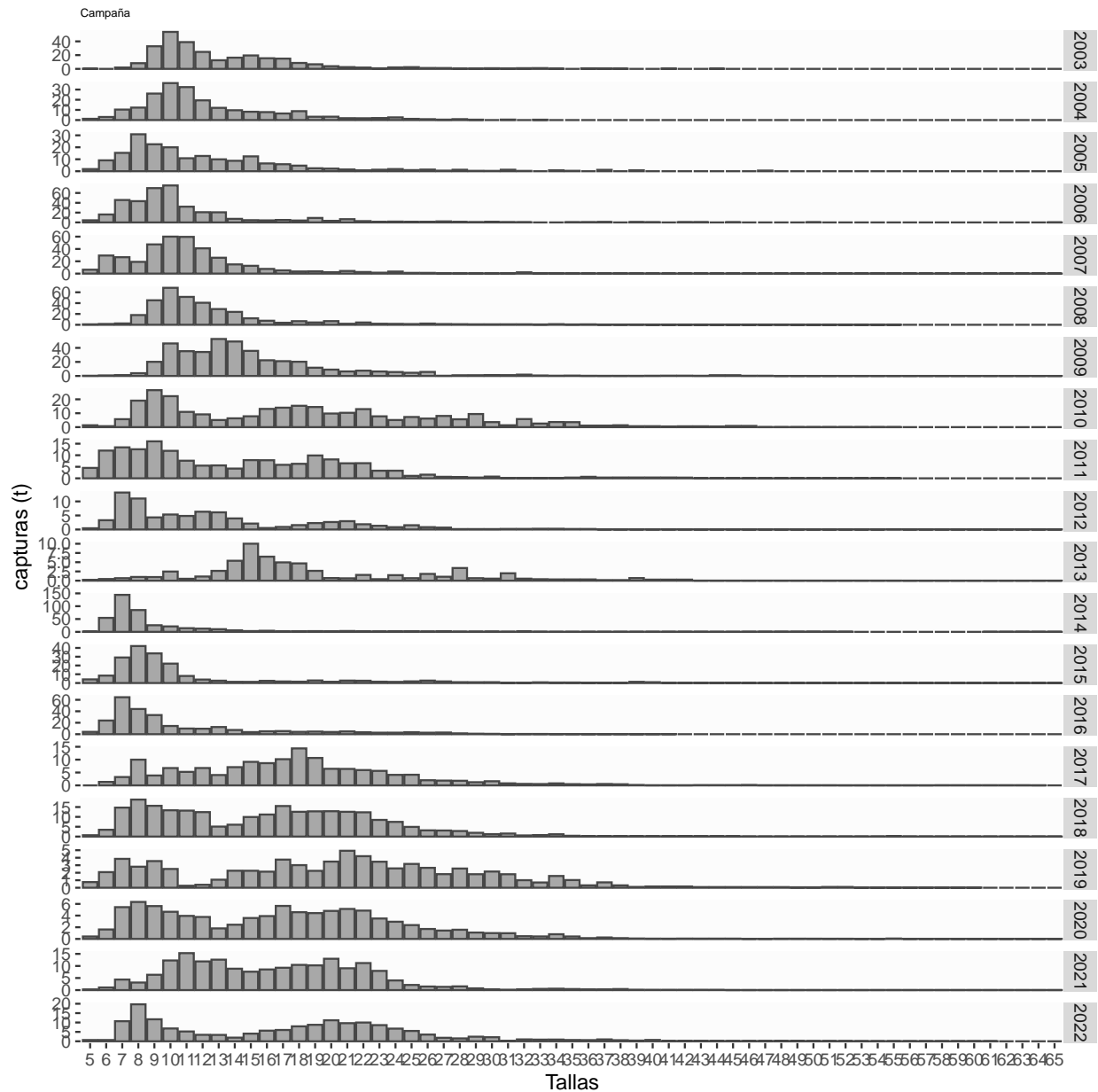
Indtallas<-read_excel(excel, sheet = "indices",col_names= TRUE,col_types=NULL,na="",skip= 1)

yearcatch<-seq(2003,2022,1)
tallas<-seq(5,65,1)
Indtallas<-Indtallas[2:62,10:29]

IndicesTallas<-data.frame(row.names = NULL,year=yearcatch,t(Indtallas))
colnames(IndicesTallas)<-c("year",tallas)
plot_Indtallas<-IndicesTallas %>% melt(id.vars=c("year"))

ggplot(plot_Indtallas) +
  geom_bar(aes(x = variable, y = value), stat="identity", fill='gray66', color = 'gray28')+
  facet_grid(rows = vars(year), as.table = TRUE,scales = "free") +
  labs(x = 'Tallas', y = 'capturas (t)') +
  theme(panel.background = element_rect(fill ="gray99")) +
  theme(panel.grid=element_line(color=NA)) +
  ggtitle('Campaña')+
  theme(plot.title = element_text(size = 6))

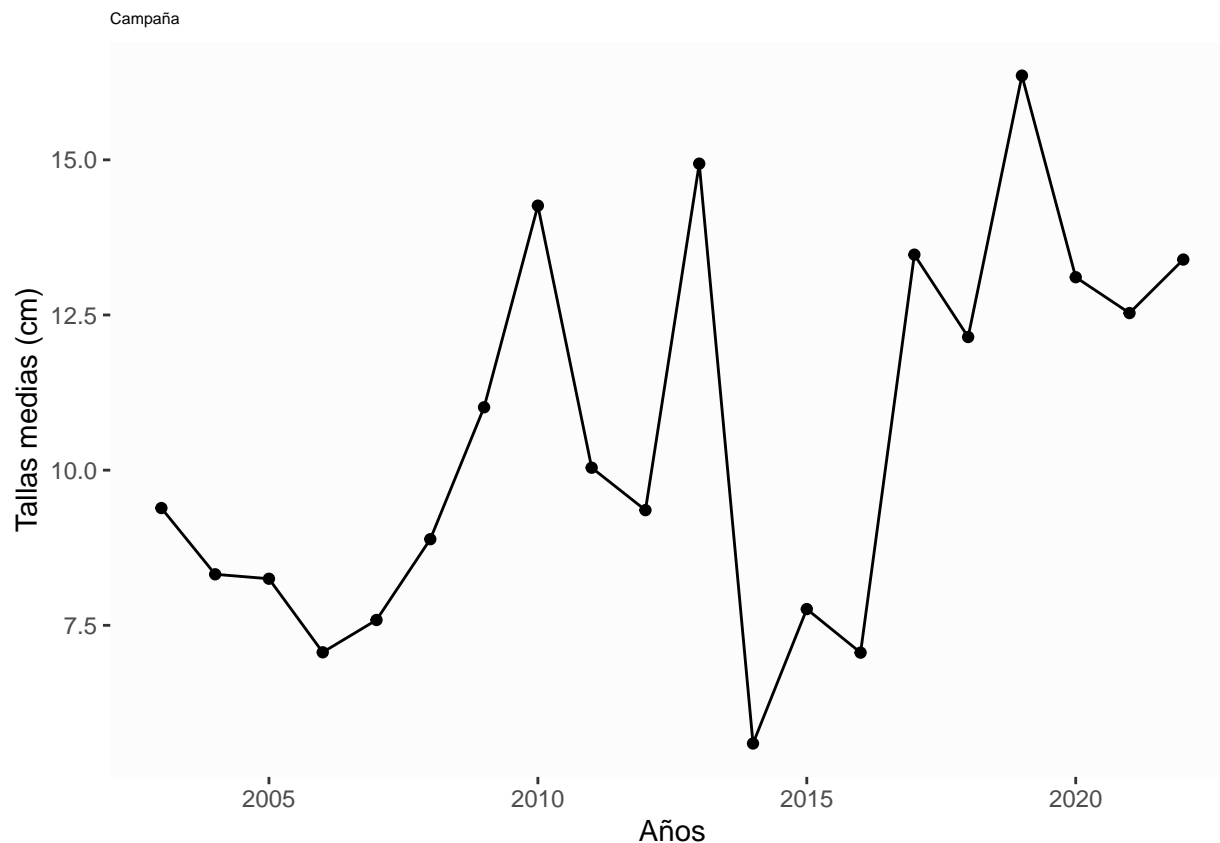
```



```
mediaTalla<-as.numeric(plot_Indtallas$variable)*as.numeric(plot_Indtallas$value)

TallasMedias_Ind<-plot_Indtallas %>% mutate(value2=mediaTalla) %>% group_by(year) %>%
  summarize(sum1 = sum(value),sum2 = sum(value2)) %>% mutate(Lmed=sum2/sum1)

ggplot(TallasMedias_Ind,aes(x=year,y=Lmed))+ geom_point()+geom_line()+
  labs(x = 'Años', y = 'Tallas medias (cm)') +
  theme(panel.background = element_rect(fill ="gray99")) +
  theme(panel.grid=element_line(color=NA)) +
  ggtitle('Campaña')+
  theme(plot.title = element_text(size = 6))
```



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$lbin_method <-2
dat1$binwidth    <-2
dat1$minimum_size <-10
dat1$maximum_size <-70
dat1$use_lencomp  <-0
```

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

`len_info` es una matriz que contiene la siguiente información:

- `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

```
len_info_mintailcomp<-rep(-1,2)
len_info_addtocomp<-rep(0.001,2)
len_info_combine_M_F<-rep(0,2)
len_info_CompressBins<-rep(0,2)
len_info_CompError<-rep(0,2)
len_info_ParmSelect<-rep(0,2)
len_info_minsamplesize<-rep(1,2)
#-----
# 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")
#-----
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
##          minsamplesize
## FISHERY              1
## SURVEY1              1
#-----
```

5.1.10.3 Especificación del vector de tallas

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

5.1.10.4 Datos de composición de tallas

lencomp es una matriz que contiene la siguiente información:

- Yr:
- Seas:
- FltSvy:
- Gender: sex codes: 0=combined; 1=use female only; 2=use male only; 3=use both as joint sexxlength distribution
- Part: partition codes: (0=combined; 1=discard; 2=retained)
- Nsamp:

```
#-----
# 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  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   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)

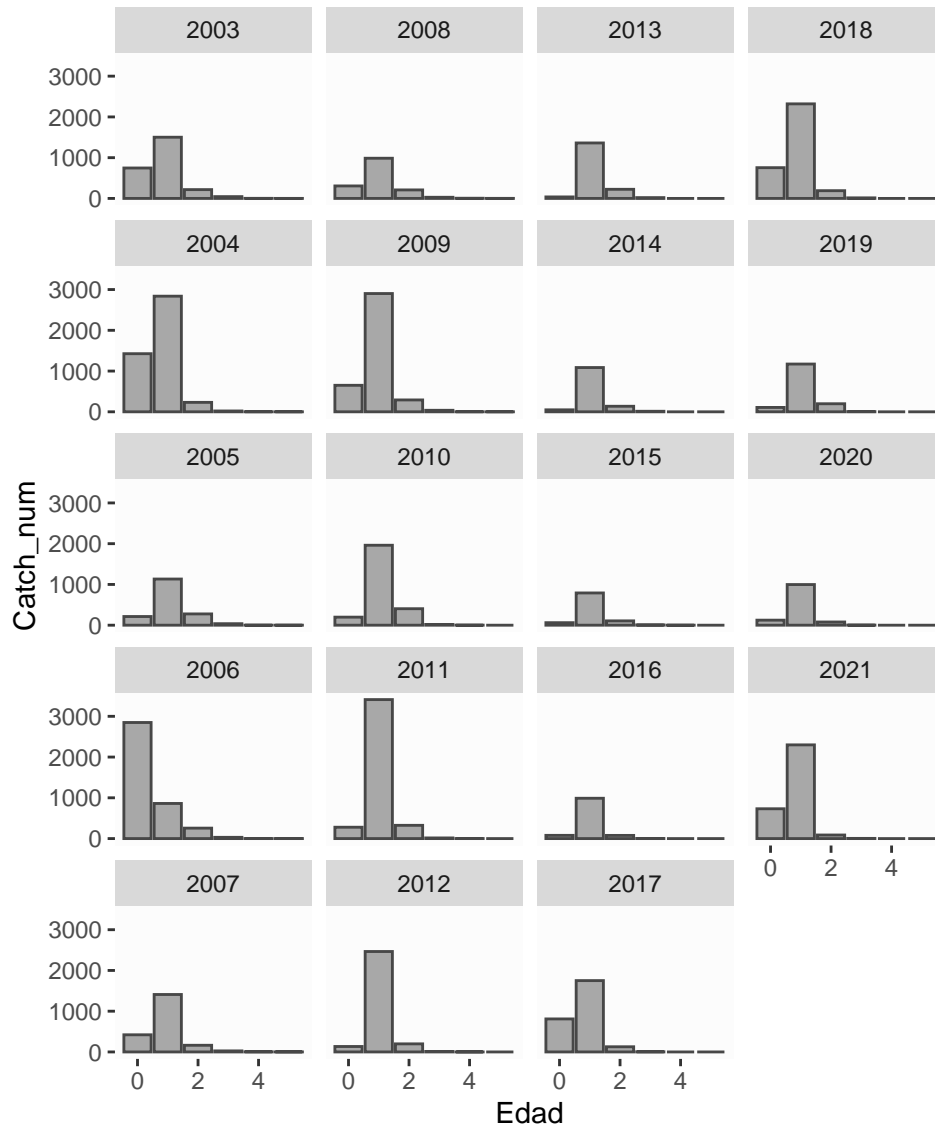
CATNUM.DAT
##      V1      V2      V3      V4      V5      V6
## 1  746.7 1502.0 216.0 44.0 2.1 0.5
## 2 1425.9 2837.4 231.9 20.5 1.7 0.5
## 3  212.8 1132.5 277.0 36.6 1.2 0.7
## 4 2848.6  861.6 256.2 30.9 1.2 1.4
## 5  421.0 1409.5 163.7 25.2 4.6 1.2
## 6  306.3  986.0 208.9 28.1 4.7 1.2
## 7  650.6 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

age<-seq(0,5,1)
years<-2003:2021
nyears<-length(years)

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

Catnum<-as.data.frame(CATNUM.DAT) %>%
  mutate(Yr=years) %>%
  melt(id.vars="Yr") %>%
  mutate(edad=rep(age,each=nyears)) %>%
  mutate(type='Catch')

ggplot(Catnum) +
  geom_bar(aes(x = edad, y = value), stat="identity", fill='gray66', color = 'gray28') +
  facet_wrap(vars(Yr), dir = 'v', as.table = TRUE) +
  labs(x = 'Edad', y = 'Catch_num') +
  theme(panel.background = element_rect(fill ="gray99")) +
  theme(panel.grid=element_line(color=NA)) + theme(plot.title = element_text(size = 12))
```



