

NOR-BTS: Norwegian bottom trawl survey data processing summary

fishglob, Aurore A. Maureaud & Juliano Palacios Abrantes

January, 2023

Contents

General info	1
Data cleaning in R	1
1. Overview of the survey data table	14
2. Summary of sampling intensity	15
3. Summary of sampling variables from the survey	16
4. Summary of biological variables	17
5. Extreme values	18
6. Summary of variables against swept area	19
7. Abundance or Weight trends of the six most abundant species	20
8. Distribution mapping	21
9. Taxonomic flagging	22
10. Spatio-temporal standardization	23
a. Standardization method 1	23
b. Standardization method 2	26
c. Standardization summary	26

General info

This document presents the cleaning code and summary of the Norwegian bottom trawl survey made available by IMR. It contains data from **1980** and up to **2017**.

Data cleaning in R

```
#####
#### R code to clean trawl survey Norwegian Sea & Barents Sea
#### Public data from IMR
#### Coding: Aurore Maureaud, old code + changes in May 2021
#### Changes in gear and haul duration: October 2022 according to Laurene
#### Update January 2023
#####

rm(list=ls())

#-----
#### LOAD LIBRARIES & CODES #####
#-----
```

```

### Libraries
library(data.table)
library(readxl) # To load extra datasets
library(tidyverse) # for data wrangling
library(janitor) # for cleaning names
library(lubridate) # for fixing dates
library(rnaturalearth) # for removing points from land
library(sf) # for removing points from land
library(sp) # for removing points from land
library(taxize) # for getting correct species names
library(magrittr) # for names wrangling
library(reshape2)
library(googledrive)
library(RODBC)

# Get WoRMS id for sourcing
wrms <- gnr_datasources() %>%
  filter(title == "World Register of Marine Species") %>%
  pull(id)

source("functions/clean_taxa.R")
source("functions/cleanspl.R")
source("functions/write_clean_data.R")
source("functions/apply_trimming_method1.R")
source("functions/apply_trimming_method2.R")
source("functions/flag_spp.R")
fishglob_data_columns <- read_excel("standard_formats/fishglob_data_columns.xlsx")

### Load files

# we create a vector list with the filenames that match with a .csv ending
files = list.files('/Volumes/Elements/fishglob data/Publicly available/NorBTS/btraal/btraal/',
                  pattern="*.csv")

# then we call a lapply function that takes x (every csv) and calls it back to a rbind.
# Check the separator to see if it's correct
norw_dat = do.call(rbind, lapply(files, function(x)
  read.csv(paste('/Volumes/Elements/fishglob data/Publicly available/NorBTS/btraal/btraal/',x,sep=''),
            stringsAsFactors = FALSE, header = TRUE, sep = ";")))
rm(files)

# change colnames from Norwegian to new names in English
setnames(norw_dat, old = c("aar","mnd","lengde","bredde","redskap","starttid","stopptid",
                         "taueT","bunndyp","opening","dist","tilstand","kvalitet",
                         "delnr","akode","art","latin","maal_fangst","fangstKvant",
                         "fangstAnt","maal_lprov","lengdemaal","lengdeProveKv",
                         "lengdeProveAnt","interv","kjonn"),
         new = c("Year","Month","ShootLong","ShootLat","Gear","ShootTimeB","ShootTimeE",
                "HaulDur","Depth","Netopening","Distance","quality_gear","quality_haul",
                "SubSampleNr", "SpecCode", "AkodeName", "ScientificName", "MeasureType",
                "Weight", "NoMeas", "MeasureType2", "LengthMethod",
                "WeightSubSample", "AbundanceSubSample", "Interv", "Sex"))

