

NIGFS: Irish Sea Bottom Trawl survey data processing summary

fishglob, Aurore A. Maureaud & Julianio Palacios Abrantes & P.D van Denderen

December, 2022

Contents

General info	1
Data cleaning in R	1
1. Overview of the survey data table	20
2. Summary of sampling intensity	21
3. Summary of sampling variables from the survey	22
4. Summary of biological variables	23
5. Extreme values	24
6. Summary of variables against swept area	25
7. Abundance or Weight trends of the six most abundant species	26
8. Distribution mapping	27
9. Taxonomic flagging	27
10. Spatio-temporal standardization	28
a. Standardization method 1	28
b. Standardization method 2	32
c. Standardization summary	32

General info

This document presents the cleaning code and summary of the Irish Sea bottom trawl survey provided by DATRAS, ICES. It contains data from **2005** and up to **2020**.

Data cleaning in R

```
#####  
#### R code to download and clean DATRAS data from ICES  
#### URL: https://datras.ices.dk/  
#### Coding: Aurore Maureaud + Julianio Palacios + Daniel van Denderen, December 2022  
#####  
rm(list=ls())  
  
date <- "14DEC2022"  
  
#####  
#### LOAD LIBRARIES & options to decide in the code  
#####  
library(data.table)  
library(tidyverse)  
library(icesDatras)  
library(worrms)  
library(curl)
```

```

library(urltools)
library(here) # for easy work around on multiple computers
library(taxize) # for getting correct species names
library(googledrive)

# load relevant functions
source("functions/write_clean_data.r")
source("functions/clean_taxa.R")
source("functions/get_length_weight_coeffs.R")
fishglob_data_columns <- read_excel("standard_formats/fishglob_data_columns.xlsx")

# should the last version of DATRAS be downloaded?
download_last_version <- FALSE

# should the data be loaded from the last version saved?
load_stored_datras <- FALSE

# should we save a new version of hh and hl?
save_hh_and_hl <- FALSE

# should we get the length-weight relationships from fishbase?
need_get_lw_rel <- FALSE

#####
#### LOAD FILES
#####

if(download_last_version == TRUE){
  last.year <- 2020

  # Haul info from Datras
  hh.ns <- getDATRAS(record='HH', survey='NS-IBTS', years=c(1967:last.year),
                    quarters=c(1,3))
  hh.baltic <- getDATRAS(record='HH', survey='BITS', years=c(1991:last.year),
                        quarters=c(1,4))
  hh.evhoe <- getDATRAS(record='HH', survey='EVHOE', years=c(1997:last.year),
                       quarters=4)
  hh.cgfs <- getDATRAS(record='HH', survey='FR-CGFS', years=c(1998:last.year),
                      quarters=4)
  hh.igfs <- getDATRAS(record='HH', survey='IE-IGFS', years=c(2003:last.year),
                      quarters=4)
  hh.nigfs <- getDATRAS(record='HH', survey='NIGFS', years=c(2005:last.year),
                       quarters=c(1:4))
  hh.pt <- getDATRAS(record='HH', survey='PT-IBTS', years=c(2002:last.year),
                    quarters=c(3:4))
  hh.rock <- getDATRAS(record='HH', survey='ROCKALL', years=c(1999:2009),
                      quarters=3)
  hh.scorock <- getDATRAS(record='HH', survey='SCOROC', years=c(2011:last.year),
                         quarters=3)
  hh.swc <- getDATRAS(record='HH', survey='SWC-IBTS', years=c(1985:2010),
                     quarters=c(1:4))
  hh.scowcgfs <- getDATRAS(record='HH', survey='SCOWCGFS', years=c(2011:last.year),

```

```

quarters=c(1:4))

write.csv(hh.ns, file = "E:/fishglob data/Publicly available/DATRAS/hh.ns.csv",
          row.names = F)
write.csv(hh.baltic, file = "E:/fishglob data/Publicly available/DATRAS/hh.baltic.csv",
          row.names = F)
write.csv(hh.evhoe, file = "E:/fishglob data/Publicly available/DATRAS/hh.evhoe.csv",
          row.names = F)
write.csv(hh.cgfs, file = "E:/fishglob data/Publicly available/DATRAS/hh.cgfs.csv",
          row.names = F)
write.csv(hh.igfs, file = "E:/fishglob data/Publicly available/DATRAS/hh.igfs.csv",
          row.names = F)
write.csv(hh.nigfs, file = "E:/fishglob data/Publicly available/DATRAS/hh.nigfs.csv",
          row.names = F)
write.csv(hh.pt, file = "E:/fishglob data/Publicly available/DATRAS/hh.pt.csv",
          row.names = F)
write.csv(hh.rock, file = "E:/fishglob data/Publicly available/DATRAS/hh.rock.csv",
          row.names = F)
write.csv(hh.scorock, file = "E:/fishglob data/Publicly available/DATRAS/hh.scorock.csv",
          row.names = F)
write.csv(hh.swc, file = "E:/fishglob data/Publicly available/DATRAS/hh.swc.csv",
          row.names = F)
write.csv(hh.scowcgfs, file = "E:/fishglob data/Publicly available/DATRAS/
          hh.scowcgfs.csv", row.names = F)

# Length info from DATRAS
hl.ns <- getDATRAS(record='HL', survey='NS-IBTS', years=c(1967:last.year),
                  quarters=c(1,3))
hl.baltic <- getDATRAS(record='HL', survey='BITS', years=c(1991:last.year),
                    quarters=c(1,4))
hl.evhoe <- getDATRAS(record='HL', survey='EVHOE', years=c(1997:last.year),
                    quarters=4)
hl.cgfs <- getDATRAS(record='HL', survey='FR-CGFS', years=c(1998:last.year),
                    quarters=4)
hl.igfs <- getDATRAS(record='HL', survey='IE-IGFS', years=c(2003:last.year),
                    quarters=4)
hl.nigfs <- getDATRAS(record='HL', survey='NIGFS', years=c(2005:last.year),
                    quarters=c(1:4))
hl.pt <- getDATRAS(record='HL', survey='PT-IBTS', years=c(2002:last.year),
                  quarters=c(3:4))
hl.rock <- getDATRAS(record='HL', survey='ROCKALL', years=c(1999:2009),
                    quarters=3)
hl.scorock <- getDATRAS(record='HL', survey='SCOROC', years=c(2011:last.year),
                      quarters=3)
hl.swc <- getDATRAS(record='HL', survey='SWC-IBTS', years=c(1985:2010),
                    quarters=c(1:4))
hl.scowcgfs <- getDATRAS(record='HL', survey='SCOWCGFS', years=c(2011:last.year),
                        quarters=c(1:4))

write.csv(hl.ns, file = "E:/fishglob data/Publicly available/DATRAS/hl.ns.csv",
          row.names = F)
write.csv(hl.baltic, file = "E:/fishglob data/Publicly available/DATRAS/hl.baltic.csv",

```

```

        row.names = F)
write.csv(hl.evhoe, file = "E:/fishglob data/Publicly available/DATRAS/hl.evhoe.csv",
        row.names = F)
write.csv(hl.cgfs, file = "E:/fishglob data/Publicly available/DATRAS/hl.cgfs.csv",
        row.names = F)
write.csv(hl.igfs, file = "E:/fishglob data/Publicly available/DATRAS/hl.igfs.csv",
        row.names = F)
write.csv(hl.nigfs, file = "E:/fishglob data/Publicly available/DATRAS/hl.nigfs.csv",
        row.names = F)
#write.csv(hl.pt, file = "E:/fishglob data/Publicly available/DATRAS/hl.pt.csv",
#row.names = F)
write.csv(hl.rock, file = "E:/fishglob data/Publicly available/DATRAS/hl.rock.csv",
        row.names = F)
write.csv(hl.scorock, file = "E:/fishglob data/Publicly available/DATRAS/hl.scorock.csv",
        row.names = F)
#write.csv(hl.swc, file = "E:/fishglob data/Publicly available/DATRAS/hl.swc.csv",
#row.names = F)
write.csv(hl.scowcgfs, file = "E:/fishglob data/Publicly available/DATRAS/hl.scowcgfs.csv",
        row.names = F)
}

if(load_stored_datras == TRUE){

hh.ns <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.ns.csv")
hh.baltic <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.baltic.csv")
hh.evhoe <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.evhoe.csv")
hh.cgfs <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.cgfs.csv")
hh.igfs <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.igfs.csv")
hh.nigfs <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.nigfs.csv")
hh.pt <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.pt.csv")
hh.rock <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.rock.csv")
hh.scorock <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.scorock.csv")
hh.swc <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.swc.csv")
hh.scowcgfs <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.scowcgfs.csv")

hl.ns <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.ns.csv")
hl.baltic <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.baltic.csv")
hl.evhoe <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.evhoe.csv")
hl.cgfs <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.cgfs.csv")
hl.igfs <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.igfs.csv")
hl.nigfs <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.nigfs.csv")
hl.pt <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.pt.csv") %>%
  dplyr::rename(Valid_Aphia = ValidAphiaID) %>%
  select(RecordType, Survey, Quarter, Country, Ship, Gear, SweepLngt, GearEx,
        DoorType, StNo, HaulNo, Year, SpecCodeType, SpecCode, SpecVal, Sex,
        TotalNo, CatIdentifier, NoMeas, SubFactor, SubWgt, CatCatchWgt, LngtCode,
        LngtClass, HLNoAtLngt, DevStage, LenMeasType, DateofCalculation,
        Valid_Aphia)
hl.rock <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.rock.csv")
hl.scorock <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.scorock.csv")
hl.swc <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.swc.csv") %>%
  dplyr::rename(Valid_Aphia = ValidAphiaID) %>%

```

```

    select(RecordType, Survey, Quarter, Country, Ship, Gear, SweepLngt, GearEx,
           DoorType, StNo, HaulNo, Year, SpecCodeType, SpecCode, SpecVal, Sex,
           TotalNo, CatIdentifier, NoMeas, SubFactor, SubWgt, CatCatchWgt, LngtCode,
           LngtClass, HLNoAtLngt, DevStage, LenMeasType, DateofCalculation,
           Valid_Aphia)
hl.scowcgfs <- read.csv("E:/fishglob data/Publicly available/DATRAS/hl.scowcgfs.csv")

hh <- rbind(hh.ns, hh.baltic, hh.evhoe, hh.cgfs, hh.igfs, hh.nigfs, hh.pt, hh.rock,
           hh.scorock, hh.swc, hh.scowcgfs)

hl <- rbind(hl.ns, hl.baltic, hl.evhoe, hl.cgfs, hl.igfs, hl.nigfs, hl.pt, hl.rock,
           hl.scorock, hl.swc, hl.scowcgfs)

rm(hl.ns, hl.baltic, hl.evhoe, hl.cgfs, hl.igfs, hl.nigfs, hl.pt, hl.rock, hl.scorock,
   hl.swc, hl.scowcgfs,
   hh.ns, hh.baltic, hh.evhoe, hh.cgfs, hh.igfs, hh.nigfs, hh.pt, hh.rock, hh.scorock,
   hh.swc, hh.scowcgfs)
#rm(ca.ns, ca.baltic, ca.evhoe, ca.cgfs, ca.igfs, ca.nigfs, ca.pt, ca.rock,
# ca.scorock, ca.swc, ca.scowcgfs)

}

if(save_hh_and_hl == TRUE){
  save(hh, file = paste0("E:/fishglob data/Publicly available/DATRAS/hh.",date,".RData"))
  save(hl, file = paste0("E:/fishglob data/Publicly available/DATRAS/hl.",date,".RData"))
}

####----- ###
# Alternative
####----- ###

# Juliano
hl <- fread("/Volumes/Enterprise/Data/FishGlob/Data/Raw/ices_hl.csv")
unique(hl$Survey)
hh <- fread("/Volumes/Enterprise/Data/FishGlob/Data/Raw/ices_hh.csv")
unique(hh$Survey)

# Aurere
load("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hl.29JUL2021.RData")
load("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hh.29JUL2021.RData")

#####
#### CREATE A UNIQUE HAUL ID
#####
hl$HaulID <- paste(hl$Survey, hl$Year,hl$Quarter, hl$Country, hl$Ship, hl$Gear, hl$StNo,
                  hl$HaulNo)
hh$HaulID <- paste(hh$Survey, hh$Year,hh$Quarter, hh$Country, hh$Ship, hh$Gear, hh$StNo,
                  hh$HaulNo)

# Is the HaulID unique?
hhn <- unique(hh$HaulID)

