NS-IBTS: North Sea Bottom Trawl survey data processing summary

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

August, 2023

Contents

General info

This document presents the cleaning code and summary of the North Sea bottom trawl survey provided by DATRAS, ICES. It contains data from **1967** and up to **2020**. We discourage users from using the NS-IBTS data for community analyses before 1983 because of sampling inconsistencies. However, the survey data prior to 1983 may be useful for species-specific studies, so we did not filter out the data of the earlier years.

Data cleaning in R

```
#########################################################
#### R code to download and clean DATRAS data from ICES
#### URL: https://datras.ices.dk/
#### Coding: Aurore Maureaud + Juliano Palacios + Daniel van Denderen, December 2022
#### Coding: + Laurene Pecuchet 2023
#########################################################
rm(list=ls())

date <- "3August2023"

#########################################################################################################
```

```r
#### 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)
library(readxl)
library(here)
library(ggplot2)

# load relevant functions
source("functions/write_clean_data.r")
source("functions/clean_taxa.R")
source("functions/write_clean_data.R")
source("functions/apply_trimming_method1.R")
source("functions/apply_trimming_method2.R")
source("functions/flag_spp.R")
source("functions/get_length_weight_coeffs_rfishbase.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

# check length types for conversion to TL?
check_TL_conversion <- FALSE

# apply TL conversion?
apply_TL_conversion <- TRUE


################################################################################################
#### 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))
hh.porc <- getDATRAS(record='HH', survey='SP-PORC', years=c(2001:last.year),
                         quarters=c(3,4))
hh.spnorth <- getDATRAS(record='HH', survey='SP-NORTH', years=c(1990:last.year),
                         quarters=c(3,4))
hh.arsa <- getDATRAS(record='HH', survey='SP-ARSA', years=c(2002:last.year),
                         quarters=c(1,4))

# write.csv(hh.ns, file = "Publicly available/DATRAS/hh.ns.csv",
#           row.names = F)
# write.csv(hh.baltic, file = "Publicly available/DATRAS/hh.baltic.csv",
#           row.names = F)
# write.csv(hh.evhoe, file = "Publicly available/DATRAS/hh.evhoe.csv",
#           row.names = F)
# write.csv(hh.cgfs, file = "Publicly available/DATRAS/hh.cgfs.csv",
#           row.names = F)
# write.csv(hh.igfs, file = "Publicly available/DATRAS/hh.igfs.csv",
#           row.names = F)
# write.csv(hh.nigfs, file = "Publicly available/DATRAS/hh.nigfs.csv",
#           row.names = F)
# write.csv(hh.pt, file = "Publicly available/DATRAS/hh.pt.csv",
#           row.names = F)
# write.csv(hh.rock, file = "Publicly available/DATRAS/hh.rock.csv",
#           row.names = F)
# write.csv(hh.scorock, file = "Publicly available/DATRAS/hh.scorock.csv",
#           row.names = F)
# write.csv(hh.swc, file = "Publicly available/DATRAS/hh.swc.csv",
#           row.names = F)
# write.csv(hh.scowcgfs, file = "Publicly available/DATRAS/hh.scowcgfs.csv",
#           row.names = F)
# write.csv(hh.porc, file = "Publicly available/DATRAS/hh.porc.csv",
#           row.names = F)
# write.csv(hh.spnorth, file = "Publicly available/DATRAS/hh.spnorth.csv",
```

```
#           row.names = F)
# write.csv(hh.arsa, file = "Publicly available/DATRAS/hh.arsa.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))
hl.porc <- getDATRAS(record='HL', survey='SP-PORC', years=c(2001:last.year),
                     quarters=c(3,4))
hl.spnorth <- getDATRAS(record='HL', survey='SP-NORTH', years=c(1990:last.year),
                    quarters=c(3,4))
hl.arsa <- getDATRAS(record='HL', survey='SP-ARSA', years=c(2002:last.year),
                         quarters=c(1,4))

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

}

if(load_stored_datras == TRUE){

  hh.ns <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hh.ns.csv")
  hh.baltic <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hh.baltic.csv")
  hh.evhoe <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hh.evhoe.csv")
  hh.cgfs <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hh.cgfs.csv")
  hh.igfs <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hh.igfs.csv")
  hh.nigfs <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hh.nigfs.csv")
  hh.pt <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hh.pt.csv")
  hh.rock <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hh.rock.csv")
  hh.scorock <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hh.scorock.csv")
  hh.swc <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hh.swc.csv")
  hh.scowcgfs <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hh.scowcgfs.csv")
  # hh.porc <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hh.porc.csv")
  # hh.spnorth <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hh.spnorth.csv")
  # hh.arsa <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hh.arsa.csv")

  hl.ns <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hl.ns.csv")
  hl.baltic <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hl.baltic.csv")
  hl.evhoe <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hl.evhoe.csv")
  hl.cgfs <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hl.cgfs.csv")
  hl.igfs <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hl.igfs.csv")
  hl.nigfs <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hl.nigfs.csv")
  hl.pt <- read.csv("/Volumes/Elements/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("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hl.rock.csv")
  hl.scorock <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hl.scorock.csv")
  hl.swc <- read.csv("/Volumes/Elements/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("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hl.scowcgfs.csv")
  # hl.porc <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hl.porc.csv")
```

```
  # hl.spnorth <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hl.spnorth.csv")
  # hl.arsa <- read.csv("/Volumes/Elements/fishglob data/Publicly available/DATRAS/hl.arsa.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, hh.porc, hh.spnorth, hh.arsa)

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

  rm(hl.ns, hl.baltic, hl.evhoe, hl.cgfs, hl.igfs, hl.nigfs, hl.pt, hl.rock, hl.scorock,
     hl.swc, hl.scowcgfs, hl.porc, hl.spnorth, hl.arsa,
     hh.ns, hh.baltic, hh.evhoe, hh.cgfs, hh.igfs, hh.nigfs, hh.pt, hh.rock, hh.scorock,
     hh.swc, hh.scowcgfs, hh.porc, hh.spnorth, hh.arsa)
  #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 = here(paste0("data/Publicly available/hh.",date,".RData")))
  save(hl, file = here(paste0("data/Publicly available/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)

# Aurore
load(here("data/Publicly available/hl.3August2023.RData"))
load(here("data/Publicly available/hh.3August2023.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()
```

```r
for (i in 1:length(hhn)){
 j <- which(hh$HaulID==hhn[i])
 if(length(j)>1){pb <- hhn[i]}
}
print(pb)
rm(hhn)

# 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.
# 69665 hauls in hh and 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$SpecCode <- hl$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
```

```r
### 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(1,10,4,5,6,7,8),
         DataType %in% c('S','R','C'))

print(length(unique(survey$HaulID)))


##############################################################################################
#### 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(# replace -9 by NAs
         HLNoAtLngt = ifelse(HLNoAtLngt == (-9), NA, HLNoAtLngt),
         TotalNo = ifelse(TotalNo == -9, NA, TotalNo),
         CatCatchWgt = ifelse(CatCatchWgt == -9, NA, CatCatchWgt),
         CatCatchWgt = ifelse(CatCatchWgt<0, NA, CatCatchWgt),
         # standardize by haul duration and rescale with subfactor
         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)),  ### What is it doing here?
         # remove hauls where not all species are recorded
         !(Survey=="BITS" & BySpecRecCode==0))

length(unique(survey$HaulID))
```

```r
##############################################################################################
#### GET THE SWEPT AREA in km2
##############################################################################################

source('cleaning_codes/source_DATRAS_wing_doorspread.R')


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

# Assess size of data without length composition or negative values
xx <- subset(survey, is.na(HLNoAtLngt) | is.na(LngtClass))
no_length_hauls <- sort(unique(xx$HaulID)) # 11,113 hauls with missing length data
print(length(no_length_hauls))
rm(no_length_hauls)

# 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
         wgth = CatCatchWgt*60/(HaulDur*1000), #weight in kg/h
         num = TotalNo, #raw number of individuals
         wgt = CatCatchWgt/1000, # raw weight in kg
         numlencpue = HLNoAtLngt/Area.swept, #abundance/km2 per length class
         numlenh = HLNoAtLngt*60/HaulDur, #abundance/h per length class
         Season = 'NA',
         SBT = replace(SBT, SBT== -9, NA),
         SST = replace(SST, SST== -9, NA),
         LngtClass = ifelse(LngtClass == -9, NA, LngtClass), #replace -9 values by NAs
         LngtCode = ifelse(LngtCode == -9, NA, LngtCode), #replace -9 by NAs
         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, Sex,
         numcpue, wtcpue, numh, wgth, num, wgt, Length, LenMeasType, numlencpue, numlenh)
survey <- data.frame(survey)


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

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

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

# Clean taxa bay of biscay
evhoe_data <- aphia_datras %>% filter(survey=="EVHOE")
clean_evhoe <- clean_taxa(evhoe_data$worms_id_datras, 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 taxa for Spanish Cantabrian Sea
spnorth_data <- aphia_datras %>% filter(survey=="SP-NORTH")
clean_spnorth <- clean_taxa(spnorth_data$worms_id_datras, input_survey = "SP-NORTH",
                       save=F, fishbase=TRUE)

# Clean taxa for Spanish Porcupine
porc_data <- aphia_datras %>% filter(survey=="SP-PORC")
clean_porc <- clean_taxa(porc_data$worms_id_datras, input_survey = "SP-PORC",
```

```
                      save=F, fishbase=TRUE)

# Clean taxa for Spanish Gulf of Cadiz
arsa_data <- aphia_datras %>% filter(survey=="SP-ARSA")
clean_arsa <- clean_taxa(arsa_data$worms_id_datras, input_survey = "SP-ARSA",
                      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, clean_spnorth,
                    clean_porc, clean_arsa) %>%
  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),
```

```
        # gobius
        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(worms_id)) # there are 622 unique inverts taxa in the surveys


##############################################################################################
#### RE-CALCULATE WEIGHTS
##############################################################################################

# 1. Check length measurement types
if(check_TL_conversion == TRUE){
  xx <- survey %>%
    filter(!is.na(LenMeasType),
           !LenMeasType %in% c(-9, 1, 12)) %>%
    group_by(Survey, taxa, LenMeasType) %>%
    summarize(n_obs = length(taxa),
              n_taxa = length(unique(taxa)))

  xx_concern <- survey %>%
    filter(taxa %in% c("Coelorinchus caelorhincus","Malacocephalus laevis",
                       "Macrourus berglax","Coryphaenoides rupestris",
                       "Coelorinchus labiatus", "Hymenocephalus italicus",
                       "Nezumia aequalis", "Nezumia bairdii", "Trachyrincus murrayi",
                       "Trachyrincus scabrus","Xenodermichthys copei",
                       "Chimaeridae","Hydrolagus mirabilis")) %>%
    group_by(Survey, taxa, LenMeasType) %>%
    summarize(n_obs = length(taxa),
              n_taxa = length(unique(taxa)))

  write.csv(xx, file = "QAQC/DATRAS/lengthtypes.csv", row.names = F)
  write.csv(xx_concern, file = "QAQC/DATRAS/lengthtypes_taxa_concern.csv", row.names = F)

