

Transición de archivos de datos GADGET a SS3 de boqueron Cádiz

abril, 21, 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	15
4.2.4	Indices de abundancia formato SS3	17
4.2.5	Descartes y tallas medias Formato SS3	19
4.2.6	Composición de tallas Formato SS3	19
4.2.7	Composición de edad Formato SS3	22
4.2.8	Otros datos	24
4.3	Escribir archivo de datos modificado con la función SS_write para el enfoque de modelación SS3	25

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

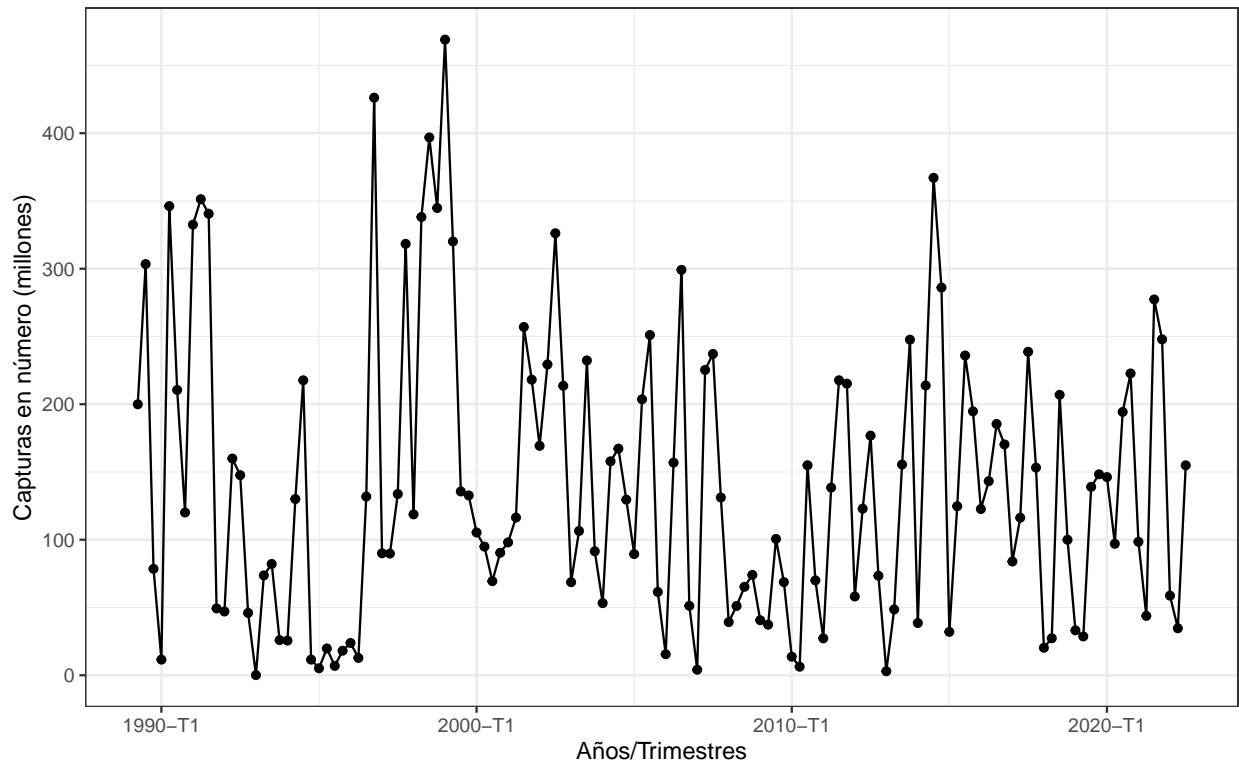
Se utilizarán los siguientes archivos de entrada de datos de Gadget:

```
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

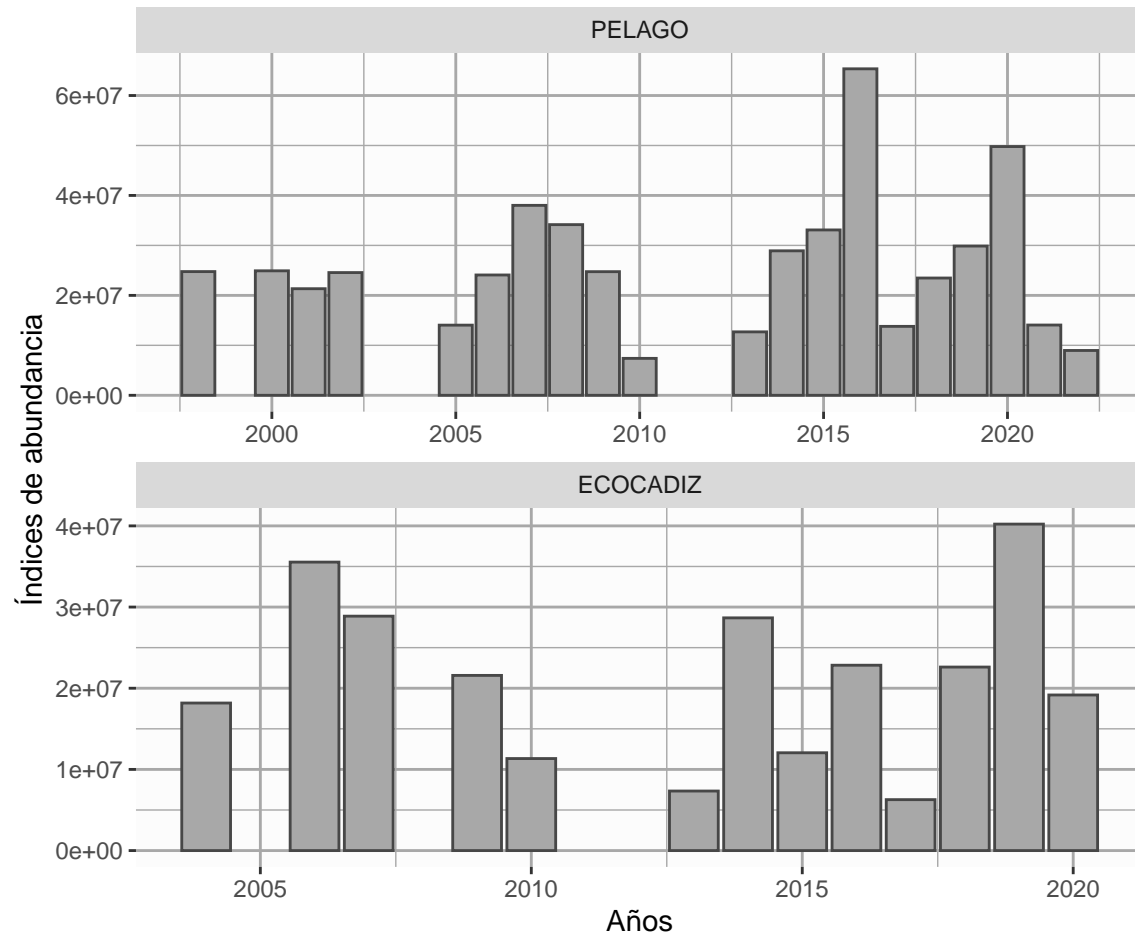
Capturas anuales del stock (toneladas)