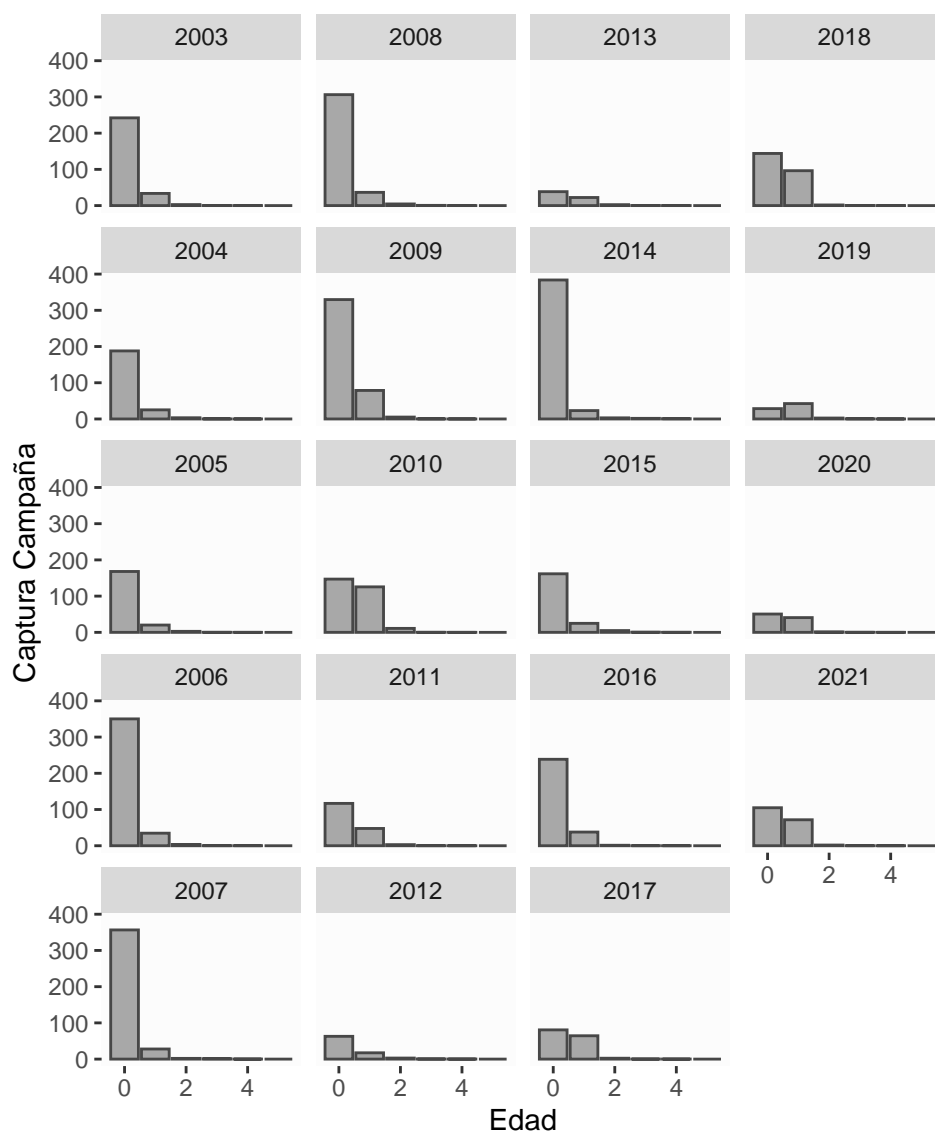
```

age<-seq(0,5,1)
years<-2003:2021
nyears<-length(years)

Indnum<-as.data.frame(TUNEFF.DAT[,2:7]) %>%
  mutate(Yr=years) %>%
  melt(id.vars="Yr") %>%
  mutate(edad=rep(age,each=nyears)) %>%
  mutate(type='Catch')

ggplot(Indnum) +
  geom_bar(aes(x = edad, y = value), stat="identity", fill='gray66', color = 'gray28') +
  facet_wrap(vars(Yr), dir = 'v', as.table = TRUE) +
  labs(x = 'Edad', y = 'Captura Campaña') +
  theme(panel.background = element_rect(fill = "gray99")) +
  theme(panel.grid=element_line(color=NA)) + theme(plot.title = element_text(size = 12))

```



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(-1,6),rep(0.001,6))
#-----
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]
## [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

age_info es una matriz que contiene las especificaciones de los datos de composición de edad. A continuación se detalla el contenido de esta matriz:

- 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

```
#-----
age_info1<-data.frame(mintailcomp=rep(-1,2),
                      addtocomp=rep(0.001,2),
                      combine_M_F=rep(0,2),
                      CompressBins=rep(0,2),
                      CompError=rep(0,2),
                      ParmSelect=rep(0,2),
                      minsamplesize=rep(1,2))

row.names(age_info1)<-c("FISHERY", "SURVEY1")
#-----
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
##          minsamplesize
## FISHERY              1
## SURVEY1              1
#-----
```

5.1.12.4 Datos de composicion de edad

Lbin_method se debe ingresar una de las siguientes opciones; 1=poplenbins; 2=datalenbins; 3=lengths
agecomp: es una matriz que contiene la siguiente información:

- Yr:
- Seas:

- FltSvy:
- Gender: sex codes: 0=combined;1=use female only; 2=use male only; 3=use both as joint sexxlength distribution
- Part: partition codes: (0=combined; 1=discard; 2=retained)
- Ageerr:
- Lbin_lo:
- Lbin_hi:
- Nsamp:

NOTA: si no se ingresan datos de edad entonces Lbin_method y agecomp = NULL

```
#-----
dat1$Lbin_method <- 1
#-----
new_agecomp<-data.frame(Yr=c(2003:2021,2003:2021),
                        Seas=rep(7,(19*2)),
                        FltSvy=c(rep(1,19), rep(2,19)),
                        Gender=rep(3,(19*2)),
                        Part=rep(0,(19*2)),
                        Ageerr=rep(2,(19*2)),
                        Lbin_lo=rep(-1,(19*2)),
                        Lbin_hi=rep(-1,(19*2)),
                        Nsamp=rep(75,(19*2)))

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

TUNEFF.DAT<-TUNEFF.DAT[,2:7]
names(TUNEFF.DAT)<-dat_rows_names

new_agecomp<-cbind(new_agecomp, rbind(CATNUM.DAT,TUNEFF.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
## 20	2003	7	2	3	0	2	-1	-1	75	242.2	33.6
## 21	2004	7	2	3	0	2	-1	-1	75	187.9	25.2
## 22	2005	7	2	3	0	2	-1	-1	75	168.1	20.2
## 23	2006	7	2	3	0	2	-1	-1	75	350.2	34.7
## 24	2007	7	2	3	0	2	-1	-1	75	356.6	27.9
## 25	2008	7	2	3	0	2	-1	-1	75	306.3	36.5
## 26	2009	7	2	3	0	2	-1	-1	75	329.6	78.8
## 27	2010	7	2	3	0	2	-1	-1	75	147.0	125.6
## 28	2011	7	2	3	0	2	-1	-1	75	116.9	47.7
## 29	2012	7	2	3	0	2	-1	-1	75	62.9	17.4
## 30	2013	7	2	3	0	2	-1	-1	75	38.4	22.3
## 31	2014	7	2	3	0	2	-1	-1	75	383.7	23.1
## 32	2015	7	2	3	0	2	-1	-1	75	161.7	24.9
## 33	2016	7	2	3	0	2	-1	-1	75	238.7	37.8
## 34	2017	7	2	3	0	2	-1	-1	75	80.7	64.4
## 35	2018	7	2	3	0	2	-1	-1	75	143.9	96.3
## 36	2019	7	2	3	0	2	-1	-1	75	28.6	42.5
## 37	2020	7	2	3	0	2	-1	-1	75	50.6	40.7
## 38	2021	7	2	3	0	2	-1	-1	75	104.9	71.7
##		E2	E3	E4	E5						
## 1		216.0	44.0	2.1	0.500						
## 2		231.9	20.5	1.7	0.500						
## 3		277.0	36.6	1.2	0.700						
## 4		256.2	30.9	1.2	1.400						
## 5		163.7	25.2	4.6	1.200						
## 6		208.9	28.1	4.7	1.200						
## 7		291.0	37.6	2.1	0.800						
## 8		404.5	18.4	2.2	0.300						
## 9		324.9	15.8	1.1	0.100						
## 10		198.4	11.7	0.7	0.100						
## 11		223.6	19.4	0.3	0.100						
## 12		136.4	13.3	0.3	0.200						
## 13		106.1	7.7	0.7	0.300						
## 14		78.1	2.7	0.1	0.100						
## 15		126.9	4.6	0.2	0.100						
## 16		189.8	13.7	0.3	0.100						
## 17		197.6	5.5	0.2	0.100						
## 18		78.3	3.0	0.3	0.100						
## 19		87.5	4.6	0.1	0.100						
## 20		2.7	0.2	0.1	0.001						
## 21		3.3	0.2	0.1	0.001						
## 22		2.8	0.2	0.1	0.001						
## 23		3.6	0.3	0.2	0.001						
## 24		2.0	1.2	0.2	0.001						
## 25		4.5	0.6	0.1	0.001						
## 26		5.2	0.3	0.1	0.001						

```
## 27 11.0 0.2 0.1 0.001
## 28 3.0 0.2 0.1 0.001
## 29 2.9 0.5 0.3 0.001
## 30 2.5 0.2 0.1 0.001
## 31 3.1 1.7 0.8 0.001
## 32 4.9 0.6 0.2 0.001
## 33 1.3 0.2 0.1 0.001
## 34 2.4 0.3 0.2 0.001
## 35 1.7 0.4 0.1 0.001
## 36 2.6 0.5 0.1 0.001
## 37 1.3 0.3 0.1 0.001
## 38 1.9 0.2 0.1 0.001
#-----
```

5.1.13 Otros datos

5.1.13.1 Tallas medias a la edad

use_MeanSize_at_Age_obs: Use_MeanSize-at-Age_obs (0/1) MeanSize_at_Age_obs es una matriz que contiene la siguiente información:

- Yr:
- Seas:
- FltSvy:
- Gender: sex codes: 0=combined;1=use female only; 2=use male only; 3=use both as joint sexlength distribution
- Part: partition codes: (0=combined; 1=discard; 2=retained
- AgeErr: ageerr codes: positive means mean length-at-age; negative means mean bodywt_at_age
- Ignore:

NOTA: si no se ingresan datos de tallas medias a la edad entonces use_MeanSize_at_Age_obs = 0 y MeanSize_at_Age_obs = NULL

```
#-----
dat1$use_MeanSize_at_Age_obs <- 0
dat1$use_MeanSize_at_Age_obs
## [1] 0
#-----
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$MeanSize_at_Age_obs<-NULL
dat1$MeanSize_at_Age_obs
## NULL
#-----
```

5.1.13.2 Datos ambientales

(tarea = buscar ejemplo donde ingresen estos datos)

```
dat1$N_environ_variables<-0
dat1$N_environ_variables
## [1] 0
```

5.1.13.3 Sizefreq data

(tarea = buscar ejemplo donde ingresen estos datos)


```
dat1$N_sizefreq_methods<-0
dat1$N_sizefreq_methods
## [1] 0
```

5.1.13.4 Datos de tags

(tarea = buscar ejemplo donde ingresen estos datos)

```
dat1$do_tags<-0
dat1$do_tags
## [1] 0
```

5.1.13.5 Datos de morfos

(tarea = buscar ejemplo donde ingresen estos datos)

```
dat1$morphcomp_data<-0
dat1$morphcomp_data
## [1] 0
```

5.1.13.5.1 Priors de selectividad

(tarea = buscar ejemplo donde ingresen estos datos)

```
dat1$use_selectivity_priors<-0
dat1$use_selectivity_priors
## [1] 0
```

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