  # according to ICES manuals
  family_concern <- survey %>%
    filter(family %in% c("Alepocephalidae","Platytroctidae","Macrouridae","Chimaeridae")) %>%
    mutate(LenMeasType = ifelse(LenMeasType == -9, NA, LenMeasType)) %>%
    group_by(Survey, taxa, family, LenMeasType) %>%
    summarize(n_obs = length(taxa),
              n_taxa = length(unique(taxa)))

  write.csv(family_concern, file = "QAQC/DATRAS/lengthtypes_family_concern.csv", row.names = F)

  taxa_not_TL <- survey %>%
```

```
    filter(family %in% c("Alepocephalidae","Platytroctidae","Macrouridae","Chimaeridae")) %>%
    mutate(LenMeasType = ifelse(LenMeasType == -9, NA, LenMeasType)) %>%
    group_by(worms_id, SpecCode, taxa, family, LenMeasType) %>%
    summarize(n_obs = length(taxa))
  write.csv(taxa_not_TL, file = "length_weight/DATRAS_taxa_not_TL.csv", row.names = F)

}


# 2. apply length conversion factors when necessary
if(apply_TL_conversion == TRUE){
  conversion_to_TL <- read.csv("length_weight/DATRAS_taxa_not_TL_conversions.csv") %>%
  filter(is.na(LenMeasType)) %>%
  select(taxa, conversion_to_TL)

  survey <- left_join(survey, conversion_to_TL, by = "taxa") %>%
    mutate(Length = ifelse(!is.na(conversion_to_TL), Length*conversion_to_TL, Length))
}


# 3. List of taxa for length-weight conversion coefficients
if(need_get_lw_rel == TRUE){
  list.taxa <- survey %>%
    select(taxa, family, genus, rank) %>%
    filter(!is.na(family)) %>%
    distinct()

  write.csv(data.frame(list.taxa), file=paste0("length_weight/taxa_DATRAS_FB_tofill_",date,".csv"),
            row.names=FALSE)

  # length-weight relationships using rfishbase
  get_coeffs(list.taxa, survey="DATRAS", date=date, save=TRUE)
}


# 4. re-calculate weights with length-weight relationships
datalw <- read.csv('length_weight/length.weight_DATRAS_3August2023.csv') %>%
  select(-X)


# summarize abundance/weight at the haul level
survey.num <- left_join(survey, datalw, by=c("taxa","family","genus","rank")) %>%
  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,Sex,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()
```

```r
survey.wgt <- left_join(survey, datalw, by=c("taxa","family","genus","rank")) %>%
  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,Sex,wtcpue,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('wtcpue', '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","rank")) %>%
  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'))


##############################################################################################
# CHECK ESTIMATES PER SURVEY AND TAXA
##############################################################################################

# correlation between abundances to check calculations are right
cor(x = survey3$numh, y = survey3$numlenh, method = 'pearson', use = "complete.obs")
xx <- subset(survey3, !is.na(numcpue))
cor(x = xx$numcpue, y = xx$numlencpue, method = 'pearson', use = "complete.obs")

# correlation between weights to check calculations are right
xx <- subset(survey3, wtcpue>0 & wgtlencpue>0)
cor(x = xx$wtcpue, y = xx$wgtlencpue, method = 'pearson', use = "complete.obs")
```

```r
xx <- subset(survey3, wgth>0 & wgtlenh>0)
cor(x = xx$wgth, y = xx$wgtlenh, method = 'pearson', use = "complete.obs")

# make per survey correlation table
surveys <- c(sort(unique(survey$Survey)),"all","all-SP")
corrs <- data.frame(surveys)
corrs$cor_num <- corrs$cor_wgt <- NA

for (i in 1:length(surveys)){

  # survey-specific data
  if(i==13){xx <- survey3
  } else if (i==14){xx <- survey3 %>% filter(!Survey %in% c("SP-NORTH","SP-ARSA","SP-PORC"))
  } else {xx <- subset(survey3, Survey == surveys[i])}

  # plots
  plot_weights <- ggplot(xx[xx$wgth>0 & xx$wgtlenh>0,], aes(x=wgth, y=wgtlenh)) + geom_point() +
    geom_abline(intercept = 0, slope = 1, color="red",
                linetype="dashed", size=1) + scale_x_log10() + scale_y_log10()  +
    theme_bw() + theme(text = element_text(size = 20)) + ggtitle("Weights per hour")

  plot_abundances <- ggplot(xx[xx$numlenh>0 & xx$num>0,], aes(x=numh, y=numlenh)) + geom_point() +
    geom_abline(intercept = 0, slope = 1, color="red",
                linetype="dashed", size=1) + scale_x_log10() + scale_y_log10() +
    theme_bw() + theme(text = element_text(size = 20)) + ggtitle("Abundances per hour")

  png(paste0("QAQC/DATRAS/",surveys[i],"_per_hour.png"), width = 18*200, height = 10*200, res = 200)
  gridExtra::grid.arrange(plot_weights, plot_abundances, ncol = 2)
  dev.off()

  # compute and save correlations
  corrs$cor_wgt[i] <- cor(x = xx$wgth, y = xx$wgtlenh, method = 'pearson', use = "complete.obs")
  corrs$cor_num[i] <- cor(x = xx$numh, y = xx$numlenh, method = 'pearson', use = "complete.obs")

  rm(xx, plot_weights, plot_abundances)
}

write.csv(corrs, file = "QAQC/DATRAS/correlations_weights.csv", row.names = F)

# 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
```

```r
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() %>%
```

```r
  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()
```

```r
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]

# SP-NORTH - PROBLEMS!
ggplot(subset(xx, Survey=='SP-NORTH'), 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()

# SP-ARSA - some outliers
ggplot(subset(xx, Survey=='SP-ARSA'), 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()

# SP-PORC - PROBLEMS!
ggplot(subset(xx, Survey=='SP-PORC'), 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()


# 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() %>%
```

```r
  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",
                             survey %in% c("SP-NORTH","SP-ARSA") ~ "spain",
                             survey == "SP-PORC" ~ "multi-countries"),
         num = numlencpue*area_swept,
         num_cpue = numlenh,
```

```
         num_cpua = numlencpue,
         wgt = wgtlencpue*area_swept,
         wgt_cpue = wgtlenh,
         wgt_cpua = wgtlencpue,
         haul_dur = haul_dur/60,
         source = "DATRAS ICES",
         timestamp = "2021-07",
         survey_unit = ifelse(survey %in% c("BITS","NS-IBTS","SWC-IBTS","SP-ARSA"),
                              paste0(survey,"-",quarter),survey),
         survey_unit = ifelse(survey %in% c("NEUS","SEUS","SCS","GMEX"),
                              paste0(survey,"-",season),survey_unit)) %>%
  # Final format
  select(fishglob_data_columns$`Column name fishglob`)



##############################################################################################
# Save database
##############################################################################################

# Just run this routine should be good for all
surveys <- sort(unique(survey4$survey))
for(i in 1:length(surveys)){
  xx <- survey4 %>%
    filter(survey == surveys[i])
  write_clean_data(data = xx, survey = surveys[i], overwrite = T,
                   rdata = TRUE)
}



# ----------------------------------------------------------------------------------------#
#### FLAGS ####
# ----------------------------------------------------------------------------------------#
#install required packages that are not already installed
required_packages <- c("data.table",
                       "devtools",
                       "dggridR",
                       "dplyr",
                       "fields",
                       "forcats",
                       "ggplot2",
                       "here",
                       "magrittr",
                       "maps",
                       "maptools",
                       "raster",
                       "rcompendium",
                       "readr",
                       "remotes",
                       "rrtools",
                       "sf",
                       "sp",
```

```
                      "tidyr",
                      "usethis")

not_installed <- required_packages[!(required_packages %in% installed.packages()[ , "Package"])]
if(length(not_installed)) install.packages(not_installed)


#load pipe operator
library(magrittr)

######### Apply taxonomic flagging per region
#get vector of regions (here the survey column)
regions <- levels(as.factor(survey4$survey))

#run flag_spp function in a loop
for (r in regions) {
  flag_spp(survey4, r)
}

######### Apply trimming per survey_unit method 1
#apply trimming for hex size 7
dat_new_method1_hex7 <- apply_trimming_per_survey_unit_method1(survey4, 7)

#apply trimming for hex size 8
dat_new_method1_hex8 <- apply_trimming_per_survey_unit_method1(survey4, 8)

######### Apply trimming per survey_unit method 2
dat_new_method2 <- apply_trimming_per_survey_unit_method2(survey4)


#-------------------------------------------------------------------------------------------------#
#### ADD STRANDARDIZATION FLAGS ####
#-------------------------------------------------------------------------------------------------#
surveys <- sort(unique(survey4$survey))
survey_units <- sort(unique(survey4$survey_unit))
survey_std <- survey4 %>%
  mutate(flag_taxa = NA_character_,
         flag_trimming_hex7_0 = NA_character_,
         flag_trimming_hex7_2 = NA_character_,
         flag_trimming_hex8_0 = NA_character_,
         flag_trimming_hex8_2 = NA_character_,
         flag_trimming_2 = NA_character_)

# integrate taxonomic flags
for(i in 1:length(surveys)){
  if(!surveys[i] %in% c("FALK","GSL-N","MRT","NZ-CHAT","SCS", "SWC-IBTS", "SP-PORC")){
    xx <- data.frame(read_delim(paste0("outputs/Flags/taxonomic_flagging/",
                                       surveys[i],"_flagspp.txt"),
                                delim=";", escape_double = FALSE, col_names = FALSE,
                                trim_ws = TRUE))
    xx <- as.vector(unlist(xx[1,]))

    survey_std <- survey_std %>%
```

```
      mutate(flag_taxa = ifelse(survey == surveys[i] & accepted_name %in% xx,
                                "TRUE",flag_taxa))

    rm(xx)
  }
}

# integrate spatio-temporal flags
for(i in 1:length(survey_units)){

  if(!survey_units[i] %in% c("DFO-SOG","IS-TAU","SCS-FALL","WBLS")){

    hex_res7_0 <- read.csv(paste0("outputs/Flags/trimming_method1/hex_res7/",
                                  survey_units[i], "_hex_res_7_trimming_0_hauls_removed.csv"),
                           sep = ";")
    hex_res7_0 <- as.vector(hex_res7_0[,1])

    hex_res7_2 <- read.csv(paste0("outputs/Flags/trimming_method1/hex_res7/",
                                  survey_units[i], "_hex_res_7_trimming_02_hauls_removed.csv"),
                           sep = ";")
    hex_res7_2 <- as.vector(hex_res7_2[,1])

    hex_res8_0 <- read.csv(paste0("outputs/Flags/trimming_method1/hex_res8/",
                                  survey_units[i], "_hex_res_8_trimming_0_hauls_removed.csv"),
                           sep= ";")
    hex_res8_0 <- as.vector(hex_res8_0[,1])

    hex_res8_2 <- read.csv(paste0("outputs/Flags/trimming_method1/hex_res8/",
                                  survey_units[i], "_hex_res_8_trimming_02_hauls_removed.csv"),
                           sep = ";")
    hex_res8_2 <- as.vector(hex_res8_2[,1])

    trim_2 <- read.csv(paste0("outputs/Flags/trimming_method2/",
                              survey_units[i],"_hauls_removed.csv"))
    trim_2 <- as.vector(trim_2[,1])

    survey_std <- survey_std %>%
      mutate(flag_trimming_hex7_0 = ifelse(survey_unit == survey_units[i] & haul_id %in% hex_res7_0,
                                           "TRUE",flag_trimming_hex7_0),
             flag_trimming_hex7_2 = ifelse(survey_unit == survey_units[i] & haul_id %in% hex_res7_2,
                                           "TRUE",flag_trimming_hex7_2),
             flag_trimming_hex8_0 = ifelse(survey_unit == survey_units[i] & haul_id %in% hex_res8_0,
                                           "TRUE",flag_trimming_hex8_0),
             flag_trimming_hex8_2 = ifelse(survey_unit == survey_units[i] & haul_id %in% hex_res8_2,
                                           "TRUE",flag_trimming_hex8_2),
             flag_trimming_2 = ifelse(survey_unit == survey_units[i] & haul_id %in% trim_2,
                                      "TRUE", flag_trimming_2)
      )
    rm(hex_res7_0, hex_res7_2, hex_res8_0, hex_res8_2, trim_2)
  }
}
```

```r
# Just run this routine should be good for all
for(i in 1:length(surveys)){
  xx <- survey_std %>%
    filter(survey == surveys[i])
  write_clean_data(data = xx, survey = paste0(surveys[i],"_std"), overwrite = T,
                   rdata = TRUE)
}
```