```
# 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

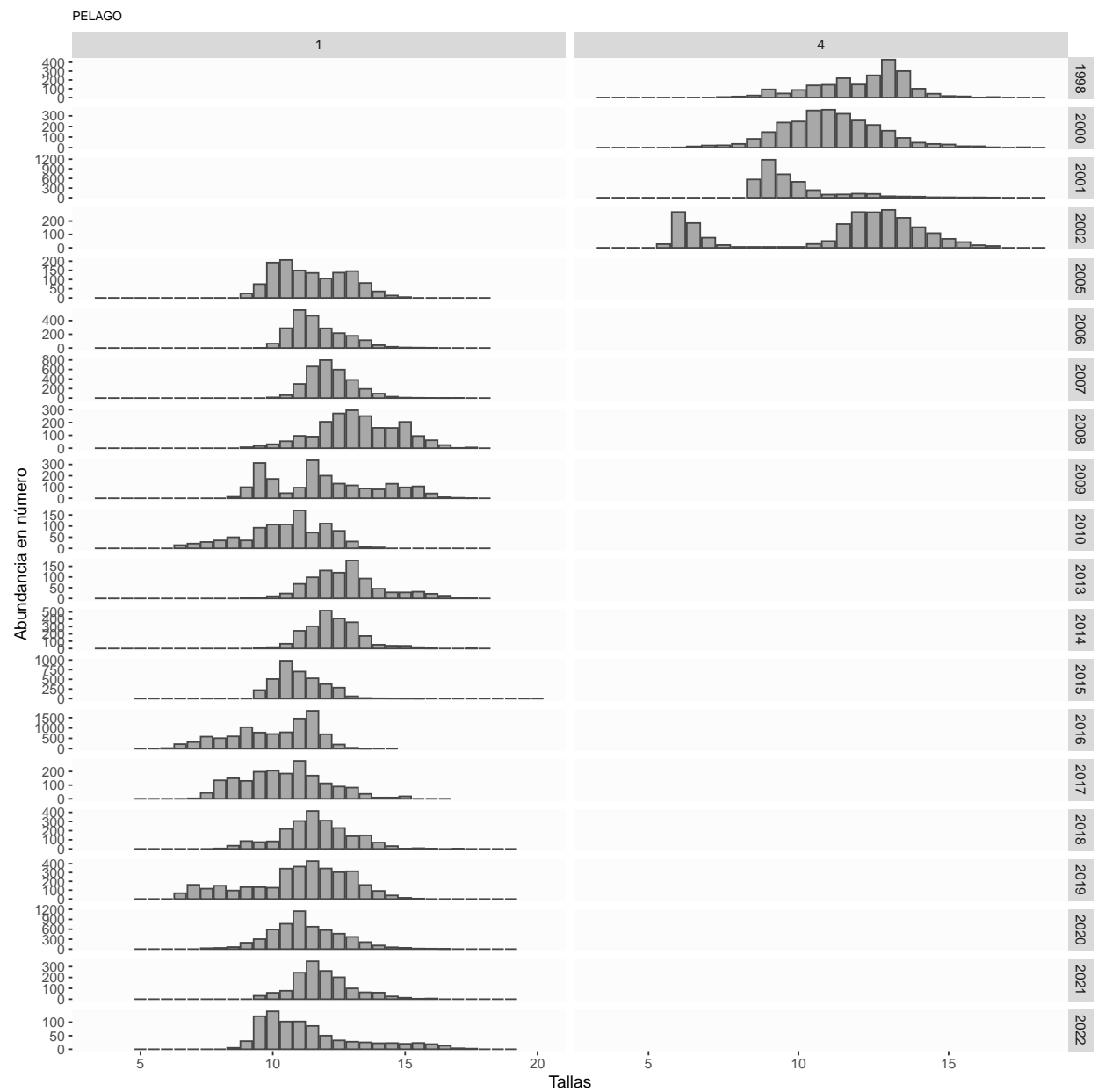
```
#-----
tallaspelago<-read.table(file=here('Data_Gadget',"catchdistribution.ldist.pelago.noage.sumofsquares"),
                        header=F,sep=" ",na='NA',fill=T,skip = 3) %>%
  magrittr::set_colnames(c('year','step','area','age','length','number')) %>%
  separate(length,into=c("text","length"),sep=3,convert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

