

Transición de archivos de datos GADGET a SS3 de boqueron Cádiz

abril, 18, 2023

Contents

1	Descripción del repositorio	2
2	Descargar archivos requeridos desde repositorio	2
3	Librerías requeridas	2
4	Descripción del Formado de entrada de datos requerido para cada enfoque de modelación	2
4.1	Identificación de los archivos de datos GADGET	2
4.1.1	Capturas en formato Gadget	2
4.1.2	Indices de abundancia formato Gadget	3
4.1.3	Composiciones de tallas Formato Gadget	4
4.1.4	Clave talla-edad formato Gadget	8
4.2	Identificación de los archivos de datos SS3	14
4.2.1	Archivos utilizado para enfoque de modelación SS3	14
4.2.2	Información general del modelo Formato SS3	14
4.2.3	Capturas en formato SS3	14
4.2.4	Indices de abundancia formato SS3	16
4.2.5	Descartes y tallas medias Formato SS3	17
4.2.6	Composición de tallas Formato SS3	18
4.2.7	Composición de edad Formato SS3	20
4.2.8	Otros datos	22
4.3	Escribir archivo de datos modificado con la función SS_write para el enfoque de modelación SS3	23

1 Descripción del repositorio

- Directorio con archivos requeridos para ejecutar GADGET
- Directorio con archivos requeridos para ejecutar SS3
- Directorio con ejecutable SS3 para tres sistemas operativos (windows, linux y mac)
- Códigos Rmarkdown (pdf o html) que permita modificar archivos SS3
 - formato data.ss
 - formato contro.ss
 - formato starter.ss
 - formato forecast.ss

2 Descargar archivos requeridos desde repositorio

Tarea pendiente. . .

3 Librerías requeridas

4 Descripción del Formado de entrada de datos requerido para cada enfoque de modelación

4.1 Identificación de los archivos de datos GADGET

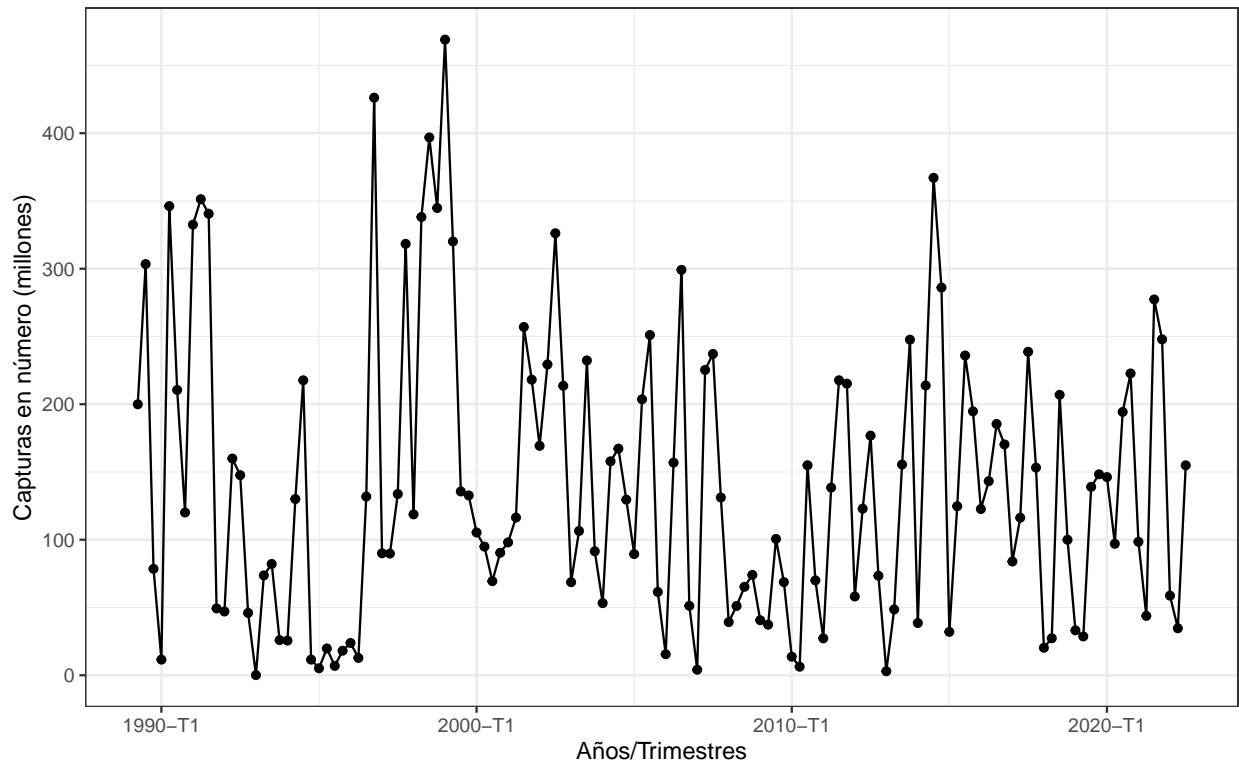
Se utilizarán los siguientes archivos de entrada de datos de Gadget:

```
dir(here('Data_Gadget'))  
## [1] "bounds.penaltyfile"  
## [2] "catchdistribution.aldist.ecocadiz.sumofsquares"  
## [3] "catchdistribution.aldist.pelago.sumofsquares"  
## [4] "catchdistribution.ldist.alkseine.sumofsquares"  
## [5] "catchdistribution.ldist.ecocadiz.noage.sumofsquares"  
## [6] "catchdistribution.ldist.pelago.noage.sumofsquares"  
## [7] "catchdistribution.ldist.seine.sumofsquares"  
## [8] "clavetalla-edad-Flota.csv"  
## [9] "fleet.ECO.data"  
## [10] "fleet.PEL.data"  
## [11] "fleet.seine.data"  
## [12] "surveyindices.ecocadiz.survey.lengths"  
## [13] "surveyindices.pelagonumber.survey.lengths"
```