```

```

length(hhn)==nrow(hh)

# check which one is not
pb <- c()
for (i in 1:length(hhn)){
  j <- which(hh$HaulID==hhn[i])
  if(length(j)>1){pb <- hhn[i]}
}

# problem with one haul in NS-IBTS
hh <- hh %>%
  filter(HaulID!=pb)
hl <- hl %>%
  filter(HaulID!=pb)

# Only keep hauls where there is the length composition.
# 70273 hauls in hh and 70273 in hl
hh <- subset(hh, hh$HaulID %in% hl$HaulID)
hl <- subset(hl, hl$HaulID %in% hh$HaulID)

#####
#### MERGE HH and HL FILES
#####

haulidhl <- sort(unique(hl$HaulID))
haulidhh <- sort(unique(hh$HaulID))
identical(haulidhh, haulidhl)
rm(haulidhh, haulidhl)

# remove some columns in hl
hl$SweepLngt <- hl$SpecCodeType <- hl$SpecCode <- hl$Sex <- hl$DateofCalculation <- NULL
hl$RecordType <- hl$GearEx <- NULL

# remove some columns in hh
hh$DateofCalculation <- hh$ThClineDepth <- hh$ThermoCline <- hh$SwellHeight <- NULL
hh$SwellDir <- hh$WindSpeed <- hh$WindDir <- hh$BotCurSpeed <- NULL
hh$BotCurDir <- hh$SurCurSpeed <- hh$SurCurDir <- hh$SpeedWater <- hh$TowDir <- NULL
hh$WgtGroundRope <- hh$KiteDim <- hh$Buoyancy <- hh$Tickler <- NULL
hh$DoorWgt <- hh$DoorSurface <- hh$WarpDen <- hh$Warpdia <- hh$Warplngt <- NULL
hh$Rigging <- hh$HydroStNo <- hh$HaulLat <- hh$HaulLong <- hh$DayNight <- NULL
hh$Stratum <- hh$TimeShot <- hh$Day <- hh$RecordType <- hh$GearExp <- hh$DoorType <- NULL

#survey <- merge(hh, hl, by='HaulID', all.x=FALSE, all.y=TRUE)
survey <- right_join(hh, hl, by=c('HaulID','Survey','Quarter','Country','Ship',
                                'Gear','StNo','HaulNo','Year'))

nrow(survey)==nrow(hl)

survey <- survey %>%
  dplyr::rename(SBT = BotTemp,
                SST = SurTemp,
                Speed = GroundSpeed,
                AphiaID = Valid_Aphia)

```

```

#### Check if the HaulID is unique
#### Not the case for the baltic sea, a lot of duplicates!!!
#ids <- unique(hh$HaulID)
#pb <- vector()
# for(i in 1:length(ids)){
#   x <- which(hh$HaulID==ids[i])
#   if(length(x)>1){pb[length(pb)+1] <- ids[i]}
# }
# print(pb) # dim 0 ok!

#####
#### REMOVE INVALID DATA
#####
survey <- survey %>%
  filter(HaulVal %in% 'V', #Remove invalid hauls
         !is.na(AphiaID), # Remove invalid species records
         SpecVal %in% c(1,10,4,7),
         DataType %in% c('S','R','C'))

#####
#### RESCALE DATA INTO ABUNDANCE FOR THE HAUL DURATION AND ABUNDANCE AT LENGTH
#####
# If Data Type=='C', abundance at length already readjusted with time so get back the
# abundance for the actual duration of the haul.
# If data type=='R', abundance at length is multiplied by subfactor and adjusted to time
survey$CatCatchWgt = as.numeric(survey$CatCatchWgt)

survey <- survey %>%
  mutate(HLNoAtLngt = case_when(DataType=='C' ~ HLNoAtLngt*SubFactor*HaulDur/60,
                                DataType %in% c('S','R') ~ HLNoAtLngt*SubFactor),
         TotalNo = case_when(DataType=='C' ~ TotalNo*HaulDur/60,
                              DataType %in% c('S','R') ~ TotalNo),
         CatCatchWgt = case_when(DataType=='C' ~ CatCatchWgt*HaulDur/60,
                                  DataType %in% c('S','R') ~ CatCatchWgt)) %>%
  select(-HaulVal, -DataType, -StdSpecRecCode, -SpecVal, -SubWgt, -SubFactor) %>%
  mutate(Survey = if_else(Survey=='SCOWCGFS', 'SWC-IBTS', Survey)) %>%
  mutate(Survey = if_else(Survey=='SCOROC', 'ROCKALL', Survey)) %>%
  filter(!(Survey=="NS-IBTS" & BySpecRecCode %in% c(0,2,3,4,5)),
         # remove hauls where not all species are recorded
         !(Survey=="BITS" & BySpecRecCode==0))

#####
#### GET THE SWEEPED AREA in km2
#####

source('cleaning.codes/source_DATRAS_wing_doorspread.R')

#####

```

```

#### GET CPUEs AND RIGHT COLUMNS NAMES
#####

# Remove data without length composition or negative values
xx <- subset(survey, HLNAtLngt<0 | is.na(LngtClass))
no_length_hauls <- sort(unique(xx$HaulID)) # 10,178 hauls without length data

# Only keep abundances/weight
survey <- survey %>%
  #filter(!(HaulID %in% no_length_hauls)) %>% # remove hauls without length data
  mutate(numcpue = TotalNo/Area.swept, # abundance/km2
         wtcpue = CatCatchWgt/(Area.swept*1000), #weight in kg/km2
         numh = (TotalNo*60)/HaulDur, # abundance/hour
         wgt = CatCatchWgt*60/(HaulDur*1000), #weight in kg/h
         num = TotalNo, #raw number of individuals
         wgt = CatCatchWgt/1000, # raw weight in kg
         numlencpue = HLNAtLngt/Area.swept, #abundance/km2 per length class
         numlenh = HLNAtLngt*60/HaulDur, #abundance/h per length class
         Season = 'NA',
         Depth = replace(Depth, Depth<0, NA),
         SBT = replace(SBT, SBT<0, NA),
         SST = replace(SST, SST<0, NA),
         LngtClass = ifelse(LngtCode %in% c('.', '0'), LngtClass*0.1, LngtClass)) %>%
  # fix unit of length class
  dplyr::rename(Length = LngtClass) %>%
  select(Survey, HaulID, StatRec, Year, Month, Quarter, Season, ShootLat, ShootLong,
         HaulDur, Area.swept, Gear, Depth, SBT, SST, AphiaID, CatIdentifier, numcpue,
         wtcpue, numh, wgt, num, wgt, Length, numlencpue, numlenh)
survey <- data.frame(survey)

#####
## fishglob taxa cleaning ##
#####

# Make AphiaID list per survey
aphia_dattras <- survey %>%
  select(Survey, AphiaID) %>%
  dplyr::rename(survey = Survey,
                worms_id_dattras = AphiaID) %>%
  distinct()

# Clean taxa north sea
ns_data <- aphia_dattras %>% filter(survey=="NS-IBTS")
clean_ns <- clean_taxa(ns_data$worms_id_dattras, input_survey = "NS-IBTS",
                      save=F, fishbase=TRUE)

# Clean taxa bay of biscay
evhoe_data <- aphia_dattras %>% filter(survey=="EVHOE")
clean_evhoe <- clean_taxa(evhoe_data$worms_id_dattras, input_survey = "EVHOE",
                          save=F, fishbase=TRUE)

# Clean taxa english channel

```



```

cgfs_data <- aphia_datras %>% filter(survey=="FR-CGFS")
clean_cgfs <- clean_taxa(cgfs_data$worms_id_datras, input_survey = "FR-CGFS",
                        save=F, fishbase=TRUE)

# Clean taxa baltic sea
bits_data <- aphia_datras %>% filter(survey=="BITS")
clean_bits <- clean_taxa(bits_data$worms_id_datras, input_survey = "BITS",
                        save=F, fishbase=TRUE)

# Clean taxa scottish sea
swc_data <- aphia_datras %>% filter(survey %in% c("SCOWCGFS","SWC-IBTS"))
clean_swc <- clean_taxa(swc_data$worms_id_datras, input_survey = "SWC-IBTS",
                        save=F, fishbase=TRUE)

# Clean taxa rockall
rock_data <- aphia_datras %>% filter(survey %in% c("SCOROC","ROCKALL"))
clean_rock <- clean_taxa(rock_data$worms_id_datras, input_survey = "ROCKALL",
                        save=F, fishbase=TRUE)

# Clean taxa irish sea
ir_data <- aphia_datras %>% filter(survey=="IE-IGFS")
clean_ir <- clean_taxa(ir_data$worms_id_datras, input_survey = "IE-IGFS",
                        save=F, fishbase=TRUE)

# Clean taxa northern ireland
nigfs_data <- aphia_datras %>% filter(survey=="NIGFS")
clean_nigfs <- clean_taxa(nigfs_data$worms_id_datras, input_survey = "NIGFS",
                        save=F, fishbase=TRUE)

# Clean taxa for portugal
pt_data <- aphia_datras %>% filter(survey=="PT-IBTS")
clean_pt <- clean_taxa(pt_data$worms_id_datras, input_survey = "PT-IBTS",
                        save=F, fishbase=TRUE)

clean_datras_taxa <- rbind(clean_bits, clean_cgfs, clean_evhoe, clean_ir, clean_nigfs,
                          clean_pt, clean_rock, clean_swc, clean_ns) %>%
  mutate(query = as.numeric(as.vector(query))) %>%
  distinct()

recoded_taxa <- c("Dipturus","Liparis","Chelon","Mustelus","Alosa","Argentina",
                  "Callionymus","Ciliata","Gaidropsarus","Sebastes","Syngnatus",
                  "Pomatoschistus","Gobius")

spp_to_recode <-c("Dipturus batis","Dipturus flossada","Dipturus batis-complex",
                  "Dipturus intermedia","Liparis montagui","Liparis liparis",
                  "Liparis liparis liparis","Chelon aurata","Chelon ramada",
                  "Mustelus mustelus/asterias","Mustelus mustelus","Mustelus asterias",
                  "Alosa alosa","Alosa fallax","Argentina silus","Argentina sphyraena",
                  "Callionymus reticulatus","Callionymus maculatus","Ciliata mustela",
                  "Ciliata septentrionalis","Gaidropsaurus macrophthalmus",
                  "Gaidropsaurus mediterraneus","Gaidropsaurus vulgaris",
                  "Sebastes norvegicus", "Sebastes mentella","Sebastes marinus",
                  "Syngnathus rostellatus", "Syngnathus acus","Syngnathus typhle",