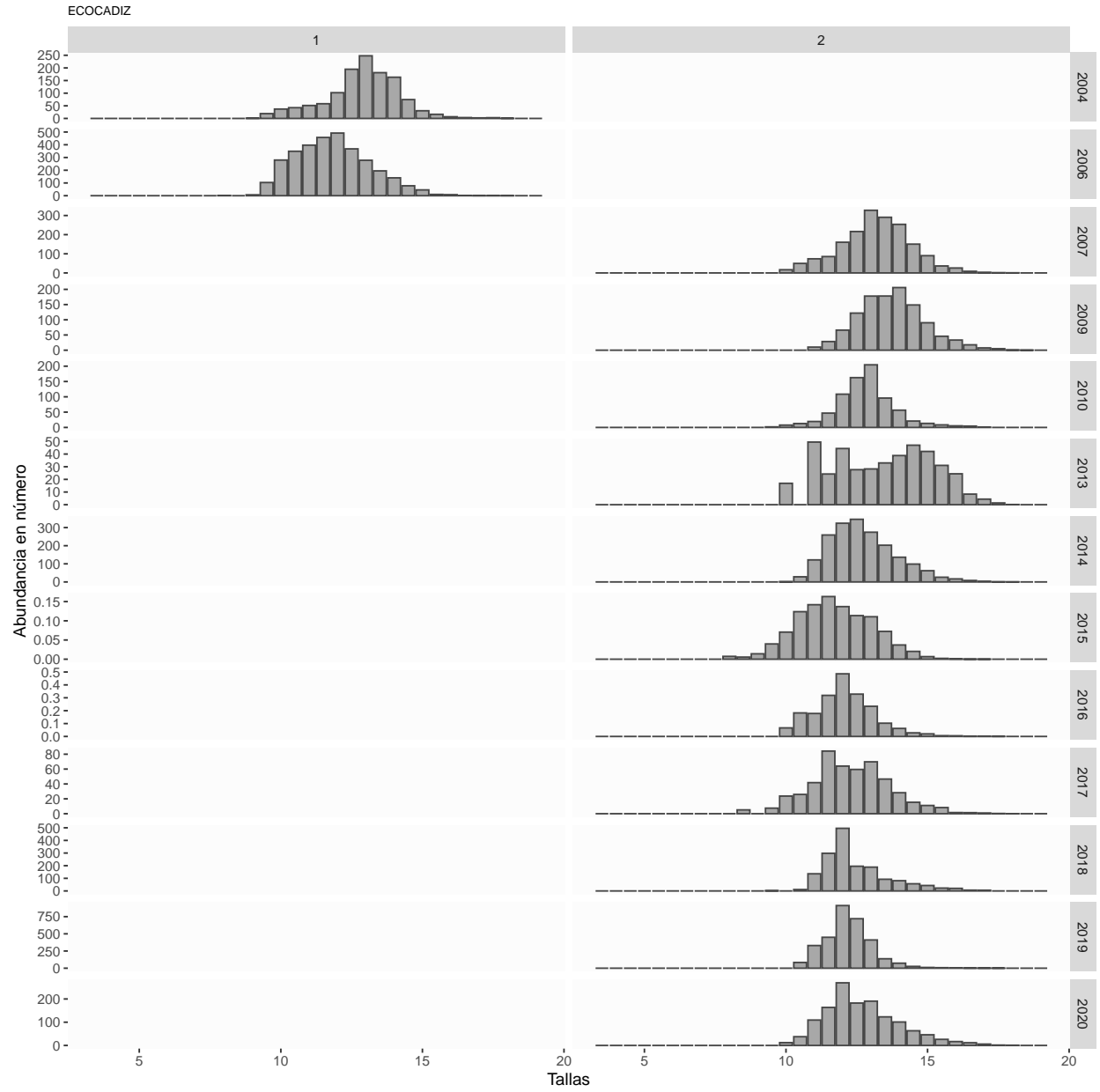
#tallaspelago
#-----
tallasecocadiz<-read.table(file=here('Data_Gadget',"catchdistribution.ldist.ecocadiz.noage.sumofsquares"),
                          header=F,sep=" ",na='NA',fill=T,skip = 3) %>%
  magrittr::set_colnames(c('year','step','area','age','length','number')) %>%
  separate(length,into=c("text","length"),sep=3,convert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

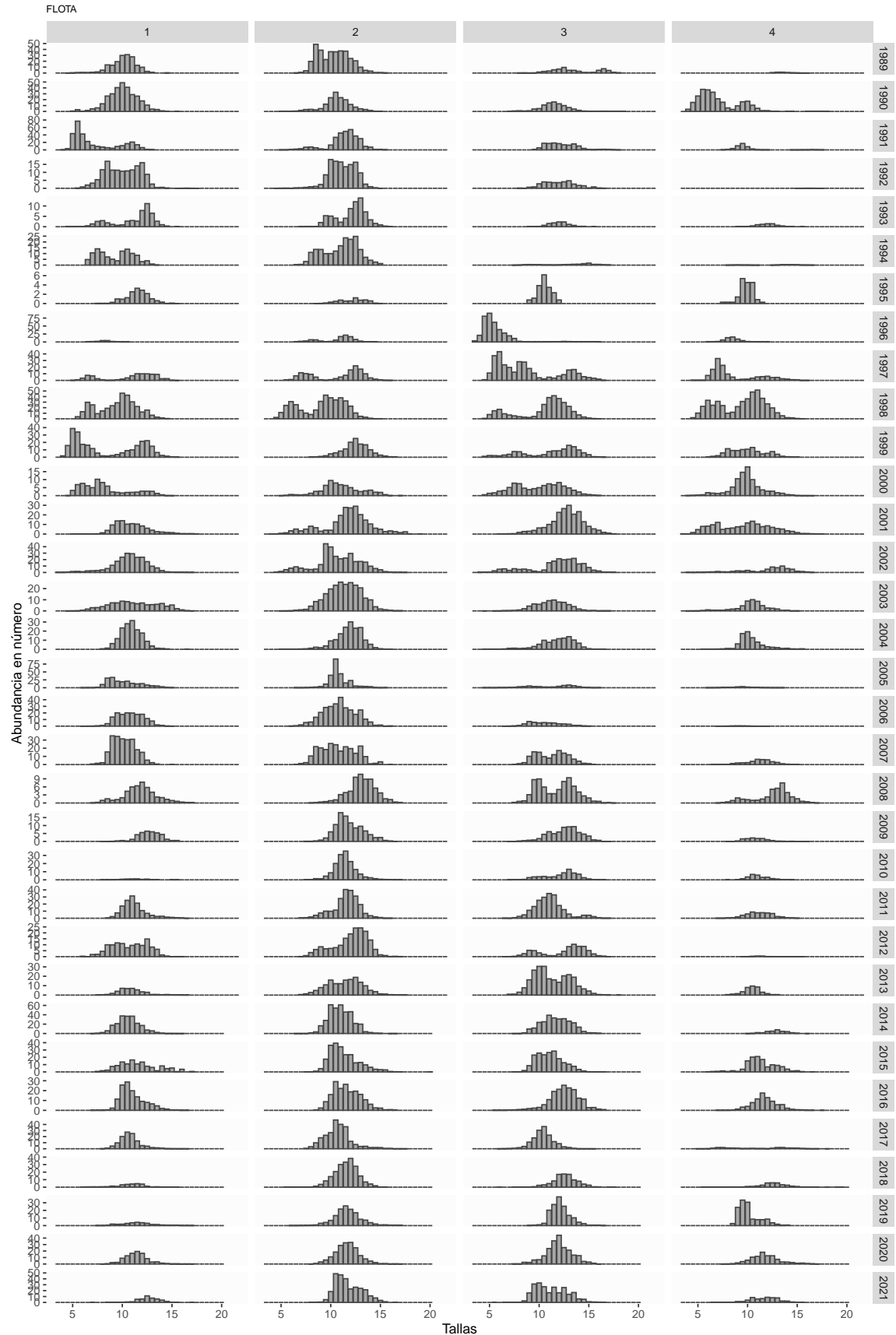
