Extracting purse-seine catch and length compostion data for yellowfin in 2000-2022

Haikun Xu

2024-08-19

This example code demonstrates how to extract the purse-seine catch and length composition data for the stock assessment of yellowfin tuna in the eastern Pacific Ocean. Data are extracted for yellowfin between 2000 and 2022 based on the R package *BSE* (version 1.2.2). The package can be installed using devtools::install\_github('HaikunXu/BSE',ref='main'). Fishery definition for this data extraction is based on the benchmark assessment conducted in 2020.

* Step 1: set up some directories and parameters for the extraction

# devtools::install\_github('HaikunXu/BSE',ref='main')   
library(BSE)  
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.2.3

## Warning: package 'ggplot2' was built under R version 4.2.3

## Warning: package 'tibble' was built under R version 4.2.3

## Warning: package 'tidyr' was built under R version 4.2.3

## Warning: package 'readr' was built under R version 4.2.3

## Warning: package 'purrr' was built under R version 4.2.3

## Warning: package 'dplyr' was built under R version 4.2.3

## Warning: package 'stringr' was built under R version 4.2.3

## Warning: package 'forcats' was built under R version 4.2.3

## Warning: package 'lubridate' was built under R version 4.2.3

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

# Load the base files (please ask Haikun to get those data)  
raw\_data\_dir <- "D:/OneDrive - IATTC/IATTC/2023/SAC14/YFT SSIs/DEL/Database/"  
# the directory where output will be saved  
save\_dir <- "D:/OneDrive - IATTC/IATTC/2022/BSE stuff from Cleridy/YFT/"  
yr.start <- 2000  
yr.end <- 2022  
Species <- "YFT"  
grow.increments <- grow.increments.betyftskj # the growth increment matrix

* Step 2: lightly process the raw data so that they can be easily used in rest of steps

# Get the total unloads for the PS fleet  
total.unlds <- read.unloads.f(raw\_data\_dir,"Unloading2000-2022.txt",yr.start,yr.end)  
# Get the CAE+IDM data  
cae <- read.cae.f(raw\_data\_dir,"CAE-LatLon2000-2022.txt",yr.start,yr.end)

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec,  
## : number of items read is not a multiple of the number of columns

# Get the length-frequency data (length in millimeters)  
lfmm <- read.lfmmdata.f(raw\_data\_dir,"LengthMM2000-2022.txt")  
# Get the grouped length-frequency output  
lfgrpd <- read.lengthfreq.f(raw\_data\_dir,"LengthFreq2000-2022.txt")

* Step 3: compile the OBJ catch and composition data for YFT

PS <- "OBJ"  
area.substitution.mat<- matrix(  
 c(1, 2, 3,  
 2, 1, 3,  
 3, 2, 1),  
 ncol = 3,  
 byrow = TRUE  
)  
  
#fishery substitute matrix  
my.FOmatrix <- matrix(paste0("FO.A", area.substitution.mat),ncol=ncol(area.substitution.mat))  
my.UNmatrix <- matrix(paste0("UN.A", area.substitution.mat),ncol=ncol(area.substitution.mat))  
my.DPmatrix <- matrix(paste0("DP.A", area.substitution.mat),ncol=ncol(area.substitution.mat))  
  
cae.stratflg <- create.strat.flg.f(cae$latc5,cae$lonc5,is.lwrght=F,cae$month,cae$setype,cae$class,PS=PS,Species=Species)