4.1.1 Capturas en formato Gadget

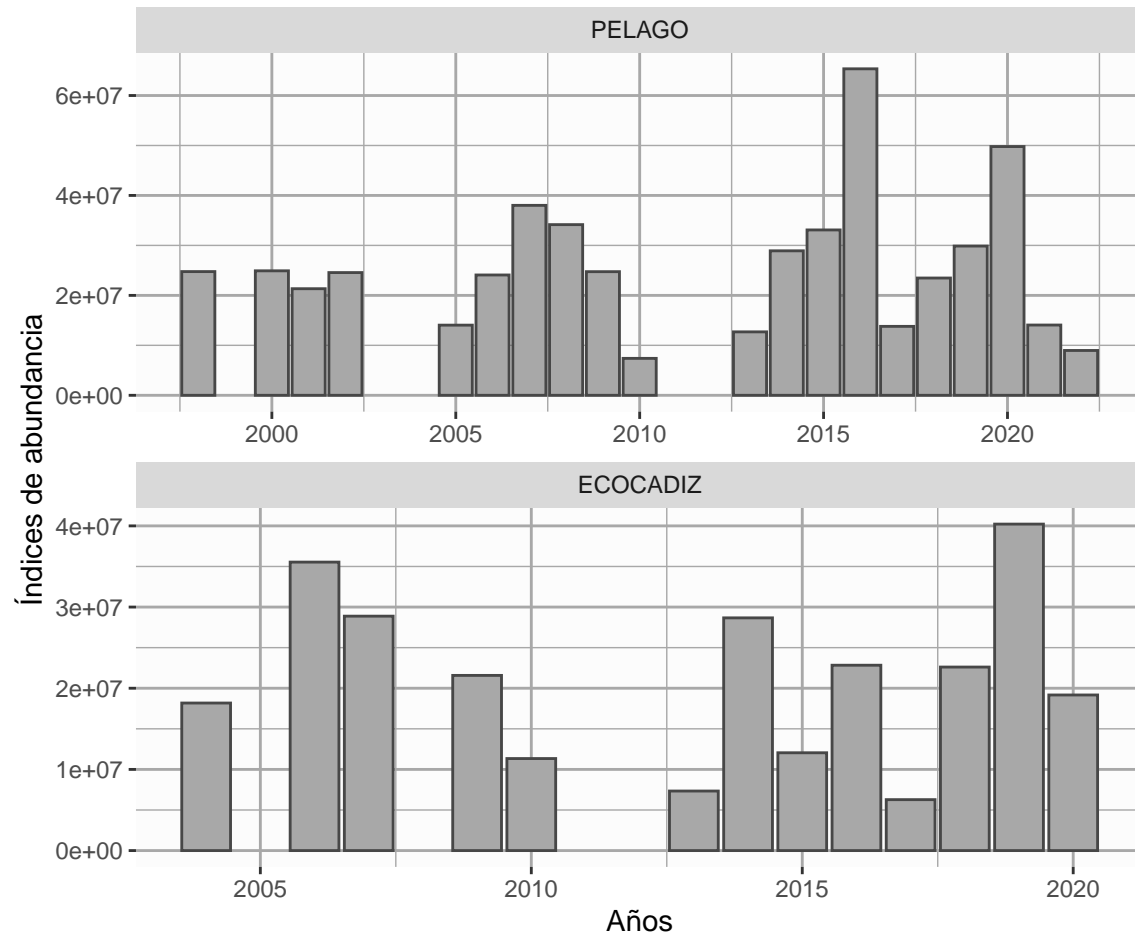
Capturas anuales del stock (toneladas)

```
# Se lee archivo de entrada Gadget  
capturas<-read.table(file=here('Data_Gadget',"fleet.seine.data"),  
                     header=F,sep=" ",na='NA',fill=T,skip = 3) %>%  
  magrittr::set_colnames(c('year','step','area','vessel','number'))  
#capturas
```



4.1.2 Índices de abundancia formato Gadget

```
#-----
pelago<-read.table(file=here('Data_Gadget',"surveyindices.pelagonumber.survey.lengths"),
  header=F,sep="",na='NA',fill=T,skip = 3) %>%
  magrittr::set_colnames(c('year','step','area','length','number'))
#pelago
#-----
ecocadiz<-read.table(file=here('Data_Gadget',"surveyindices.ecocadiz.survey.lengths"),
  header=F,sep="",na='NA',fill=T,skip = 3)%>%
  magrittr::set_colnames(c('year','step','area','length','number'))
#ecocadiz
#-----
```