#tallasecocadiz
#-----
tallasflota<-read.table(file=here('Data_Gadget',"catchdistribution.ldist.seine.sumofsquares"),
                      header=F,sep=" ",na='NA',fill=T,skip = 3) %>%
  magrittr::set_colnames(c('year','step','area','age','length','number')) %>%
  separate(length,into=c("text","length"),sep=3,convert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

#tallasflota

#-----







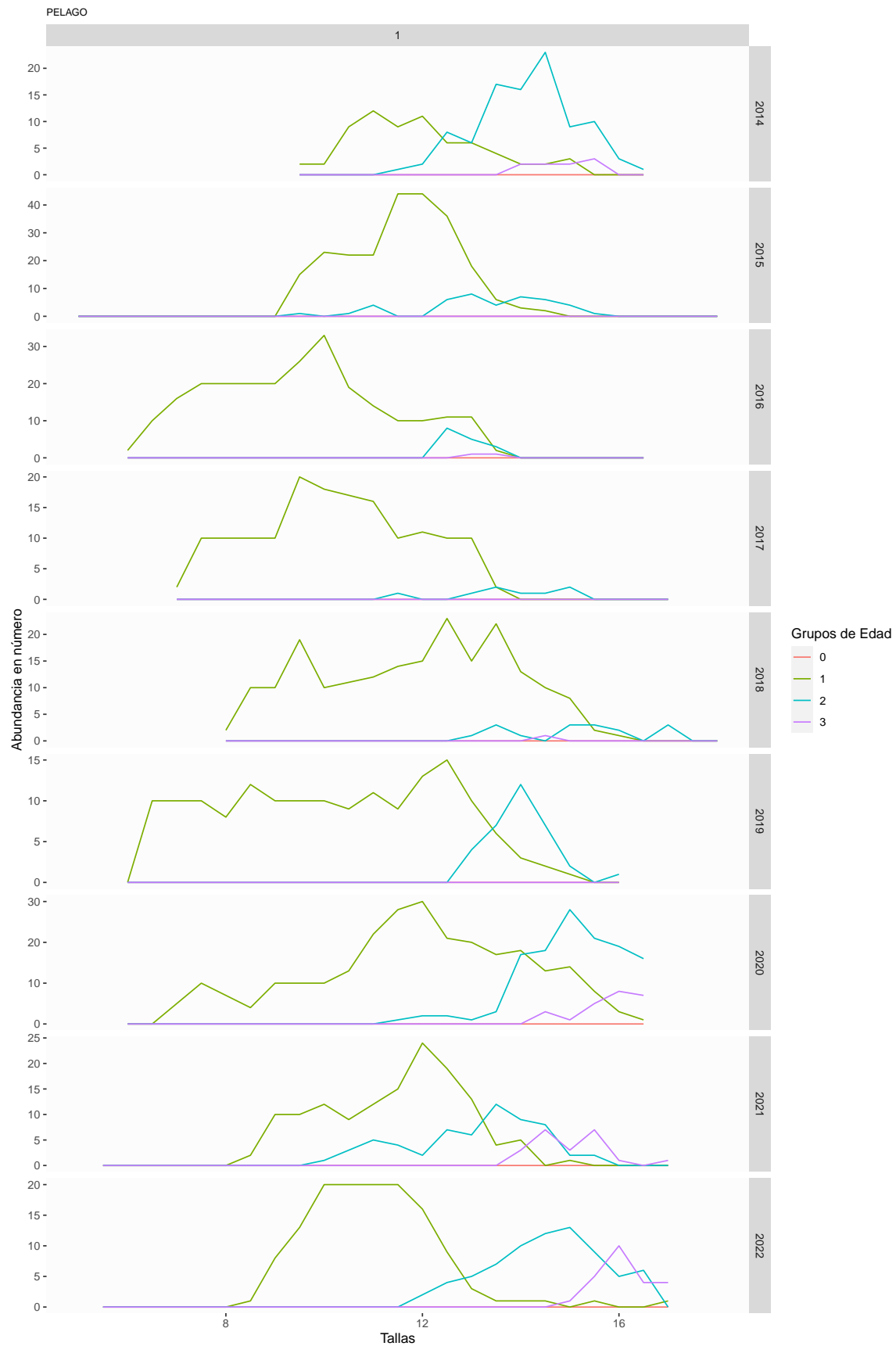
4.1.4 Clave talla-edad formato Gadget