```

```

#####
##### CREATE HAULD ID #####
#####

# Give survey name
norw_dat$Survey <- rep("NOR-BTS", each=length(unique(rownames(norw_dat)))))

# Haulid
norw_dat$HaulID <- paste(norw_dat$Survey, norw_dat$Year, norw_dat$Month, norw_dat$Gear,
                           norw_dat$ShootLong, norw_dat$ShootLat, norw_dat$Depth,
                           norw_dat$ShootTimeB)

# Recalculate the haul duration because the column has weird values
# start time: ShootTimeB in XYZW where XY are hours from 0 to 24 and ZW are minutes
# from 0 to 59
# end time: ShootTimeE
norw_dat[norw_dat$ShootTimeE==1,]$ShootTimeE <- 'NA'
norw_dat[norw_dat$ShootTimeB==1,]$ShootTimeB <- 'NA'
norw_dat$ShootTimeB <- as.numeric(as.vector(norw_dat$ShootTimeB))
norw_dat$ShootTimeE <- as.numeric(as.vector(norw_dat$ShootTimeE))

times <- data.frame(cbind(norw_dat$HaulID, norw_dat$ShootTimeB, norw_dat$ShootTimeE))
names(times) <- c('HaulID', 'ShootTimeB', 'ShootTimeE')
times <- subset(times, !is.na(times$ShootTimeB))
times <- subset(times, !is.na(times$ShootTimeE))
for(i in 1:ncol(times)){times[,i] <- as.character(times[,i])}
# add 0 as characters to have length 4 of times
times[nchar(times$ShootTimeB)==2,]$ShootTimeB <-
  paste('00',times[nchar(times$ShootTimeB)==2,]$ShootTimeB, sep=' ')
times[nchar(times$ShootTimeB)==3,]$ShootTimeB <-
  paste('0',times[nchar(times$ShootTimeB)==3,]$ShootTimeB, sep=' ')
times[nchar(times$ShootTimeB)==1,]$ShootTimeB <-
  paste('000',times[nchar(times$ShootTimeB)==1,]$ShootTimeB, sep=' ')
times[nchar(times$ShootTimeE)==2,]$ShootTimeE <-
  paste('00',times[nchar(times$ShootTimeE)==2,]$ShootTimeE, sep=' ')
times[nchar(times$ShootTimeE)==3,]$ShootTimeE <-
  paste('0',times[nchar(times$ShootTimeE)==3,]$ShootTimeE, sep=' ')
times[nchar(times$ShootTimeE)==1,]$ShootTimeE <-
  paste('000',times[nchar(times$ShootTimeE)==1,]$ShootTimeE, sep=' ')

# count minutes and hours for begining and end
times$minB <- as.numeric(as.vector(substr(
  times$ShootTimeB, start=1, stop=2)))*60+as.numeric(as.vector(substr(
  times$ShootTimeB, start=3, stop=4)))
times$minE <- as.numeric(as.vector(substr(
  times$ShootTimeE, start=1, stop=2)))*60+as.numeric(as.vector(substr(
  times$ShootTimeE, start=3, stop=4)))
times$duration <- times$minE-times$minB
times[times$minB>1320 & times$minE<120,]$duration <-
  times[times$minB>1320 & times$minE<120,]$minE-times[times$minB>1320 & times$minE<120,]$minB+1440
times[times$minB>1080 & times$minE<420,]$duration <-
  times[times$minB>1080 & times$minE<420,]$minE-times[times$minB>1080 & times$minE<420,]$minB+1440
# all remaining times are too long or start before begining time -> to be removed

```

```

times <- subset(times, times$duration>0)
# let's check the very high times: higher than 8h?
times <- unique(times)
times$ShootTimeB <- times$ShootTimeE <- times$minB <- times$minE <- NULL
setnames(times, old='duration', new='HaulDur2')

# join back with norw_dat
norw_dat0 <- left_join(norw_dat, times, by='HaulID')
nrow(norw_dat)==nrow(norw_dat0)
norw_dat <- norw_dat0

#####
##### SELECT GEAR TYPES #####
#####

# Keep only hauls done with "correct" gear types. Keeping shrimp trawls
keep_gear <- c("3236", #Campelen 1800 shrimp trawl with 35 mm mesh Reketrål. Campelen 1800 ma 35 mm m/4
              "3270", #Campelen 1800 shrimp trawl with 22mm mesh size. Reketrål. Campelen 1800 ma 20 mm m/4
              "3271" #Like 3270 with strapping Reketrål. Campelen 1800 ma 20 mm m/40 m sveiper. Rockh
)
rm(keep_gear)

#####
##### REMOVE BAD QUALITY HAULS #####
#####

# Remove bad quality hauls and gears
norw_dat <- subset(norw_dat, norw_dat$quality_gear %in% c(1,2))
norw_dat <- subset(norw_dat, norw_dat$quality_haul %in% c(1,2))

# Is there still empty species names and abundances?
check.sp <- subset(norw_dat, norw_dat$ScientificName=='')
# all hauls from 1981 and 1982 with no ab/weight/spp specified
norw_dat <- subset(norw_dat, norw_dat$ScientificName!= '') # remove rows with empty rows

check.ab <- subset(norw_dat, is.na(norw_dat$NoMeas)) # ok
check.sub.ab <- subset(norw_dat, is.na(norw_dat$AbundanceSubSample))
check.sum <- subset(norw_dat, is.na(norw_dat$Sum))
check.sub.w <- subset(norw_dat, is.na(norw_dat$WeightSubSample)) # same as abundance

#####
##### STANDARDIZE UNITS AND REMOVE NEGATIVE VALUES #####
#####

# HaulDuration: if the range 1-60m then minutes. If 0-1, in hours
# ICES data in minutes, convert all in minutes 1h <-> 60min
# -1, data unavailable, so insert NA

norw_dat[norw_dat$HaulDur<=1,$HaulDur <- norw_dat[norw_dat$HaulDur<=1,$HaulDur*60