4.1.3 Composiciones de tallas Formato Gadget

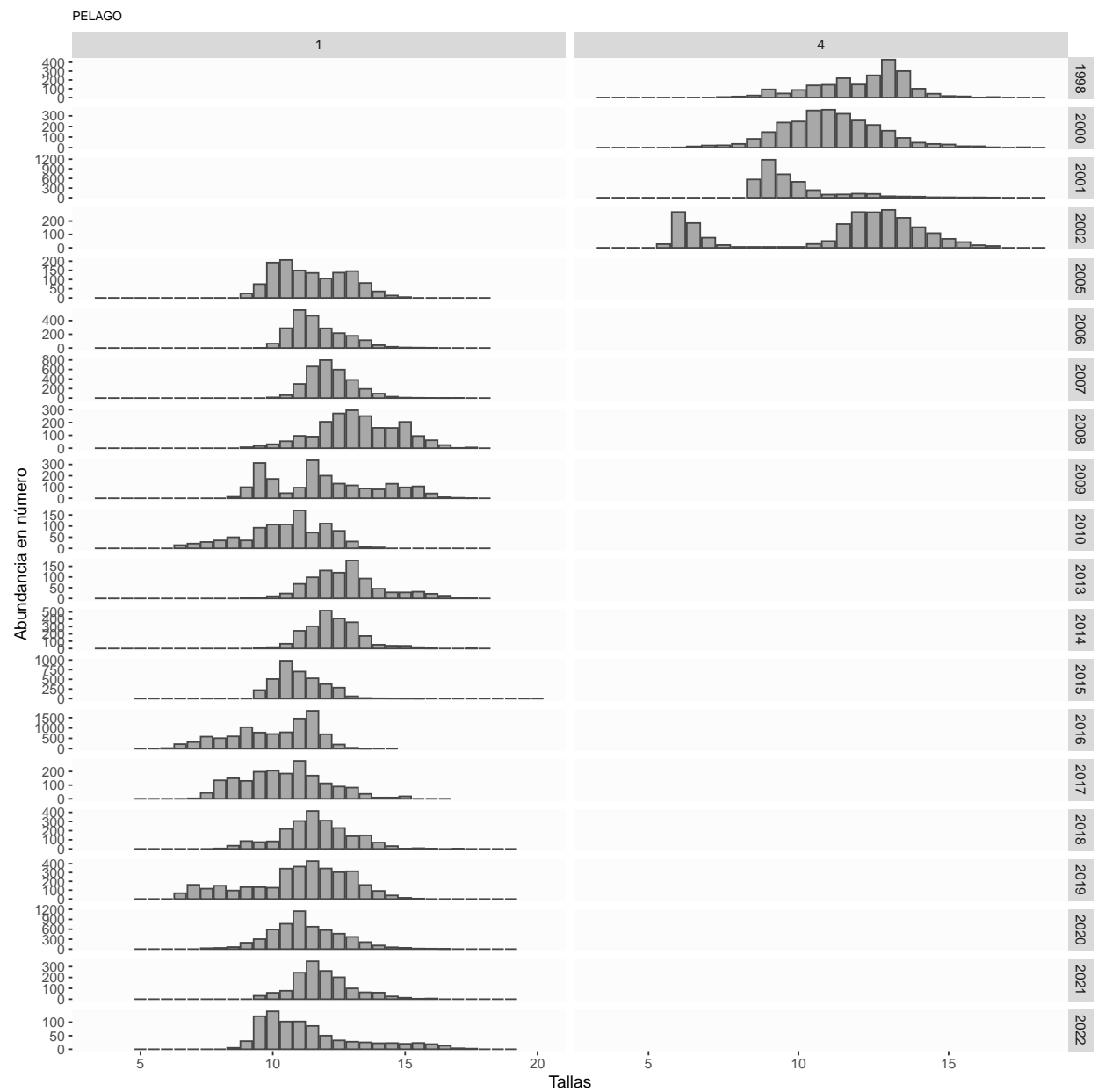
```
#-----
tallaspelago<-read.table(file=here('Data_Gadget',"catchdistribution.ldist.pelago.noage.sumofsquares"),
                        header=F,sep=" ",na='NA',fill=T,skip = 3) %>%
  magrittr::set_colnames(c('year','step','area','age','length','number')) %>%
  separate(length,into=c("text","length"),sep=3,convert =TRUE)

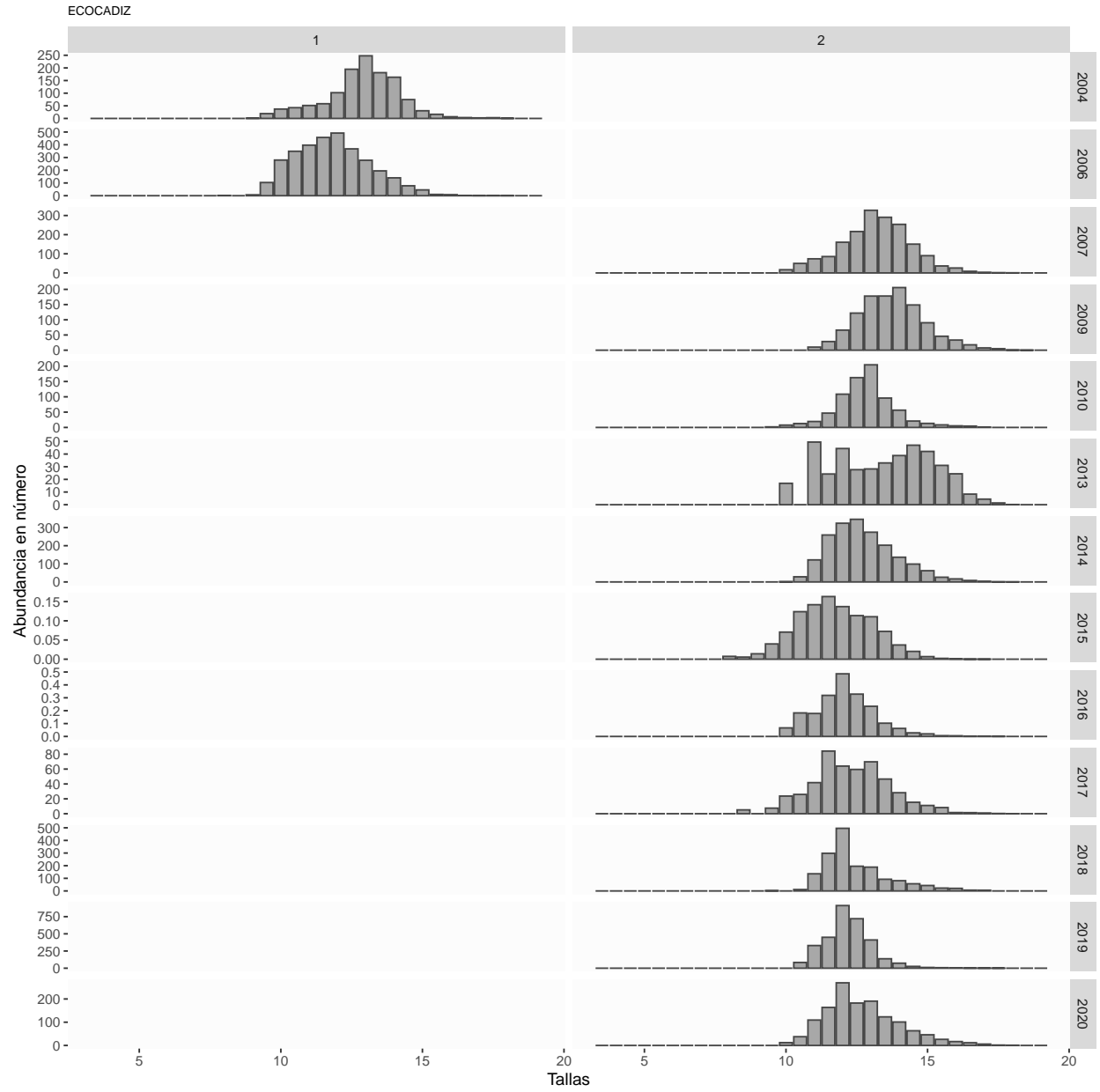
#tallaspelago
#-----
tallasecocadiz<-read.table(file=here('Data_Gadget',"catchdistribution.ldist.ecocadiz.noage.sumofsquares"),