```
ctl1 <- r4ss::SS_readctl(here(dirname.base, "control.ss"),
                        verbose = FALSE,
                        use_datlist = TRUE,
                        datlist = dat) #base

ctl1<-ctl1 # para modificar
names(ctl1) # muestra los objetos de la lista
## [1] "warnings" "Comments"
## [3] "nseas" "N_areas"
## [5] "Nages" "Nsexes"
## [7] "Npopbins" "Nfleets"
## [9] "Do_AgeKey" "fleetnames"
## [11] "sourcefile" "type"
## [13] "ReadVersion" "eof"
## [15] "EmpiricalWAA" "N_GP"
## [17] "N_platoon" "recr_dist_method"
## [19] "recr_global_area" "recr_dist_read"
## [21] "recr_dist_inx" "recr_dist_pattern"
## [23] "N_Block_Designs" "blocks_per_pattern"
## [25] "Block_Design" "time_vary_adjust_method"
## [27] "time_vary_auto_generation" "natM_type"
## [29] "GrowthModel" "Growth_Age_for_L1"
## [31] "Growth_Age_for_L2" "Exp_Decay"
## [33] "Growth_Placeholder" "N_natMparms"
## [35] "SD_add_to_LAA" "CV_Growth_Pattern"
## [37] "maturity_option" "First_Mature_Age"
## [39] "fecundity_option" "hermaphroditism_option"
## [41] "parameter_offset_approach" "MG_parms"
## [43] "MGparm_seas_effects" "SR_function"
## [45] "Use_steep_init_equi" "Sigma_R_FofCurvature"
## [47] "SR_parms" "do_recdev"
## [49] "MainRdevYrFirst" "MainRdevYrLast"
## [51] "recdev_phase" "recdev_adv"
## [53] "recdev_early_start" "recdev_early_phase"
## [55] "Fcast_recdev_phase" "lambda4Fcast_recdev_like"
## [57] "last_early_yr_nobias_adj" "first_yr_fullbias_adj"
## [59] "last_yr_fullbias_adj" "first_recent_yr_nobias_adj"
## [61] "max_bias_adj" "period_of_cycles_in_recdev"
## [63] "min_rec_dev" "max_rec_dev"
## [65] "N_Read_recdevs" "F_ballpark"
## [67] "F_ballpark_year" "F_Method"
## [69] "maxF" "F_iter"
## [71] "Q_options" "Q_parms"
## [73] "size_selex_types" "age_selex_types"
## [75] "age_selex_parms" "Use_2D_AR1_selectivity"
## [77] "TG_custom" "DoVar_adjust"
## [79] "maxlambdaphase" "sd_offset"
## [81] "lambdas" "N_lambdas"
## [83] "more_stddev_reporting" "stddev_reporting_specs"
## [85] "stddev_reporting_selex" "stddev_reporting_N_at_A"
```

5.2.1 Especificaciones iniciales

```
#ctl1$warnings
ctl1$Comments<-"#C 2023 Merluza archivo control modelo con datos de edad"
ctl1$nseas      <-1
ctl1$N_areas    <-1
ctl1$Nages      <-6
ctl1$Nsexes     <-1
ctl1$Npopbins   <-70
ctl1$Nfleets    <-2
ctl1$Do_AgeKey<-0
ctl1$fleetnames <- c("FISHERY", "SURVEY1")
#ctl1$sourcefile
#ctl1$type
#ctl1$ReadVersion
ctl1$eof        <- TRUE
```

5.2.2 Especificaciones del crecimiento

Consulte la Guía de usuario de SS3: [8.4.1 Weight-at-Age](#).

En este caso, no se usa, por lo que se establece en 0.

Si se usara el peso empírico por edad, SS3 ignoraría todas las entradas relacionadas con el crecimiento, la madurez y la fecundidad que se especifican más adelante en el archivo de **control.ss** (aunque todavía espera entradas).

Luego están las opciones para el número de patrones de crecimiento y platoons. Estos se establecen en 1 porque asumimos que toda la población tiene el mismo patrón de crecimiento y no hay platoons dentro de los patrones de crecimiento.

Opciones y especificaciones:

- EmpiricalWAA = 0 means do not read wtatage.ss; 1 means read and use wtatage.ss and also read and use growth parameters
- N_GP = N_Growth_Patterns (Growth Patterns, Morphs, Bio Patterns, GP are terms used interchangeably in SS3)
- N_platoon = N_platoons_Within_GrowthPattern

```
ctl1$EmpiricalWAA    <- 1
ctl1$N_GP            <- 1
ctl1$N_platoon       <- 1
```

5.3 Distribución del reclutamiento

Consulte la Guía de usuario de SS3: [8.4.2 Settlement Timing for Recruits and Distribution](#).

Luego se especifica la distribución del reclutamiento. Suponemos que para esta población solo hay 1 patrón de crecimiento, 1 tiempo de asentamiento (settlement) y 1 área, por lo que podemos usar la opción 4. Esta es la más simple, porque no requiere líneas completas de parámetros más adelante en el archivo **control.ss**.

Tenga en cuenta que también usamos 1 asignación de asentamiento (settlement) y especificamos que los peces se establecieron en enero (1) a la edad 0 (año calendario). Esto podría modificarse si los peces se establecieron a una edad o momento diferente.

Opciones y especificaciones:

- recr_dist_method = 2=main effects for GP, Area, Settle timing; 3=each Settle entity; 4=none (only when N_GP $N_{settlepop}$ ==1)

- `recr_global_area` = 1=global; 2=by area
- `recr_dist_read` = number of recruitment settlement assignments
- `recr_dist_inx` = unused option

`recr_dist_pattern` es una Matriz que contiene la siguiente información:

- `GPattern`:
- `month`:
- `area`:
- `age`:

(for each settlement assignment)

```
ctl1$recr_dist_method <- 2
ctl1$recr_global_area <- 1
ctl1$recr_dist_read <- 1
ctl1$recr_dist_inx <- 0
#-----
rec_pattern<-data.frame(row.names="recr_dist_pattern1",
                        "GPattern" = 1,
                        "month"    = 1,
                        "area"     = 1,
                        "age"      = 0)
ctl1$recr_dist_pattern <- rec_pattern
#-----
```

5.3.1 Bloques

No estamos usando bloques de tiempo, pero hay algunos especificados en el modelo. Tenga en cuenta que cualquier bloque puede especificarse y no usarse (los bloques se especifican aquí, pero no se usan hasta que se especifica dentro de una línea de parámetro), por lo que podría haber cualquier cantidad de bloques especificados aquí.

[Consulte la Guía de usuario de SS3: 8.4.4 Time Blocks.](#)

Opciones y especificaciones:

- `N_Block_Designs` = Number of block patterns. These patterns can be referred to in the parameter sections to create a separate parameter value for each block. Following inputs are omitted if the number of block patterns equals 0.
- `blocks_per_pattern` =
- `Block_Design` =

```
ctl1$N_Block_Designs <- 1
ctl1$blocks_per_pattern <- 1
ctl1$Block_Design[[1]] <- c(2003,2003)
```

5.3.2 Parámetros que varían en el tiempo

Así mismo, no se utilizan parámetros variables en el tiempo, por lo que no es importante lo que se especifica en los parámetros de control variables en el tiempo.

[Consulte la Guía de usuario de SS3: 8.4.5 Auto-generation.](#)

Opciones y especificaciones:

```
ctl1$time_vary_adjust_method <- 1
#-----
time_auto<-data.frame(matrix(rep(1,5),nrow=1,ncol=5))
```

```
colnames(time_auto)<-paste("time_vary_auto_generation_",seq(1,5,1),sep="")
```

```
ctl1$time_vary_auto_generation <-time_auto
```

```
ctl1$time_vary_auto_generation
```

```
##   time_vary_auto_generation_1 time_vary_auto_generation_2
```

```
## 1                               1                               1
```

```
##   time_vary_auto_generation_3 time_vary_auto_generation_4
```

```
## 1                               1                               1
```

```
##   time_vary_auto_generation_5
```

```
## 1                               1
```

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

5.3.3 Parámetros biológicos

Consulte la Guía de usuario de SS3: [8.5 Biology](#).

Las líneas de parámetros resultantes de la mortalidad natural, el crecimiento y la madurez (esta sección a veces se denomina parámetros MG) se especifican a continuación. El número de líneas de parámetros depende de las opciones seleccionadas en las líneas de especificación. Los parámetros también deben especificarse en un orden particular, con los parámetros de las hembras antes que los de machos en un modelo de 2 sexos.

Tenga en cuenta que la primera línea en el bloque de entrada SS anterior muestra los encabezados de columna. Todas las secciones con líneas de parámetros largas dentro del archivo de control tienen estos mismo encabezados. Hay muchas especificaciones en estas largas líneas de parámetros, pero algunas de particular importancia son:

- Todo lo que tenga fase negativa (séptimo valor en una línea larga de parámetros) no se estima y se establece en el valor inicial (tercer valor en la línea), mientras que las fases positivas sí se estiman.
- La mortalidad natural tanto para machos como para hembras se especifican en 0,1.
- Se estiman los tres parámetros de la curva de crecimiento de von Bertalanffy.

A continuación se presentan algunas opciones no utilizadas, porque no hay estacionalidad en los parámetros biológicos:

Opciones y especificaciones:

- **natM_type:** 0=1Parm;1=N_breakpoints;2=Lorenzen;3=agespecific;4=agespec_withseasinterpolate;[8.5.1 Natural Mortality](#). 5=BETA: _Maunder_link_to_maturity;6=Lorenzen_rangeno additional input for selected M option; read 1P per morph
- **GrowthModel:** 1=vonBert with L1&L2; 2=Richards with L1&L2; 3=age_specific_K_incr; 4=age_specific_K_decr; 5=age_specific_K_each; 6=NA; 7=NA; 8=growth cessation
- **Growth_Age_for_L1:** Age(post-settlement)_for_L1;linear growth below this
- **Growth_Age_for_L2:** 999 to use as Linf
- **Exp_Decay:** exponential decay for growth above maxage (value should approx initial Z; -999 replicates 3.24; -998 to not allow growth above maxage)
- **Growth_Placeholder:** placeholder for future growth feature
- **N_natMparms:**
- **SD_add_to_LAA:** SD_add_to_LAA (set to 0.1 for SS2 V1.x compatibility)
- **CV_Growth_Pattern:** 0=(CV=f(LAA)); 1=(CV=F(A)); 2=(SD=F(LAA)); 3=(SD=F(A)); 4=(logSD=F(A))
- **maturity_option:** 1=length logistic; 2=age logistic; 3=read age-maturity matrix by growth_pattern; 4=read age-fecundity; 5=disabled; 6=read length-maturity
- **First_Mature_Age:**
- **fecundity_option:** (1)eggs=Wt(a+bWt); (2)eggs=aL^b; (3)eggs=aWt^b; (4)eggs=a+bL; (5)eggs=a+bW
- **hermaphroditism_option:** 0=none; 1=female-to-male age-specific fxn; -1=male-to-female age-specific fxn
- **parameter_offset_approach:** 1- direct, no offset**; 2- male=fem_parmexp(male_parm); 3: male=femaleexp(parm) then old=young*exp(parm)

El siguiente código permite modificar las opciones antes especificadas:

```
# ctl1$natM_type          <- 3
# #incluir vector de mortalidad natural-----
# # leer vector de M natural desde archivos de GADGET
# medad0 <- list()
# medad1<-data.frame(matrix(c(2.21,1.3,1.3,1.3),nrow=1,ncol=4),row.names="#_natM1")
# colnames(medad1)<-paste("Age_",seq(0,3,1),sep="")
#
# medad0[[1]] <- medad1
```

```

# names(medad0)<-"natM"
# ctl1 <- append(ctl1, medad0, after = 28)

ctl1$natM_type      <- 0

#Crecimiento ----
ctl1$GrowthModel    <- 1 # Modelo de crecimiento
ctl1$Growth_Age_for_L1 <- 1 #Edad de referencia para el parámetro de primera talla a la edad (posterior)
ctl1$Growth_Age_for_L2 <- 25 #Edad de referencia para el segundo parámetro de talla por edad (999 para posterior)
ctl1$Exp_Decay      <- -999 #Replicar el cálculo más simple realizado en SS3 v.3.24.
ctl1$Growth_Placeholder <- 0
ctl1$SD_add_to_LAA   <- 0
ctl1$CV_Growth_Pattern <- 0 #(no puede ser variable)

#Madurez-fecundidad ----
ctl1$maturity_option <- 5
ctl1$First_Mature_Age <- 2
ctl1$fecundity_option <- 1
ctl1$hermaphroditism_option <- 0
ctl1$parameter_offset_approach <- 1

```

La función principal del archivo de control es definir los parámetros que utilizará el modelo. A continuación se indican las líneas de parámetros biológicos que serán modificadas:

```

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"

```

La definición de cada línea es la siguiente:

- NatM_p_1_Fem_GP_1 = Mortalidad natural para el patrón de crecimiento de hembras (Fem) 1, donde el número de parámetros de mortalidad natural depende de la opción seleccionada.
- L_at_Amin_Fem_GP_1 = Longitud en Amin (unidades en cm) para hembra, patrón de crecimiento 1.
- L_at_Amax_Fem_GP_1 = Longitud en Amax (unidades en cm) para hembra, patrón de crecimiento 1.
- VonBert_K_Fem_GP_1 = Coeficiente de crecimiento de von Bertalanffy (las unidades son por año) para las hembras, patrón de crecimiento 1.
- CV_young_Fem_GP_1 = variabilidad para el tamaño a la edad \leq Amin para las hembras, patrón de crecimiento 1. Tenga en cuenta que CV no puede variar con el tiempo, por lo que no figure env-link o un vector de desviación. Además, las unidades son como CV o como desviación estándar, dependiendo del valor asignado del patrón CV.
- CV_old_Fem_GP_1 = variabilidad para el tamaño a la edad \geq Amax para las hembras, patrón de crecimiento 1. Para edades intermedias, haga una interpolación lineal de CV en el tamaño medio a la edad. Tenga en cuenta que las unidades para CV dependerán del patrón CV y del valor del parámetro mortalidad-crecimiento como compensación. El valor del CV no puede variar con el tiempo.
- Wtlen_1_Fem_GP_1 = coeficiente para convertir la longitud en cm en peso en kg para las hembras.
- Wtlen_2_Fem_GP_1 = exponente en convertir la longitud a peso para hembras.

- `Mat50%_Fem_GP_1` = inflexión logística de madurez (en cm o años) donde la madurez femenina en logitud (o edad) es una función logística.
- `Mat_slope_Fem_GP_1` = pendiente logística (debe tener valor negativo).
- `Eggs_alpha_Fem_GP_1` = parámetros de fecundidad. El uso depende de la opción de fecundidad seleccionada.
- `Eggs_beta_Fem_GP_1`

Cada línea de parámetro biológicos contiene la siguiente información:

```
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"
```

La definición de cada línea es la siguiente:

- `LO` = un valor mínimo para el parámetro
- `HI` = un valor máximo para el parámetro
- `INIT` = valor inicial para el parámetro. Si la fase (descrita a continuación) para el parámetro es negativa, el parámetro se fija en este valor. Si se lee el archivo `ss.par`, sobrescribe estos valores `INIT`.
- `PRIOR` = valor esperado para el parámetro. Este valor se ignora si el `PR_type` es 0 (no prior) o 1 (symmetric beta). Si `PR_type` es lognormal (descrito a continuación).
- `PR_SD` = desviación standar de la `PRIOR`, utilizado para calcular la likelihood del valor del parámetro actual. Este valor es ignorado si la `PR_type` es 0.
- `PR_type` = tipo de distribución de error del valor esperado:
 - 0 = ninguno
 - 1 = symmetric beta
 - 2 = full beta
 - 3 = lognormal sin ajuste de sesgo
 - 4 = lognormal con ajuste de sesgo
 - 5 = gamma, y
 - 6 = normal.
- `PHASE` = fase en la que se empieza a estimar el parámetro. Un valor negativo hace que el parámetro conserve su valor `INIT` (o valor leído del archivo `ss.par`)
- `env_var&link` = crea un vinculo a una serie temporal ingresada al archivo de datos
- `dev_link` = invoca el uso del vector de desviación en la función linkage
- `dev_minyr`= año de inicio del vector de desviación
- `dev_maxyr`= año final para el vector de desviación
- `dev_PH` = fase de estimación para elementos en el vector de desviación
- `Block` = bloque de tiempo o tendencia a aplicar
- `Block_Fxn`= forma funcional para el desplazamiento de bloques

5.3.4 Mortalidad natural

La opción 0 se usa para la mortalidad natural porque solo se asume 1 valor.

5.3.5 Crecimiento

Consulte la Guía de usuario de SS3: 8.5.2 Growth.

El modelo de crecimiento 1 se usa para especificar un modelo de crecimiento de von Bertalanffy, seguido de líneas que especifican detalles sobre el crecimiento.

5.3.6 Madurez-fecundidad

Consulte la Guía de usuario de SS3: 8.5.3 Maturity-Fecundity.

5.3.7 Hermafroditismo

Consulte la Guía de usuario de SS3: 8.5.4 Hermaphroditism.

```
##-----
MG_parms1<-data.frame("L0"          = 0.05,
                      "HI"          = 0.4,
                      "INIT"        = 0.18,
                      "PRIOR"       = -1.60944,
                      "PR_SD"       = 0.1,
                      "PR_type"     = 0,
                      "PHASE"       = -4,
                      "env_var&link" = 0,
                      "dev_link"    = 0,
                      "dev_minyr"   = 0,
                      "dev_maxyr"   = 0,
                      "dev_PH"      = 0,
                      "Block"       = 0,
                      "Block_Fxn"   = 0)

##-----
ctl1$MG_parms["NatM_p_1_Fem_GP_1",] <- MG_parms1
##-----

MG_parms2<-data.frame("L0"          = 2,
                      "HI"          = 15,
                      "INIT"        = 5,
                      "PRIOR"       = 32,
                      "PR_SD"       = 99,
                      "PR_type"     = 0,
                      "PHASE"       = -5,
                      "env_var&link" = 0,
                      "dev_link"    = 0,
                      "dev_minyr"   = 0,
                      "dev_maxyr"   = 0,
                      "dev_PH"      = 0,
                      "Block"       = 0,
                      "Block_Fxn"   = 0)

##-----
ctl1$MG_parms["L_at_Amin_Fem_GP_1",] <-MG_parms2
##-----

MG_parms3<-data.frame("L0"          = 45,
```

```

        "HI"            = 60,
        "INIT"          = 53,
        "PRIOR"         = 50,
        "PR_SD"         = 99,
        "PR_type"       = 0,
        "PHASE"         = -3,
        "env_var&link"   = 0,
        "dev_link"      = 0,
        "dev_minyr"     = 0,
        "dev_maxyr"     = 0,
        "dev_PH"        = 0,
        "Block"         = 0,
        "Block_Fxn"     = 0)

##-----
ctl1$MG_parms["L_at_Amax_Fem_GP_1",] <- MG_parms3
##-----

MG_parms4<-data.frame("L0"            = 0.2,
        "HI"            = 0.4,
        "INIT"          = 0.3,
        "PRIOR"         = 0.3,
        "PR_SD"         = 99,
        "PR_type"       = 0,
        "PHASE"         = -3,
        "env_var&link"   = 0,
        "dev_link"      = 0,
        "dev_minyr"     = 0,
        "dev_maxyr"     = 0,
        "dev_PH"        = 0,
        "Block"         = 0,
        "Block_Fxn"     = 0)

##-----
ctl1$MG_parms["VonBert_K_Fem_GP_1",]<-MG_parms4
##-----

MG_parms5<-data.frame("L0"            = 0.03,
        "HI"            = 0.16,
        "INIT"          = 0.066,
        "PRIOR"         = 0.1,
        "PR_SD"         = 99,
        "PR_type"       = 0,
        "PHASE"         = -5,
        "env_var&link"   = 0,
        "dev_link"      = 0,
        "dev_minyr"     = 0,
        "dev_maxyr"     = 0,
        "dev_PH"        = 0,
        "Block"         = 0,
        "Block_Fxn"     = 0)

##-----
ctl1$MG_parms["CV_young_Fem_GP_1",] <- MG_parms5
##-----

```

```

MG_parms6<-data.frame("LO"          = 0.03,
                      "HI"          = 0.16,
                      "INIT"        = 0.062,
                      "PRIOR"       = 0.1,
                      "PR_SD"       = 99,
                      "PR_type"     = 0,
                      "PHASE"       = -5,
                      "env_var&link" = 0,
                      "dev_link"    = 0,
                      "dev_minyr"   = 0,
                      "dev_maxyr"   = 0,
                      "dev_PH"      = 0,
                      "Block"       = 0,
                      "Block_Fxn"   = 0)

##-----
ctl1$MG_parms["CV_old_Fem_GP_1",] <- MG_parms6
##-----

MG_parms7<-data.frame("LO"          = -3,
                      "HI"          = 3,
                      "INIT"        = 7e-06,
                      "PRIOR"       = 7e-06,
                      "PR_SD"       = 99,
                      "PR_type"     = 0,
                      "PHASE"       = -50,
                      "env_var&link" = 0,
                      "dev_link"    = 0,
                      "dev_minyr"   = 0,
                      "dev_maxyr"   = 0,
                      "dev_PH"      = 0,
                      "Block"       = 0,
                      "Block_Fxn"   = 0)

##-----
ctl1$MG_parms["Wtlen_1_Fem_GP_1",] <- MG_parms7
##-----

MG_parms8<-data.frame("LO"          = -3,
                      "HI"          = 3,
                      "INIT"        = 2.9624,
                      "PRIOR"       = 2.9624,
                      "PR_SD"       = 99,
                      "PR_type"     = 0,
                      "PHASE"       = -50,
                      "env_var&link" = 0,
                      "dev_link"    = 0,
                      "dev_minyr"   = 0,
                      "dev_maxyr"   = 0,
                      "dev_PH"      = 0,
                      "Block"       = 0,
                      "Block_Fxn"   = 0)

##-----
ctl1$MG_parms["Wtlen_2_Fem_GP_1",] <-MG_parms8
##-----

```

```

MG_parms9<-data.frame("LO"          = -3,
                      "HI"          = 43,
                      "INIT"        = 37,
                      "PRIOR"       = 37,
                      "PR_SD"       = 99,
                      "PR_type"     = 0,
                      "PHASE"       = -50,
                      "env_var&link" = 0,
                      "dev_link"    = 0,
                      "dev_minyr"   = 0,
                      "dev_maxyr"   = 0,
                      "dev_PH"      = 0,
                      "Block"       = 0,
                      "Block_Fxn"   = 0)

##-----
ctl1$MG_parms["Mat50%_Fem_GP_1",]<- MG_parms9
##-----

MG_parms10<-data.frame("LO"          = -3,
                       "HI"          = 3,
                       "INIT"        = -0.48,
                       "PRIOR"       = -0.48,
                       "PR_SD"       = 99,
                       "PR_type"     = 0,
                       "PHASE"       = -50,
                       "env_var&link" = 0,
                       "dev_link"    = 0,
                       "dev_minyr"   = 0,
                       "dev_maxyr"   = 0,
                       "dev_PH"      = 0,
                       "Block"       = 0,
                       "Block_Fxn"   = 0)

##-----
ctl1$MG_parms["Mat_slope_Fem_GP_1",]<-MG_parms10
##-----

MG_parms11<-data.frame("LO"          = -3,
                       "HI"          = 3,
                       "INIT"        = 1,
                       "PRIOR"       = 1,
                       "PR_SD"       = 99,
                       "PR_type"     = 0,
                       "PHASE"       = -50,
                       "env_var&link" = 0,
                       "dev_link"    = 0,
                       "dev_minyr"   = 0,
                       "dev_maxyr"   = 0,
                       "dev_PH"      = 0,
                       "Block"       = 0,
                       "Block_Fxn"   = 0)

##-----
ctl1$MG_parms["Eggs/kg_inter_Fem_GP_1",] <-MG_parms11
##-----

```

```

MG_parms12<-data.frame("LO"          = -3,
                        "HI"          = 3,
                        "INIT"         = 0,
                        "PRIOR"        = 0,
                        "PR_SD"        = 99,
                        "PR_type"       = 0,
                        "PHASE"        = -50,
                        "env_var&link" = 0,
                        "dev_link"     = 0,
                        "dev_minyr"    = 0,
                        "dev_maxyr"    = 0,
                        "dev_PH"       = 0,
                        "Block"        = 0,
                        "Block_Fxn"    = 0)

##-----
ctl1$MG_parms["Eggs/kg_slope_wt_Fem_GP_1",] <- MG_parms12
##-----

MG_parms21<-data.frame("LO"          = 0,
                        "HI"          = 2,
                        "INIT"         = 1,
                        "PRIOR"        = 1,
                        "PR_SD"        = 99,
                        "PR_type"       = 0,
                        "PHASE"        = -50,
                        "env_var&link" = 0,
                        "dev_link"     = 0,
                        "dev_minyr"    = 0,
                        "dev_maxyr"    = 0,
                        "dev_PH"       = 0,
                        "Block"        = 0,
                        "Block_Fxn"    = 0)

##-----
ctl1$MG_parms["RecrDist_GP_1",] <- MG_parms21
##-----

MG_parms22<-data.frame("LO"          = 0,
                        "HI"          = 2,
                        "INIT"         = 1,
                        "PRIOR"        = 1,
                        "PR_SD"        = 99,
                        "PR_type"       = 0,
                        "PHASE"        = -50,
                        "env_var&link" = 0,
                        "dev_link"     = 0,
                        "dev_minyr"    = 0,
                        "dev_maxyr"    = 0,
                        "dev_PH"       = 0,
                        "Block"        = 0,
                        "Block_Fxn"    = 0)

##-----
ctl1$MG_parms["RecrDist_Area_1",] <- MG_parms22
##-----

```

```

MG_parms23<-data.frame("LO"           = 0,
                      "HI"           = 2,
                      "INIT"         = 1,
                      "PRIOR"        = 1,
                      "PR_SD"        = 99,
                      "PR_type"      = 0,
                      "PHASE"        = -50,
                      "env_var&link" = 0,
                      "dev_link"     = 0,
                      "dev_minyr"    = 0,
                      "dev_maxyr"    = 0,
                      "dev_PH"       = 0,
                      "Block"        = 0,
                      "Block_Fxn"    = 0)

##-----
ctl1$MG_parms["RecrDist_month_1",]<- MG_parms23
##-----

MG_parms24<-data.frame("LO"           = 1,
                      "HI"           = 1,
                      "INIT"         = 1,
                      "PRIOR"        = 1,
                      "PR_SD"        = 1,
                      "PR_type"      = 0,
                      "PHASE"        = -1,
                      "env_var&link" = 0,
                      "dev_link"     = 0,
                      "dev_minyr"    = 0,
                      "dev_maxyr"    = 0,
                      "dev_PH"       = 0,
                      "Block"        = 0,
                      "Block_Fxn"    = 0)

##-----
ctl1$MG_parms["CohortGrowDev",] <- MG_parms24
##-----

MG_parms25<-data.frame("LO"           = 0.000001,
                      "HI"           = 0.999999,
                      "INIT"         = 0.999999,
                      "PRIOR"        = 0.5,
                      "PR_SD"        = 0.5,
                      "PR_type"      = 0,
                      "PHASE"        = -99,
                      "env_var&link" = 0,
                      "dev_link"     = 0,
                      "dev_minyr"    = 0,
                      "dev_maxyr"    = 0,
                      "dev_PH"       = 0,
                      "Block"        = 0,
                      "Block_Fxn"    = 0)