```

```

      "Nerophis ophidion", "Pomatoschistus microps", "Pomatoschistus minutus",
      "Pomatoschistus pictus", "Gobius cobitis", "Gobius niger",
      "Leusueurigobius friesii", "Neogobius melanostomus")

alphaid <- get_wormsid(recoded_taxa)
alphaid <- tibble(taxa = recoded_taxa,
                  worms_id = alphaid[1:length(recoded_taxa)])
clean_manual_recoded <- clean_taxa(alphaid$worms_id, input_survey = "recoded",
                                   save = F, fishbase=TRUE)

clean_datras_taxa <- clean_datras_taxa %>%
  select(-survey) %>%
  mutate(SpecCode = ifelse(taxa %in% spp_to_recode, NA, SpecCode),
         rank = ifelse(taxa %in% spp_to_recode, "Genus", rank),
         #dipturus
         worms_id = ifelse(taxa %in% c("Dipturus batis", "Dipturus flossada",
                                       "Dipturus batis-complex", "Dipturus intermedia"),
                           105762, worms_id),
         taxa = ifelse(taxa %in% c("Dipturus batis", "Dipturus flossada",
                                   "Dipturus batis-complex", "Dipturus intermedia"),
                       "Dipturus", taxa),
         # liparis
         worms_id = ifelse(taxa %in% c("Liparis montagui", "Liparis liparis",
                                       "Liparis liparis liparis"), 126160, worms_id),
         taxa = ifelse(taxa %in% c("Liparis montagui", "Liparis liparis",
                                   "Liparis liparis liparis"), "Liparis", taxa),
         # chelon
         worms_id = ifelse(taxa %in% c("Chelon aurata", "Chelon ramada"), 126030, worms_id),
         taxa = ifelse(taxa %in% c("Chelon aurata", "Chelon ramada"), "Chelon", taxa),
         #mustelus
         worms_id = ifelse(taxa %in% c("Mustelus mustelus/asterias", "Mustelus mustelus",
                                       "Mustelus asterias"), 105732, worms_id),
         taxa = ifelse(taxa %in% c("Mustelus mustelus/asterias", "Mustelus mustelus",
                                   "Mustelus asterias"), "Mustelus", taxa),
         #alosa
         worms_id = ifelse(taxa %in% c("Alosa alosa", "Alosa fallax"), 125715, worms_id),
         taxa = ifelse(taxa %in% c("Alosa alosa", "Alosa fallax"), "Alosa", taxa),
         #argentina
         worms_id = ifelse(taxa %in% c("Argentina silus", "Argentina sphyraena"),
                           125885, worms_id),
         taxa = ifelse(taxa %in% c("Argentina silus", "Argentina sphyraena"),
                       "Argentina", taxa),
         # callionymus
         worms_id = ifelse(taxa %in% c("Callionymus reticulatus", "Callionymus maculatus"),
                           125930, worms_id),
         taxa = ifelse(taxa %in% c("Callionymus reticulatus", "Callionymus maculatus"),
                       "Callionymus", taxa),
         # ciliata
         worms_id = ifelse(taxa %in% c("Ciliata mustela", "Ciliata septentrionalis"),
                           125741, worms_id),
         taxa = ifelse(taxa %in% c("Ciliata mustela", "Ciliata septentrionalis"),
                       "Ciliata", taxa),
         # gaidropsarus

```

```

worms_id = ifelse(taxa %in% c("Gaidropsaurus macrophthalmus",
                             "Gaidropsaurus mediterraneus",
                             "Gaidropsaurus vulgaris"),
                  125743,worms_id),
taxa = ifelse(taxa %in% c("Gaidropsaurus macrophthalmus",
                          "Gaidropsaurus mediterraneus",
                          "Gaidropsaurus vulgaris"),"Gaidropsarus",taxa),

# sebastes
worms_id = ifelse(taxa %in% c("Sebastes norvegicus","Sebastes mentella",
                              "Sebastes marinus"),
                  126175,worms_id),
taxa = ifelse(taxa %in% c("Sebastes norvegicus","Sebastes mentella",
                          "Sebastes marinus"),
              "Sebastes",taxa),

# syngnathus
worms_id = ifelse(taxa %in% c("Syngnathus rostellatus","Syngnathus acus",
                              "Syngnathus typhle","Nerophis ophidion"),
                  126227,worms_id),
taxa = ifelse(taxa %in% c("Syngnathus rostellatus","Syngnathus acus",
                          "Syngnathus typhle","Nerophis ophidion"),
              "Syngnathus",taxa),

# pomatosc
worms_id = ifelse(taxa %in% c("Pomatoschistus microps","Pomatoschistus minutus",
                              "Pomatoschistus pictus"),125999,worms_id),
taxa = ifelse(taxa %in% c("Pomatoschistus microps","Pomatoschistus minutus",
                          "Pomatoschistus pictus"),"Pomatoschistus",taxa),

# gobioid
worms_id = ifelse(taxa %in% c("Gobius cobitis","Gobius niger",
                              "Leusueurigobius friesii",
                              "Neogobius melanostomus"),125988,worms_id),
taxa = ifelse(taxa %in% c("Gobius cobitis","Gobius niger",
                          "Leusueurigobius friesii",
                          "Neogobius melanostomus"),"Gobius",taxa),

) %>%
distinct()

# add taxonomy to data
survey <- left_join(survey, clean_datras_taxa, by=c("AphiaID" = "query")) %>%
  filter(!is.na(AphiaID))

#####
#### RE-CALCULATE WEIGHTS
#####
detach(package:worms)
detach(package:plyr)

# select only certain gears
# 1. summary of gears per survey
gears <- data.frame(survey) %>%
  group_by(Survey, Gear) %>%
  summarise(hauls = length(unique(HaulID)), years = length(unique(Year))) %>%
  select(Survey, Gear, hauls, years)

```

```

# 2. only select certain gears per survey (GOV and/or most dominant in cases without GOV)
survey <- survey %>%
  filter(!(Survey=="NS-IBTS" & Gear %in% c('ABD', 'BOT', 'DHT', 'FOT', 'GRT',
                                           'H18', 'HOB', 'HT', 'KAB', 'VIN'))),
         !(Survey=="BITS" & Gear %in% c('CAM', 'CHP', 'DT', 'EGY', 'ESB', 'EXP',
                                           'FOT', 'GRT', 'H2O', 'HAK', 'LBT', 'SON'))),
         !(Survey=="PT-IBTS" & Gear=="CAR"),
         !(Survey=="Can-Mar" & Gear=="Y36'))

# 3. associate an LME to each haul and make final list of species
### Prepare list for estimating length-weight parameters
list.taxa <- survey %>%
  mutate(species = str_split(taxa, pattern = " ", simplify = TRUE)[,2]) %>%
  select(HaulID, Survey, ShootLat, ShootLong, family, genus, species) %>%
  distinct()

# get LME
library(rgdal)
shape1 <- readOGR(dsn = "length.weight/LME66", layer="LMEs66")
coords <- list.taxa %>%
  dplyr::select(ShootLat, ShootLong, Survey) %>%
  distinct()
str(coords)

coordinates(coords) <- ~ ShootLong + ShootLat
proj4string(coords) <- proj4string(shape1)
lme <- over(coords, shape1)

coords <- list.taxa %>%
  dplyr::select(ShootLat, ShootLong, Survey) %>%
  distinct()
coords <- cbind(coords, lme$LME_NUMBER)
setnames(coords, old='lme$LME_NUMBER', new='lme')

coords$lme <- as.factor(coords$lme)
#Select from each LME 50 long and lat
ind <- c()
for (i in 1:nlevels(coords$lme)){
  ind <- c(ind, sample(which(coords$lme==levels(coords$lme)[i]), 50,
                       replace = FALSE))
}
long50 <- coords$ShootLong[ind]
lat50 <- coords$ShootLat[ind]
lme50 <- rep(levels(coords$lme), each=50)
#For each haul without LME find a close LME that has an LME number already
nlme <- subset(coords, is.na(lme)) # many hauls without LME 710
nlme$ShootLat <- as.numeric(as.vector(nlme$ShootLat))
nlme$ShootLong <- as.numeric(as.vector(nlme$ShootLong))
long50 <- as.numeric(as.vector(long50))
lat50 <- as.numeric(as.vector(lat50))
dilme <- c()
for (i in 1:length(lme50)){

```

```

  dilme <- cbind(dilme, (nlme$ShootLat-lat50[i])**2 + (nlme$ShootLong-long50[i])**2)
}
mindl <- apply(dilme, 1, which.min)
coords$lme[is.na(coords$lme)] <- lme50[mindl]
# assign the closest LME number to each haul without LME

#Check
coords$ShootLat <- as.numeric(as.vector(coords$ShootLat))
coords$ShootLong <- as.numeric(as.vector(coords$ShootLong))

# rockall not assigned to Faroe plateau but to celtic sea LME
coords$lme <- as.character(coords$lme)
coords <- coords %>%
  mutate(lme = replace(lme, Survey == 'ROCKALL', '60')) %>%
  as.data.frame()

survey <- left_join(survey, coords, by=c('ShootLat', 'ShootLong', 'Survey'))

if(need_get_lw_rel == TRUE){
  list.taxa <- survey %>%
    select(taxa, family, genus, taxa, lme, rank, Survey) %>%
    filter(!is.na(taxa),
           rank!="Suborder") %>%
    rename(survey = Survey) %>%
    distinct()

  write.csv(data.frame(list.taxa), file="length.weight/taxa.DATRAS.FB.tofill.csv",
            row.names=FALSE)
  save.image("length.weight/DATRAS_before_lw_4AUG2021.RData")

  ### run length-weight relationships open R 32bit
  list.taxa <- read.csv("length.weight/taxa.DATRAS.FB.tofill.csv")
  get_coeffs(list.taxa, survey="DATRAS", save=TRUE)
}

# 4. re-calculate weights with length-weight relationships
#load("length.weight/DATRAS_before_lw_4AUG2021.RData")
datalw <- read.csv('length.weight/length.weight_DATRAS.csv') %>%
  rename(Survey = survey) %>%
  select(-X) %>%
  mutate(lme = as.character(lme))

survey <- survey %>%
  filter(rank!="Suborder")

# summarize abundance/weight at the haul level
survey.num <- left_join(survey, datalw, by=c("taxa", "family", "genus", "lme",
      "rank", "Survey")) %>%
  select(Survey, HaulID, StatRec, Year, Month, Quarter, Season, ShootLat, ShootLong,
    HaulDur, Area.swept, Gear, Depth, SBT, SST, family, genus, taxa, AphiaID, worms_id,
    SpecCode, kingdom, class, order, phylum, rank,