                          header=F,sep=" ",na='NA',fill=T,skip = 3) %>%
  magrittr::set_colnames(c('year','step','area','age','length','number')) %>%
  separate(length,into=c("text","length"),sep=3,convert =TRUE)

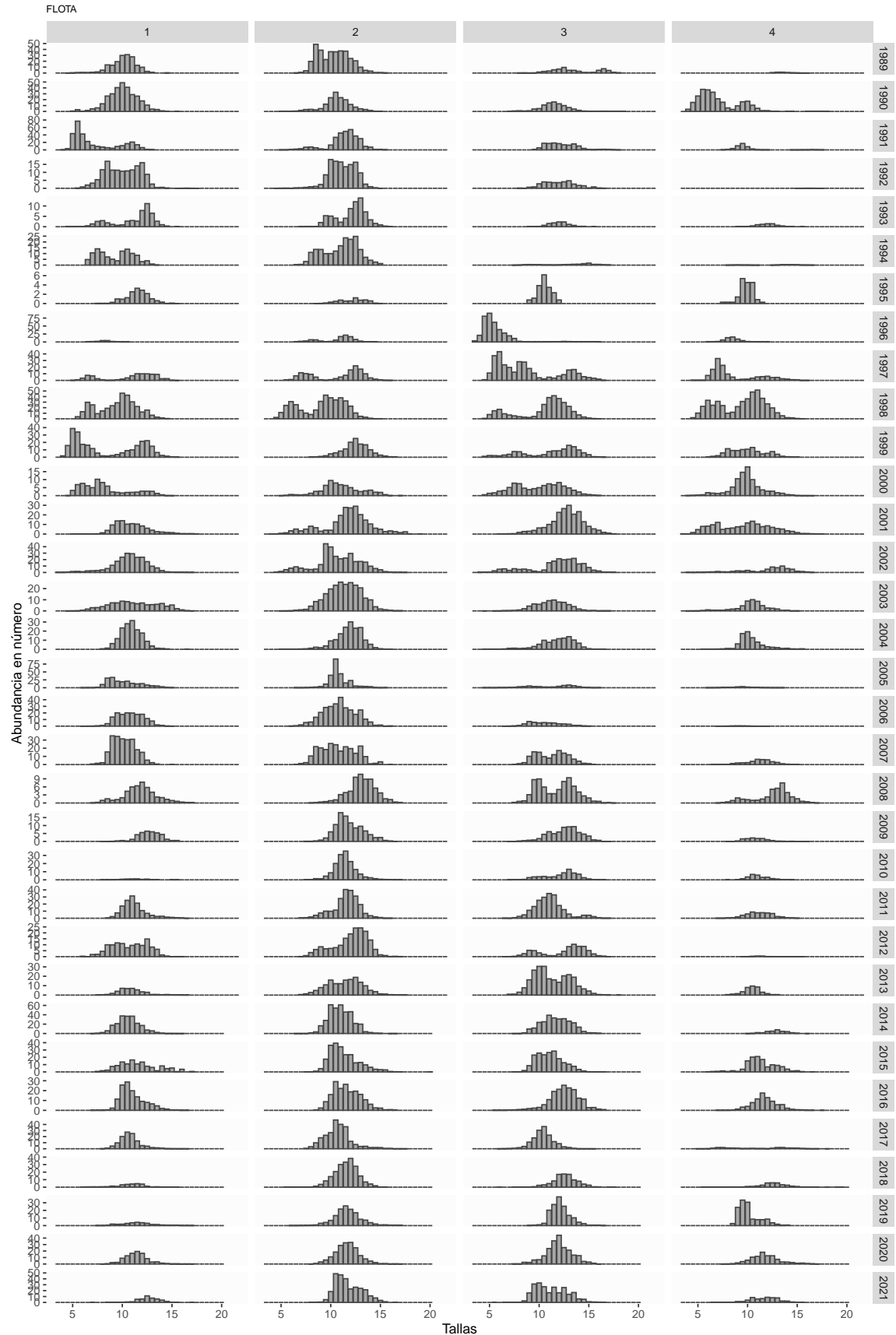
#tallasecocadiz
#-----
tallasflota<-read.table(file=here('Data_Gadget',"catchdistribution.ldist.seine.sumofsquares"),
                       header=F,sep=" ",na='NA',fill=T,skip = 3) %>%
  magrittr::set_colnames(c('year','step','area','age','length','number')) %>%
  separate(length,into=c("text","length"),sep=3,convert =TRUE)
```