##-----
ctl1$MG_parms["FracFemale_GP_1",] <-MG_parms25
##-----

```

```
##-----
#elimino los parámetros que no se utilizaran en el modelo trimestral por el momento
patron_eliminar<-rownames(ctl1$MG_parms)[grep("_Mal_",rownames(ctl1$MG_parms))]
ctl1$MG_parms <- subset(ctl1$MG_parms, !rownames(ctl1$MG_parms) %in%
                        c(patron_eliminar,
                          "Eggs_beta_Fem_GP_1","Eggs_alpha_Fem_GP_1"))

# revisa salidas primeras columnas

ctl1$MG_parms
##
```

	LO	HI	INIT	PRIOR	PR_SD	PR_type
## NatM_p_1_Fem_GP_1	5e-02	0.400000	0.180000	-1.609440	0.1	0
## L_at_Amin_Fem_GP_1	2e+00	15.000000	5.000000	32.000000	99.0	0
## L_at_Amax_Fem_GP_1	4e+01	90.000000	71.649200	70.000000	10.0	6
## VonBert_K_Fem_GP_1	2e-01	0.400000	0.300000	0.300000	99.0	0
## CV_young_Fem_GP_1	3e-02	0.160000	0.066000	0.100000	99.0	0
## CV_old_Fem_GP_1	3e-02	0.160000	0.062000	0.100000	99.0	0
## Wtlen_1_Fem_GP_1	-3e+00	3.000000	0.000007	0.000007	99.0	0
## Wtlen_2_Fem_GP_1	-3e+00	3.000000	2.962400	2.962400	99.0	0
## Mat50%_Fem_GP_1	-3e+00	43.000000	37.000000	37.000000	99.0	0
## Mat_slope_Fem_GP_1	-3e+00	3.000000	-0.480000	-0.480000	99.0	0
## RecrDist_GP_1	0e+00	2.000000	1.000000	1.000000	99.0	0
## RecrDist_Area_1	0e+00	2.000000	1.000000	1.000000	99.0	0
## RecrDist_month_1	0e+00	2.000000	1.000000	1.000000	99.0	0
## CohortGrowDev	1e+00	1.000000	1.000000	1.000000	1.0	0
## FracFemale_GP_1	1e-06	0.999999	0.999999	0.500000	0.5	0
## Eggs/kg_inter_Fem_GP_1	-3e+00	3.000000	1.000000	1.000000	99.0	0
## Eggs/kg_slope_wt_Fem_GP_1	-3e+00	3.000000	0.000000	0.000000	99.0	0

```
##
```

	PHASE	env_var&link	dev_link	dev_minyr	dev_maxyr
## NatM_p_1_Fem_GP_1	-4	0	0	0	0
## L_at_Amin_Fem_GP_1	-5	0	0	0	0
## L_at_Amax_Fem_GP_1	-4	0	0	0	0
## VonBert_K_Fem_GP_1	-3	0	0	0	0
## CV_young_Fem_GP_1	-5	0	0	0	0
## CV_old_Fem_GP_1	-5	0	0	0	0
## Wtlen_1_Fem_GP_1	-50	0	0	0	0
## Wtlen_2_Fem_GP_1	-50	0	0	0	0
## Mat50%_Fem_GP_1	-50	0	0	0	0
## Mat_slope_Fem_GP_1	-50	0	0	0	0
## RecrDist_GP_1	-50	0	0	0	0
## RecrDist_Area_1	-50	0	0	0	0
## RecrDist_month_1	-50	0	0	0	0
## CohortGrowDev	-1	0	0	0	0
## FracFemale_GP_1	-99	0	0	0	0
## Eggs/kg_inter_Fem_GP_1	-50	0	0	0	0
## Eggs/kg_slope_wt_Fem_GP_1	-50	0	0	0	0

```
##
```

	dev_PH	Block	Block_Fxn
## NatM_p_1_Fem_GP_1	0	0	0
## L_at_Amin_Fem_GP_1	0	0	0
## L_at_Amax_Fem_GP_1	0	0	0
## VonBert_K_Fem_GP_1	0	0	0
## CV_young_Fem_GP_1	0	0	0
## CV_old_Fem_GP_1	0	0	0


```
## Wtlen_1_Fem_GP_1      0      0      0
## Wtlen_2_Fem_GP_1      0      0      0
## Mat50%_Fem_GP_1      0      0      0
## Mat_slope_Fem_GP_1    0      0      0
## RecrDist_GP_1         0      0      0
## RecrDist_Area_1       0      0      0
## RecrDist_month_1      0      0      0
## CohortGrowDev         0      0      0
## FracFemale_GP_1       0      0      0
## Eggs/kg_inter_Fem_GP_1 0      0      0
## Eggs/kg_slope_wt_Fem_GP_1 0      0      0
```

5.3.7.0.1 Seasonal_effects_on_biology_parms

Los efectos estacionales están disponibles para los parámetros de peso-longitud, madurez, fecundidad y para el parámetro de crecimiento K. Los valores de los parámetros estacionales ajustan el valor del parámetro base para esa temporada.

MGparm_seas_effects: Estacionalidad para parámetros biológicos seleccionados (no una entrada condicional). Lea 10 enteros para especificar qué parámetros biológicos tienen estacionalidad: female-wtlen1, female-wtlen2, maturity1, maturity2, fecundity1, fecundity2, male-wtlen1, male-wtlen2, L1, K. La lectura de un valor positivo selecciona ese factor para la estacionalidad.

Consulte la Guía de usuario de SS3: 8.5.12 Seasonal Biology Parameters.

```
MGparm_seas_effects1<-data.frame(matrix(rep(0,10),nrow=1,ncol=10))
colnames(MGparm_seas_effects1)<-paste("MGparm_seas_effects_",seq(1,10,1),sep="")

ctl1$MGparm_seas_effects <-MGparm_seas_effects1
ctl1$MGparm_seas_effects
## MGparm_seas_effects_1 MGparm_seas_effects_2 MGparm_seas_effects_3
## 1      0      0      0
## MGparm_seas_effects_4 MGparm_seas_effects_5 MGparm_seas_effects_6
## 1      0      0      0
## MGparm_seas_effects_7 MGparm_seas_effects_8 MGparm_seas_effects_9
## 1      0      0      0
## MGparm_seas_effects_10
## 1      0
##-----
```

5.3.8 Relación stock recluta

Consulte la Guía de usuario de SS3: 8.6 Spawner-Recruitment.

- SR_function: 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
- Use_steep_init_equi: 0/1 to use steepness in initial equ recruitment calculation
- ctl1\$Sigma_R_FofCurvature: future feature: 0/1 to make realized sigmaR a function of SR curvature
- Sigma_R_FofCurvature: future feature: 0/1 to make realized sigmaR a function of SR curvature

```
ctl1$SR_function      <-4
ctl1$Use_steep_init_equi <-0
ctl1$Sigma_R_FofCurvature <-0
```

```
##-----
SR_parms1<-data.frame("L0"      = 5,
```

```

        "HI"            = 30,
        "INIT"          = 20,
        "PRIOR"         = 15,
        "PR_SD"         = 99,
        "PR_type"       = 0,
        "PHASE"         = 1,
        "env_var&link"   = 0,
        "dev_link"      = 0,
        "dev_minyr"     = 0,
        "dev_maxyr"     = 0,
        "dev_PH"        = 0,
        "Block"         = 0,
        "Block_Fxn"     = 0)
ct11$SR_parms[1,]<-SR_parms1
rownames(ct11$SR_parms)[1]<-"SR_LN(R0)"

##-----
SR_parms2<-data.frame("L0"            = 0.2,
        "HI"            = 1,
        "INIT"          = 0.88,
        "PRIOR"         = 0.777,
        "PR_SD"         = 0.113,
        "PR_type"       = 2,
        "PHASE"         = -4,
        "env_var&link"   = 0,
        "dev_link"      = 0,
        "dev_minyr"     = 0,
        "dev_maxyr"     = 0,
        "dev_PH"        = 0,
        "Block"         = 0,
        "Block_Fxn"     = 0)

##-----
ct11$SR_parms[2,] <-SR_parms2
rownames(ct11$SR_parms)[2]<-"SR_SCAA_null"
##-----

SR_parms3<-data.frame("L0"            = 0.3,
        "HI"            = 1.6,
        "INIT"          = 0.6,
        "PRIOR"         = 1.1,
        "PR_SD"         = 99,
        "PR_type"       = 0,
        "PHASE"         = -6,
        "env_var&link"   = 0,
        "dev_link"      = 0,
        "dev_minyr"     = 0,
        "dev_maxyr"     = 0,
        "dev_PH"        = 0,
        "Block"         = 0,
        "Block_Fxn"     = 0)

##-----
ct11$SR_parms[3,] <-SR_parms3
rownames(ct11$SR_parms)[3]<-"SR_sigmaR"

```

```
##-----
SR_parms4<-data.frame("LO"           = -5,
                      "HI"           = 5,
                      "INIT"          = 0,
                      "PRIOR"         = 0,
                      "PR_SD"         = 99,
                      "PR_type"        = 0,
                      "PHASE"         = -50,
                      "env_var&link"  = 0,
                      "dev_link"      = 0,
                      "dev_minyr"     = 0,
                      "dev_maxyr"     = 0,
                      "dev_PH"        = 0,
                      "Block"         = 0,
                      "Block_Fxn"     = 0)

##-----
ctl1$SR_parms[4,] <-SR_parms4
rownames(ctl1$SR_parms)[4]<-"SR_regime"
##-----

SR_parms5<-data.frame("LO"           = 0,
                      "HI"           = 2,
                      "INIT"          = 0,
                      "PRIOR"         = 1,
                      "PR_SD"         = 99,
                      "PR_type"        = 0,
                      "PHASE"         = -50,
                      "env_var&link"  = 0,
                      "dev_link"      = 0,
                      "dev_minyr"     = 0,
                      "dev_maxyr"     = 0,
                      "dev_PH"        = 0,
                      "Block"         = 0,
                      "Block_Fxn"     = 0)

ctl1$SR_parms[5,] <- SR_parms5
rownames(ctl1$SR_parms)[5]<-"SR_autocorr"
##-----
ctl1$SR_parms
##
##      LO    HI   INIT  PRIOR  PR_SD  PR_type  PHASE  env_var&link  dev_link
## SR_LN(R0)  5.0 30.0 20.00 15.000 99.000      0      1           0          0
## SR_SCAA_null 0.2  1.0  0.88  0.777  0.113      2     -4           0          0
## SR_sigmaR    0.3  1.6  0.60  1.100 99.000      0     -6           0          0
## SR_regime   -5.0  5.0  0.00  0.000 99.000      0    -50           0          0
## SR_autocorr  0.0  2.0  0.00  1.000 99.000      0    -50           0          0
##
##      dev_minyr dev_maxyr dev_PH Block Block_Fxn
## SR_LN(R0)           0           0      0      0
## SR_SCAA_null        0           0      0      0
## SR_sigmaR           0           0      0      0
## SR_regime           0           0      0      0
## SR_autocorr         0           0      0      0
```

5.3.9 Desvíos de los reclutamientos

Consulte la Guía de usuario de SS3: 8.6.5 Recruitment Deviation Setup.

Las opciones de desviación de reclutamiento se especifican después de los parámetros de SR. Primero están las opciones de desviaciones estándar de reclutamiento:

- `do_recdev`: 0=none; 1=devvector ($R=F(SSB)+dev$); 2=deviations ($R=F(SSB)+dev$); 3=deviations ($R=R0*dev$; $dev2=R-f(SSB)$); 4=like 3 with `sum(dev2)` adding penalty
- `MainRdevYrFirst`: first year of main `recr_devs`; early devs can precede this era
- `MainRdevYrLast`: last year of main `recr_devs`; forecast devs start in following year
- `recdev_phase`:

Estos definen las principales desviaciones de reclutamientos, que en este caso duran desde el primer año del modelo hasta el último año. También se leen opciones avanzadas:

- `recdev_adv`: (0/1) to read 13 advanced options
- `recdev_early_start`: (0=none; neg value makes relative to `recdev_start`)
- `recdev_early_phase`:
- `Fcast_recr_phase`: (incl. late recr) (0 value resets to `maxphase+1`)
- `lambda4Fcast_recr_like`: lambda for `Fcast_recr_like` occurring before `endyr+1`
- `last_early_yr_nobias_adj`: `last_yr_nobias_adj_in_MPD`; begin of ramp
- `first_yr_fullbias_adj`: `first_yr_fullbias_adj_in_MPD`; begin of plateau
- `last_yr_fullbias_adj`: `last_yr_fullbias_adj_in_MPD`
- `first_recent_yr_nobias_adj`: `end_yr_for_ramp_in_MPD` (can be in forecast to shape ramp, but SS3 sets `bias_adj` to 0.0 for fcast yrs)
- `max_bias_adj`: `max_bias_adj_in_MPD`; (typical ~0.8; -3 sets all years to 0.0; -2 sets all non-forecast yrs w/ estimated `recdevs` to 1.0; -1 sets `biasadj=1.0` for all yrs w/ `recdevs`).
- `period_of_cycles_in_recr`: period of cycles in recruitment (N parms read below)
- `min_rec_dev`:
- `max_rec_dev`:
- `N_Read_recdevs`:

Las opciones avanzadas permiten al usuario sesgar el ajuste de las desviaciones de reclutamiento. Hay más información sobre el ajuste del sesgo en el manual del usuario de SS, pero la idea general es tener en cuenta el hecho de que las desviaciones de reclutamiento anteriores y posteriores probablemente tengan menos información que las del medio.

La rampa de ajuste de sesgo tiene esto en cuenta y, por lo general, se “ajusta” observando la rampa de sesgo en los resultados del modelo después de ejecutarlo, volviendo a especificar la rampa de sesgo según sea necesario y volviendo a ejecutar el modelo