```

```

      CatIdentifier,numcpue,numh,num) %>%
distinct() %>%
group_by(Survey,HaulID,StatRec,Year,Month,Quarter,Season,ShootLat,ShootLong,
      HaulDur,Area.swept,Gear,Depth,SBT,SST,family,genus,taxa,AphiaID,
      worms_id,SpecCode,kingdom,class,order,phylum,rank) %>%
summarize_at(.vars=c('numcpue','numh','num'), .funs = function(x) sum(x)) %>%
ungroup()

survey.wgt <- left_join(survey, datalw, by=c("taxa","family","genus","lme",
      "rank","Survey")) %>%

select(Survey,HaulID,StatRec,Year,Month,Quarter,Season,ShootLat,ShootLong,HaulDur,
      Area.swept,Gear,Depth,SBT,SST,family,genus,taxa,AphiaID,worms_id,SpecCode,
      kingdom,class,order,phylum,rank,
      CatIdentifier,wtpcue,wgth,wgt) %>%
distinct() %>%
group_by(Survey,HaulID,StatRec,Year,Month,Quarter,Season,ShootLat,ShootLong,
      HaulDur,Area.swept,Gear,Depth,SBT,SST,family,genus,taxa,AphiaID,worms_id,
      SpecCode,kingdom,class,order,phylum,rank) %>%
summarize_at(.vars=c('wtpcue','wgth','wgt'), .funs = function(x) sum(x)) %>%
ungroup()

survey1 <- full_join(survey.num, survey.wgt,
      by=c('Survey','HaulID','StatRec','Year','Month','Quarter',
      'Season','ShootLat','ShootLong','HaulDur','Area.swept',
      'Gear','Depth','SBT','SST','family','genus','taxa','AphiaID',
      'worms_id','SpecCode',
      'kingdom','phylum','class','order','rank'))

# summarize abundance/weight from length data
survey2 <- left_join(survey, datalw, by=c("taxa","family","genus","lme",
      "rank","Survey")) %>%

mutate(wgtlencpue = numlencpue*a*Length^b/1000, # divide by 1000 to get kg/km2
      wgtlenh = numlenh*a*Length^b/1000) %>% # divide by 1000 to get kg/h
group_by(Survey,HaulID,StatRec,Year,Month,Quarter,Season,ShootLat,ShootLong,HaulDur,
      Area.swept,Gear,Depth,SBT,SST,family,genus,taxa,AphiaID,worms_id,SpecCode,a,b,
      kingdom,class,order,phylum,rank) %>%
summarize_at(.vars=c('numlencpue','numlenh','wgtlencpue','wgtlenh'),
      .funs=function(x) sum(x)) %>%
ungroup()

# merge both and compare
nrow(survey1)==nrow(survey2)
survey3 <- full_join(survey1, survey2, by=c('Survey','HaulID','StatRec','Year','Month',
      'Quarter','Season','ShootLat','ShootLong',
      'HaulDur','Area.swept','Gear','Depth',
      'SBT','SST','family','genus','taxa',
      'AphiaID','worms_id','SpecCode',
      'kingdom','phylum','class','order','rank'))

library(ggplot2)

# correlation between abundances to check calculations are right
cor(x = survey3$numh, y = survey3$numlenh, method = 'pearson')

```

```

xx <- subset(survey3, !is.na(numcpue))
cor(x = xx$numcpue, y = xx$numlencpue, method = 'pearson')

# check weights
xx <- subset(survey3, wtcpue > 0 & wgtlencpue > 0)
cor(x = xx$wtcpue, y = xx$wgtlencpue, method = 'pearson')

xx <- subset(survey3, wgth > 0 & wgtlenh > 0)
cor(x = xx$wgth, y = xx$wgtlenh, method = 'pearson')

### cor = 0.92 and 0.90 so something does not work.

#####
# CHECK PER SURVEY
#####

# no zeros
xx <- subset(survey3, wgth > 0 & wgtlenh > 0)

# rockall looks OK
ggplot(subset(xx, Survey=='ROCKALL'), aes(x=wgth, y=wgtlenh)) + geom_point() +
  geom_abline(intercept = 0, slope = 1, color="red",
    linetype="dashed", size=0.5) + scale_x_log10() + scale_y_log10()

# IE-IGFS looks OK
ggplot(subset(xx, Survey=='IE-IGFS'), aes(x=wgth, y=wgtlenh)) + geom_point() +
  geom_abline(intercept = 0, slope = 1, color="red",
    linetype="dashed", size=0.5) + scale_x_log10() + scale_y_log10()

# NIGFS looks OK
ggplot(subset(xx, Survey=='NIGFS'), aes(x=wgth, y=wgtlenh)) + geom_point() +
  geom_abline(intercept = 0, slope = 1, color="red",
    linetype="dashed", size=0.5) + scale_x_log10() + scale_y_log10()

# PT-IBTS looks OK
ggplot(subset(xx, Survey=='PT-IBTS'), aes(x=wgth, y=wgtlenh)) + geom_point() +
  geom_abline(intercept = 0, slope = 1, color="red",
    linetype="dashed", size=0.5) + scale_x_log10() + scale_y_log10()

# FR-CGFS looks OK
ggplot(subset(xx, Survey=='FR-CGFS'), aes(x=wgth, y=wgtlenh)) + geom_point() +
  geom_abline(intercept = 0, slope = 1, color="red",
    linetype="dashed", size=0.5) + scale_x_log10() + scale_y_log10()

# SWC-IBTS issue
ggplot(subset(xx, Survey=='SWC-IBTS'), aes(x=wgth, y=wgtlenh)) + geom_point() +
  geom_abline(intercept = 0, slope = 1, color="red",
    linetype="dashed", size=0.5) + scale_x_log10() + scale_y_log10()

comp <- subset(xx, Survey=='SWC-IBTS') %>%
  select(HaulID, wgtlenh, wgth) %>%
  distinct() %>%
  group_by(HaulID) %>%

```

```

summarize_at(.vars=c('wgtlenh', 'wgth'), .funs = function(x) sum(x)) %>%
ungroup() %>%
as.data.frame()

ggplot(comp, aes(x=wgth, y=wgtlenh)) + geom_point() +
  geom_abline(intercept = 0, slope = 1, color="red",
    linetype="dashed", size=0.5) + scale_x_log10() + scale_y_log10()

comp$factor <- comp$wgtlenh / comp$wgth
plot(comp$factor)
resc <- comp$HaulID[comp$factor > 40]

# after check with original haul length data (HL) for some resc haulid, weight
# is clearly wrong factor 100
survey3 <- survey3 %>%
  mutate(wtcpue = if_else(HaulID %in% resc, wtcpue*100, wtcpue),
    wgth = if_else(HaulID %in% resc, wgth*100, wgth),
    wgt = if_else(HaulID %in% resc, wgt*100, wgt))

# BITS issue
ggplot(subset(xx, Survey=='BITS'), aes(x=wgth, y=wgtlenh)) + geom_point() +
  geom_abline(intercept = 0, slope = 1, color="red",
    linetype="dashed", size=0.5) + scale_x_log10() + scale_y_log10()

comp <- subset(xx, Survey=='BITS') %>%
  select(HaulID, wgtlenh, wgth) %>%
  distinct() %>%
  group_by(HaulID) %>%
  summarize_at(.vars=c('wgtlenh', 'wgth'), .funs = function(x) sum(x)) %>%
  ungroup() %>%
  as.data.frame()

ggplot(comp, aes(x=wgth, y=wgtlenh)) + geom_point() +
  geom_abline(intercept = 0, slope = 1, color="red",
    linetype="dashed", size=0.5) + scale_x_log10() + scale_y_log10()

comp$factor <- comp$wgtlenh / comp$wgth
plot(comp$factor)
resc <- comp$HaulID[comp$factor > 40]

# after check with original haul length data (HL) for some resc haulid, weight
# is clearly wrong factor 100
survey3 <- survey3 %>%
  mutate(wtcpue = if_else(HaulID %in% resc, wtcpue*100, wtcpue),
    wgth = if_else(HaulID %in% resc, wgth*100, wgth),
    wgt = if_else(HaulID %in% resc, wgt*100, wgt))

# EVHOE may have an issue, no changes as not very clear
ggplot(subset(xx, Survey=='EVHOE'), aes(x=wgth, y=wgtlenh)) + geom_point() +
  geom_abline(intercept = 0, slope = 1, color="red",
    linetype="dashed", size=0.5) + scale_x_log10() + scale_y_log10()

comp <- subset(xx, Survey=='EVHOE') %>%

```



```

select(HaulID,wgtlenh,wgth) %>%
distinct() %>%
group_by(HaulID) %>%
summarize_at(.vars=c('wgtlenh', 'wgth'), .funs = function(x) sum(x)) %>%
ungroup() %>%
as.data.frame()

ggplot(comp, aes(x=wgth, y=wgtlenh)) + geom_point() +
  geom_abline(intercept = 0, slope = 1, color="red",
             linetype="dashed", size=0.5) + scale_x_log10() + scale_y_log10()

comp$factor <- comp$wgtlenh / comp$wgth
plot(comp$factor)

# NS - IBTS issue
ggplot(subset(xx, Survey=='NS-IBTS'), aes(x=wgth, y=wgtlenh)) + geom_point() +
  geom_abline(intercept = 0, slope = 1, color="red",
             linetype="dashed", size=0.5) + scale_x_log10() + scale_y_log10()

comp <- subset(xx, Survey=='NS-IBTS') %>%
select(HaulID,wgtlenh,wgth) %>%
distinct() %>%
group_by(HaulID) %>%
summarize_at(.vars=c('wgtlenh', 'wgth'), .funs = function(x) sum(x)) %>%
ungroup() %>%
as.data.frame()

ggplot(comp, aes(x=wgth, y=wgtlenh)) + geom_point() +
  geom_abline(intercept = 0, slope = 1, color="red",
             linetype="dashed", size=0.5) + scale_x_log10() + scale_y_log10()

comp$factor <- comp$wgtlenh / comp$wgth
comp$uni <- c(1:nrow(comp))
plot(comp$factor~comp$uni,ylim=c(0,120))
points(comp$factor[comp$factor > 20]~comp$uni[comp$factor > 20],col="red")
points(comp$factor[comp$factor > 8 & comp$factor <20]~
       comp$uni[comp$factor > 8 & comp$factor <20],col="blue")

# two issues - one estimate 100 times higher based on length, the other 10 times
resc <- comp$HaulID[comp$factor > 20]
resc2 <- comp$HaulID[comp$factor > 8 & comp$factor <20]

# after check with original haul length data (HL) for some resc haulid, weight
# is clearly wrong factor 100
# and also a cluster of factor 10
survey3 <- survey3 %>%
  mutate(wtcpue = if_else(HaulID %in% resc, wtcpue*100,wtcpue),
         wgth = if_else(HaulID %in% resc , wgth*100,wgth),
         wgt = if_else(HaulID %in% resc , wgt*100,wgt))

survey3 <- survey3 %>%
  mutate(wtcpue = if_else(HaulID %in% resc2, wtcpue*10,wtcpue),
         wgth = if_else(HaulID %in% resc2 , wgth*10,wgth),

```

```

    wgt = if_else(HaulID %in% resc2 , wgt*10,wgt))

# check again correlations
xx <- subset(survey3, wtcpue> 0 & wgtlencpue>0)
cor(x = xx$wtcpue , y = xx$wgtlencpue, method = 'pearson') # looks better

xx <- subset(survey3, wgth>0 & wgtlenh>0)
cor(x = xx$wgth, y = xx$wgtlenh, method = 'pearson') # looks better

# now check per haul without zeros, NAs
xx <- subset(survey3, wtcpue>0 & wgtlencpue>0)

comp <- xx %>%
  select(HaulID,wgtlencpue,wtcpue) %>%
  distinct() %>%
  group_by(HaulID) %>%
  summarize_at(.vars=c('wgtlencpue', 'wtcpue'), .funs = function(x) sum(x)) %>%
  ungroup() %>%
  as.data.frame()

ggplot(comp, aes(x=wtcpue, y=wgtlencpue)) + geom_point() +
  geom_abline(intercept = 0, slope = 1, color="red",
             linetype="dashed", size=0.5) + scale_x_log10() + scale_y_log10()

cor(x = xx$wtcpue , y = xx$wgtlencpue, method = 'pearson')
# [1] 0.9635742

#####
#### Fishglob format
#####

survey4 <- survey3 %>%
  rename(survey = Survey,
        haul_id = HaulID,
        stat_rec = StatRec,
        year = Year,
        month = Month,
        quarter = Quarter,
        season = Season,
        latitude = ShootLat,
        longitude = ShootLong,
        haul_dur = HaulDur,
        area_swept = Area.swept,
        gear = Gear,
        depth = Depth,
        sbt = SBT,
        sst = SST,
        verbatim_aphia_id = AphiaID,
        aphia_id = worms_id,
        accepted_name = taxa,
        ) %>%
  mutate(day = NA_integer_,