```
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
```

```

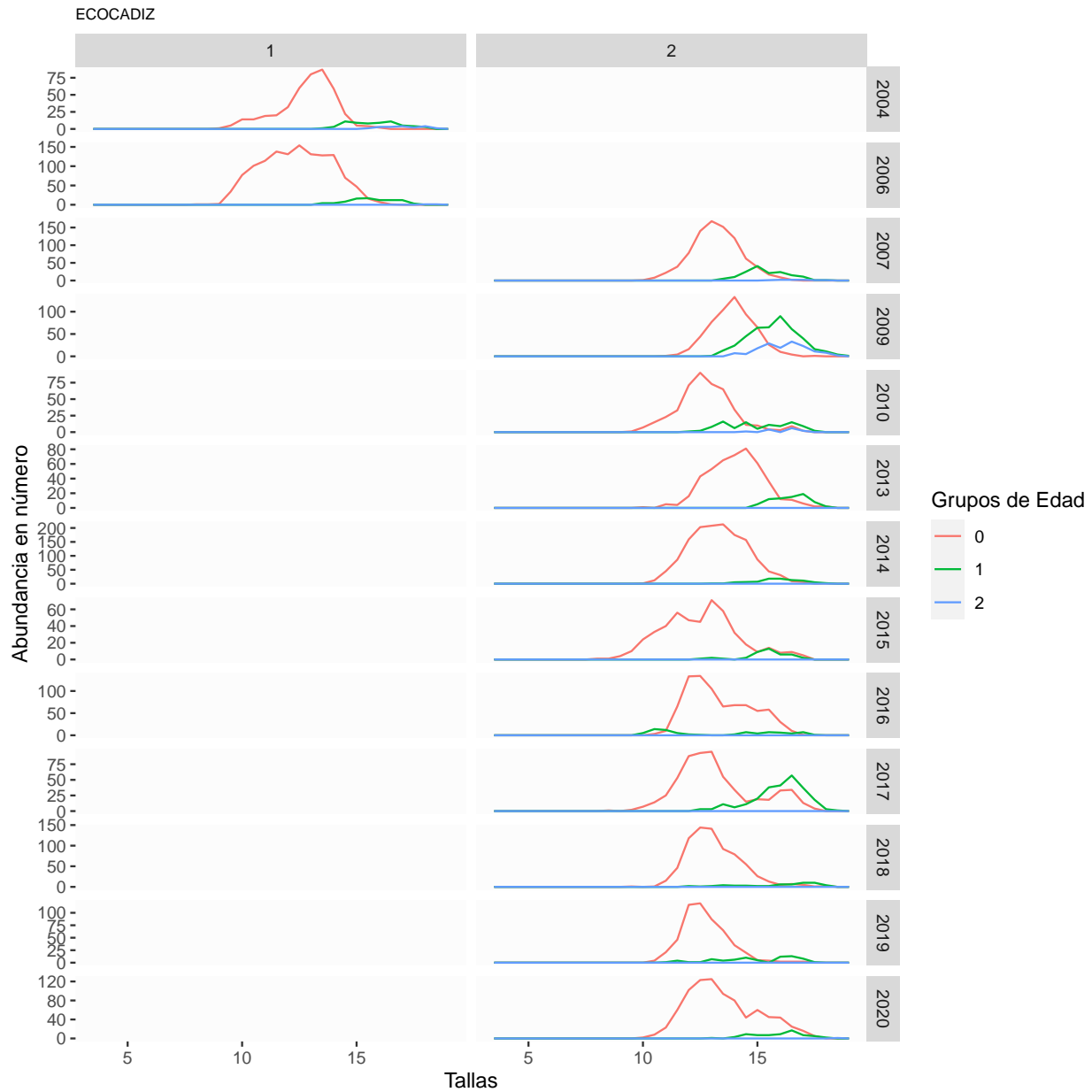
ecocadiz.aldist<-read.table(file=here("Data_Gadget","catchdistribution.aldist.ecocadiz.sumofsquares"),
                           header=F,sep=" ",na='NA',fill=T,skip = 3)
names(ecocadiz.aldist)<-c('year','step','area','age','length','number')