```
ctl1$do_recdev      <- 1
ctl1$MainRdevYrFirst <- 2003
ctl1$MainRdevYrLast  <- 2021
ctl1$recdev_phase    <- 1
ctl1$recdev_adv      <- 0
ctl1$recdev_early_start <- NULL
ctl1$recdev_early_phase <- NULL
ctl1$Fcast_recr_phase  <- NULL
ctl1$lambda4Fcast_recr_like <-NULL
```

```
ctl1$last_early_yr_nobias_adj <-NULL
ctl1$first_yr_fullbias_adj <-NULL
ctl1$last_yr_fullbias_adj <-NULL
ctl1$first_recent_yr_nobias_adj <-NULL
ctl1$max_bias_adj <-NULL
ctl1$period_of_cycles_in_recr <-NULL
ctl1$min_rec_dev <- NULL
ctl1$max_rec_dev <- NULL
ctl1$N_Read_recdevs <- NULL
```

5.3.10 Mortalidad por pesca

Consulte la Guía de usuario de SS3: [8.7 Fishing Mortality Method](#).

- `F_ballpark`: F ballpark value in units of `annual_F`
- `F_ballpark_year`: F ballpark year (neg value to disable)
- `F_Method`: 1=Pope midseason rate; 2=F as parameter; 3=F as hybrid; 4=fleet-specific parm/hybrid (#4 is superset of #2 and #3 and is recommended)
- `maxF`: max F (methods 2-4) or harvest fraction (method 1)
- `F_iter`: N iterations for tuning in hybrid mode; recommend 3 (faster) to 5 (more precise if many fleets)

```
#-----  
ctl1$F_ballpark <- 0.1  
ctl1$F_ballpark_year <- -2003  
ctl1$F_Method <- 3  
ctl1$maxF <- 0.95  
ctl1$F_iter <- NULL  
#-----
```

5.3.11 Capturabilidad

Consulte la Guía de usuario de SS3: 8.8 Catchability.

5.3.12 Opciones de capturabilidad por flota (CPUE o CAMPAÑAS)

- `fleet`: number
- `link_type`: (1=simple q, 1 parm;2=mirror simple q, 1 mirrored parm;3=q and power, 2 parm;4=mirror with offset, 2 parm)
- `link_info`: extra input for link, i.e. mirror fleet# or dev index number
- `extra_se`: 0/1 to select extra sd parameter
- `biasadj`: 0/1 for biasadj or not
- `float`: 0/1 to float

```
#-----
Q_options1<-data.frame("fleet"      = 2,
                      "link"       = 1,
                      "link_info"  = 0,
                      "extra_se"   = 0,
                      "biasadj"    = 0,
                      "float"      = 0)

ctl1$Q_options[1,]<-Q_options1
rownames(ctl1$Q_options)[1]<-"SURVEY1"
```

5.3.12.1 Parámetros

```
#-----
Q_parms1<-data.frame( "LO"          = -7,
                     "HI"          = 5,
                     "INIT"        = 0.516018,
                     "PRIOR"       = 0,
                     "PR_SD"       = 1,
                     "PR_type"     = 0,
                     "PHASE"       = 1,
                     "env_var&link" = 0,
                     "dev_link"    = 0,
                     "dev_minyr"   = 0,
                     "dev_maxyr"   = 0,
                     "dev_PH"      = 0,
                     "Block"       = 0,
                     "Block_Fxn"   = 0)

ctl1$Q_parms[1,]<-Q_parms1
rownames(ctl1$Q_parms)[1]<-"LnQ_base_SURVEY1(2)"
#-----

#elimino los parámetros que no se utilizaran en el modelo por el momento
patron_eliminar<-rownames(ctl1$Q_parms)[grep("_SURVEY2",rownames(ctl1$Q_parms))]
ctl1$Q_parms<- subset(ctl1$Q_parms, !rownames(ctl1$Q_parms) %in%
                     c(patron_eliminar,"Q_extraSD_SURVEY1(2)"))
```

5.3.13 Selectividad

Consulte la Guía de usuario de SS3: 8.9 Selectivity and Discard.

[ayuda con parámetros de selectividad](#)

5.3.14 tipos de selectividad a la talla

- 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 6=1 does desc as offset
- Pattern:_21; parm=2+special; non-parm len selex, read as pairs of size, then selex
- 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 version of 24 with 3.30.18 and older
- 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 bin range)
- discard_options:_0=none;_1=define_retention;_2=retention&mortality;_3=all_discarded_dead;_4=define_dome-shaped_retention

```
#-----
size_selex_types1<-data.frame("Pattern" = 0,
                             "Discard" = 0,
                             "Male"    = 0,
                             "Special" = 0)
ctl1$size_selex_types[1,]<-size_selex_types1
rownames(ctl1$size_selex_types)[1]<-"FISHERY"
#-----
size_selex_types2<-data.frame("Pattern" = 0,
                             "Discard" = 0,
                             "Male"    = 0,
                             "Special" = 0)
ctl1$size_selex_types[2,]<-size_selex_types2
rownames(ctl1$size_selex_types)[2]<-"SURVEY1"
#-----
#elimino los parámetros que no se utilizaran en el modelo por el momento
patron_eliminar<-rownames(ctl1$size_selex_types)[grep("SURVEY2",rownames(ctl1$size_selex_types))]
ctl1$size_selex_types<- subset(ctl1$size_selex_types, !rownames(ctl1$size_selex_types) %in%
                             c(patron_eliminar))
```

5.3.15 tipos de selectividad a la edad

- Pattern:_0; parm=0; selex=1.0 for ages 0 to maxage
- Pattern:_10; parm=0; selex=1.0 for ages 1 to maxage
- Pattern:_11; parm=2; selex=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 select
- Pattern:_16; parm=2; Coleraine - Gaussian
- Pattern:_17; parm=nages+1; empirical as random walk N parameters to read can be overridden by setting special to non-zero
- 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 range)
- Age patterns entered with value >100 create Min_selage from first digit and pattern from remainder

```
#-----
age_selex_types1<-data.frame( Pattern = 11,
                             Discard = 0,
                             Male    = 0,
                             Special = 0)
ctl1$age_selex_types[1,]<-age_selex_types1
rownames(ctl1$age_selex_types)[1]<-"FISHERY"
#-----
age_selex_types2<-data.frame( Pattern = 11,
                             Discard = 0,
                             Male    = 0,
                             Special = 0)
ctl1$age_selex_types[2,]<-age_selex_types2
rownames(ctl1$age_selex_types)[2]<-"SURVEY1"
#-----
#elimino los parámetros que no se utilizaran en el modelo por el momento
patron_eliminar<-rownames(ctl1$age_selex_types)[grep("SURVEY2",rownames(ctl1$age_selex_types))]
ctl1$age_selex_types<- subset(ctl1$age_selex_types, !rownames(ctl1$age_selex_types) %in%
                             c(patron_eliminar))

ctl1$age_selex_types
##           Pattern Discard Male Special
## FISHERY      11      0    0      0
## SURVEY1      11      0    0      0
```

5.3.15.1 parametros de selectividad a la talla

```
# #-----
# size_selex_parms1<-data.frame("LO"           = 19.00,
#                               "HI"           = 80,
#                               "INIT"         = 53.64110,
#                               "PRIOR"         = 50,
#                               "PR_SD"         = 0.01,
#                               "PR_type"       = 1,
#                               "PHASE"         = 2,
#                               "env_var$link" = 0,
#                               "dev_link"      = 0,
#                               "dev_minyr"     = 0,
```

```

#           "dev_maxyr"      = 0,
#           "dev_PH"         = 0,
#           "Block"          = 0,
#           "Block_Fxn"      = 0)
# ctl1$size_selex_parms[1,]<-size_selex_parms1
# rownames(ctl1$size_selex_parms)[1]<-"SizeSel_P_1_FISHERY(1)"
# #-----
# size_selex_parms2<-data.frame("LO"          = 0.01,
#           "HI"              = 60,
#           "INIT"            = 18.92320,
#           "PRIOR"           = 15,
#           "PR_SD"           = 0.01,
#           "PR_type"         = 1,
#           "PHASE"           = 3,
#           "env_var&link"    = 0,
#           "dev_link"        = 0,
#           "dev_minyr"       = 0,
#           "dev_maxyr"       = 0,
#           "dev_PH"          = 0,
#           "Block"           = 0,
#           "Block_Fxn"       = 0)
# ctl1$size_selex_parms[2,]<-size_selex_parms2
# rownames(ctl1$size_selex_parms)[2]<-"SizeSel_P_2_FISHERY(1)"
# #-----
# size_selex_parms3<-data.frame("LO"          = 19.00,
#           "HI"              = 70,
#           "INIT"            = 36.65300,
#           "PRIOR"           = 30,
#           "PR_SD"           = 0.01,
#           "PR_type"         = 1,
#           "PHASE"           = 2,
#           "env_var&link"    = 0,
#           "dev_link"        = 0,
#           "dev_minyr"       = 0,
#           "dev_maxyr"       = 0,
#           "dev_PH"          = 0,
#           "Block"           = 0,
#           "Block_Fxn"       = 0)
# ctl1$size_selex_parms[3,]<-size_selex_parms3
# rownames(ctl1$size_selex_parms)[3]<-"SizeSel_P_1_SURVEY1(2)"
# #-----
# size_selex_parms4<-data.frame(row.names="SizeSel_P_2_SURVEY1(2)",
#           "LO"              = 0.01,
#           "HI"              = 60,
#           "INIT"            = 6.59179,
#           "PRIOR"           = 10,
#           "PR_SD"           = 0.01,
#           "PR_type"         = 1,
#           "PHASE"           = 3,
#           "env_var&link"    = 0,
#           "dev_link"        = 0,
#           "dev_minyr"       = 0,
#           "dev_maxyr"       = 0,

```

```

#           "dev_PH"      = 0,
#           "Block"       = 0,
#           "Block_Fxn"   = 0)
#   ctl1$size_selex_parms[4,]<-size_selex_parms4
#   rownames(ctl1$size_selex_parms)[4]<-"SizeSel_P_2_SURVEY1(2)"
#   #-----

#ctl1$size_selex_parms

```

5.3.15.2 parametros de selectividad a la edad

```

#-----
age_selex_parms1<-data.frame("LO"           = 0,
                             "HI"           = 40,
                             "INIT"         = 0,
                             "PRIOR"        = 5,
                             "PR_SD"        = 99,
                             "PR_type"      = 0,
                             "PHASE"        = -99,
                             "env_var&link" = 0,
                             "dev_link"     = 0,
                             "dev_minyr"    = 0,
                             "dev_maxyr"    = 0,
                             "dev_PH"       = 0,
                             "Block"        = 0,
                             "Block_Fxn"    = 0)
ctl1$age_selex_parms[1,]<-age_selex_parms1
rownames(ctl1$age_selex_parms)[1]<-"AgeSel_P_1_FISHERY(1)"
#-----
age_selex_parms2<-data.frame("LO"           = 0,
                             "HI"           = 40,
                             "INIT"         = 40,
                             "PRIOR"        = 6,
                             "PR_SD"        = 99,
                             "PR_type"      = 0,
                             "PHASE"        = -99,
                             "env_var&link" = 0,
                             "dev_link"     = 0,
                             "dev_minyr"    = 0,
                             "dev_maxyr"    = 0,
                             "dev_PH"       = 0,
                             "Block"        = 0,
                             "Block_Fxn"    = 0)
ctl1$age_selex_parms[2,]<-age_selex_parms2
rownames(ctl1$age_selex_parms)[2]<-"AgeSel_P_2_FISHERY(1)"
#-----
age_selex_parms3<-data.frame(row.names="AgeSel_P_1_SURVEY1(2)",
                             "LO"           = 0,
                             "HI"           = 40,
                             "INIT"         = 0,
                             "PRIOR"        = 5,
                             "PR_SD"        = 99,
                             "PR_type"      = 0,

```

```

        "PHASE"          = -99,
        "env_var&link"   = 0,
        "dev_link"       = 0,
        "dev_minyr"      = 0,
        "dev_maxyr"      = 0,
        "dev_PH"         = 0,
        "Block"          = 0,
        "Block_Fxn"      = 0)
ctl1$age_selex_parms[12,]<-age_selex_parms3
rownames(ctl1$age_selex_parms)[12]<-"AgeSel_P_1_SURVEY1(2)"
#-----
age_selex_parms4<-data.frame(row.names="AgeSel_P_2_SURVEY1(2)",
        "LO"            = 0,
        "HI"            = 40,
        "INIT"          = 40,
        "PRIOR"         = 6,
        "PR_SD"         = 99,
        "PR_type"       = 0,
        "PHASE"         = -99,
        "env_var&link"   = 0,
        "dev_link"       = 0,
        "dev_minyr"      = 0,
        "dev_maxyr"      = 0,
        "dev_PH"         = 0,
        "Block"          = 0,
        "Block_Fxn"      = 0)
ctl1$age_selex_parms[13,]<-age_selex_parms4
rownames(ctl1$age_selex_parms)[13]<-"AgeSel_P_2_SURVEY1(2)"
#-----
#elimino los parámetros que no se utilizaran en el modelo por el momento
patron_eliminar<-rownames(ctl1$age_selex_parms)[c(3:11,14:24)]
ctl1$age_selex_parms<- subset(ctl1$age_selex_parms, !rownames(ctl1$age_selex_parms) %in%
        c(patron_eliminar))

ctl1$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
##              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

```

5.3.16 otros parámetros

- Use_2D_AR1_selectivity: use 2D_AR1 selectivity(0/1)
- TG_custom: 0=no read and autogen if tag data exist; 1=read
- DoVar_adjust: 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
- Variance_adjustment_list:
- maxlambdaphase:
- sd_offset: must be 1 if any growthCV, sigmaR, or survey extraSD is an estimated parameter
- maxlambdaphase:
- sd_offset:
- lambdas:
- N_lambdas:
- more_stddev_reporting:
- stddev_reporting_specs:
- stddev_reporting_selex:
- stddev_reporting_growth:
- stddev_reporting_N_at_A:

```
ctl1$Use_2D_AR1_selectivity <-0
ctl1$TG_custom <- 0
ctl1$DoVar_adjust <- 0
ctl1$maxlambdaphase <- 1
ctl1$sd_offset <- 1
ctl1$lambdas <-NULL
ctl1$N_lambdas <- 0
ctl1$more_stddev_reporting <- 0
#-----
ctl1$stddev_reporting_specs <-""
ctl1$stddev_reporting_selex <-""
ctl1$stddev_reporting_N_at_A <-""
```

5.3.17 Escribir archivo de control modificado con la función SS_write para el enfoque de modelación SS3

```
#-----
r4ss::SS_writectl(ctl1,
                  outfile=here(dirname.simple_mod,"control.ss"),
                  overwrite = TRUE)
#-----
```

5.4 Archivo starter.ss

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

```
#-----
start <- r4ss::SS_readstarter(here(dirname.base,"starter.ss"),
                             verbose = FALSE) #base
#-----

start1<-start # para modificar
#names(start1) # muestra los objetos de la lista

start1$datfile          <-"data.ss"
start1$ctlfile          <-"control.ss"
start1$init_values_src  <-0
start1$run_display_detail <-0
start1$detailed_age_structure<-1
start1$checkup          <-0
start1$parmtrace        <-0
start1$cumreport        <-0
start1$prior_like       <-1
start1$soft_bounds      <-1
start1$N_bootstraps     <-1
start1$last_estimation_phase<-10
start1$MCMCburn         <-0
start1$MCMCthin         <-1
start1$jitter_fraction <-0
start1$minyr_sdreport   <-2000
start1$maxyr_sdreport   <-2023
start1$N_STD_yrs        <-0
start1$converge_criterion <-1e-04
start1$retro_yr         <-0
start1$min_age_summary_bio<-1
start1$depl_basis       <-2
start1$depl_denom_frac  <-1
start1$SPR_basis        <-4
start1$F_report_units   <-3
start1$F_age_range      <-"#"
start1$F_report_basis   <-0
start1$MCMC_output_detail <-0
start1$ALK_tolerance    <-0
start1$seed             <- -1
start1$final            <-3.3
```

5.4.1 Escribir archivo de starter modificado con la función SS_write para el enfoque de modelación SS3

```
r4ss::SS_writestarter(mylist = start1,
                      dir = here(dirname.simple_mod),
                      file = "starter.ss",
                      overwrite = TRUE,
                      verbose = TRUE)
```

5.5 Archivo forecast.ss

Revisamos los nombres de los componentes de la lista del archivo forecast

```
fore <- r4ss::SS_readforecast(here(dirname.base,"forecast.ss"),
                             verbose = FALSE) #base
fore1<-fore # para modificar
names(fore1) # muestra los objetos de la lista
## [1] "warnings"
## [2] "SSversion"
## [3] "sourcefile"
## [4] "type"
## [5] "benchmarks"
## [6] "MSY"
## [7] "SPRtarget"
## [8] "Btarget"
## [9] "Bmark_years"
## [10] "Bmark_relF_Basis"
## [11] "Forecast"
## [12] "Nforecasttyrs"
## [13] "F_scalar"
## [14] "Fcast_years"
## [15] "Fcast_selex"
## [16] "ControlRuleMethod"
## [17] "BforconstantF"
## [18] "BfornoF"
## [19] "Flimitfraction"
## [20] "N_forecast_loops"
## [21] "First_forecast_loop_with_stochastic_recruitment"
## [22] "fcast_rec_option"
## [23] "fcast_rec_val"
## [24] "Forecast_loop_control_5"
## [25] "FirstYear_for_caps_and_allocations"
## [26] "stddev_of_log_catch_ratio"
## [27] "Do_West_Coast_gfish_rebuilder_output"
## [28] "Ydecl"
## [29] "Yinit"
## [30] "fleet_relative_F"
## [31] "basis_for_fcast_catch_tuning"
## [32] "N_allocation_groups"
## [33] "InputBasis"
## [34] "eof"

fore1$benchmarks <- 1
fore1$MSY <- 2
fore1$SPRtarget <- 0.4
fore1$Btarget <- 0.4
#-----
Bmark_years1<-data.frame(matrix(rep(0,10),nrow=1,ncol=10))
colnames(Bmark_years1)<-paste("#_Bmark_years_",seq(1,10,1),sep="")
fore1$Bmark_years <- Bmark_years1
#-----
fore1$Bmark_relF_Basis <- 1
fore1$Forecast <- 1
fore1$Nforecasttyrs <- 10
```

```

fore1$F_scalar <- 0.2
#-----
Fcast_years1<-data.frame(matrix(c(-5,0,-5,0,-999,0),nrow=1,ncol=6))
colnames(Fcast_years1)<-paste("#_Fcast_years_",seq(1,6,1),sep="")
fore1$Fcast_years <-Fcast_years1
#-----
fore1$Fcast_selex <- 0
fore1$ControlRuleMethod <- 1
fore1$BforconstantF <- 0.4
fore1$BfornoF <- 0.1
fore1$Flimitfraction <- 0.75
fore1$N_forecast_loops <- 3
fore1$First_forecast_loop_with_stochastic_recruitment <- 3
fore1$fcast_rec_option <- 1
fore1$fcast_rec_val <- 1
fore1$Forecast_loop_control_5 <- 0
fore1$FirstYear_for_caps_and_allocations <- 2020
fore1$stddev_of_log_catch_ratio <- 0
fore1$Do_West_Coast_gfish_rebuilder_output <- 0
fore1$Ydecl <- 2000
fore1$Yinit <- 2021
fore1$fleet_relative_F <- 1
fore1$basis_for_fcast_catch_tuning <- 2
fore1$N_allocation_groups <- 0
fore1$InputBasis <- 2
fore1$eof <- TRUE

```

5.5.1 Escribir archivo de forecast modificado con la función SS_write para el enfoque de modelación SS3

```

r4ss::SS_writeforecast(mylist = fore1,
                        dir = here(dirname.simple_mod),
                        file = "forecast.ss",
                        overwrite = TRUE,
                        verbose = TRUE)

```


5.6 Archivo watatege

5.6.1 Formato a4a

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

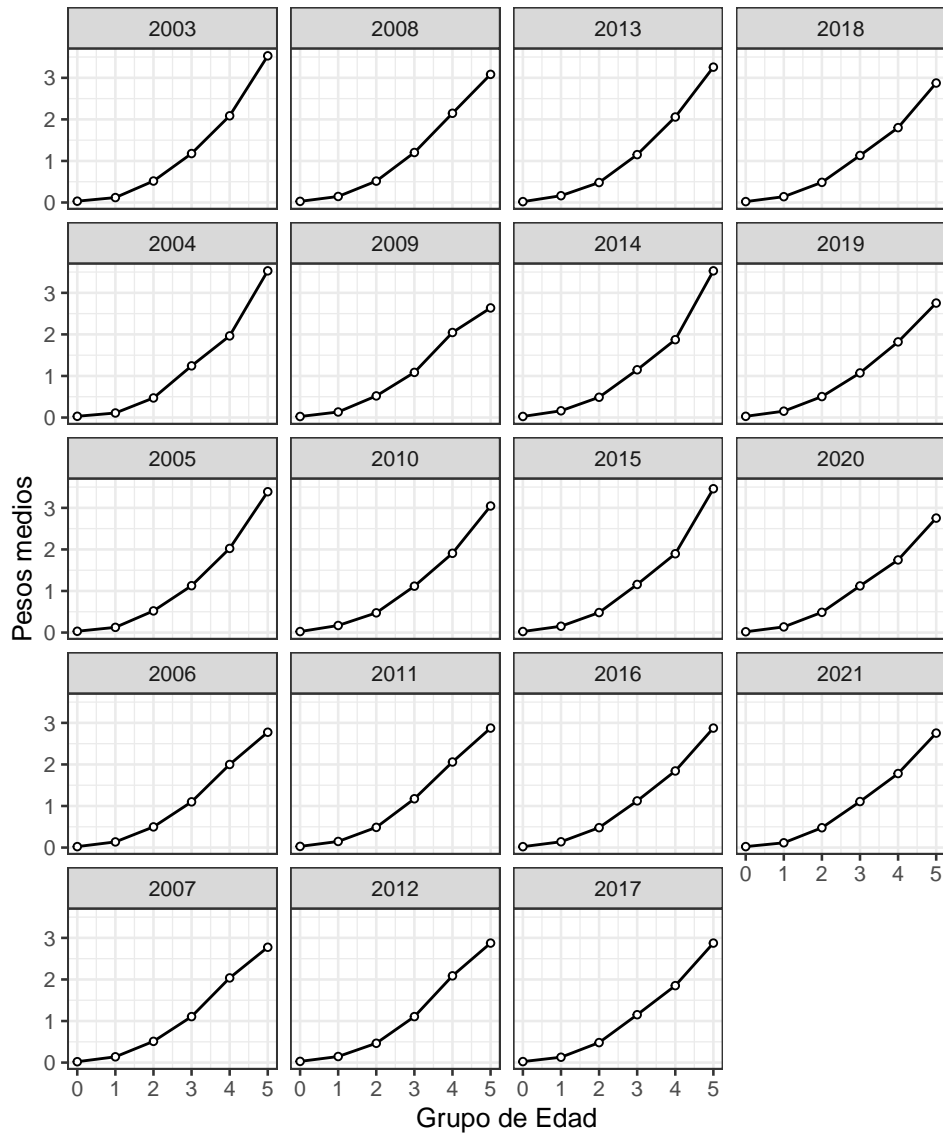
```
dir(dir_GSA1)[3]
## [1] "CATWT.DAT"
CATWT.DAT <- read.table(paste(dir_GSA1,dir(dir_GSA1)[3],sep="/"),
                        sep=" ",na="NA",fill=T,skip = 5)
CATWT.DAT
##      V1      V2      V3      V4      V5      V6
## 1 0.032 0.119 0.517 1.178 2.086 3.529
## 2 0.030 0.109 0.471 1.243 1.964 3.529
## 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
```

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

```
age<-seq(0,5,1)
years<-2003:2021
nyears<-length(years)
#dir(dir_GSA1)[11]
STOCWT.DAT <- read.table(paste(dir_GSA1,dir(dir_GSA1)[11],sep="/"),
                        sep=" ",na="NA",fill=T,skip = 5)
#STOCWT.DAT

stockWt<-as.data.frame(STOCWT.DAT) %>%
  mutate(Yr=years) %>%
  melt(id.vars="Yr") %>%
  mutate(edad=rep(age,each=nyears)) %>%
  mutate(type='Wt')

ggplot(stockWt, aes(edad,value))+
  geom_line()+
  geom_point(size=1,shape=21,fill="white")+
  facet_wrap(vars(Yr),dir='v',as.table=TRUE)+
  labs(x="Grupo de Edad", y="Pesos medios") +
  theme(panel.background = element_rect(fill ="gray95")) +
  theme_bw(base_size=10)
```

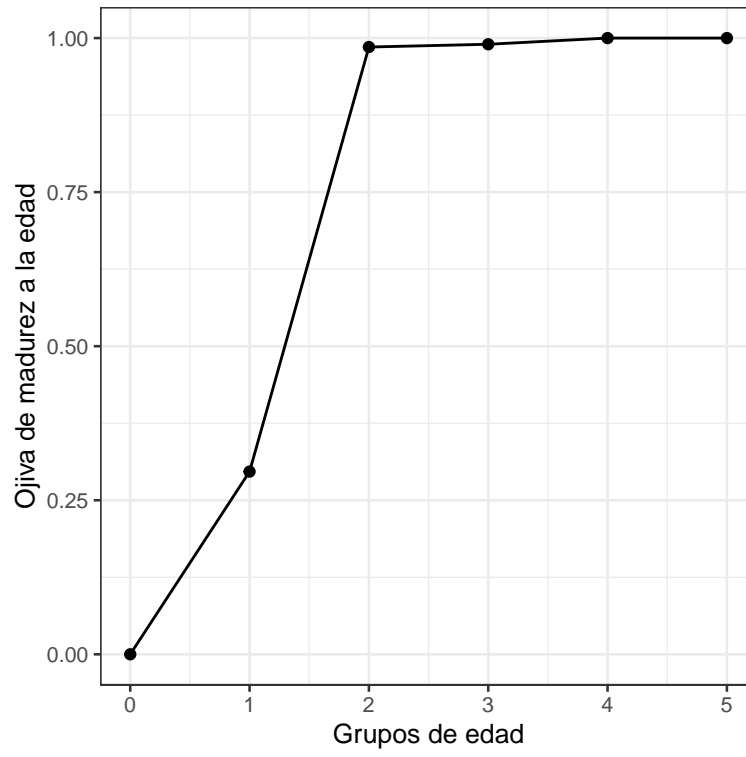


5.6.1.3 Ojiva de madurez por edad.