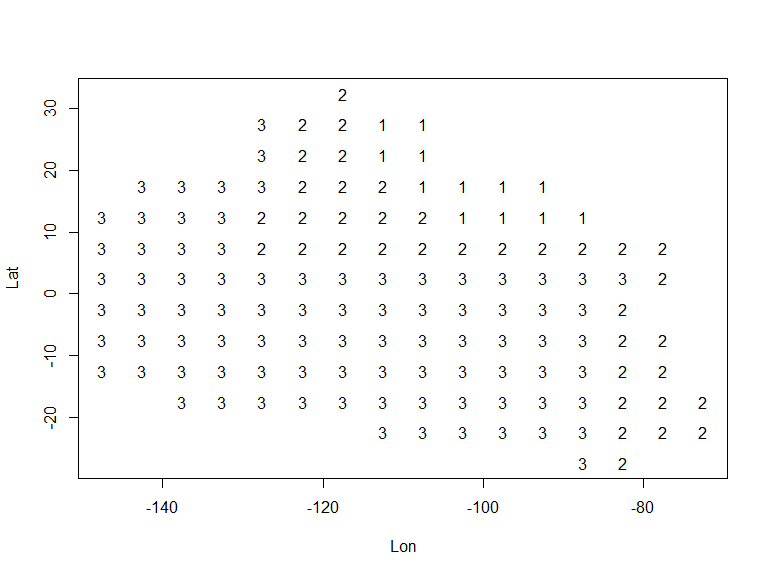
## Joining with `by = join\_by(lat, lon)`

lfgrpd.stratflg <- create.strat.flg.f(lfgrpd$lat.5deg,lfgrpd$lon.5deg,is.lwrght=T,floor(lfgrpd$moda/100),lfgrpd$setype,lfgrpd$class,PS=PS,Species=Species)

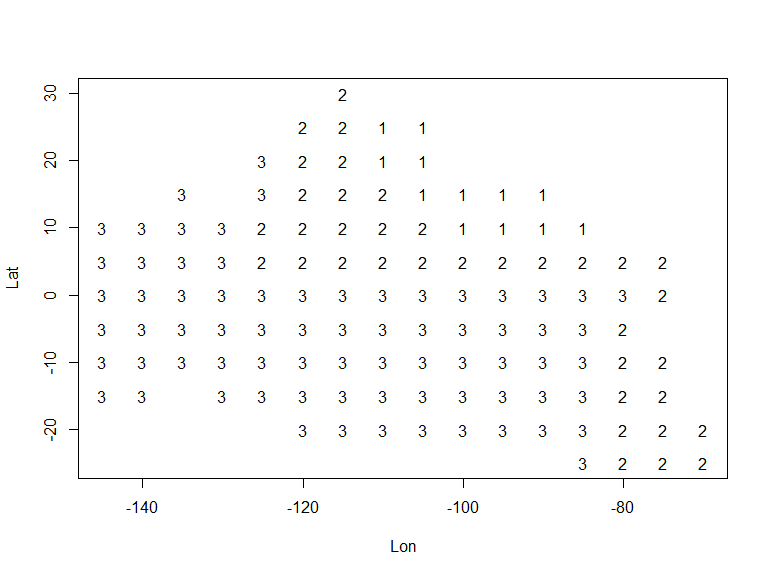
## Joining with `by = join\_by(lat, lon)`

Check the strata definition for OBJ in both cae and lf data sets to make sure that they are correct

check.strat.flg.f(cae$latc5,cae$lonc5,cae.stratflg)



check.strat.flg.f(lfgrpd$lat.5deg,lfgrpd$lon.5deg,lfgrpd.stratflg)



Loop through every year between yr.start and yr.end to get catch and composition data for YFT in the OBJ fishery

for(year in yr.start:yr.end) {  
 # print(paste0("Year: ",year))  
   
 # print("Step 1: get well estimates")  
 well.estimates <- well.estimates.f(lfgrpd[lfgrpd$year.firstset==year,],lfmm)  
   
 # print("Step 2: get catch estimates")  
 catch.estimates <- get.catch.estimates.f(cae,cae.stratflg,total.unlds,lfgrpd,lfgrpd.stratflg,lfmm,year,2,well.estimates,area.substitution.mat,grow.increments,PS=PS,Species=Species,my.FOmatrix,my.UNmatrix,my.DPmatrix)  
   
 # print("Step 3: get fishery estimates")  
 fishery.estimates <- fishery.estimates.f(catch.estimates$stratum.estimates.withsamps,catch.estimates$stratum.estimates.NOsamps,year,PS=PS,Species=Species)  
   
 assign(paste0("fishery.estimates.", year), fishery.estimates, pos=1)  
}

## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`

# save middle-step data as a record  
save(list=objects(pat="fishery.estimates"),file=paste0(save\_dir,"YFT\_",PS,"\_2000-2022.RData"))