#tallasflota

#-----







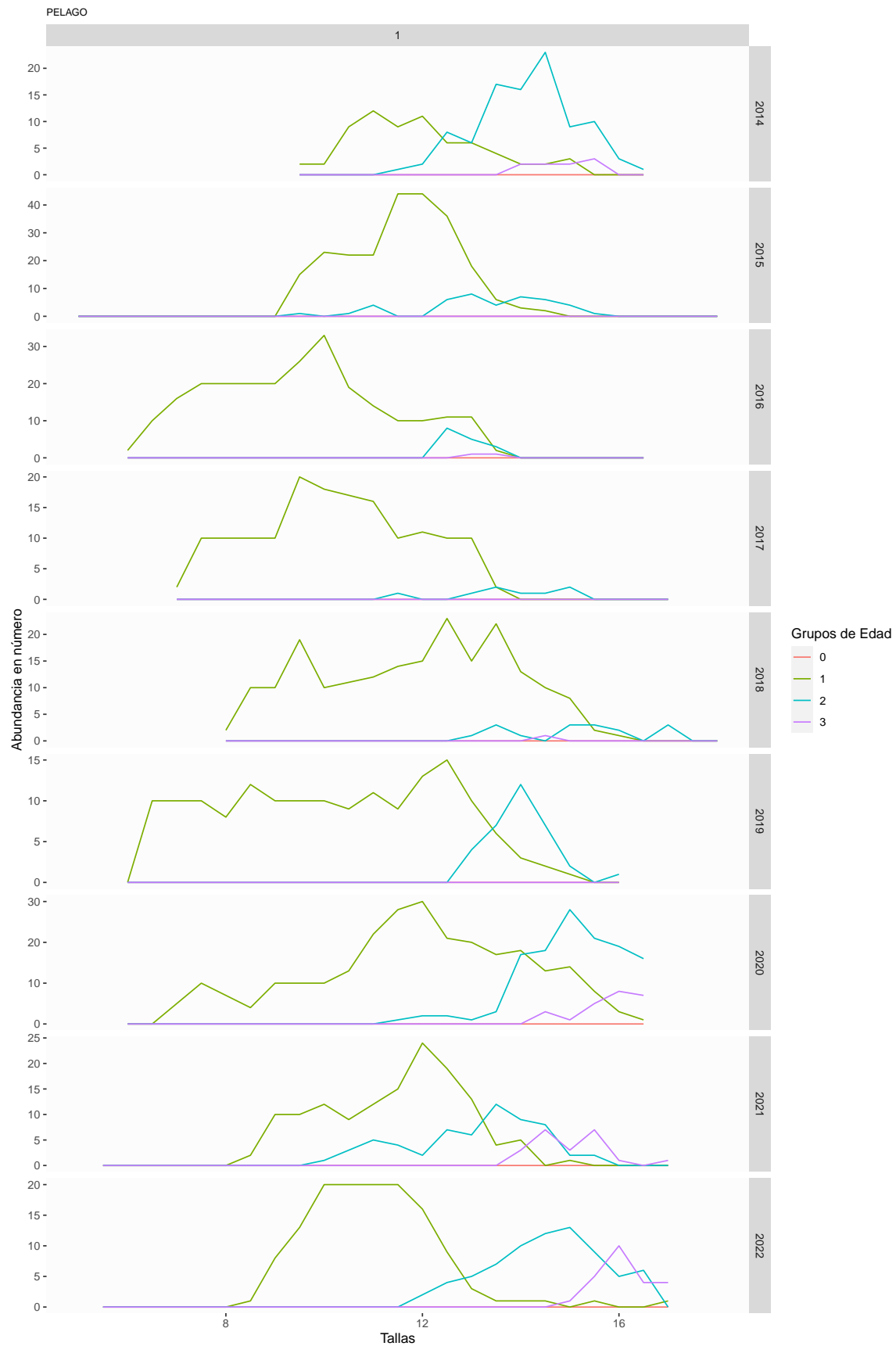
4.1.4 Clave talla-edad formato Gadget

```
pelago.aldist<-read.table(file=here("Data_Gadget","catchdistribution.aldist.pelago.sumofsquares"),
                          header=F,sep=" ",na='NA',fill=T,skip = 3)
names(pelago.aldist)<-c('year','step','area','age','length','number')

pelago.aldist$fleet    <- 2
pelago.aldist$sex      <- 0
pelago.aldist$part     <- 0
pelago.aldist$ageerr   <- 2
pelago.aldist$Lbin_lo  <- pelago.aldist$length
pelago.aldist$Lbin_hi  <- pelago.aldist$length
pelago.aldist$Nsamp    <- 10

pelago.aldist_xxx<- pelago.aldist %>%
  separate(length,into=c("text","length"),sep=3,convert=TRUE)

# Se transforma a formato SS3
pelago.aldistSS3<- pelago.aldist %>%
  separate(length,into=c("text","length"),sep=3,convert=TRUE)%>%
  separate(Lbin_lo,into=c("text","Lbin_lo"),sep=3,convert=TRUE)%>%
  separate(Lbin_hi,into=c("text","Lbin_hi"),sep=3,convert=TRUE)%>%
  arrange("Lbin_lo","Lbin_hi",'year','step','part','ageerr','Nsamp')%>%
  spread(age,number)%>%
  select(-area,-length,-text)
pelago.aldistSS3[is.na(pelago.aldistSS3)] <- 0
```

```

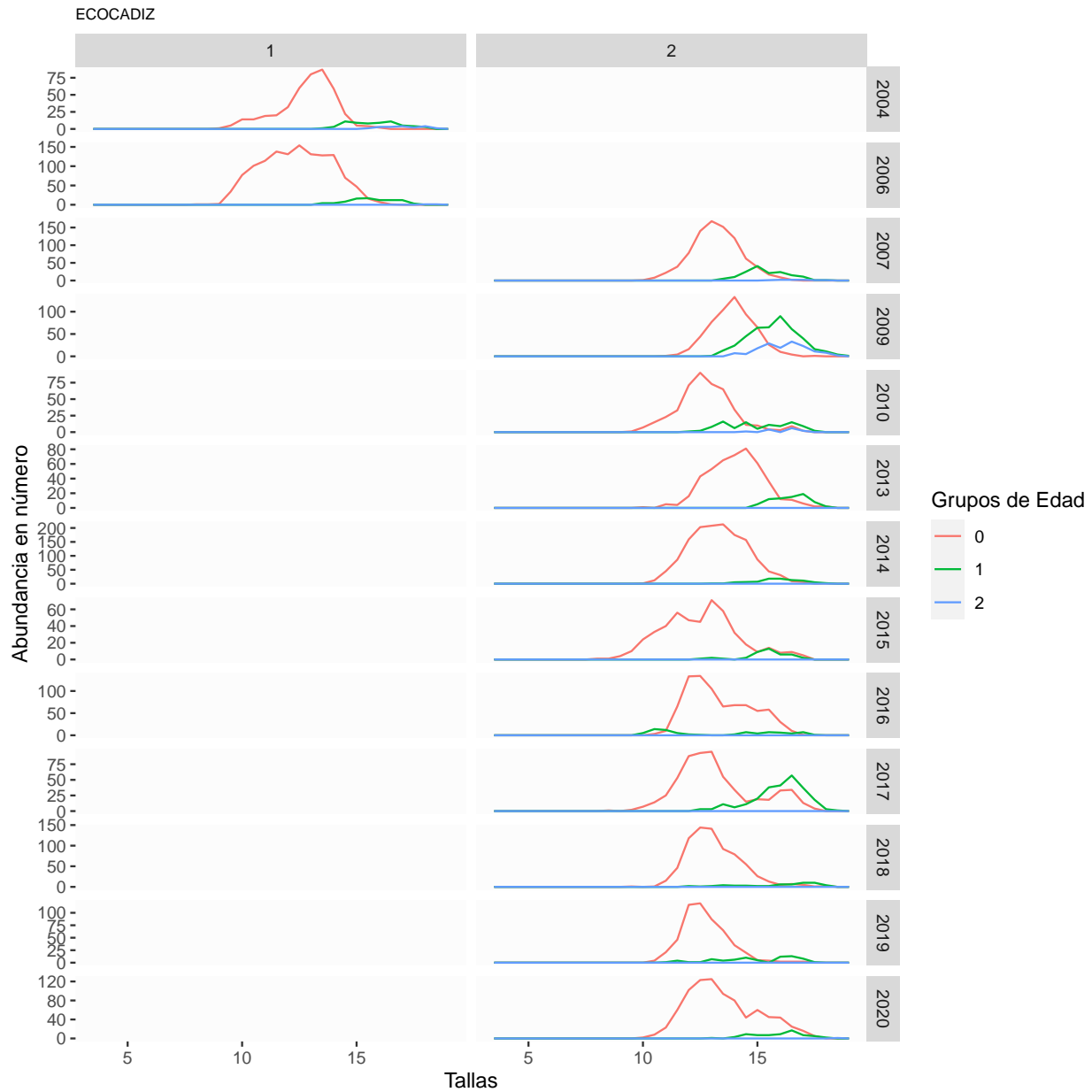
ecocadiz.aldist<-read.table(file=here("Data_Gadget","catchdistribution.aldist.ecocadiz.sumofsquares"),
                           header=F,sep=" ",na='NA',fill=T,skip = 3)
names(ecocadiz.aldist)<-c('year','step','area','age','length','number')

ecocadiz.aldist$fleet    <- 3
ecocadiz.aldist$sex      <- 0
ecocadiz.aldist$part     <- 0
ecocadiz.aldist$ageerr   <- 2
ecocadiz.aldist$Lbin_lo  <- ecocadiz.aldist$length
ecocadiz.aldist$Lbin_hi  <- ecocadiz.aldist$length
ecocadiz.aldist$Nsamp    <- 10

ecocadiz.aldist_xxx<- ecocadiz.aldist %>%
  separate(length,into=c("text","length"),sep=3,convert=TRUE)

# Se transforma a formato SS3
ecocadiz.aldistSS3 <-ecocadiz.aldist %>%
  separate(length,into=c("text","length"),sep=3,convert=TRUE)%>%
  separate(Lbin_lo,into=c("text","Lbin_lo"),sep=3,convert=TRUE)%>%
  separate(Lbin_hi,into=c("text","Lbin_hi"),sep=3,convert=TRUE)%>%
  arrange("Lbin_lo","Lbin_hi",'year','step','part','ageerr','Nsamp')%>%
  spread(age,number)%>%
  select(-area,-length,-text)
ecocadiz.aldistSS3[is.na(ecocadiz.aldistSS3)] <- 0