1. Overview of the survey data table

survey	source	timestamp	haul_id	country	sub_area	con
NS-IBTS	DATRAS ICES	2021-07	NS-IBTS 1967 1 GB-SCT 74EX GOV 31 1	multi-countries	NA	eur
NS-IBTS	DATRAS ICES	2021-07	NS-IBTS 1967 1 GB-SCT 74EX GOV 31 1	multi-countries	NA	eur
NS-IBTS	DATRAS ICES	2021-07	NS-IBTS 1967 1 GB-SCT 74EX GOV 31 1	multi-countries	NA	eur
NS-IBTS	DATRAS ICES	2021-07	NS-IBTS 1967 1 GB-SCT 74EX GOV 31 1	multi-countries	NA	eur
NS-IBTS	DATRAS ICES	2021-07	NS-IBTS 1967 1 GB-SCT 74EX GOV 31 1	multi-countries	NA	eur

station	stratum	year	month	day	quarter	season
NA	NA	1967	3	NA	1	NA
NA	NA	1967	3	NA	1	NA
NA	NA	1967	3	NA	1	NA
NA	NA	1967	3	NA	1	NA
NA	NA	1967	3	NA	1	NA

latitude	longitude	haul_dur	area_swept	gear	depth
56.5167	-2.3333	1	0.121119	GOV	46
56.5167	-2.3333	1	0.121119	GOV	46
56.5167	-2.3333	1	0.121119	GOV	46
56.5167	-2.3333	1	0.121119	GOV	46
56.5167	-2.3333	1	0.121119	GOV	46

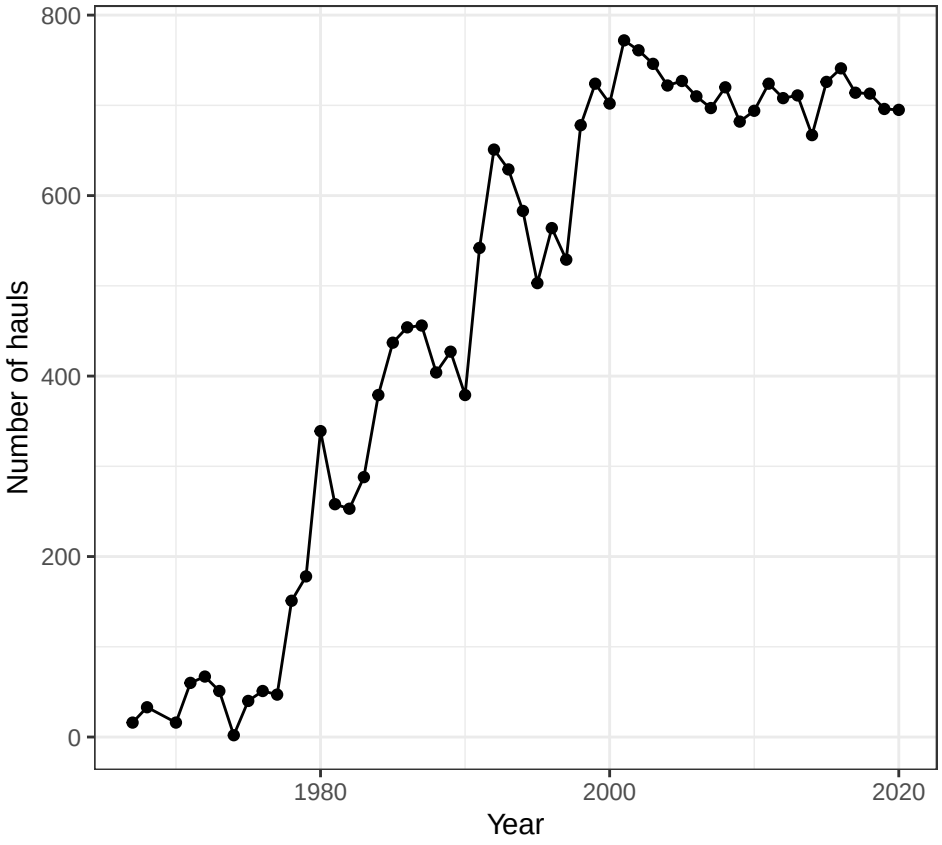
sbt	sst	num	num_cpue	num_cpua	wgt
NA	NA	6	6	49.538075	0.3348526
NA	NA	14	14	115.588842	0.0761275
NA	NA	4	4	33.025383	0.3040554
NA	NA	15	15	123.845187	0.5031525
NA	NA	1	1	8.256346	0.0204853

wgt_cpue	wgt_cpua	verbatim_name	verbatim_aphia_id	accepted_name
0.3348526	2.7646592	NA	126417	Clupea harengus
0.0761275	0.6285354	NA	126425	Sprattus sprattus
0.3040554	2.5103864	NA	126437	Melanogrammus aeglefinus
0.5031525	4.1542011	NA	126438	Merlangius merlangus
0.0204853	0.1691334	NA	127137	Hippoglossoides platessoides

aphia_id	SpecCode	kingdom	phylum	class	order	family
126417	24	Animalia	Chordata	Teleostei	Clupeiformes	Clupeidae
126425	1357	Animalia	Chordata	Teleostei	Clupeiformes	Clupeidae
126437	1381	Animalia	Chordata	Teleostei	Gadiformes	Gadidae
126438	29	Animalia	Chordata	Teleostei	Gadiformes	Gadidae
127137	4239	Animalia	Chordata	Teleostei	Pleuronectiformes	Pleuronectidae

2. Summary of sampling intensity

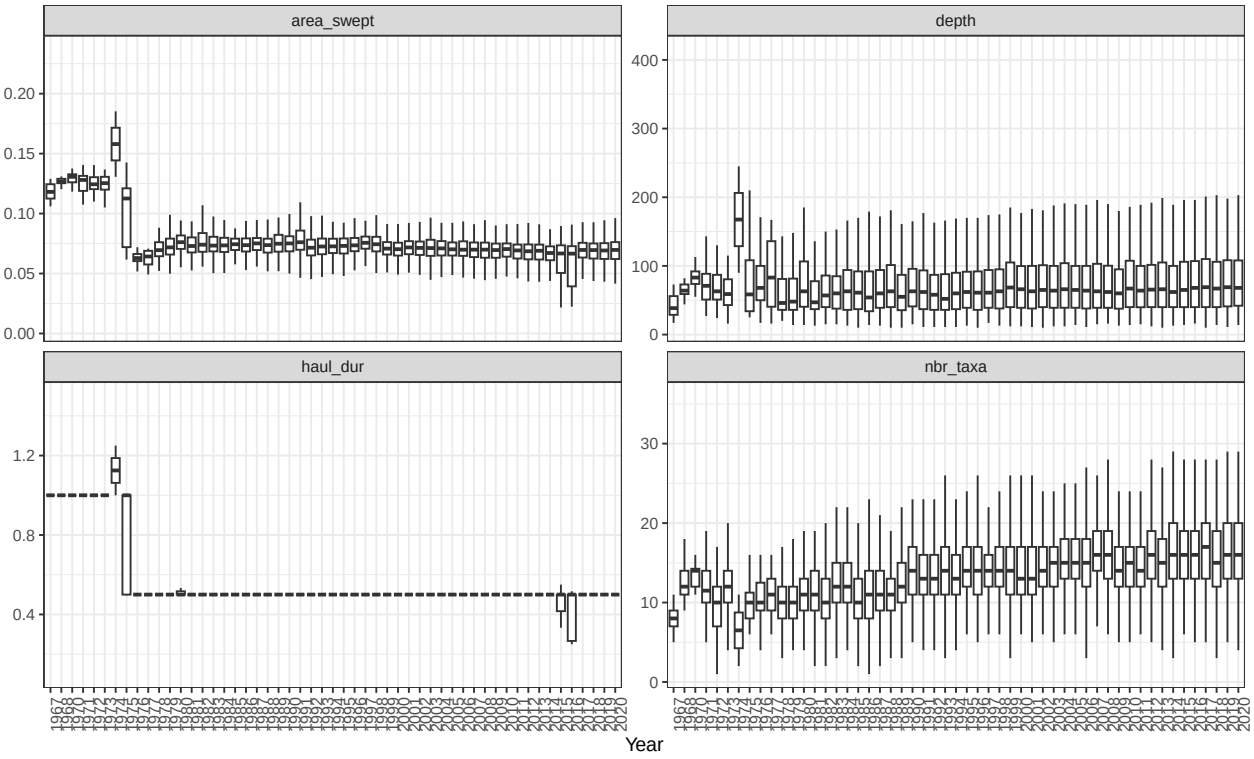
Number of hauls per year performed during the survey after data processing.