Get final OBJ catch and comp output for the stock assessment

YFT.OBJ.Catch.20002022<-compile.catch.output.f(yr.start,yr.end,PS=PS,Species=Species,c("A1","A2","A3")) # five OBJ fisheries  
YFT.OBJ.Comp.20002022<-compile.sizecomps.output.f(yr.start,yr.end,PS=PS,Species=Species)

* Step 4: compile the NOA catch and composition data for YFT

PS <- "NOA"  
  
cae.stratflg <- create.strat.flg.f(cae$latc5,cae$lonc5,is.lwrght=F,cae$month,cae$setype,cae$class,PS=PS,Species=Species)

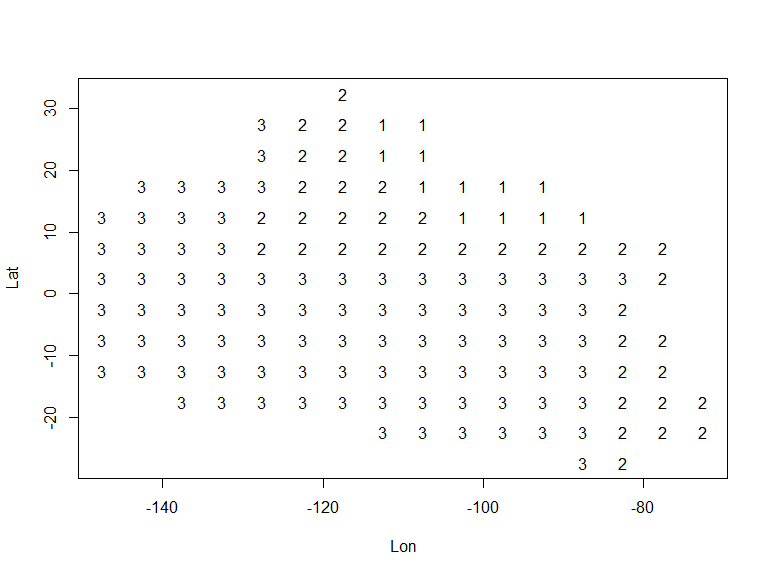
## Joining with `by = join\_by(lat, lon)`

lfgrpd.stratflg <- create.strat.flg.f(lfgrpd$lat.5deg,lfgrpd$lon.5deg,is.lwrght=T,floor(lfgrpd$moda/100),lfgrpd$setype,lfgrpd$class,PS=PS,Species=Species)

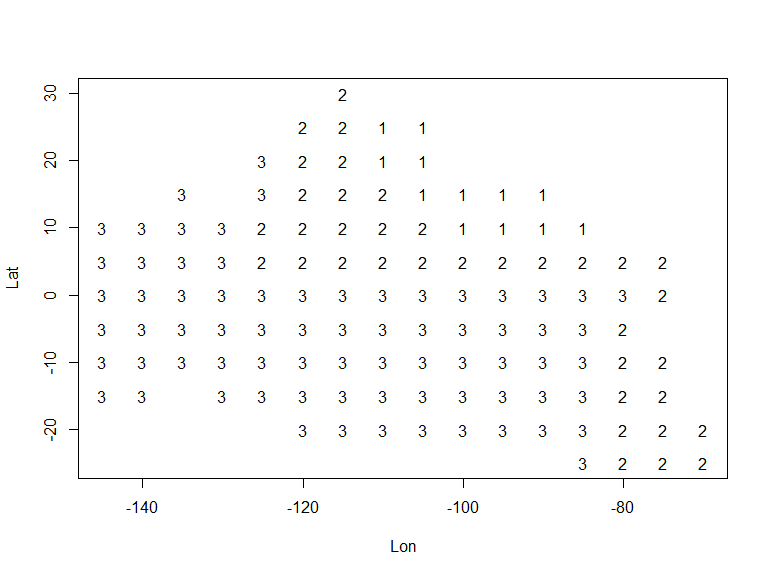
## Joining with `by = join\_by(lat, lon)`

Check the strata definition for NOA in both cae and lf data sets to make sure that they are correct

check.strat.flg.f(cae$latc5,cae$lonc5,cae.stratflg)



check.strat.flg.f(lfgrpd$lat.5deg,lfgrpd$lon.5deg,lfgrpd.stratflg)