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')

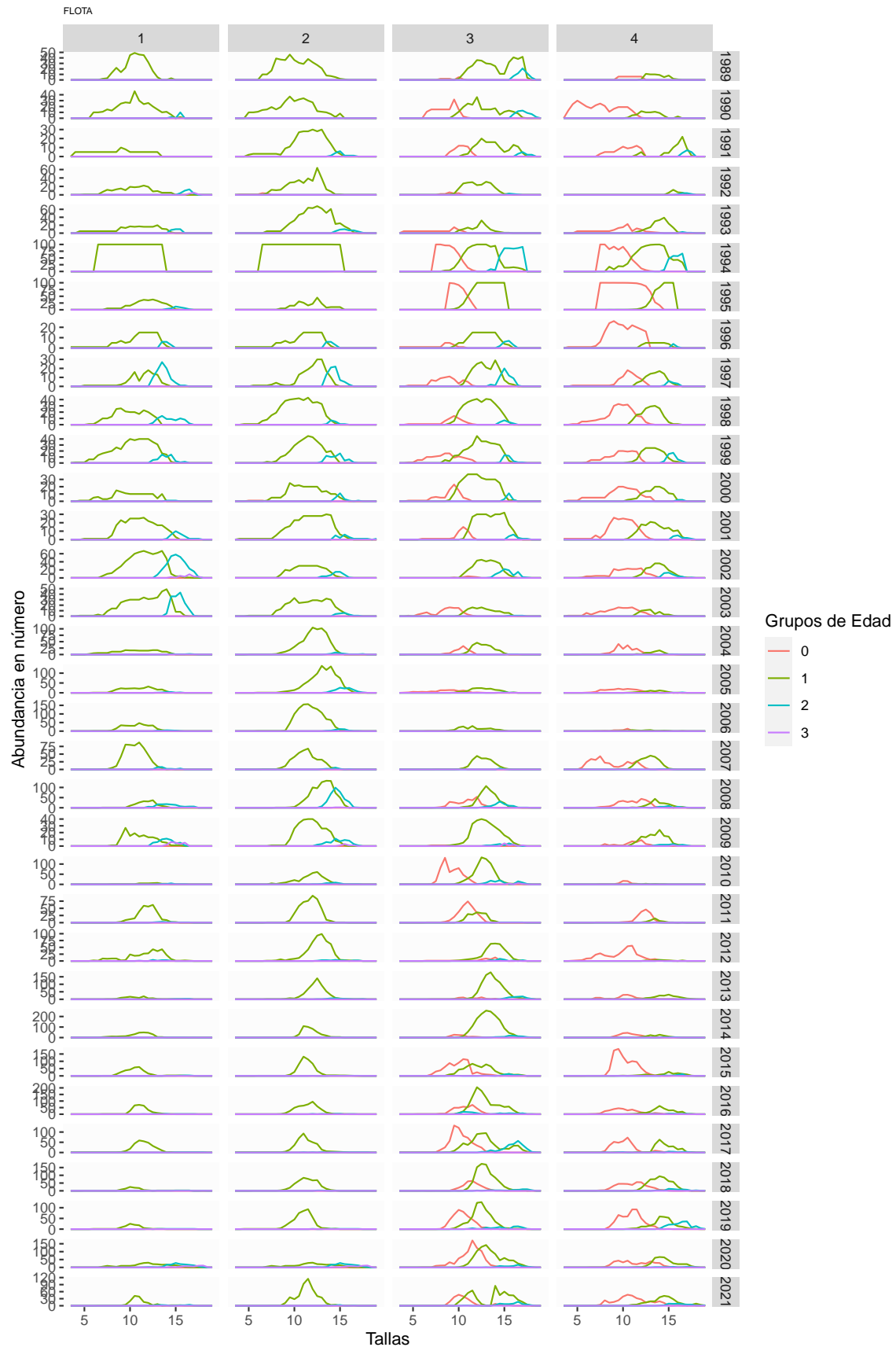
fleet.ldist.alkseine$fleet <- 3
fleet.ldist.alkseine$sex   <- 0
fleet.ldist.alkseine$part  <- 0
fleet.ldist.alkseine$ageerr <- 2
fleet.ldist.alkseine$Lbin_lo <- fleet.ldist.alkseine$length
fleet.ldist.alkseine$Lbin_hi <- fleet.ldist.alkseine$length
fleet.ldist.alkseine$Nsamp  <- 10

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

```
#bajarlo desde Github SS3
dirname.base <- here("modelos_SS3","simple")
#dirname.base <- here("10a_anchcadiz")
```

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,"monthanch.dat")) #base
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$Comments      <- "#C 2022 Anchoveta Cadiz data file MODELO TRIMESTRAL"
dat1$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)

dat1[1:15]
## $sourcefile
## [1] "C:/SS3_boqueron/SS3_ane27.9a/modelos_SS3/simple/data.ss"
```