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:

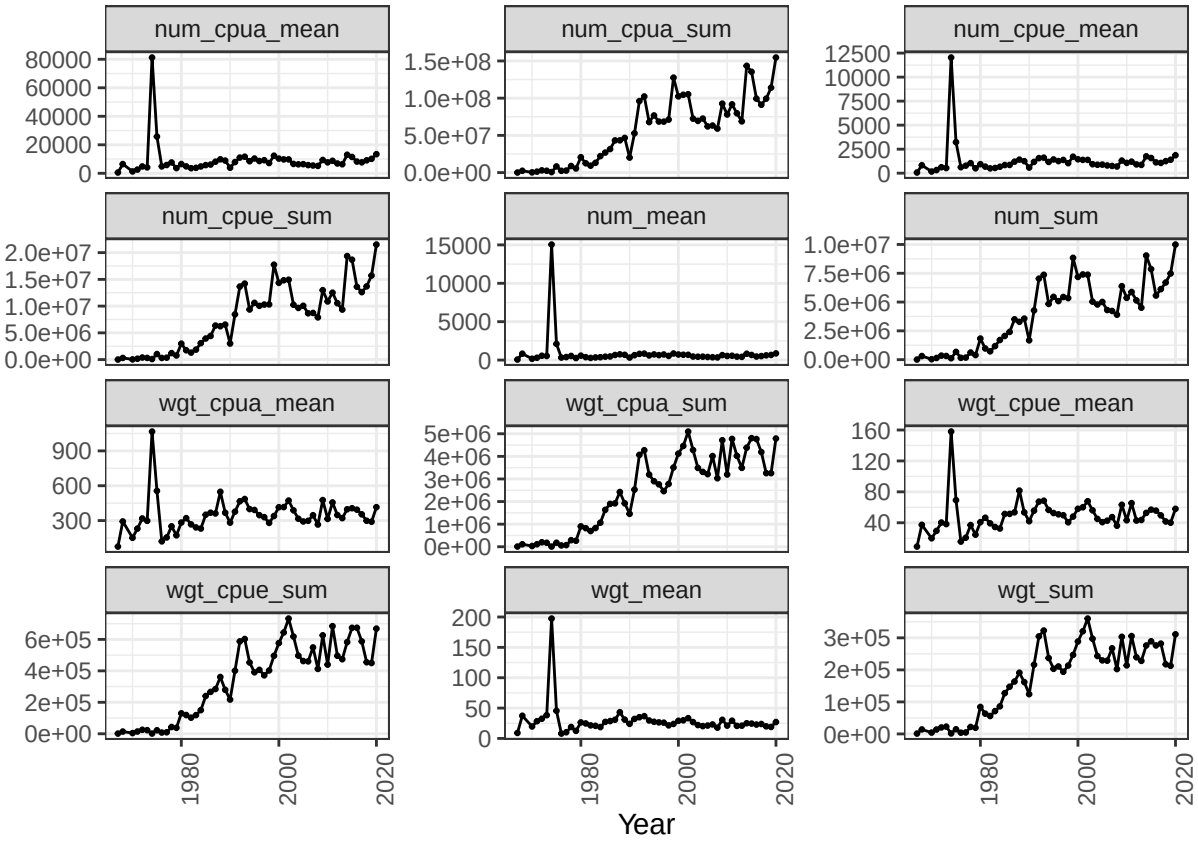
- *area_swept*, swept area by the bottom trawl gear km^2
- *depth*, sampling depth in m
- *haul_dur*, haul sampling duration *hours*
- *number of marine fish taxa*, taxa were cleaned following the last version of taxonomy from the World Register of Marine Species (https://www.marinespecies.org/, October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



4. Summary of biological variables

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

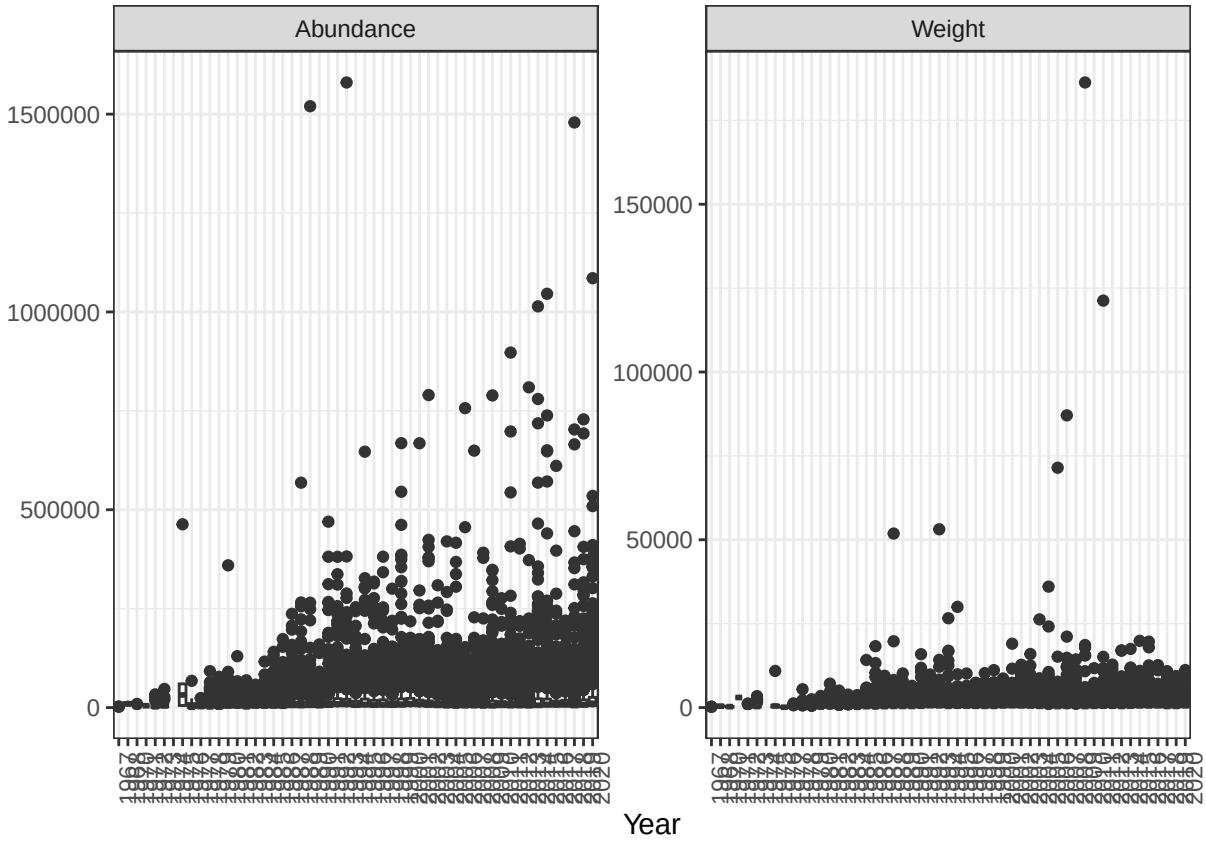
- *num_cpue*, number of individuals (abundance) in $\frac{individuals}{km^2}$
- *num_h*, number of individuals (abundance) in $\frac{individuals}{h}$
- *num*, number of individuals (abundance)
- *wgt_cpue*, weight in $\frac{kg}{km^2}$
- *wgt_h*, weight in $\frac{kg}{h}$
- *wgt*, weight in kg



5. Extreme values

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

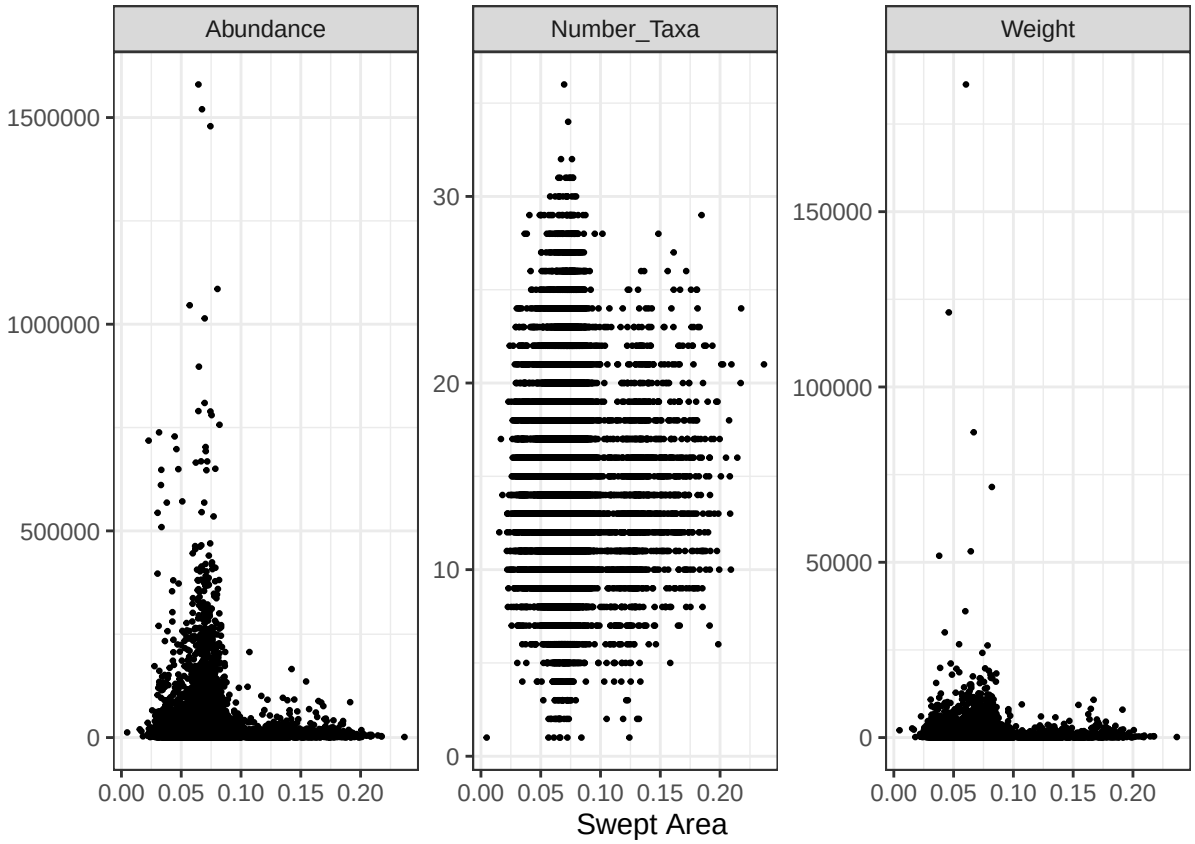
- *wgt*, total weight in *kg* per haul and year per haul and year, if available in the survey data
- *num*, total number of individuals, if available in the survey data



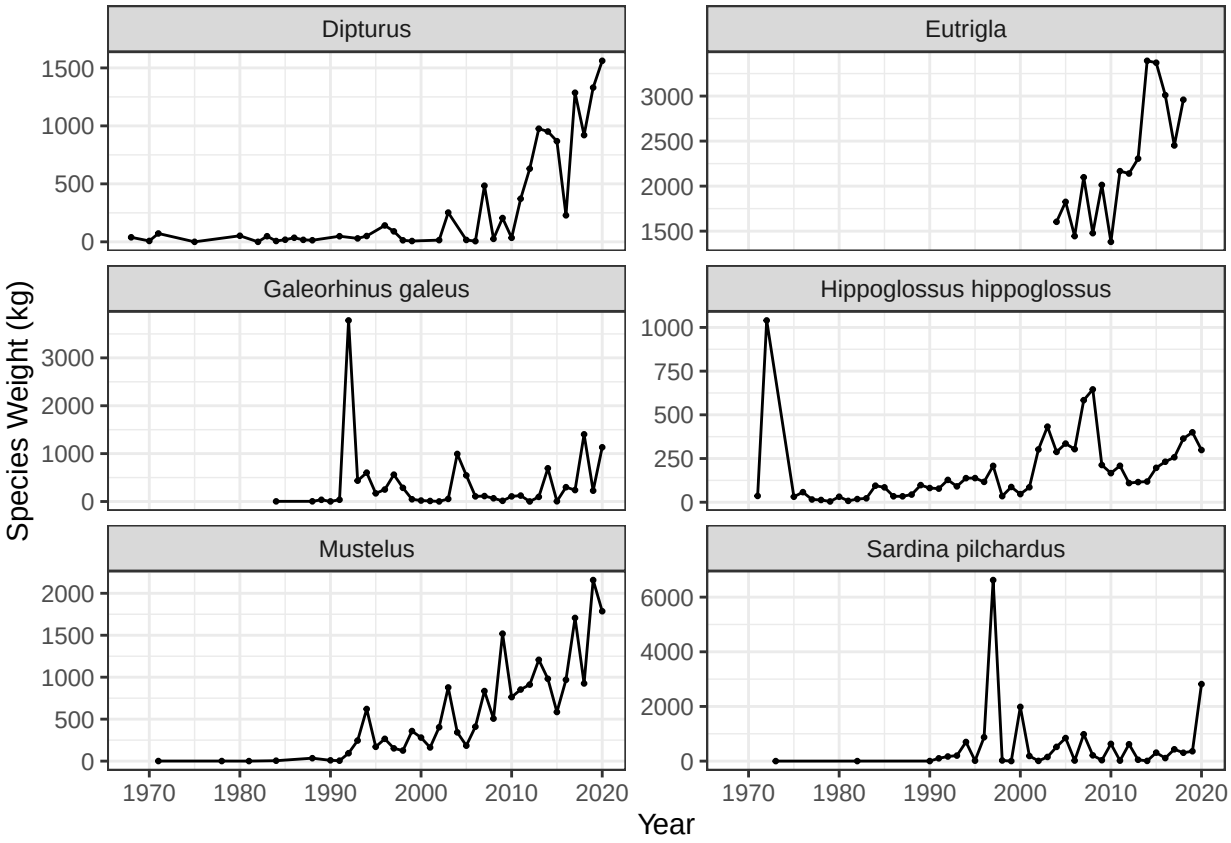
6. Summary of variables against swept area