Loop through every year between yr.start and yr.end to get catch and composition data for YFT in the NOA fishery

for(year in yr.start:yr.end) {  
 # print(paste0("Year: ",year))  
   
 # print("Step 1: get well estimates")  
 well.estimates <- well.estimates.f(lfgrpd[lfgrpd$year.firstset==year,],lfmm)  
   
 # print("Step 2: get catch estimates")  
 catch.estimates <- get.catch.estimates.f(cae,cae.stratflg,total.unlds,lfgrpd,lfgrpd.stratflg,lfmm,year,2,well.estimates,area.substitution.mat,grow.increments,PS=PS,Species=Species,my.FOmatrix,my.UNmatrix,my.DPmatrix)  
   
 # print("Step 3: get fishery estimates")  
 fishery.estimates <- fishery.estimates.f(catch.estimates$stratum.estimates.withsamps,catch.estimates$stratum.estimates.NOsamps,year,PS=PS,Species=Species)  
   
 assign(paste0("fishery.estimates.", year), fishery.estimates, pos=1)  
}

## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`

# save middle-step data as a record  
save(list=objects(pat="fishery.estimates"),file=paste0(save\_dir,"YFT\_",PS,"\_2000-2022.RData"))

Get final NOA catch and comp output for the stock assessment

YFT.NOA.Catch.20002022<-compile.catch.output.f(yr.start,yr.end,PS=PS,Species=Species,c("A1","A2","A3")) # four NOA fisheries  
YFT.NOA.Comp.20002022<-compile.sizecomps.output.f(yr.start,yr.end,PS=PS,Species=Species)

* Step 5: compile the DEL catch and composition data for YFT

PS <- "DEL"  
  
cae.stratflg <- create.strat.flg.f(cae$latc5,cae$lonc5,is.lwrght=F,cae$month,cae$setype,cae$class,PS=PS,Species=Species)

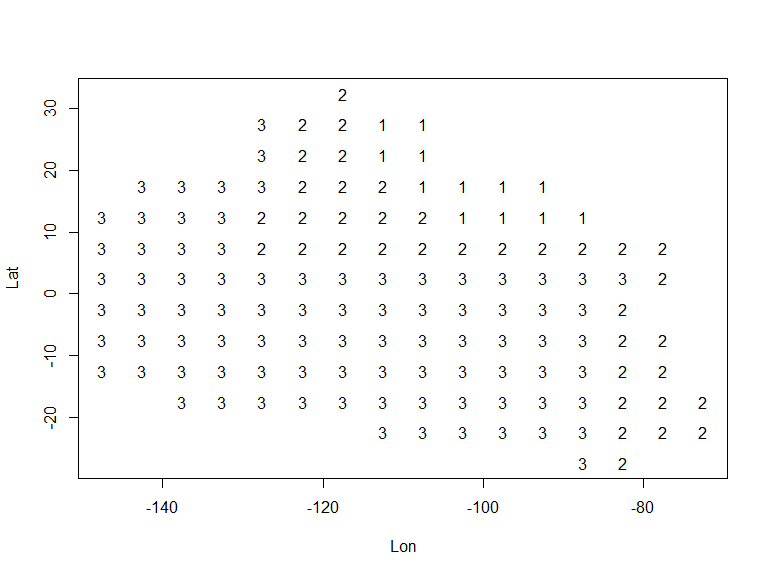
## Joining with `by = join\_by(lat, lon)`

lfgrpd.stratflg <- create.strat.flg.f(lfgrpd$lat.5deg,lfgrpd$lon.5deg,is.lwrght=T,floor(lfgrpd$moda/100),lfgrpd$setype,lfgrpd$class,PS=PS,Species=Species)