```

```

    verbatim_name = NA_character_,
    station = NA_character_,
    stratum = NA_character_,
    sub_area = NA_character_,
    continent = "europe",
    country = case_when(survey=="PT-IBTS" ~ "portugal",
                        survey=="EVHOE" ~ "france",
                        survey=="IE-IGFS" ~ "ireland",
                        survey %in% c("ROCKALL","SWC-IBTS","NIGFS") ~ "uk",
                        survey=="FR-CGFS" ~ "france",
                        survey %in% c("NS-IBTS","BITS") ~ "multi-countries"),
    num = numlencpue*area_swept,
    num_h = numlenh,
    num_cpue = numlencpue,
    wgt = wgtlencpue*area_swept,
    wgt_h = wgtlenh,
    wgt_cpue = wgtlencpue,
    haul_dur = haul_dur/60,
    source = "DATRAS ICES",
    timestamp = "2021-07") %>%
# Final format
select(fishglob_data_columns$`Column name fishglob`)

#####
# Save database in Google drive
#####

# Just run this routine should be good for all
write_clean_data(data = survey4, survey = "DATRAS_v3", overwrite = T)

```

1. Overview of the survey data table

survey	source	timestamp	haul_id	country	sub_area	continent	sta
NIGFS	DATRAS ICES	2021-07	NIGFS 2005 1 GB-NIR 74RY ROT 35 1	uk	NA	europe	38
NIGFS	DATRAS ICES	2021-07	NIGFS 2006 1 GB-NIR 74RY ROT 35 1	uk	NA	europe	38
NIGFS	DATRAS ICES	2021-07	NIGFS 2006 4 GB-NIR 74RY ROT 35 1	uk	NA	europe	38
NIGFS	DATRAS ICES	2021-07	NIGFS 2007 1 GB-NIR 74RY ROT 86 1	uk	NA	europe	38
NIGFS	DATRAS ICES	2021-07	NIGFS 2007 4 GB-NIR 74RY ROT 245 1	uk	NA	europe	36

station	stratum	year	month	day	quarter	season
NA	NA	2005	3	NA	1	NA
NA	NA	2006	3	NA	1	NA
NA	NA	2006	10	NA	4	NA
NA	NA	2007	3	NA	1	NA
NA	NA	2007	10	NA	4	NA

latitude	longitude	haul_dur	area_swept	gear	depth
54.7175	-5.6901	0.9833333	0.0904319	ROT	21
54.7168	-5.6908	0.6666667	0.0547375	ROT	22
54.7195	-5.6776	0.3333333	0.0299577	ROT	18
54.5726	-5.4245	1.0000000	0.1004743	ROT	47
53.5150	-4.2383	0.3500000	0.0365526	ROT	43

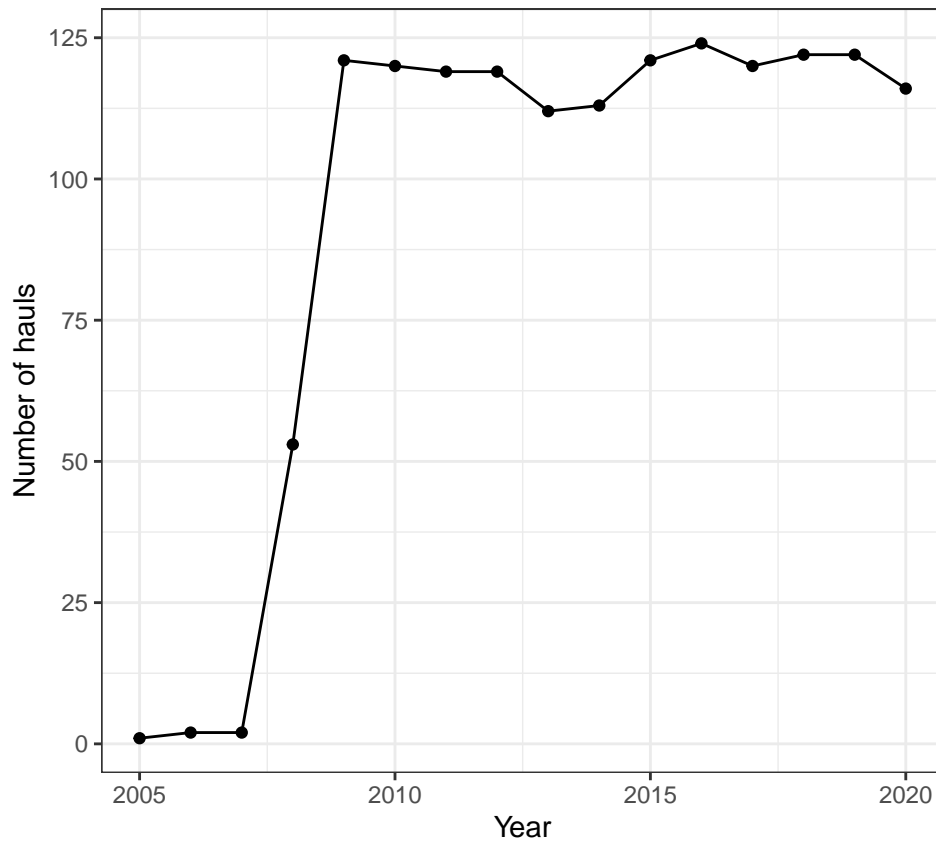
sbt	sst	num	num_h	num_cpue	wgt
NA	NA	1.0000	1.016949	11.058043	0.0300088
NA	NA	2.9194	4.379100	53.334526	0.0521340
NA	NA	1.0000	3.000000	33.380421	1.3815813
NA	NA	1.0000	1.000000	9.952797	0.0148495
NA	NA	1.0000	2.857143	27.357837	0.3029693

wgt_h	wgt_cpue	verbatim_name	verbatim_aphia_id	accepted_name
0.0305175	0.3318391	NA	125909	Ammodytes
0.0782010	0.9524363	NA	126417	Clupea harengus
4.1447438	46.1177648	NA	105814	Scyliorhinus canicula
0.0148495	0.1477942	NA	105814	Scyliorhinus canicula
0.8656267	8.2885855	NA	105814	Scyliorhinus canicula

aphia_id	SpecCode	kingdom	phylum	class	order	family
125909	NA	Animalia	Chordata	Actinopteri	Perciformes	Ammodytidae
126417	24	Animalia	Chordata	Actinopteri	Clupeiformes	Clupeidae
105814	845	Animalia	Chordata	Elasmobranchii	Carcharhiniformes	Scyliorhinidae
105814	845	Animalia	Chordata	Elasmobranchii	Carcharhiniformes	Scyliorhinidae
105814	845	Animalia	Chordata	Elasmobranchii	Carcharhiniformes	Scyliorhinidae

2. Summary of sampling intensity

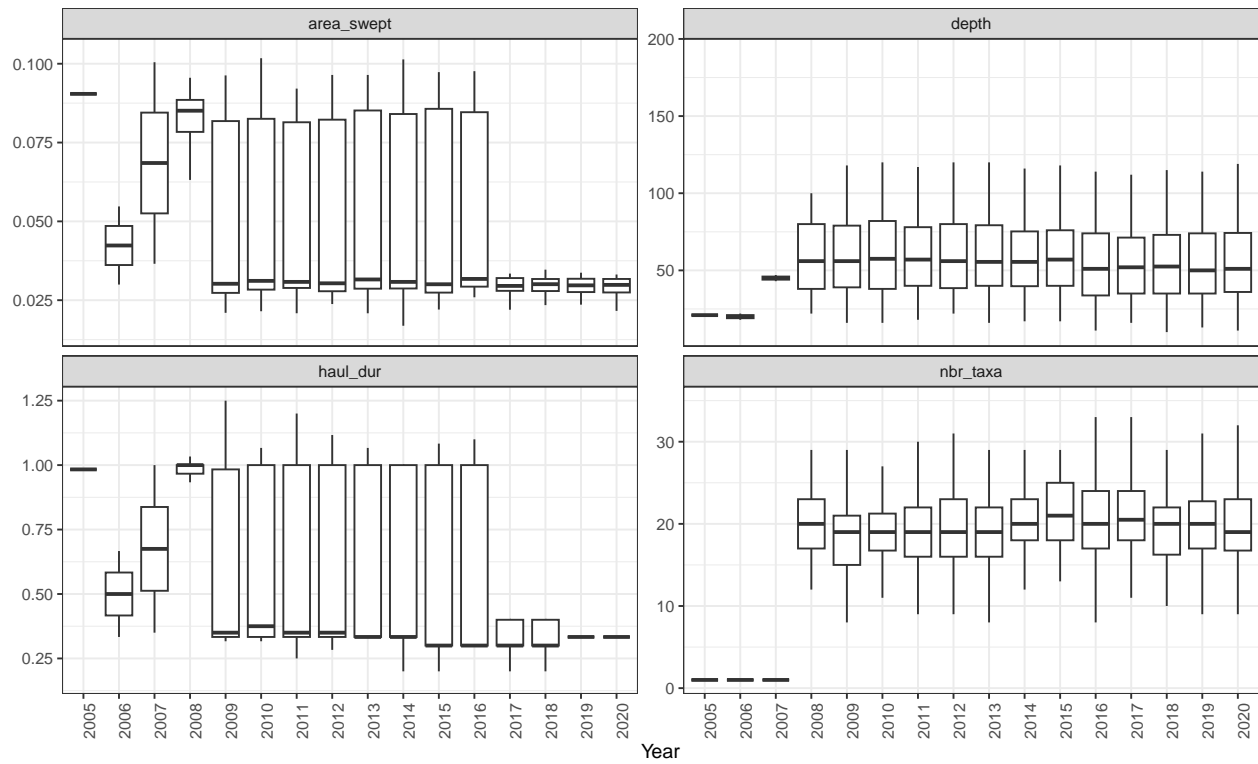
Number of hauls per year performed during the survey after data processing.