```
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)

ojivamadurez<-data.frame(edad=seq(0,5,1),
                          prop=as.numeric(PROPMAT.DAT[1,]))

ggplot(ojivamadurez,aes(edad,prop))+
  geom_line()+
  geom_point()+
  labs(x="Grupos de edad", y="Ojiva de madurez a la edad") +
  theme(panel.background = element_rect(fill ="gray95")) +
  theme_bw(base_size=10)
```



5.6.2 Formato SS3

```
wtat <- r4ss::SS_read(here("empirical_wtatage_and_age_selex"),
                     verbose = FALSE) #base
wtat1<-wtat$wtatage # para modificar
#names(wtat1) # muestra los objetos de la lista

maturity1<-as.numeric(PROPMAT.DAT[1,])
data_wt<-as.data.frame(CATWT.DAT)

wt_flt<-data.frame(do.call(rbind,replicate(2,data_wt,simplify = FALSE))) %>% mutate(type=c(rep("#wt_flt",
colnames(wt_flt)<-c("E0", "E1", "E2", "E3", "E4", "E5", "#type")

fecundity<-data.frame(do.call(rbind,replicate(nyears,maturity1,simplify = FALSE))) %>% mutate(type=c(r
colnames(fecundity)<-c("E0", "E1", "E2", "E3", "E4", "E5", "#type")

popwt<-data.frame(do.call(rbind,replicate(2,data_wt,simplify = FALSE))) %>% mutate(type=c(rep("#popwt_b
colnames(popwt)<-c("E0", "E1", "E2", "E3", "E4", "E5", "#type")

wtatege_f1<-data.frame(Yr=c(rep(years,5)),
                      Seas=1,
                      Sex=1,
                      Bio_Pattern=1,
                      BirthSeas=1,
                      Fleet=c(rep(1,nyears),rep(2,nyears),
                              rep(-2,nyears),
                              rep(0,nyears),rep(-1,nyears)))

wtatege_f1<-cbind(wtatege_f1,rbind(wt_flt,fecundity,popwt))

wtat1<-wtatege_f1[order(wtatege_f1$Yr),]
```

5.6.3 Escribir archivo de wtatage modificado con la función SS_write para el enfoque de modelación SS3

```
r4ss::SS_writewtatage(mylist = wtat1,
                     dir = here(dirname.simple_mod),
                     file = "wtatage.ss",
                     overwrite = TRUE,
                     verbose = TRUE)

## 6 # maxage
## # if Yr is negative, then fill remaining years for that Seas, growpattern, Bio_Pattern, Fleet
## # if season is negative, then fill remaining fleets for that Seas, Bio_Pattern, Sex, Fleet
## # will fill through forecast years, so be careful
## # fleet 0 contains begin season pop WT
## # fleet -1 contains mid season pop WT
## # fleet -2 contains maturity*fecundity
##   #Yr Seas Sex Bio_Pattern BirthSeas Fleet   E0   E1   E2   E3   E4   E5   #type
##   2003   1   1         1         1   -2 0.000 0.2965 0.9855 0.990 1.000 1.000 #fecundity
##   2003   1   1         1         1   -1 0.032 0.1190 0.5170 1.178 2.086 3.529 #popwt_mid
##   2003   1   1         1         1    0 0.032 0.1190 0.5170 1.178 2.086 3.529 #popwt_beg
##   2003   1   1         1         1    1 0.032 0.1190 0.5170 1.178 2.086 3.529 #wt_flt_1
```

##	2003	1	1	1	1	2	0.032	0.1190	0.5170	1.178	2.086	3.529	#wt_flt_2
##	2004	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity
##	2004	1	1	1	1	-1	0.030	0.1090	0.4710	1.243	1.964	3.529	#popwt_mid
##	2004	1	1	1	1	0	0.030	0.1090	0.4710	1.243	1.964	3.529	#popwt_beg
##	2004	1	1	1	1	1	0.030	0.1090	0.4710	1.243	1.964	3.529	#wt_flt_1
##	2004	1	1	1	1	2	0.030	0.1090	0.4710	1.243	1.964	3.529	#wt_flt_2
##	2005	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity
##	2005	1	1	1	1	-1	0.031	0.1250	0.5210	1.127	2.024	3.388	#popwt_mid
##	2005	1	1	1	1	0	0.031	0.1250	0.5210	1.127	2.024	3.388	#popwt_beg
##	2005	1	1	1	1	1	0.031	0.1250	0.5210	1.127	2.024	3.388	#wt_flt_1
##	2005	1	1	1	1	2	0.031	0.1250	0.5210	1.127	2.024	3.388	#wt_flt_2
##	2006	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity
##	2006	1	1	1	1	-1	0.023	0.1360	0.4970	1.100	1.999	2.775	#popwt_mid
##	2006	1	1	1	1	0	0.023	0.1360	0.4970	1.100	1.999	2.775	#popwt_beg
##	2006	1	1	1	1	1	0.023	0.1360	0.4970	1.100	1.999	2.775	#wt_flt_1
##	2006	1	1	1	1	2	0.023	0.1360	0.4970	1.100	1.999	2.775	#wt_flt_2
##	2007	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity
##	2007	1	1	1	1	-1	0.021	0.1390	0.5110	1.106	2.037	2.775	#popwt_mid
##	2007	1	1	1	1	0	0.021	0.1390	0.5110	1.106	2.037	2.775	#popwt_beg
##	2007	1	1	1	1	1	0.021	0.1390	0.5110	1.106	2.037	2.775	#wt_flt_1
##	2007	1	1	1	1	2	0.021	0.1390	0.5110	1.106	2.037	2.775	#wt_flt_2
##	2008	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity
##	2008	1	1	1	1	-1	0.028	0.1450	0.5150	1.203	2.147	3.083	#popwt_mid
##	2008	1	1	1	1	0	0.028	0.1450	0.5150	1.203	2.147	3.083	#popwt_beg
##	2008	1	1	1	1	1	0.028	0.1450	0.5150	1.203	2.147	3.083	#wt_flt_1
##	2008	1	1	1	1	2	0.028	0.1450	0.5150	1.203	2.147	3.083	#wt_flt_2
##	2009	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity
##	2009	1	1	1	1	-1	0.025	0.1330	0.5200	1.086	2.046	2.637	#popwt_mid
##	2009	1	1	1	1	0	0.025	0.1330	0.5200	1.086	2.046	2.637	#popwt_beg
##	2009	1	1	1	1	1	0.025	0.1330	0.5200	1.086	2.046	2.637	#wt_flt_1
##	2009	1	1	1	1	2	0.025	0.1330	0.5200	1.086	2.046	2.637	#wt_flt_2
##	2010	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity
##	2010	1	1	1	1	-1	0.024	0.1700	0.4740	1.116	1.907	3.045	#popwt_mid
##	2010	1	1	1	1	0	0.024	0.1700	0.4740	1.116	1.907	3.045	#popwt_beg
##	2010	1	1	1	1	1	0.024	0.1700	0.4740	1.116	1.907	3.045	#wt_flt_1
##	2010	1	1	1	1	2	0.024	0.1700	0.4740	1.116	1.907	3.045	#wt_flt_2
##	2011	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity
##	2011	1	1	1	1	-1	0.028	0.1470	0.4860	1.174	2.059	2.875	#popwt_mid
##	2011	1	1	1	1	0	0.028	0.1470	0.4860	1.174	2.059	2.875	#popwt_beg
##	2011	1	1	1	1	1	0.028	0.1470	0.4860	1.174	2.059	2.875	#wt_flt_1
##	2011	1	1	1	1	2	0.028	0.1470	0.4860	1.174	2.059	2.875	#wt_flt_2
##	2012	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity
##	2012	1	1	1	1	-1	0.029	0.1460	0.4650	1.106	2.087	2.875	#popwt_mid
##	2012	1	1	1	1	0	0.029	0.1460	0.4650	1.106	2.087	2.875	#popwt_beg
##	2012	1	1	1	1	1	0.029	0.1460	0.4650	1.106	2.087	2.875	#wt_flt_1
##	2012	1	1	1	1	2	0.029	0.1460	0.4650	1.106	2.087	2.875	#wt_flt_2
##	2013	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity
##	2013	1	1	1	1	-1	0.021	0.1640	0.4810	1.152	2.057	3.257	#popwt_mid
##	2013	1	1	1	1	0	0.021	0.1640	0.4810	1.152	2.057	3.257	#popwt_beg
##	2013	1	1	1	1	1	0.021	0.1640	0.4810	1.152	2.057	3.257	#wt_flt_1
##	2013	1	1	1	1	2	0.021	0.1640	0.4810	1.152	2.057	3.257	#wt_flt_2
##	2014	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity
##	2014	1	1	1	1	-1	0.026	0.1620	0.4880	1.149	1.873	3.529	#popwt_mid

##	2014	1	1	1	1	0	0.026	0.1620	0.4880	1.149	1.873	3.529	#popwt_beg
##	2014	1	1	1	1	1	0.026	0.1620	0.4880	1.149	1.873	3.529	#wt_flt_1
##	2014	1	1	1	1	2	0.026	0.1620	0.4880	1.149	1.873	3.529	#wt_flt_2
##	2015	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity
##	2015	1	1	1	1	-1	0.025	0.1520	0.4810	1.157	1.896	3.460	#popwt_mid
##	2015	1	1	1	1	0	0.025	0.1520	0.4810	1.157	1.896	3.460	#popwt_beg
##	2015	1	1	1	1	1	0.025	0.1520	0.4810	1.157	1.896	3.460	#wt_flt_1
##	2015	1	1	1	1	2	0.025	0.1520	0.4810	1.157	1.896	3.460	#wt_flt_2
##	2016	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity
##	2016	1	1	1	1	-1	0.022	0.1390	0.4790	1.123	1.843	2.875	#popwt_mid
##	2016	1	1	1	1	0	0.022	0.1390	0.4790	1.123	1.843	2.875	#popwt_beg
##	2016	1	1	1	1	1	0.022	0.1390	0.4790	1.123	1.843	2.875	#wt_flt_1
##	2016	1	1	1	1	2	0.022	0.1390	0.4790	1.123	1.843	2.875	#wt_flt_2
##	2017	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity
##	2017	1	1	1	1	-1	0.024	0.1290	0.4810	1.152	1.851	2.875	#popwt_mid
##	2017	1	1	1	1	0	0.024	0.1290	0.4810	1.152	1.851	2.875	#popwt_beg
##	2017	1	1	1	1	1	0.024	0.1290	0.4810	1.152	1.851	2.875	#wt_flt_1
##	2017	1	1	1	1	2	0.024	0.1290	0.4810	1.152	1.851	2.875	#wt_flt_2
##	2018	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity
##	2018	1	1	1	1	-1	0.022	0.1400	0.4850	1.132	1.801	2.875	#popwt_mid
##	2018	1	1	1	1	0	0.022	0.1400	0.4850	1.132	1.801	2.875	#popwt_beg
##	2018	1	1	1	1	1	0.022	0.1400	0.4850	1.132	1.801	2.875	#wt_flt_1
##	2018	1	1	1	1	2	0.022	0.1400	0.4850	1.132	1.801	2.875	#wt_flt_2
##	2019	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity
##	2019	1	1	1	1	-1	0.029	0.1530	0.5040	1.073	1.820	2.754	#popwt_mid
##	2019	1	1	1	1	0	0.029	0.1530	0.5040	1.073	1.820	2.754	#popwt_beg
##	2019	1	1	1	1	1	0.029	0.1530	0.5040	1.073	1.820	2.754	#wt_flt_1
##	2019	1	1	1	1	2	0.029	0.1530	0.5040	1.073	1.820	2.754	#wt_flt_2
##	2020	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity
##	2020	1	1	1	1	-1	0.019	0.1360	0.4870	1.122	1.747	2.754	#popwt_mid
##	2020	1	1	1	1	0	0.019	0.1360	0.4870	1.122	1.747	2.754	#popwt_beg
##	2020	1	1	1	1	1	0.019	0.1360	0.4870	1.122	1.747	2.754	#wt_flt_1
##	2020	1	1	1	1	2	0.019	0.1360	0.4870	1.122	1.747	2.754	#wt_flt_2
##	2021	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity
##	2021	1	1	1	1	-1	0.023	0.1150	0.4770	1.107	1.781	2.754	#popwt_mid
##	2021	1	1	1	1	0	0.023	0.1150	0.4770	1.107	1.781	2.754	#popwt_beg
##	2021	1	1	1	1	1	0.023	0.1150	0.4770	1.107	1.781	2.754	#wt_flt_1
##	2021	1	1	1	1	2	0.023	0.1150	0.4770	1.107	1.781	2.754	#wt_flt_2
##	-9999	1	1	1	1	-2	0.000	0.2965	0.9855	0.990	1.000	1.000	#fecundity

6 comprobar si el modelo corre al modificar este archivo

```
exe_path <- here("Ejecutables_SS3", "3.30.18_release")
ss_exe_mac <- paste(exe_path, "ss_osx", sep= "/")

r4ss::run(
  dir = dirname.simple_mod,
  exe = ss_exe_mac,
  extras = "",
  skipfinished = FALSE,
  show_in_console = TRUE,
  verbose = TRUE
)
```

)

6.0.0.1 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
```