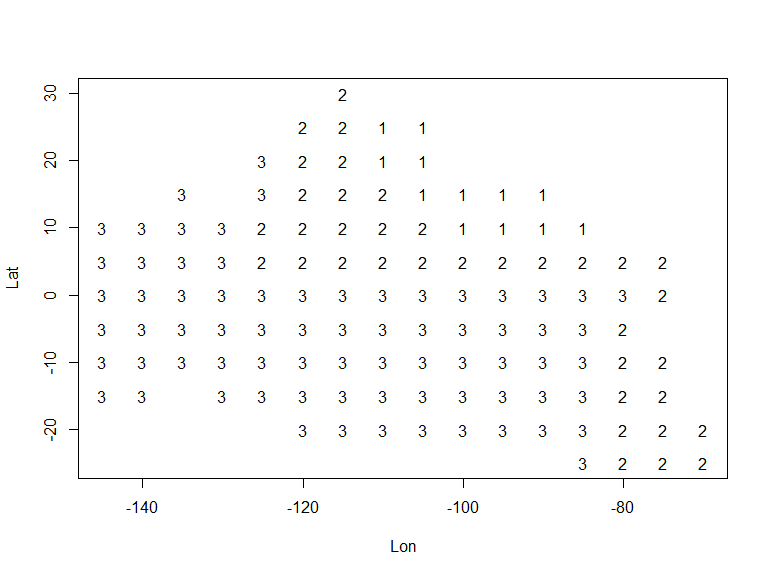
## Joining with `by = join\_by(lat, lon)`

Check the strata definition for DEL in both cae and lf data sets to make sure that they are correct

check.strat.flg.f(cae$latc5,cae$lonc5,cae.stratflg)



check.strat.flg.f(lfgrpd$lat.5deg,lfgrpd$lon.5deg,lfgrpd.stratflg)



Loop through every year between yr.start and yr.end to get catch and composition data for YFT in the DEL fishery

for(year in yr.start:yr.end) {  
 # print(paste0("Year: ",year))  
   
 # print("Step 1: get well estimates")  
 well.estimates <- well.estimates.f(lfgrpd[lfgrpd$year.firstset==year,],lfmm)  
   
 # print("Step 2: get catch estimates")  
 catch.estimates <- get.catch.estimates.f(cae,cae.stratflg,total.unlds,lfgrpd,lfgrpd.stratflg,lfmm,year,2,well.estimates,area.substitution.mat,grow.increments,PS=PS,Species=Species,my.FOmatrix,my.UNmatrix,my.DPmatrix)  
   
 # print("Step 3: get fishery estimates")  
 fishery.estimates <- fishery.estimates.f(catch.estimates$stratum.estimates.withsamps,catch.estimates$stratum.estimates.NOsamps,year,PS=PS,Species=Species)  
   
 assign(paste0("fishery.estimates.", year), fishery.estimates, pos=1)  
}

## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`  
## Joining with `by = join\_by(lat, lon)`

# save middle-step data as a record  
save(list=objects(pat="fishery.estimates"),file=paste0(save\_dir,"YFT\_",PS,"\_2000-2022.RData"))

Get final DEL catch and comp output for the stock assessment

YFT.DEL.Catch.20002022<-compile.catch.output.f(yr.start,yr.end,PS=PS,Species=Species,c("A1","A2","A3")) # two DEL fisheries  
YFT.DEL.Comp.20002022<-compile.sizecomps.output.f(yr.start,yr.end,PS=PS,Species=Species)

* Step 6: save all results for YFT as csv files

write.csv(YFT.OBJ.Catch.20002022,file=paste0(save\_dir,"YFT.OBJ.Catch.20002022.csv"),row.names = FALSE)  
write.csv(YFT.OBJ.Comp.20002022,file=paste0(save\_dir,"YFT.OBJ.Comp.20002022.csv"),row.names = FALSE)  
write.csv(YFT.NOA.Catch.20002022,file=paste0(save\_dir,"YFT.NOA.Catch.20002022.csv"),row.names = FALSE)  
write.csv(YFT.NOA.Comp.20002022,file=paste0(save\_dir,"YFT.NOA.Comp.20002022.csv"),row.names = FALSE)  
write.csv(YFT.DEL.Catch.20002022,file=paste0(save\_dir,"YFT.DEL.Catch.20002022.csv"),row.names = FALSE)  
write.csv(YFT.DEL.Comp.20002022,file=paste0(save\_dir,"YFT.DEL.Comp.20002022.csv"),row.names = FALSE)