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_cpue*, number of individuals (abundance) in $\frac{individuals}{km^2}$
- *wgt_cpue*, weight in $\frac{kg}{km^2}$

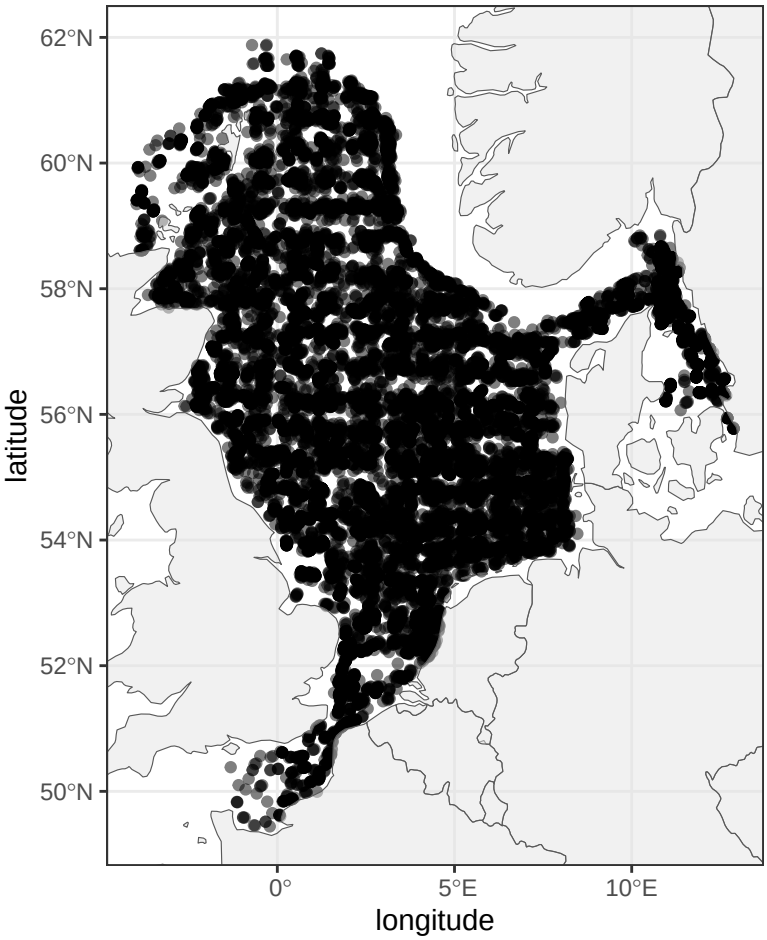


7. Abundance or Weight trends of the six most abundant species



8. Distribution mapping

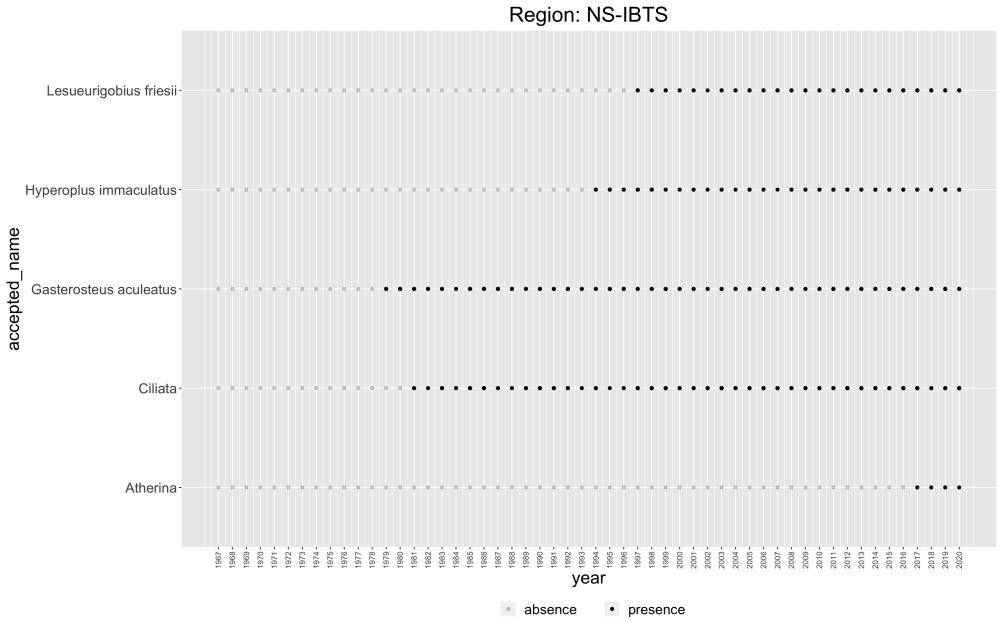
Map of the sampling distribution in space. Note that we only show one year per coordinate.



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

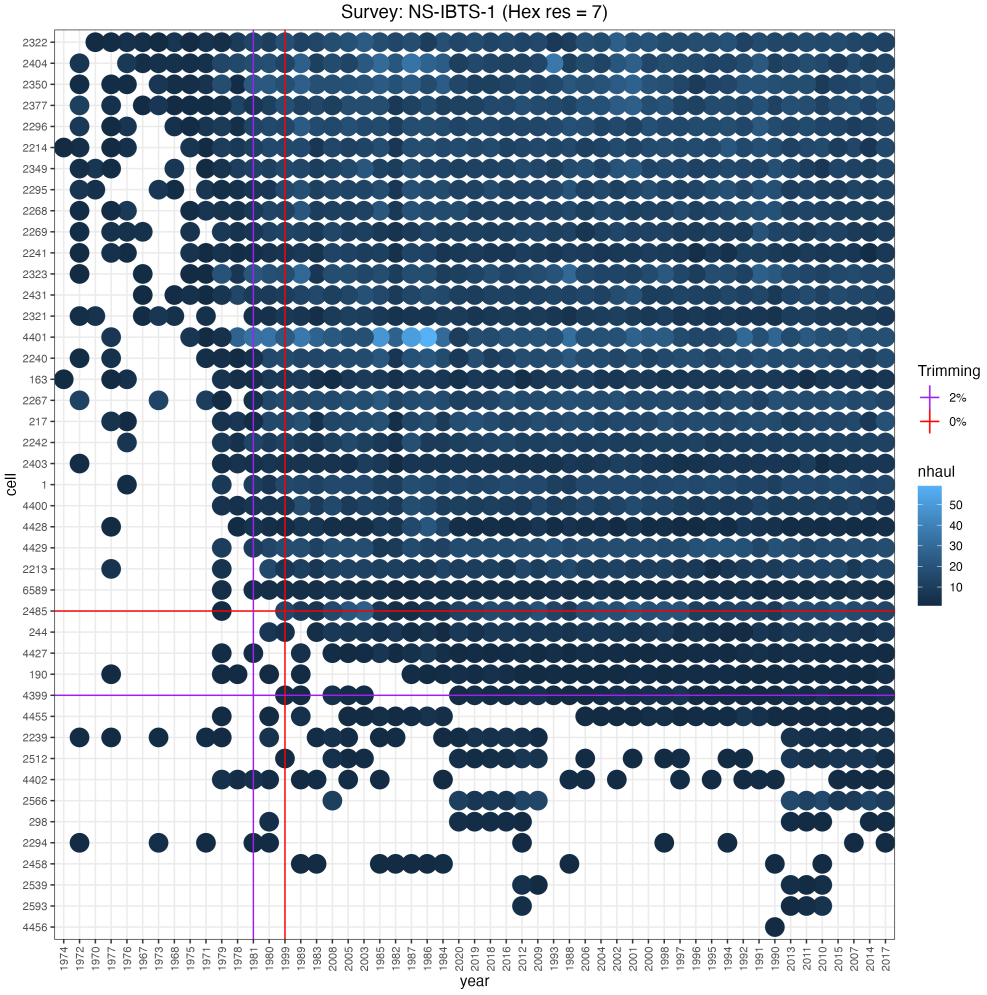
Total number of species	245
Percentage of species flagged	2