3. Summary of sampling variables from the survey

Here we show the yearly total and average of the following variables reported in the survey data:

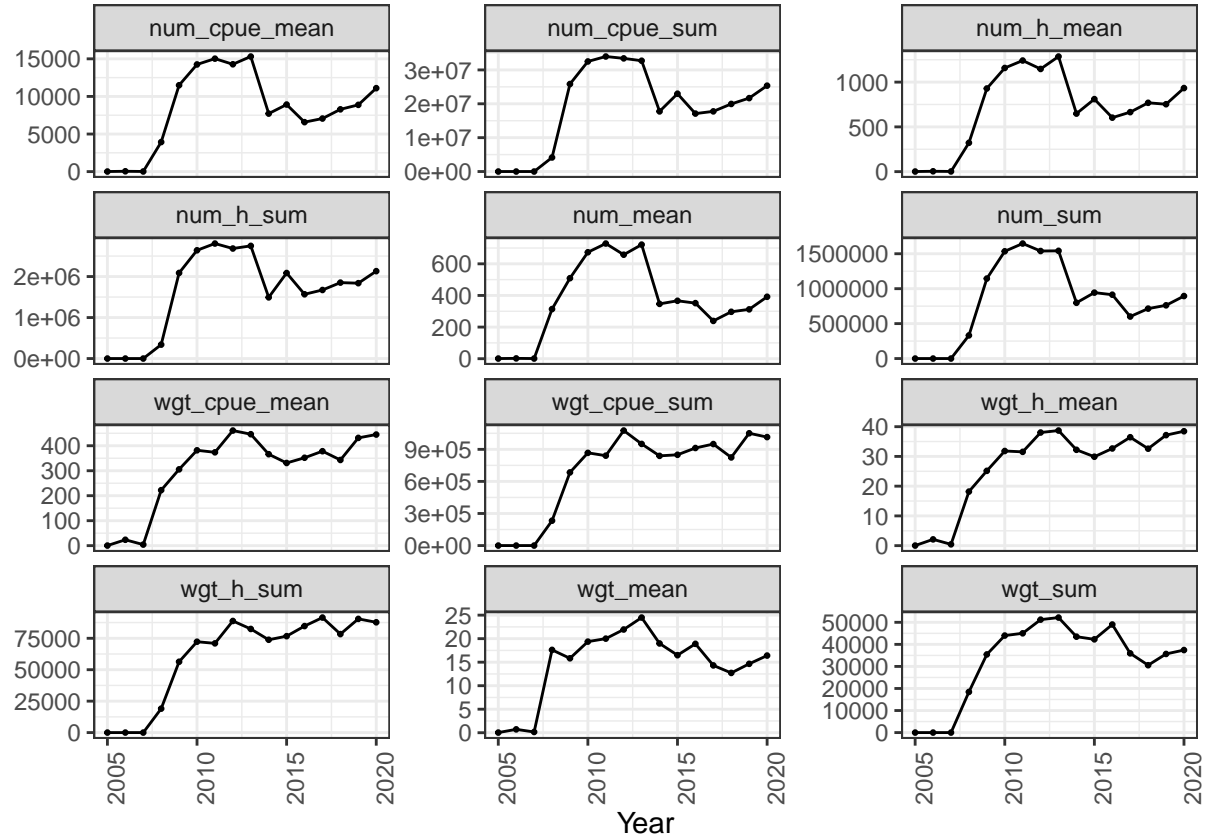
- *area_swept*, swept area by the bottom trawl gear km^2
- *depth*, sampling depth in *m*
- *haul_dur*, haul sampling duration *hours*
- *number of marine fish taxa*, taxa were cleaned following the last version of taxonomy from the World Register of Marine Species (<https://www.marinespecies.org/>, October 2021)



4. Summary of biological variables

Here we display the yearly total and average across hauls of the following variables recorded in the data:

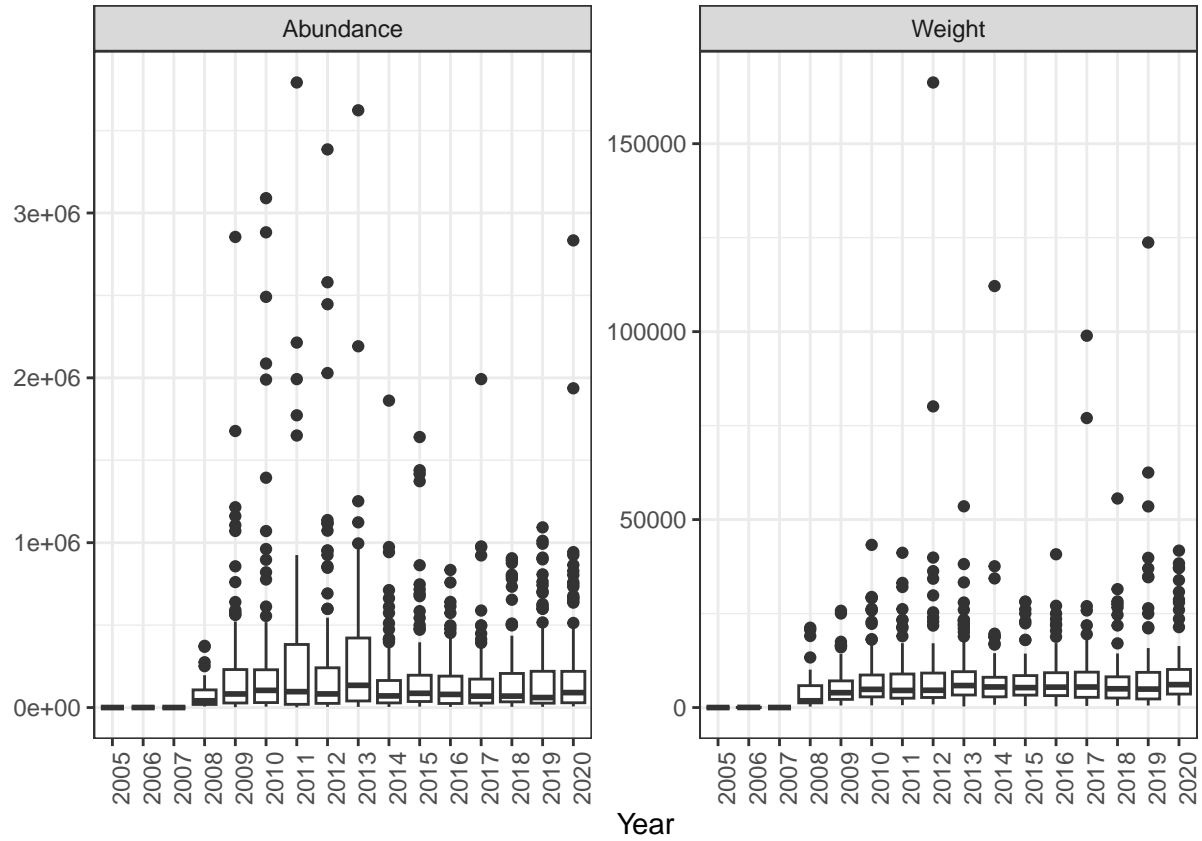
- *num_cpue*, number of individuals (abundance) in $\frac{\text{individuals}}{\text{km}^2}$
- *num_h*, number of individuals (abundance) in $\frac{\text{individuals}}{h}$
- *num*, number of individuals (abundance)
- *wgt_cpue*, weight in $\frac{kg}{\text{km}^2}$
- *wgt_h*, weight in $\frac{kg}{h}$
- *wgt*, weight in *kg*



5. Extreme values

Here we show a yearly total distribution of the biomass data to visualize outliers:

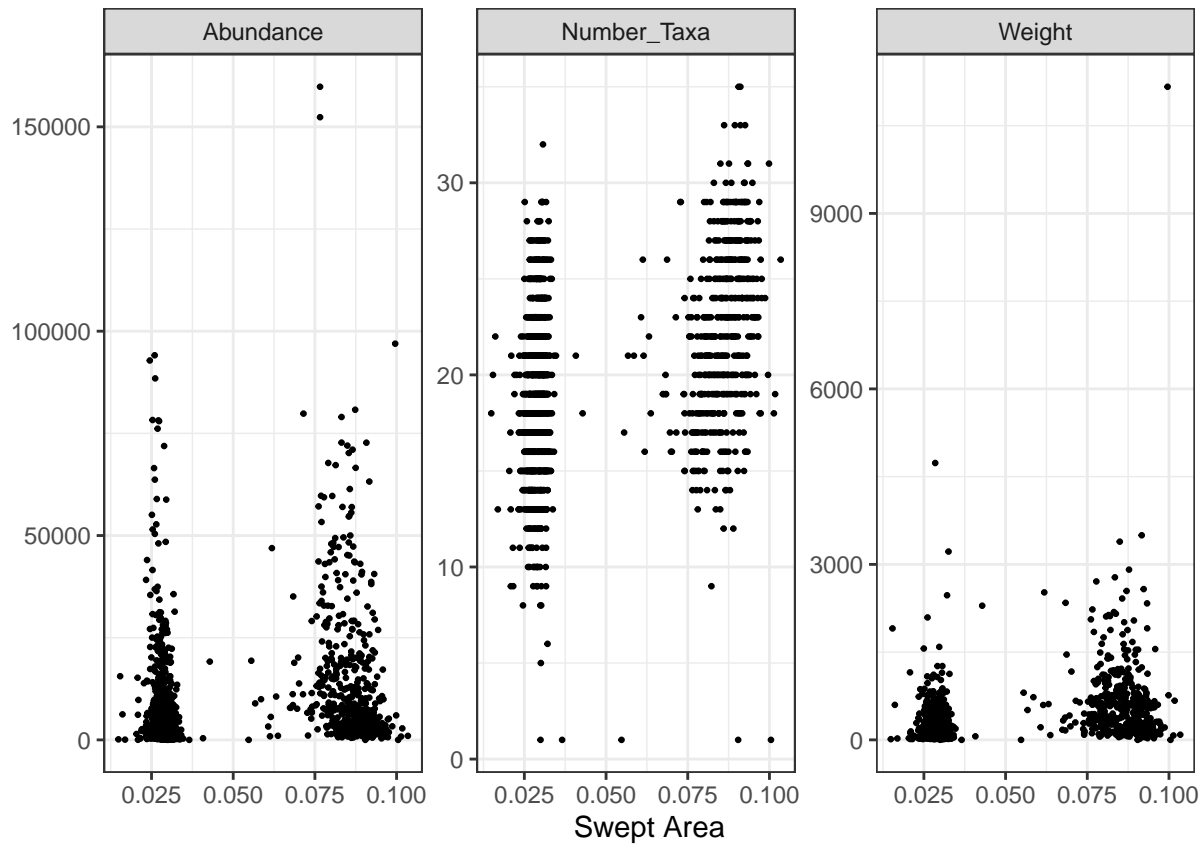
- *wgt*, total weight in *kg* per haul and year per haul and year, if available in the survey data
- *num*, total number of individuals, if available in the survey data