```



```
fleet.ldist.alkseine<-read.table(file=here("Data_Gadget","catchdistribution.ldist.alkseine.sumofsquares",
                                         header=F,sep=" ",na='NA',fill=T,skip = 3)
names(fleet.ldist.alkseine)<-c('year','step','area','age','length','number')

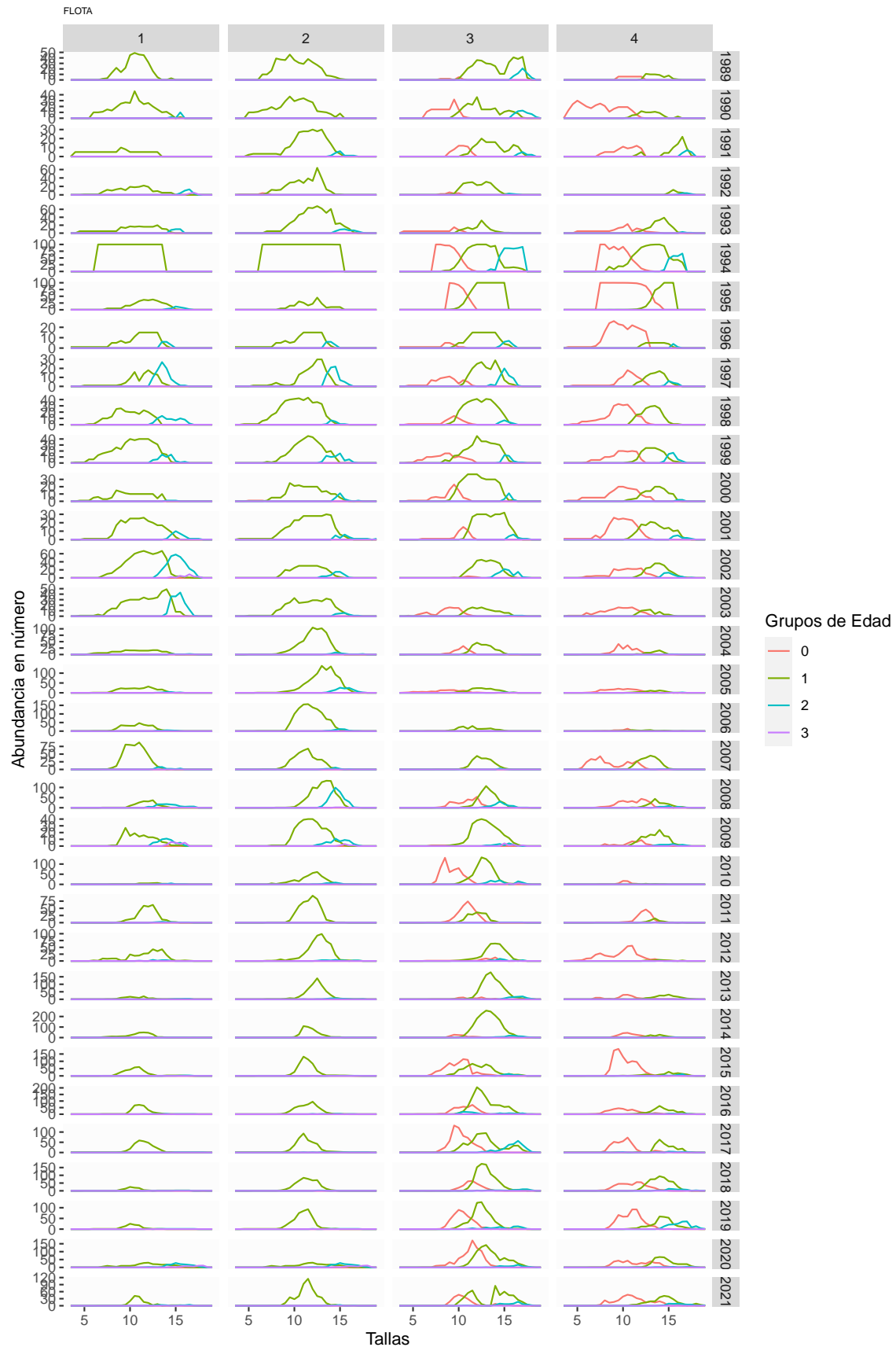
fleet.ldist.alkseine$fleet    <- 3
fleet.ldist.alkseine$sex     <- 0
fleet.ldist.alkseine$part    <- 0
fleet.ldist.alkseine$ageerr  <- 2
fleet.ldist.alkseine$Lbin_lo <- fleet.ldist.alkseine$length
fleet.ldist.alkseine$Lbin_hi <- fleet.ldist.alkseine$length
fleet.ldist.alkseine$Nsamp   <- 10

fleet.ldist.alkseine_xxx <- fleet.ldist.alkseine %>%
  separate(length,into=c("text","length"),sep=3,convert=TRUE)
```

```

# Se transforma a formato SS3
fleet.ldist.alkseineSS3 <- fleet.ldist.alkseine %>%
  separate(length,into=c("text","length"),sep=3,convert=TRUE)%>%
  separate(Lbin_lo,into=c("text","Lbin_lo"),sep=3,convert=TRUE)%>%
  separate(Lbin_hi,into=c("text","Lbin_hi"),sep=3,convert=TRUE)%>%
  arrange("Lbin_lo","Lbin_hi",'year','step','part','ageerr','Nsamp')%>%
  spread(age,number)%>%
  select(-area,-length,-text)
fleet.ldist.alkseineSS3[is.na(fleet.ldist.alkseineSS3)] <- 0