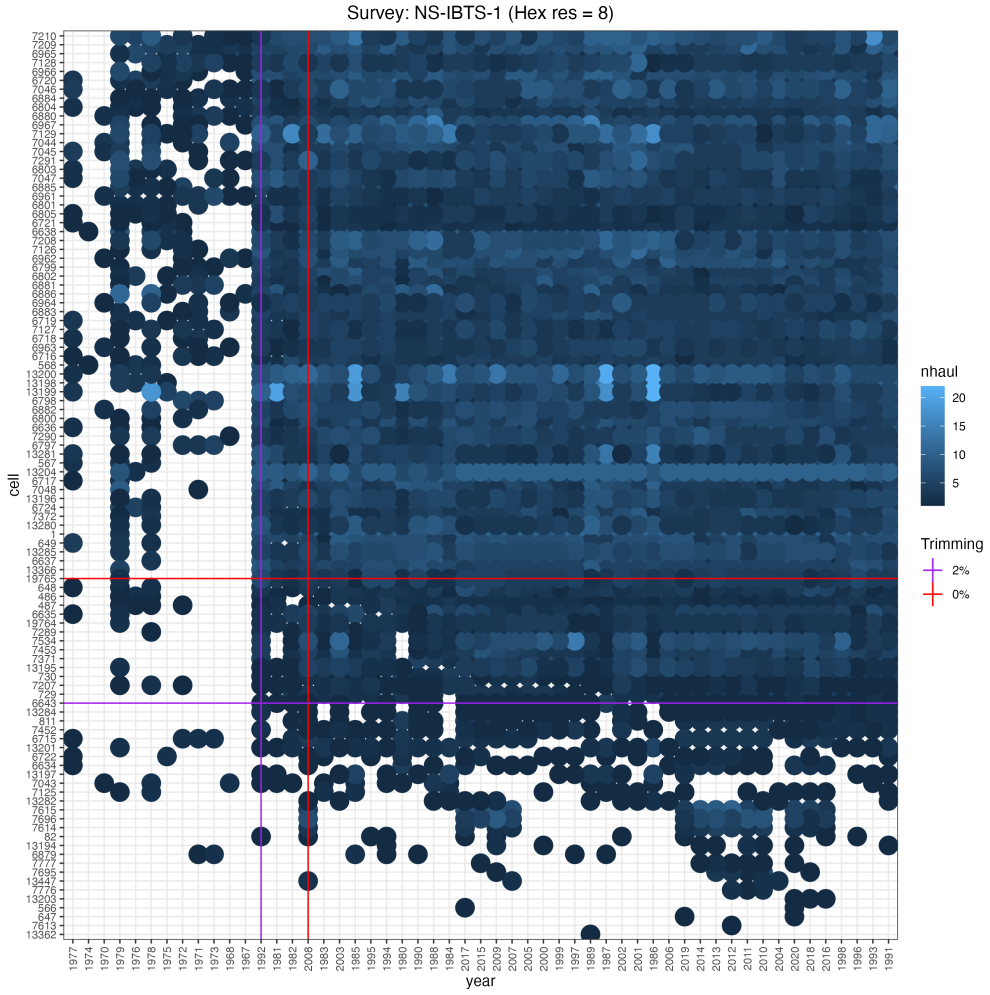
10. Spatio-temporal standardization: NS-IBTS-1

a. Standardization method 1

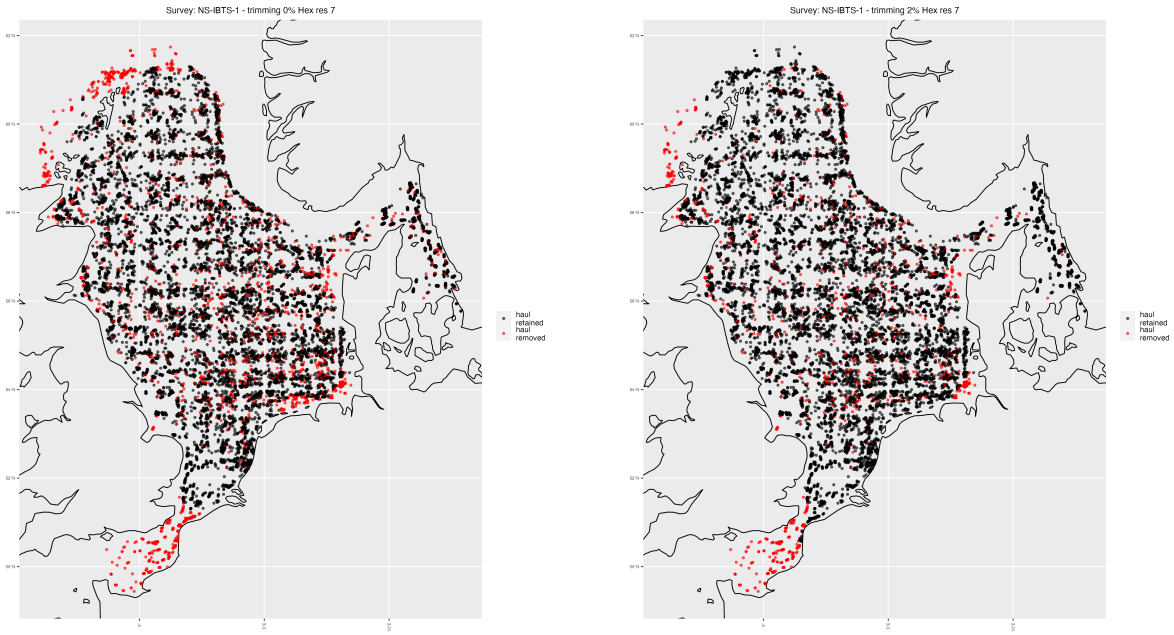
This standardization method was adapted from https://github.com/zoekitchel/trawl_spatial_turnover/blob/master/data_prep_code/species/explore_NorthSea_trimming.Rmd
It was run for hex resolution 7 and 8.

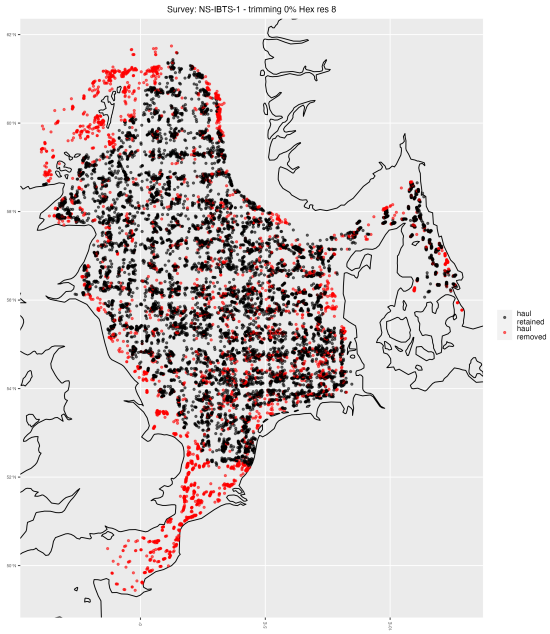
Plot of number of cells x years with overlaid flagging options

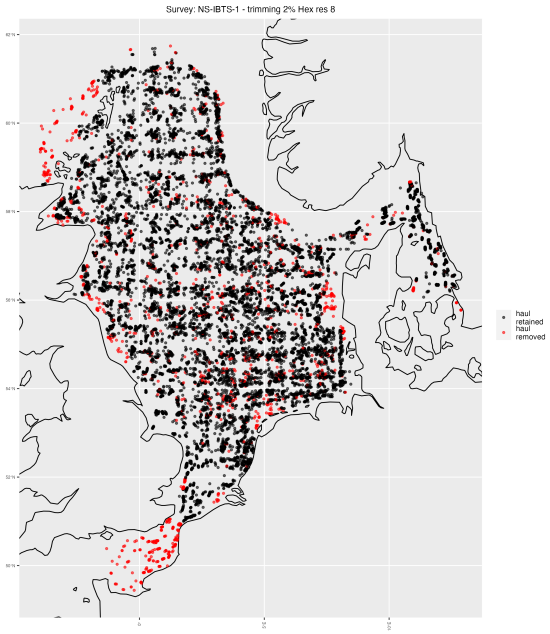




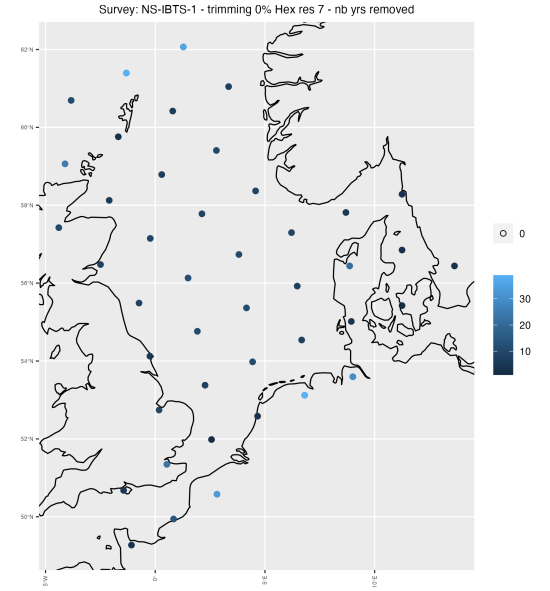
Map of hauls retained and removed per flagging method and threshold