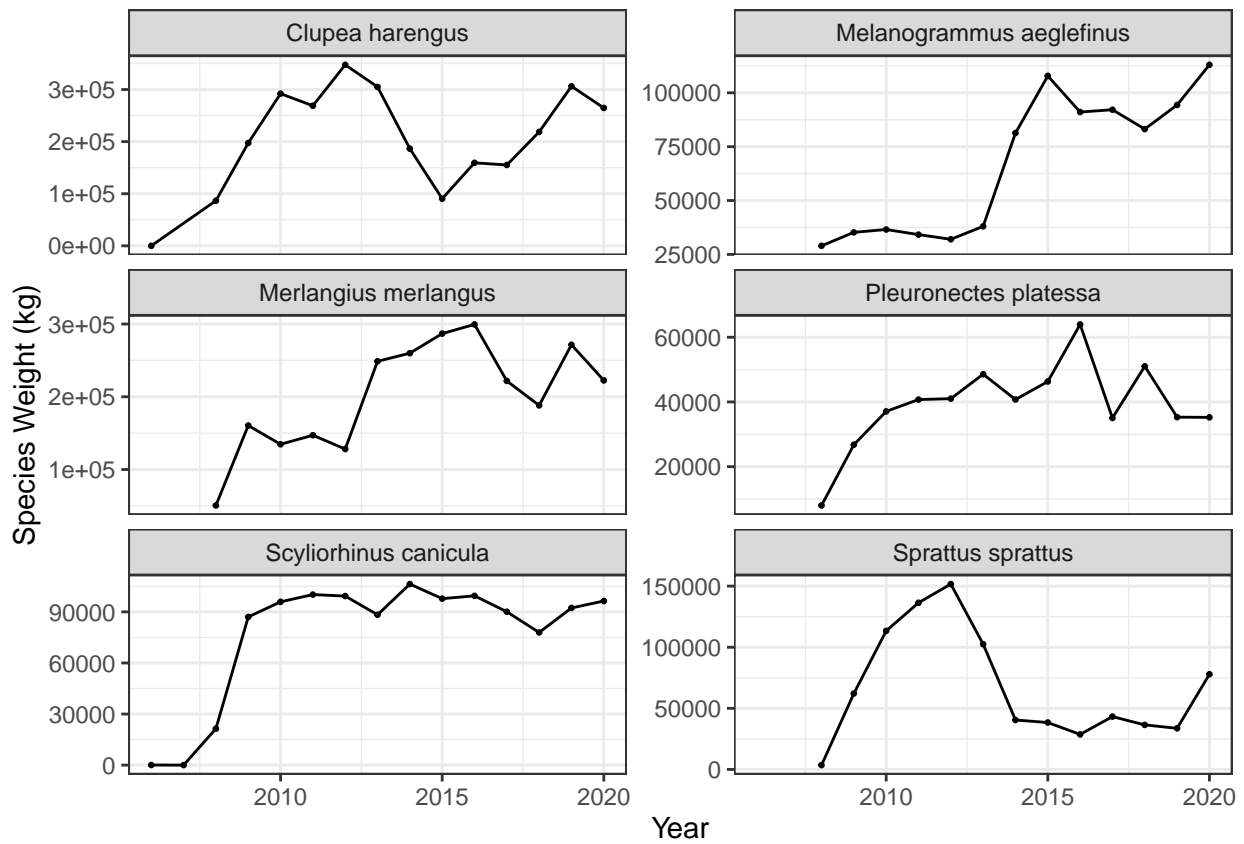
6. Summary of variables against swept area

Here we show the total abundance and number of taxa relationships with the area swept:

- *nbr_taxa*, number of marine fish taxa after taxonomic data cleaning
- *num*, number of individuals, if available in the survey data
- *wgt*, weight in *kg*, if available in the survey data

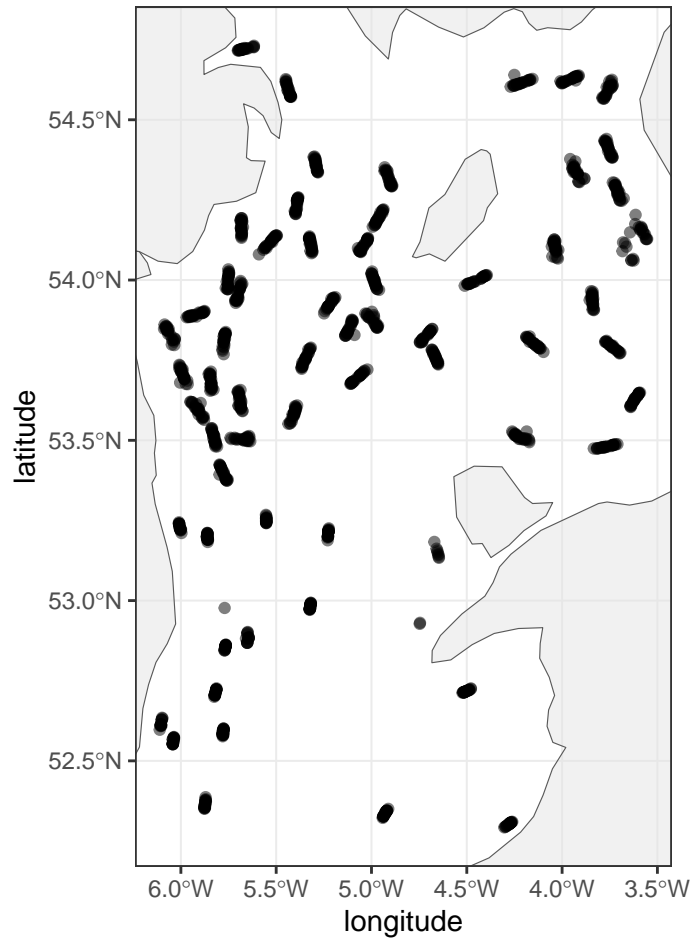


7. Abundance or Weight trends of the six most abundant species



8. Distribution mapping

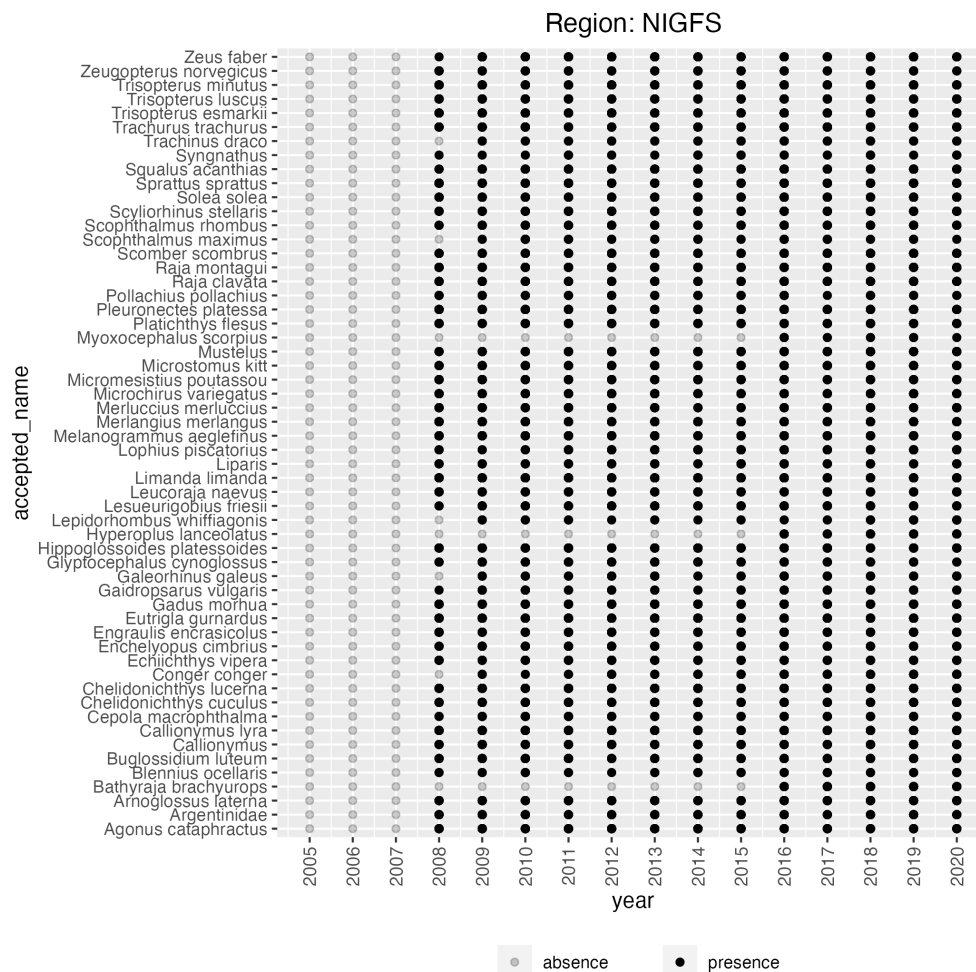
Map of the sampling distribution in space. Note that we only show one year per coordinate.



9. Taxonomic flagging

This species flagging method was adapted from <https://github.com/pinskylab/OceanAdapt/blob/master/R/add-spp-to-taxonomy.Rmd#L33>

Visualization of flagged taxa



Statistics related to the taxonomic flagging outputs

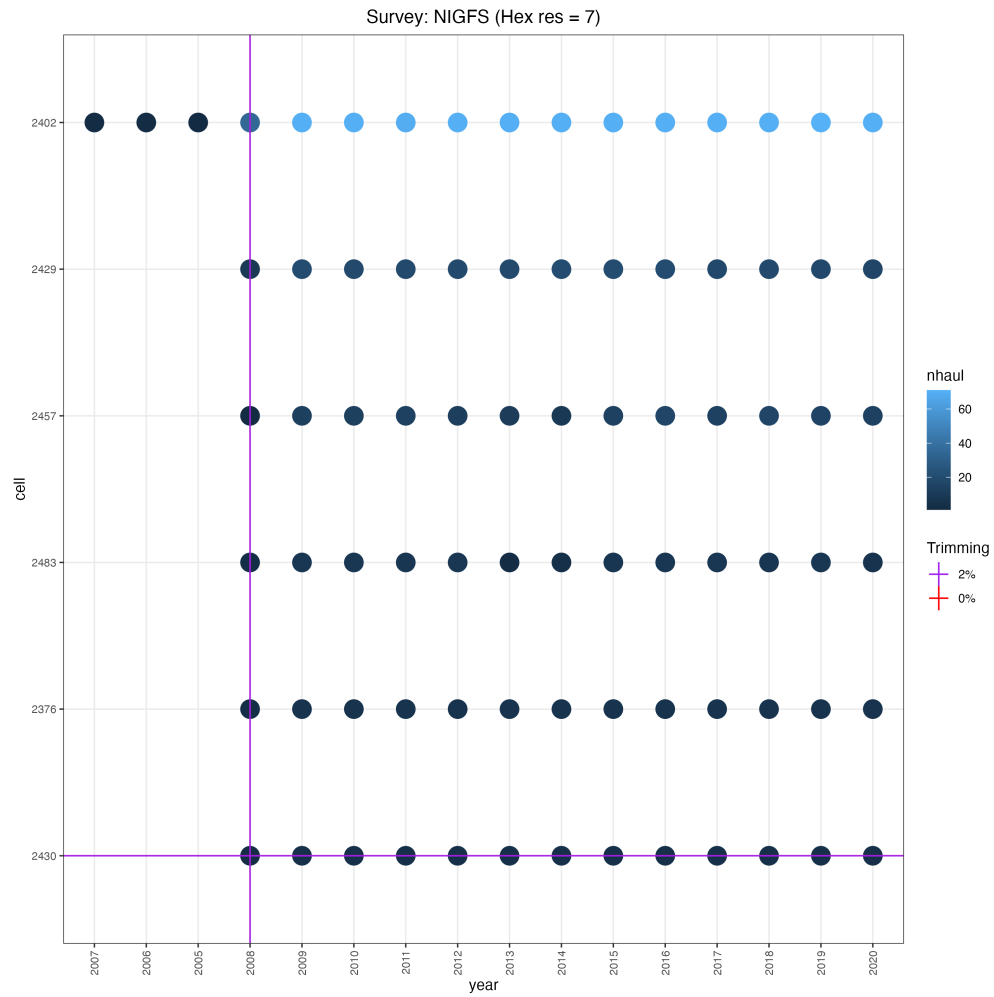
Total number of species	96.0
Percentage of species flagged	58.3