```



4.2 Identificación de los archivos de datos SS3

4.2.1 Archivos utilizado para enfoque de modelación SS3

4.2.1.1 Identificamos el directorio donde se encuentra el modelo base

```
dirname.base <- here("modelos_SS3", "simple")
```

4.2.1.2 Creamos un nuevo directorio para la nueva versión del modelo modificado

```
dirname.simple_mod <- here("boqueron_SS3")  
dir.create(path=dirname.simple_mod, showWarnings = TRUE, recursive = TRUE)
```

4.2.1.3 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
```

4.2.2 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  
## Char version is 3.30  
## Numeric version is 3.3  
dat1<-dat # para modificar  
#names(dat1) # muestra los objetos de la lista  
  
#Especificaciones iniciales  
dat1$styr <-1989 #_StartYr  
dat1$endyr <-2022 #_EndYr  
dat1$nseas <-4 #_Nseas  
dat1$months_per_seas<-c(3,3,3,3) #_months/season  
dat1$Nsubseasons <-2 #_Nsubseasons (even number, minimum is 2)  
dat1$spawn_month <-8 #_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$Nages <-4 #_Nages=accumulator age, first age is always age 0  
dat1$N_areas <-1 #_Nareas  
dat1$Nfleets <-3 #_Nfleets (including surveys)
```

4.2.3 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”.](#)

4.2.3.1 Especificaciones de datos de captura

```
#-----
#_fleet_type: 1=catch fleet; 2=bycatch only fleet; 3=survey; 4=ignore
#_sample_timing: -1 for fishing fleet to use season-long catch-at-age for
#observations, or 1 to use observation month; (always 1 for surveys)
#_fleet_area: area the fleet/survey operates in
#_units of catch: 1=bio; 2=num (ignored for surveys; their units read later)
#_catch_mult: 0=no; 1=yes
#_rows are fleets
#_fleet_type fishery_timing area catch_units need_catch_mult fleetname
#-----
# Arreglo de datos
fleetnames1 <-c("FLOTA", "PELAGO", "ECOCADIZ")
type1 <-c(1,3,3)
surveytiming1 <-c(-1,0.08,0.4)
units_of_catch1 <-c(1,1,1)
areas1 <-c(1,1,1)
need_catch_mult1 <-c(0,0,0)
#-----
# crear data.frame
fleetinfo1<-data.frame(type = type1,
                      surveytiming = surveytiming1,
                      area = areas1,
                      units = units_of_catch1,
                      need_catch_mult = need_catch_mult1,
                      fleetname = fleetnames1)
#-----
dat1$fleetinfo<-fleetinfo1
#-----
```

4.2.3.2 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 <- capturas$year
nyear <- length(year)

catch_year <- c(rep(-999,4),year)
catch_seas <- c(seq(1,4,1),capturas$step)
catch_fleet <- rep(1,nyear+4)
catch_catch <- c(rep(0,4),capturas$number)
catch_catch_se <- rep(0.01,nyear+4) # 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
```

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

- 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.

4.2.4 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”.](#)

4.2.4.1 Especificaciones de los índices de abundancia

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

CPUEinfo_names<-fleetnames1
#-----
# 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
#-----
```

- 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:

4.2.4.2 Datos de índices de abundancia

```
#-----  
# Arreglo de datos  
  
CPUE_year<-c(pelago$year,ecocadiz$year)  
#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(pelago$year)),  
               rep(3,length(ecocadiz$year)))  
CPUE_obs<-c(pelago$number,  
            ecocadiz$number)  
CPUE_se_log<-c(rep(0.3,length(pelago$year)),  
               rep(0.3,length(ecocadiz$year)))  
  
#-----  
# 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  
#-----
```

4.2.5 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”.](#)

4.2.5.1 Descarte

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

4.2.5.2 Tallas medias

```

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

```

4.2.6 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”.