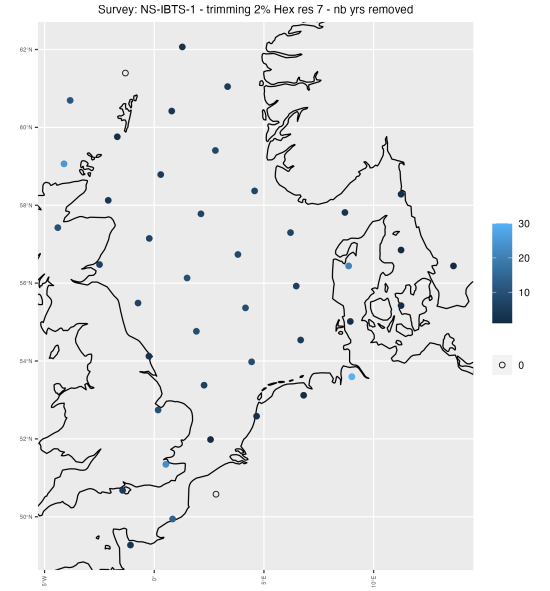


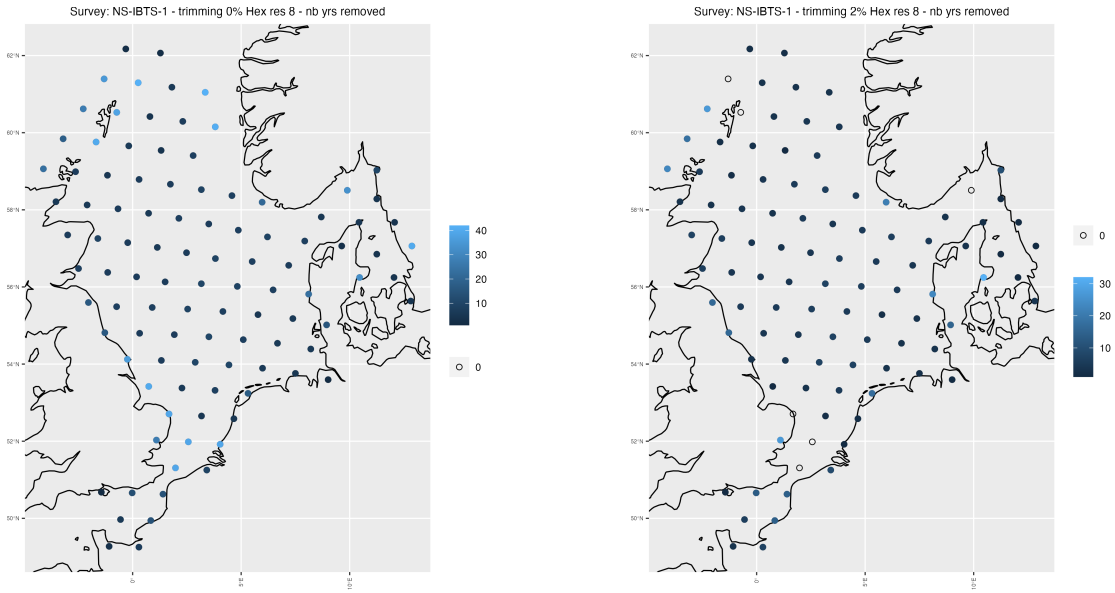




Map of numbers of years removed per grid cell and flagging method/threshold



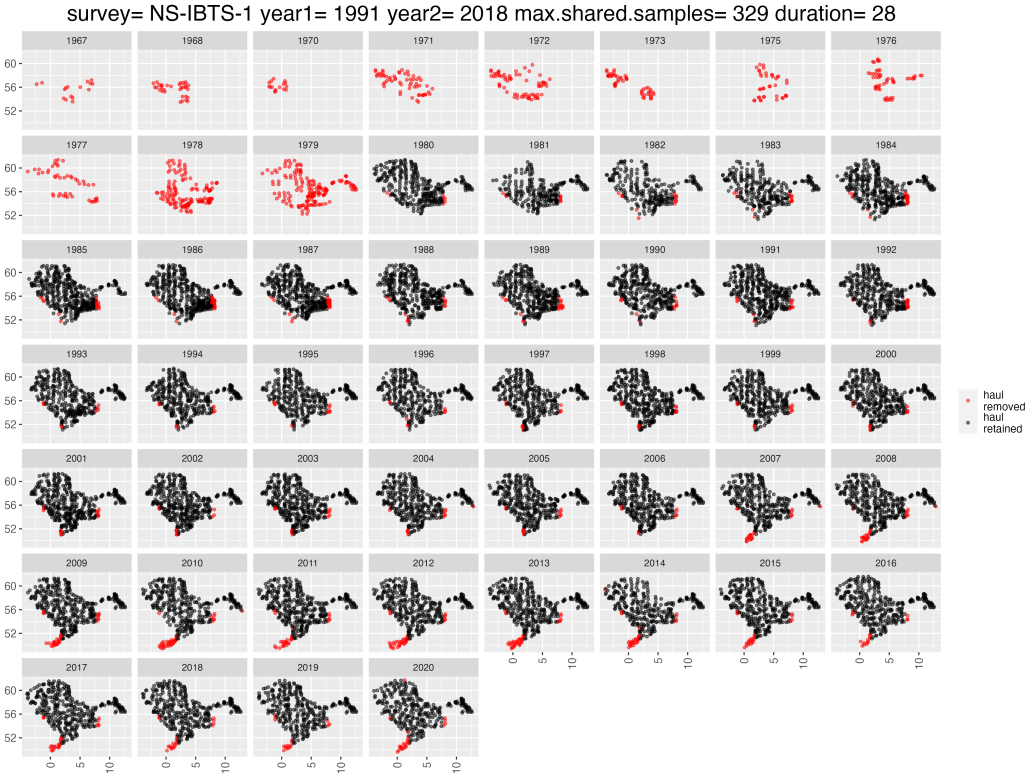




b. Standardization method 2

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

Map of hauls retained and removed



c. Standardization summary

Statistics of hauls removed for each standardization method

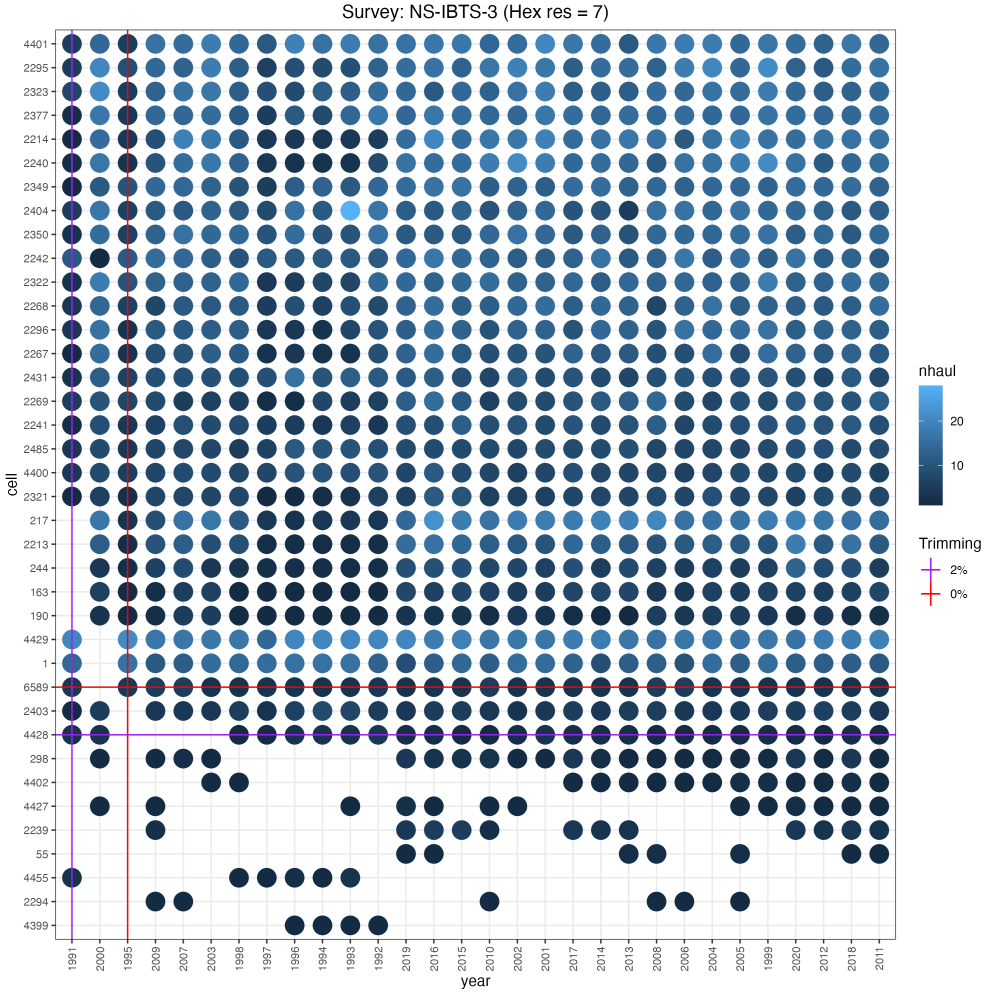
summary	grid cell 7, 0% threshold	grid cell 7, 2% threshold	grid cell 8, 0% threshold	grid cell 8, 2% threshold	method 2 (biotime)
number of hauls removed	2160.0	1157.0	3683.0	1350.0	19362.0
percentage of hauls removed	13.4	7.2	22.8	8.4	8.5

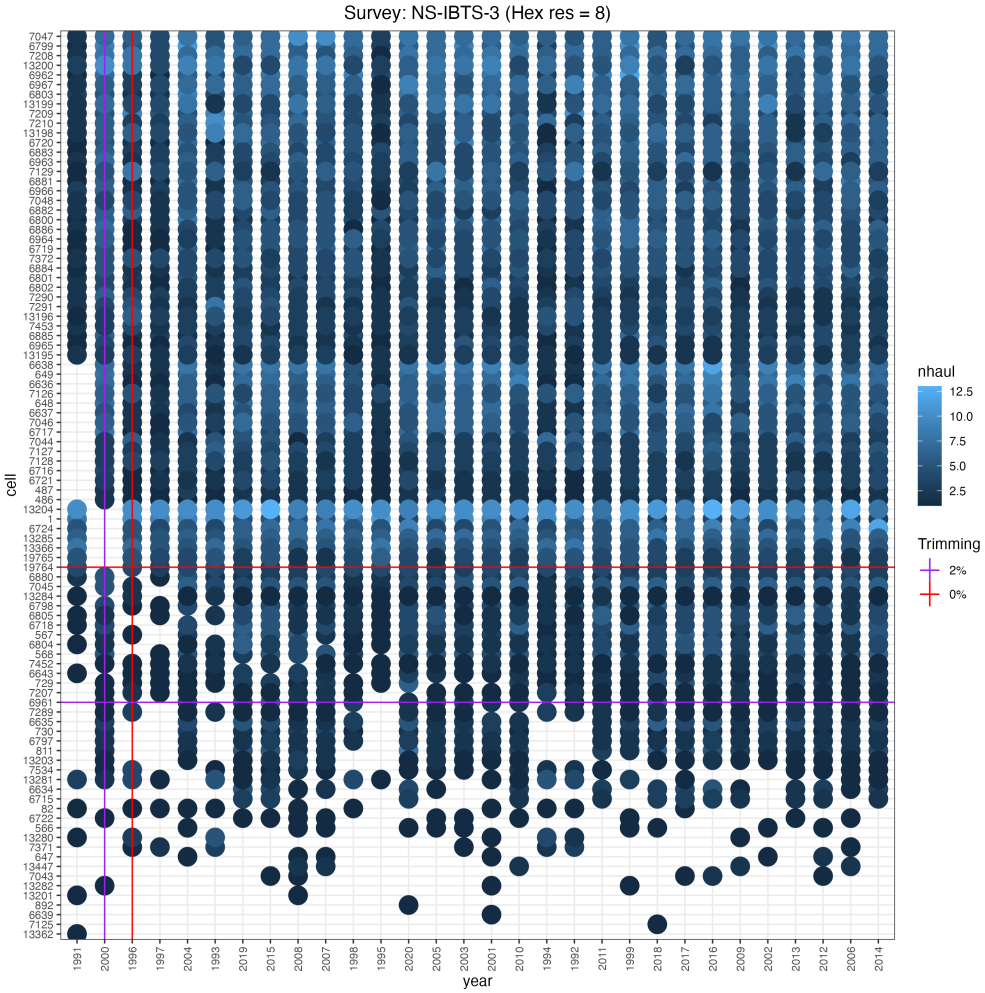
11. Spatio-temporal standardization: NS-IBTS-3

a. Standardization method 1

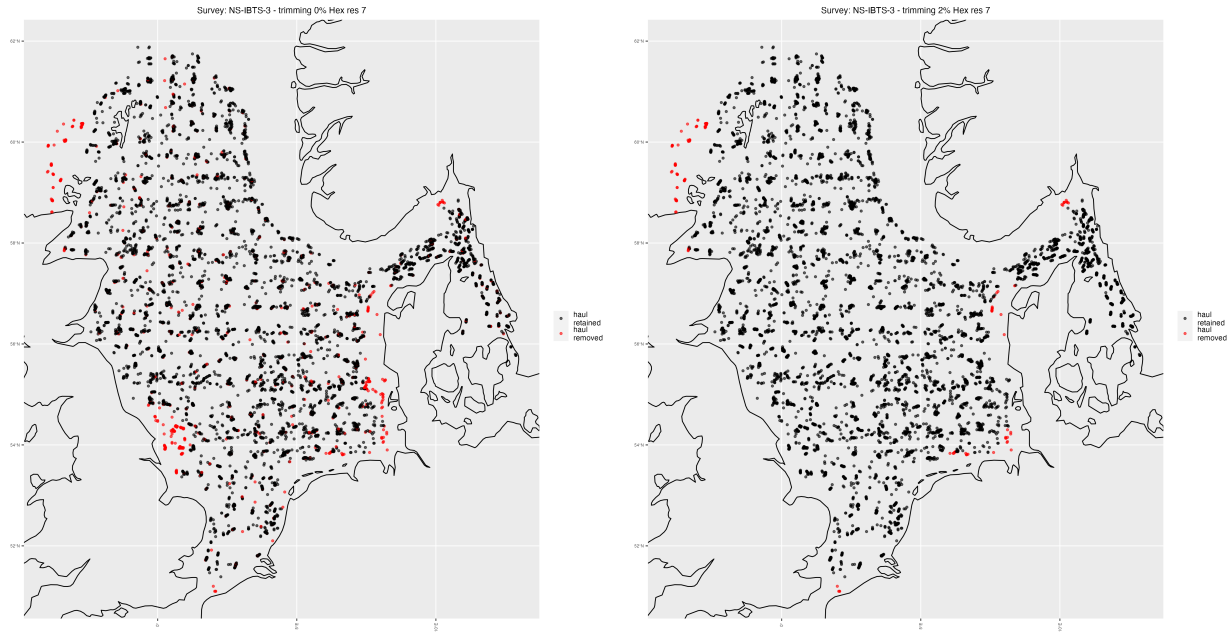
This standardization method was adapted from https://github.com/zoekitchel/trawl_spatial_turnover/blob/master/data_prep_code/species/explore_NorthSea_trimming.Rmd
It was run for hex resolution 7 and 8.