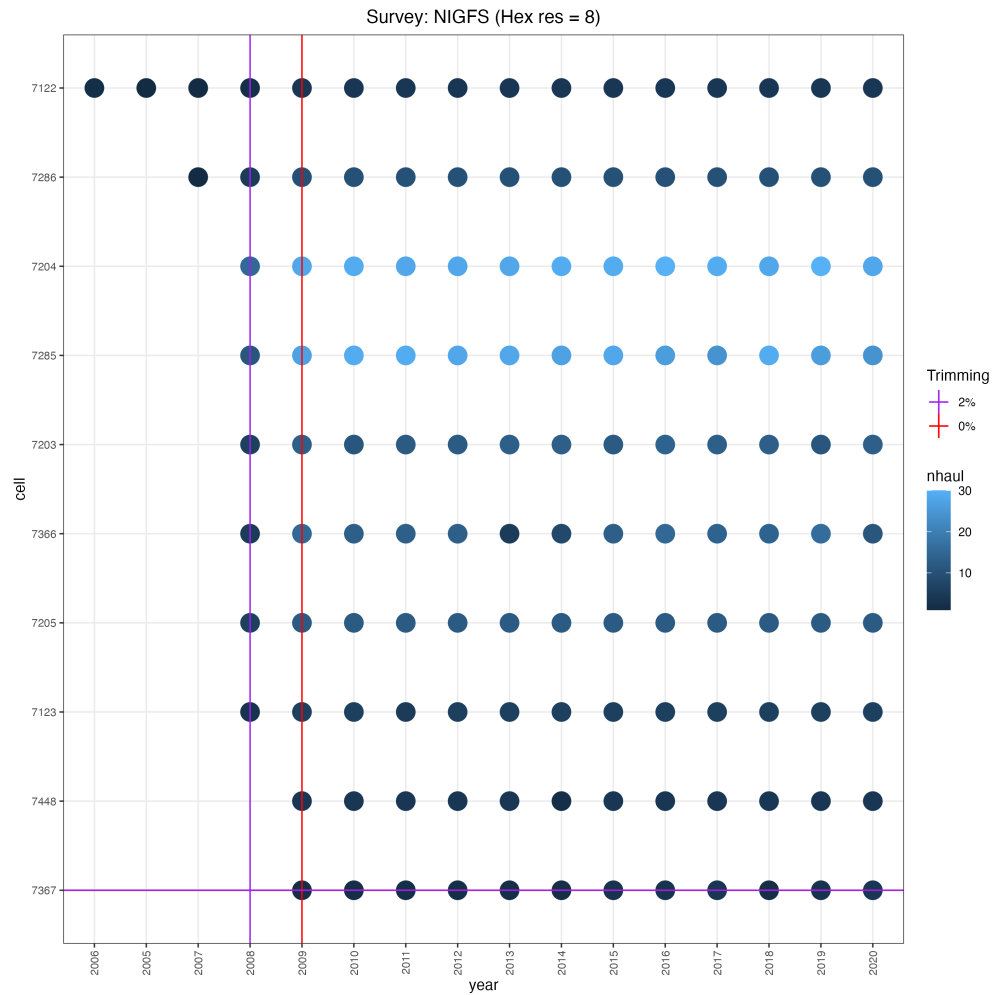
10. Spatio-temporal standardization

a. Standardization method 1

This standardization method was adapted from https://github.com/zoekitchel/rawl_spatial_turnover/blob/master/data_prep_code/species/explore_NorthSea_trimming.Rmd. It was run for hex resolution 7 and 8.

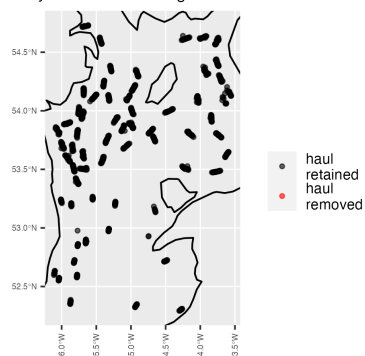
Plot of number of cells x years with overlaid flagging options



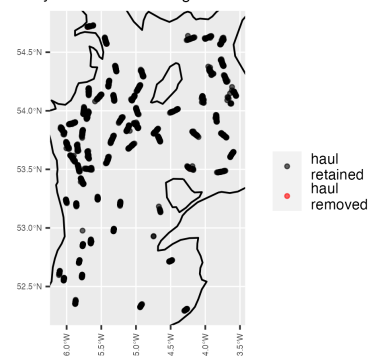


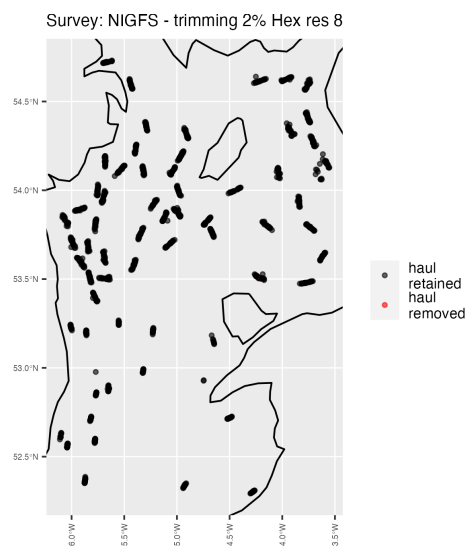
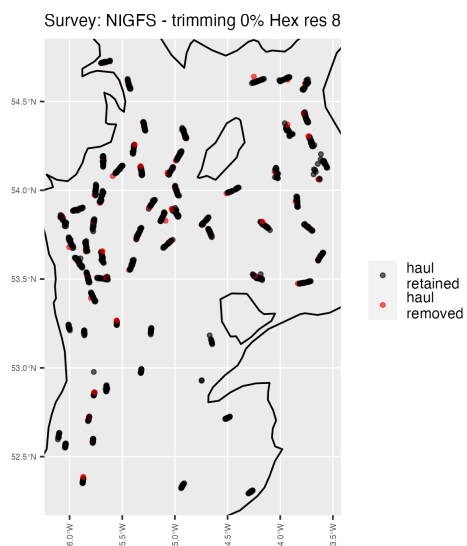
Map of hauls retained and removed per flagging method and threshold

Survey: NIGFS - trimming 0% Hex res 7

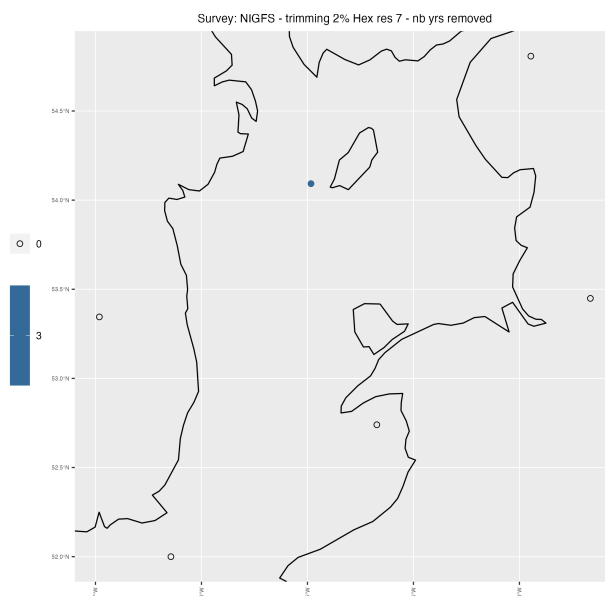
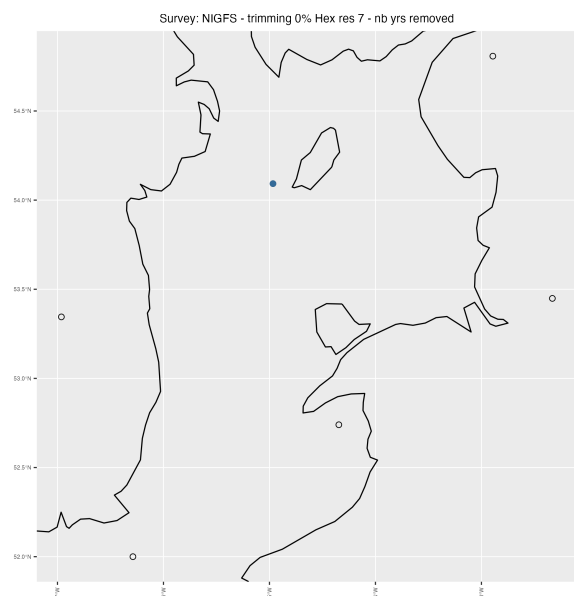


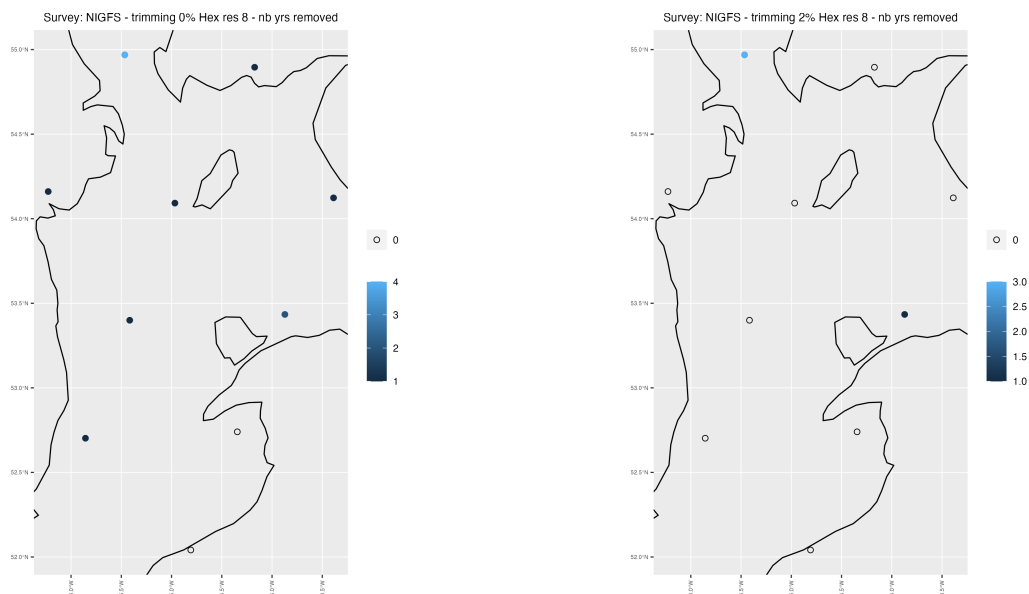
Survey: NIGFS - trimming 2% Hex res 7





Map of numbers of years removed per grid cell and flagging method/threshold



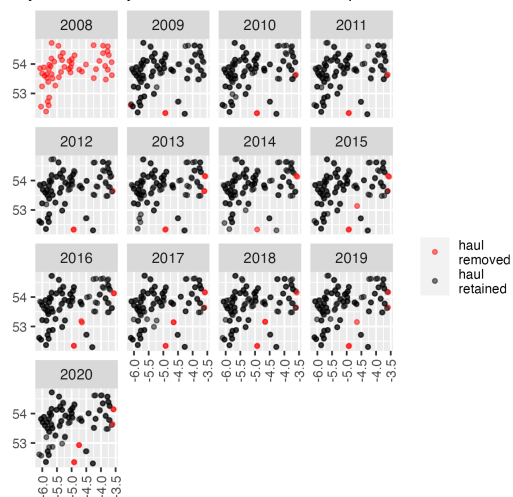


b. Standardization method 2

This standardization method was adapted from BioTIME code from https://github.com/Wubing-Xu/Range_size_winners_losers

Map of hauls retained and removed

survey= NIGFS year1= 2009 year2= 2020 max.shared.samples= 106 duration= 12



c. Standardization summary

Statistics of hauls removed for each standardization method

summary	grid cell 7, 0% threshold	grid cell 7, 2% threshold	grid cell 8, 0% threshold	grid cell 8, 2% threshold	method 2 (biotime)
number of hauls removed	5.0	5.0	58.0	5.0	2261.0
percentage of hauls removed	0.3	0.3	3.9	0.3	7.7