4.2.6.1 Bins tallas

```

# set up population length bin structure (note - irrelevant if not using size
#data and using empirical wtatage
dat1$lbmethod <-2 # length bin method:
# 1=use databins;
# 2=generate from binwidth,min,max below;
# 3=read vector
dat1$binwidth <-0.5 # binwidth for population size comp
dat1$minimum_size <-3.5 # minimum size in the population
# (lower edge of first bin and size at age 0.00)
dat1$maximum_size <-21.5 # maximum size in the population (lower edge of last bin)
dat1$use_lencomp <-1 # use length composition data (0/1)

```

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

4.2.6.2 Especificación composición de tallas

```

#-----
#_mintailcomp: upper and lower distribution for females and males separately are
# accumulated until exceeding this level.
#_addtocomp: after accumulation of tails; this value added to all bins
#_combM+F: males and females treated as combined gender below this bin number
#_compressbins: accumulate upper tail by this number of bins; acts simultaneous with
# mintailcomp; set=0 for no forced accumulation
#_Comp_Error: 0=multinomial,
# 1=dirichlet using Theta*n,
# 2=dirichlet using beta,
# 3=MV_Tweedie
#_ParmSelect: consecutive index for dirichlet or MV_Tweedie
#_minsamplesize: minimum sample size; set to 1 to match 3.24, minimum value is 0.001
#
#_mintailcomp addtocomp combM+F CompressBins CompError ParmSelect minsamplesize
#-----
# Arreglo de datos
nfleets<-dat1$Nfleets
len_info_mintailcomp <-rep(-1,nfleets)

```

```

len_info_addtocomp      <-rep(0.001,nfleets)
len_info_combine_M_F    <-rep(0,nfleets)
len_info_CompressBins   <-rep(0,nfleets)
len_info_CompError      <-rep(0,nfleets)
len_info_ParmSelect     <-rep(0,nfleets)
len_info_minsamplesize  <-rep(1,nfleets)
#-----
# 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)<-fleetnames1
#-----
dat1$len_info<-len_info1 #data.frame
#dat1$len_info
#-----

```

4.2.6.3 Especificación del vector de tallas

```

dat1$N_lbins<-37
dat1$lbin_vector<-seq(3.5,21.5,0.5)

```

4.2.6.4 Datos de composición de tallas

```

#-----
# sex codes:  0=combined;
#             1=use female only;
#             2=use male only;
#             3=use both as joint sexlength distribution
# partition codes:  (0=combined;
#                   1=discard;
#                   2=retained)
#-----
# Arreglo de datos
#-----
tallasplot<-merge(tallasflota,
                  merge(tallaspelago,tallasecocadiz,
                        by=c('year','step','area','age','length',"text"),all=TRUE),
                  by=c('year','step','area','age','length',"text"),all=TRUE) %>%
select(-area, -age,-text)%>%
magrittr::set_colnames(c('year','step','length','FLOTA','PELAGO','ECOCADIZ')) %>%
mutate(Gender=0,
       Part=0,
       Nsamp=100) %>%
melt(id.vars=c('year','step','length','Gender','Part','Nsamp')) %>%
spread(length,value)

tallasSS3<-tallasplot[order(tallasplot$variable),]
tallasSS3[is.na(tallasSS3)] <- 0

```

```

#-----
nyear<-length(tallasSS3$year)/3

new_lencomp <- data.frame(Yr      = tallasSS3$year,
                          Seas   = tallasSS3$step,
                          FltSvy = c(rep(1,nyear),rep(2,nyear),rep(3,nyear)),
                          Gender = tallasSS3$Gender,
                          Part    = tallasSS3$Part,
                          Nsamp  = tallasSS3$Nsamp)

dat_rows_names <- paste("L",seq(3.5,21.5,0.5) ,sep="")

dat_rows <- tallasSS3[,7:43]
names(dat_rows)<-dat_rows_names
#-----
# crear data.frame
new_lencomp1<-cbind(new_lencomp, dat_rows)
#-----
dat1$lencomp<-new_lencomp1
#dat1$lencomp
#-----

```

4.2.7 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”.

4.2.7.1 Bins de edad

```

dat1$N_agebins<-0
dat1$agebin_vector<-""

```

4.2.7.2 Datos de error edad

```

# Arreglo de datos
matrix_ageerror<-""
#-----
dat1$N_ageerror_definitions<- "" #_N_ageerror_definitions
dat1$ageerror<-matrix_ageerror #_ageerror_definitions

```

4.2.7.3 Especificaciones de los datos de composicion de edad

```

#-----
#_mintailcomp: upper and lower distribution for females and males separately are
#               accumulated until exceeding this level.
#_addtocomp:   after accumulation of tails; this value added to all bins
#_combM+F:     males and females treated as combined gender below this bin number
#_compressbins: accumulate upper tail by this number of bins; acts simultaneous
#               with mintailcomp; set=0 for no forced accumulation
#_Comp_Error:  0=multinomial,
#               1=dirichlet using Theta*n,
#               2=dirichlet using beta,

```

```

#           3=MV_Tweedie
#_ParmSelect: consecutive index for dirichlet or MV_Tweedie
#_minsamplesize: minimum sample size; set to 1 to match 3.24, minimum value is 0.001
#
#_mintailcomp addtocomp combM+F CompressBins CompError ParmSelect minsamplesize
#-----
# crear data.frame
age_info1<-data.frame(mintailcomp="#",
                      addtocomp="#",
                      combine_M_F="#",
                      CompressBins="#",
                      CompError="#",
                      ParmSelect="#",
                      minsamplesize="#")

row.names(age_info1)<-"#"
#-----
dat1$age_info<-age_info1
#-----

```

4.2.7.4 Datos de composicion de edad

```

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

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

#names(data.age)<-dat_rows_names

new_agecomp<- "#" #cbind(new_agecomp, data.age)
#-----
dat1$agecomp<-new_agecomp
#-----

```

4.2.8 Otros datos

4.2.8.1 Tallas medias a la edad

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

dat_rows_names <- "##" #c(paste("L",seq(0,5,1),sep=""),paste("nmL",seq(0,5,1),sep="")) # los datos de es
dat_rows <- as.data.frame(matrix(data = 0,
                                nrow = nrow(MeanSize_at_Age_obs_esp),
                                ncol = length(dat_rows_names)))
names(dat_rows)<-dat_rows_names
#-----
# crear data.frame
MeanSize_at_Age_obs1 <-cbind(MeanSize_at_Age_obs_esp, dat_rows)
#-----
dat1$MeanSize_at_Age_obs<-MeanSize_at_Age_obs1
#-----
```

4.2.8.2 Datos ambientales

(tarea = buscar ejemplo donde ingresen estos datos)

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

4.2.8.3 Sizefreq data

(tarea = buscar ejemplo donde ingresen estos datos)

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

4.2.8.4 Datos de tags

(tarea = buscar ejemplo donde ingresen estos datos)

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

4.2.8.5 Datos de morfos

(tarea = buscar ejemplo donde ingresen estos datos)

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

4.2.8.5.1 Priors de selectividad

(tarea = buscar ejemplo donde ingresen estos datos)

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

4.2.8.6 Final de archivo data.ss

imprime línea final

```
dat1$eof<-TRUE
```

4.3 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)
```