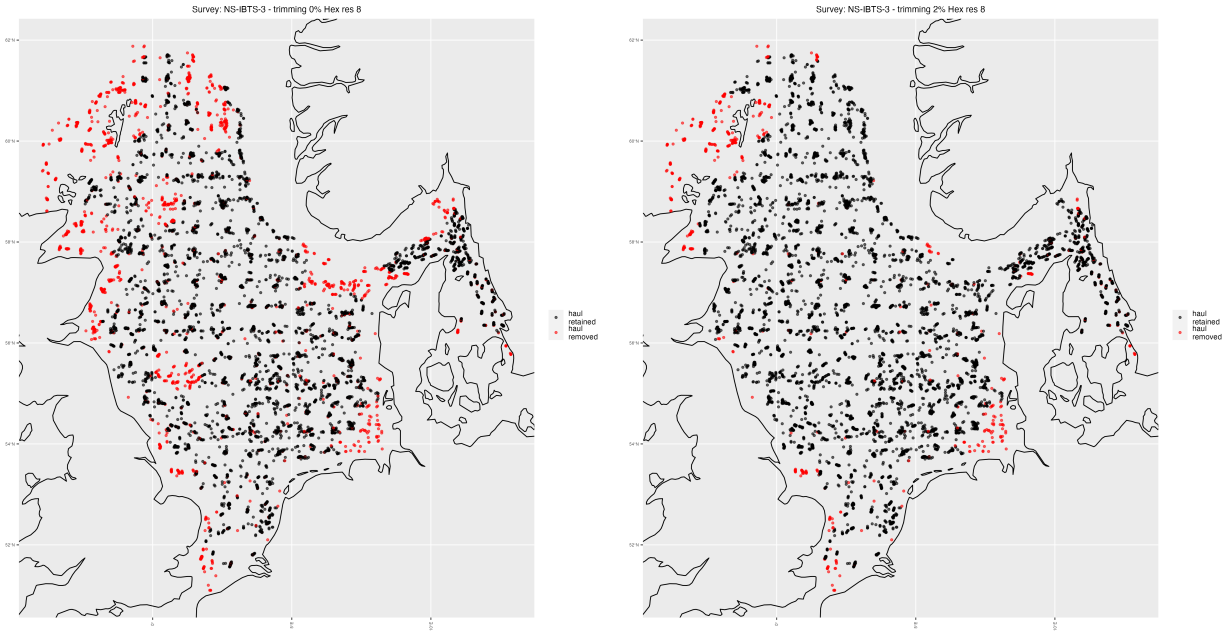
Plot of number of cells x years with overlaid flagging options



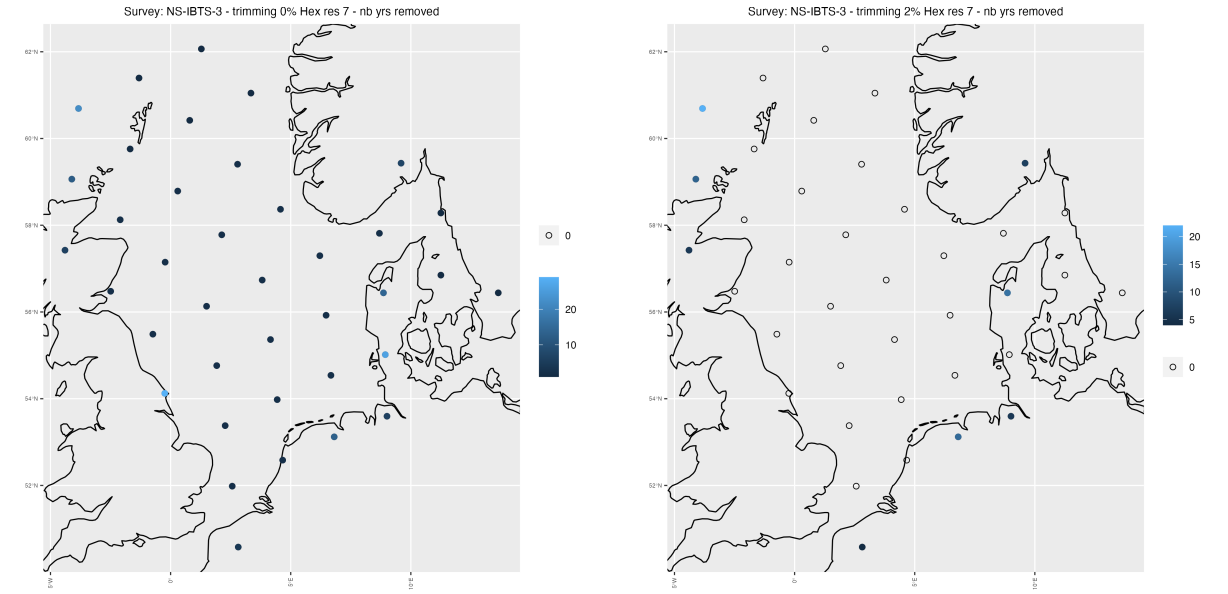


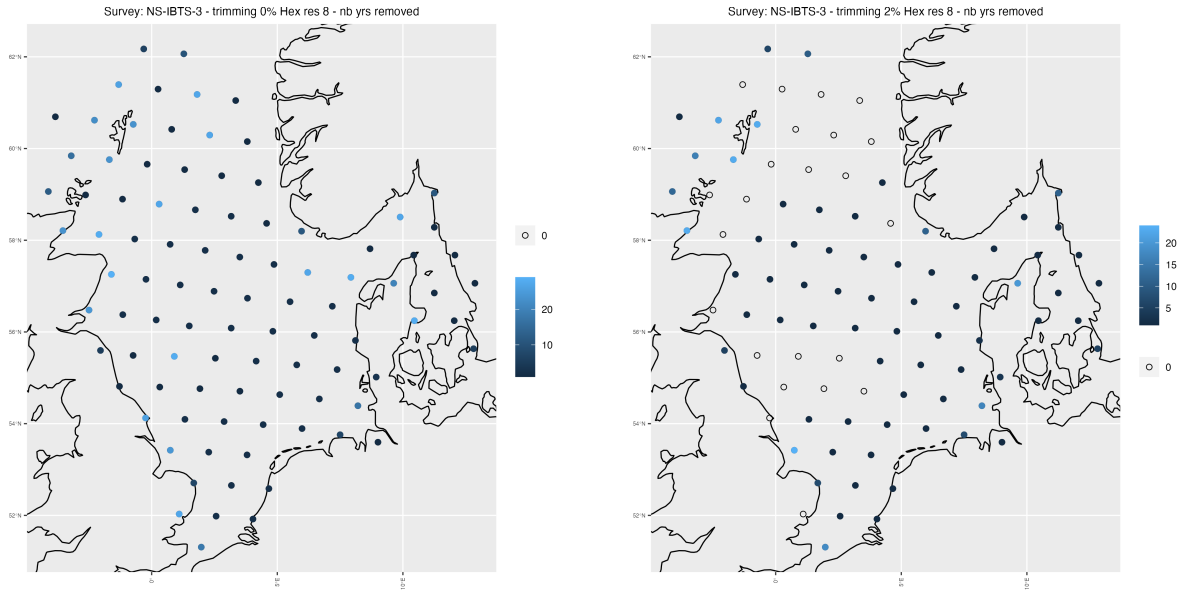
Map of hauls retained and removed per flagging method and threshold





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

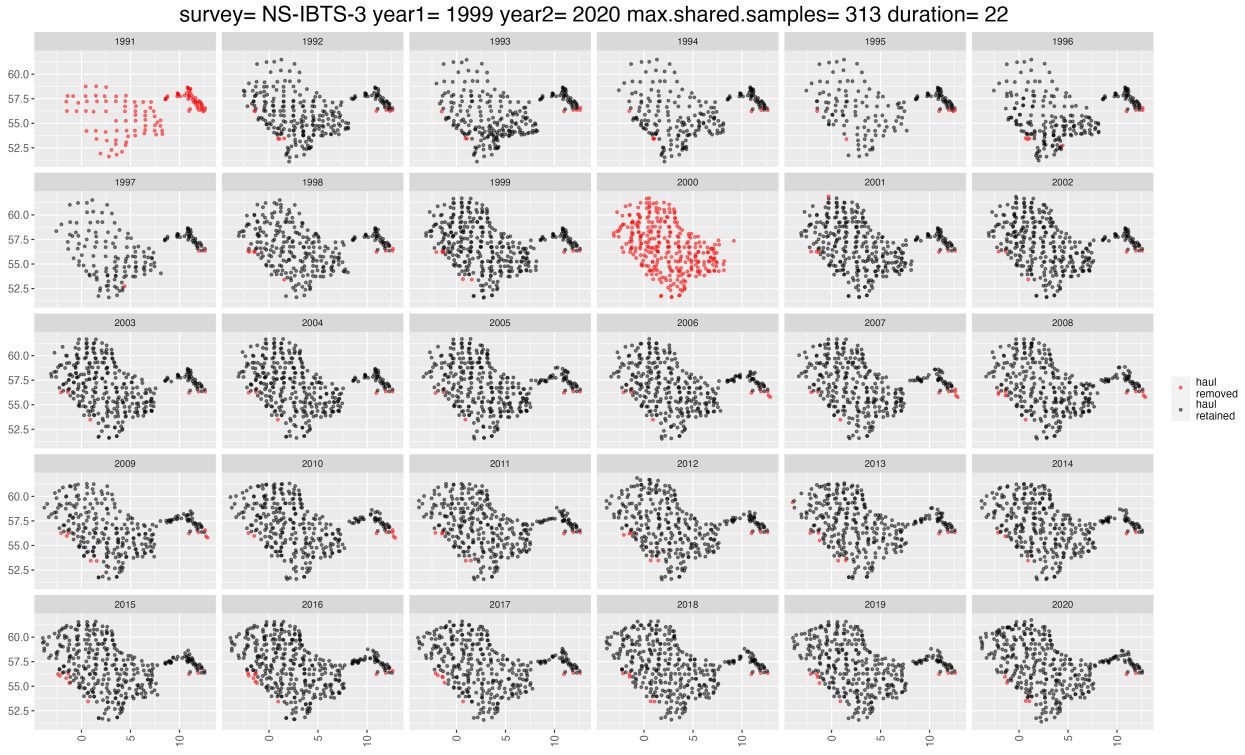




b. Standardization method 2

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

Map of hauls retained and removed



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	746.0	147.0	1931.0	649.0	8905.0
percentage of hauls removed	8.2	1.6	21.3	7.1	6.6