```
##
## $type
## [1] "Stock_Synthesis_data_file"
##
## $ReadVersion
## [1] "3.30"
##
## $Comments
## [1] "#C 2022 Anchoveta Cadiz data file MODELO TRIMESTRAL"
##
## $styr
## [1] 1989
##
## $endyr
## [1] 2022
##
## $nseas
## [1] 4
##
## $months_per_seas
## [1] 3 3 3 3
##
## $Nsubseasons
## [1] 2
##
## $spawn_month
## [1] 8
##
## $Ngenders
## [1] 1
##
## $Nsexes
## [1] 1
##
## $Nages
## [1] 4
##
## $N_areas
## [1] 1
##
## $Nfleets
## [1] 3
```

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

`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**:

```
#-----
# Arreglo de datos
fleetnames1      <-c("FLOTA", "PELAGO", "ECOCADIZ")
type1            <-c(1,3,3)
surveytiming1     <-c(-1,0.08,0.4)
units_of_catch1  <-c(2,2,2) # flota comercial y ambas campañas en número
areas1           <-c(1,1,1)
need_catch_mult1 <-c(0,0,0)

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

dat1$fleetinfo<-fleetinfo1
#-----
# revisar salida final
dat1$fleetinfo
##   type surveytiming area units need_catch_mult fleetname
## 1    1      -1.00    1     2                0    FLOTA
## 2    3       0.08    1     2                0    PELAGO
## 3    3       0.40    1     2                0    ECOCADIZ
#-----
```

4.2.3.2 Datos de captura

catch es una matriz que contiene la siguiente información;

- **year**: años de información de capturas de la flota comercial
- **seas**: mes de captura
- **fleet**: número de flota
- **catch**: datos de captura en número o toneladas según como se ha especificado anteriormente
- **catch_se**: standard error of log(catch)

NOTE: catch data is ignored for survey fleets

```
#-----
# Arreglo de Datos
year      <- capturas$year # esta información se extrae de los archivos de GADGET
nyear     <- length(year)

catch_year <- c(rep(-999,4),year)
catch_seas <- c(seq(1,4,1),capturas$step) # esta información se extrae de los archivos de GADGET
catch_fleet <- rep(1,nyear+4)
catch_catch <- c(rep(0,4),capturas$number) # esta información se extrae de los archivos de GADGET
catch_catch_se <- rep(0.01,nyear+4) # se asume cv = 0.01 Revisar!!!!
catch1<-data.frame(year      = catch_year,
                   seas      = catch_seas,
```



```

        fleet      = catch_fleet,
        catch       = catch_catch,
        catch_se    = catch_catch_se)
dat1$catch<-catch1
#-----
# revisar salida final
head(dat1$catch, n=15)
##   year seas fleet      catch catch_se
## 1  -999   1    1         0     0.01
## 2  -999   2    1         0     0.01
## 3  -999   3    1         0     0.01
## 4  -999   4    1         0     0.01
## 5  1989   1    1 199957658     0.01
## 6  1989   2    1 303404539     0.01
## 7  1989   3    1  78515361     0.01
## 8  1989   4    1  11546955     0.01
## 9  1990   1    1 346182244     0.01
## 10 1990   2    1 210525329     0.01
## 11 1990   3    1 120108618     0.01
## 12 1990   4    1 332530366     0.01
## 13 1991   1    1 351328902     0.01
## 14 1991   2    1 340602150     0.01
## 15 1991   3    1 49312629      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.

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 y/o Campañas)

CPUEinfo Es una matriz que contiene la siguiente información;

- Fleet:
- 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

```

#-----
# Arreglo de datos
CPUEinfo_Fleet      <- c(1,2,3)
CPUEinfo_Units      <- c(1,1,1)
CPUEinfo_Errtype    <- c(0,0,0)
CPUEinfo_SD_Report  <- c(0,0,0)
CPUEinfo_names      <-fleetnames1
CPUEinfo1<-data.frame(Fleet      = CPUEinfo_Fleet,
                      Units      = CPUEinfo_Units,

```

```

Errtype = CPUEinfo_Errtype,
SD_Report = CPUEinfo_SD_Report)
row.names(CPUEinfo1)<-CPUEinfo_names
dat1$CPUEinfo<-CPUEinfo1
#-----
# revisar salida final
dat1$CPUEinfo
##      Fleet Units Errtype SD_Report
## FLOTA      1      1      0         0
## PELAGO      2      1      0         0
## ECOCADIZ     3      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:

4.2.4.2 Datos de índices de abundancia

CPUE es una matriz que contiene la siguiente información:

- year:
- seas:
- index:
- obs:
- se_log:

```

#-----
# Arreglo de datos
CPUE_year <-c(pelago$year,ecocadiz$year)
CPUE_seas <-rep(1,length(CPUE_year))
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)))

CPUE1<-data.frame(year = CPUE_year,
                  seas = CPUE_seas,
                  index = CPUE_index,
                  obs = CPUE_obs,
                  se_log = CPUE_se_log)
dat1$CPUE<-CPUE1
#-----
# revisar salida final
head(dat1$CPUE,n=10)
##   year seas index      obs se_log
## 1 1998    1     2 24763000    0.3
## 2 2000    1     2 24913000    0.3
## 3 2001    1     2 21335000    0.3
## 4 2002    1     2 24565000    0.3

```

```
## 5 2005 1 2 14041000 0.3
## 6 2006 1 2 24082000 0.3
## 7 2007 1 2 38020000 0.3
## 8 2008 1 2 34162000 0.3
## 9 2009 1 2 24745000 0.3
## 10 2010 1 2 7395000 0.3
#-----
```

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

N_discard_fleets:

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

4.2.5.2 Tallas medias

use_meanbodywt:

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

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)

- `lbin_method`: length bin method: 1=use databins; 2=generate from binwidth,min,max below; 3=read vector
- `binwidth`: binwidth for population size comp
- `minimum_size`: minimum size in the population (lower edge of first bin and size at age 0.00)
- `maximum_size`: maximum size in the population (lower edge of last bin)
- `use_lencomp`: use length composition data (0/1)

```
#-----
dat1$lbin_method <-2
dat1$binwidth <-0.5
dat1$minimum_size <-0
```

```

dat1$maximum_size <-21.5
dat1$use_lencomp <-1
#-----
#revisar salida final
dat1[26:30]
## $lbin_method
## [1] 2
##
## $binwidth
## [1] 0.5
##
## $minimum_size
## [1] 0
##
## $maximum_size
## [1] 21.5
##
## $use_lencomp
## [1] 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

`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 Θ^n , 2=dirichlet using β , 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

```

#-----
# 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)
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$len_info<-len_info1
#-----
# revisar salida final
dat1$len_info
##          mintailcomp addtocomp combine_M_F CompressBins CompError ParmSelect
## FLOTA             -1    0.001           0           0           0           0
## PELAGO             -1    0.001           0           0           0           0
## ECOCADIZ           -1    0.001           0           0           0           0
##          minsamplesize
## FLOTA                1
## PELAGO                1
## ECOCADIZ              1
#-----

```

4.2.6.3 Especificación del vector de tallas

```

#-----
dat1$N_lbins<-37
dat1$lbin_vector<-seq(3.5,21.5,0.5)
#-----
# revisar salida final
dat1[32:33]
## $N_lbins
## [1] 37
##
## $lbin_vector
## [1] 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9.0 9.5 10.0 10.5
## [16] 11.0 11.5 12.0 12.5 13.0 13.5 14.0 14.5 15.0 15.5 16.0 16.5 17.0 17.5 18.0
## [31] 18.5 19.0 19.5 20.0 20.5 21.0 21.5
#-----

```

4.2.6.4 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
#-----
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
new_lencomp1<-cbind(new_lencomp, dat_rows)

dat1$lencomp<-new_lencomp1
#-----
# revisar salida final
dat1$lencomp[1:10,1:10]
##      Yr Seas FltSvy Gender Part Nsamp L3.5      L4      L4.5      L5
## 1  1989   1     1     0    0   100   0      0    127000  452000
## 4  1989   2     1     0    0   100   0      0         0      0
## 7  1989   3     1     0    0   100   0      0         0      0
## 10 1989   4     1     0    0   100   0      0         0      0
## 13 1990   1     1     0    0   100   0      0         0  686000
## 16 1990   2     1     0    0   100   0      0         0  11000
## 19 1990   3     1     0    0   100   0      0         0      0
## 22 1990   4     1     0    0   100   0 3707000 15341000 26269000
## 25 1991   1     1     0    0   100   0  258000  3306000 43707000
## 28 1991   2     1     0    0   100   0      0         0  107000
#-----

```

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$N_agebins<-0
dat1$N_agebins
## [1] 0
dat1$agebin_vector<-NULL
dat1$agebin_vector
## NULL
#-----

```

4.2.7.2 Datos de error edad

Aquí se ingresa la matriz se error de lectura de edad, si no se ingresa datos de edad entonces de deja NULL
ageerror es una matriz que contiene la siguiente información:

```
#-----  
dat1$N_ageerror_definitions<- NULL  
dat1$N_ageerror_definitions  
## NULL  
dat1$ageerror<-NULL  
dat1$ageerror  
## NULL  
#-----
```

4.2.7.3 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

```
#-----  
dat1$age_info<-NULL  
dat1$age_info  
## NULL  
#-----
```

4.2.7.4 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$Lbin_method <- NULL  
dat1$Lbin_method  
## NULL  
#-----
```

```

# Arreglo de datos
new_agecomp<-data.frame(Yr=0,
                        Seas=0,
                        FltSvy=0,
                        Gender=0,
                        Part=0,
                        Ageerr=0,
                        Lbin_lo=0,
                        Lbin_hi=0,
                        Nsamp=0)
dat_rows_names<- paste("E",seq(0,5,1),sep="")
#names(data.age)<-dat_rows_names
dat1$agecomp<-NULL
dat1$agecomp
## NULL
#-----
#-----

```

4.2.8 Otros datos

4.2.8.1 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
#-----
# Arreglo de datos
MeanSize_at_Age_obs_esp<-data.frame(Yr=0,
                                    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=""))
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$MeanSize_at_Age_obs<-NULL #MeanSize_at_Age_obs1
dat1$MeanSize_at_Age_obs
## NULL
#-----

```

4.2.8.2 Datos ambientales

```

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

```

4.2.8.3 Sizefreq data

```

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

```

4.2.8.4 Datos de tags

```

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

```

4.2.8.5 Datos de morfos

```

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

```

4.2.8.5.1 Priors de selectividad

```

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

```

4.2.8.6 Final de archivo data.ss

imprime línea final

```

dat1$eof<-TRUE
dat1$eo
## [1] 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)

```