```

```

norw_dat[norw_dat$HaulDur<10 & norw_dat$Distance>2,]$HaulDur <-
  norw_dat[norw_dat$HaulDur<10 & norw_dat$Distance>2,]$HaulDur*60
norw_dat[norw_dat$HaulDur<0,]$HaulDur <- NA

# Transform distance nautical miles to km
# 1nm <-> 1.852km

norw_dat$Distance <- norw_dat$Distance*1.852/1
norw_dat[norw_dat$Distance<0,]$Distance <- NA

# Change net opening to DoorSpread
setnames(norw_dat, old = "Netopening", new="DoorSpread")
norw_dat[norw_dat$DoorSpread<0,]$DoorSpread <- NA
norw_dat$DoorSpread <- norw_dat$DoorSpread/1000 # transform m into km

# Transform abundance and weight into the same units, transform weight measures all in kg
# for column Weight, use MeasureType
# for column NoMeas, use MeasureType as is category 6, *1000 individuals
# No document for conversion factors from L weight measurements!!!
# No liters measurements after 2001, so ok if we only select from 2005
# Two rows are in MeasureType or MeasureType==6, but in 1993 and 1995, so will be removed
norw_dat[norw_dat$MeasureType==5,]$Weight <- norw_dat[norw_dat$MeasureType==5,]$Weight*1000
norw_dat[norw_dat$MeasureType==6,]$Weight <- norw_dat[norw_dat$MeasureType==6,]$Weight*1000*1000
norw_dat[norw_dat$MeasureType==6,]$NoMeas <- norw_dat[norw_dat$MeasureType==6,]$NoMeas*1000
norw_dat[norw_dat$MeasureType==7,]$Weight <- norw_dat[norw_dat$MeasureType==7,]$Weight*1000
norw_dat[norw_dat$MeasureType==8,]$Weight <- norw_dat[norw_dat$MeasureType==8,]$Weight*1000
norw_dat[norw_dat$MeasureType==9,]$Weight <- norw_dat[norw_dat$MeasureType==9,]$Weight/1000

# Correction factors for gutted/without head and L transfo. might exist, but cannot find it

# Transform units from the sub-samples not possible because of NAs
norw_dat[is.na(norw_dat$WeightSubSample),]$WeightSubSample <- -1
norw_dat[is.na(norw_dat$AbundanceSubSample),]$AbundanceSubSample <- -1

norw_dat[norw_dat$MeasureType2==5,]$WeightSubSample <-
  norw_dat[norw_dat$MeasureType2==5,]$WeightSubSample*1000
norw_dat[norw_dat$MeasureType2==6,]$WeightSubSample <-
  norw_dat[norw_dat$MeasureType2==6,]$WeightSubSample*1000*1000
norw_dat[norw_dat$MeasureType2==6,]$AbundanceSubSample <-
  norw_dat[norw_dat$MeasureType2==6,]$AbundanceSubSample*1000
norw_dat[norw_dat$MeasureType2==7,]$WeightSubSample <-
  norw_dat[norw_dat$MeasureType2==7,]$WeightSubSample*1000
norw_dat[norw_dat$MeasureType2==8,]$WeightSubSample <-
  norw_dat[norw_dat$MeasureType2==8,]$WeightSubSample*1000
norw_dat[norw_dat$MeasureType2==9,]$WeightSubSample <-
  norw_dat[norw_dat$MeasureType2==9,]$WeightSubSample/1000

# Replace all -1 by NAs
norw_dat[norw_dat$WeightSubSample==(-1),]$WeightSubSample <- NA
norw_dat[norw_dat$AbundanceSubSample==(-1),]$AbundanceSubSample <- NA
norw_dat[norw_dat$Weight==(-1),]$Weight <- NA
norw_dat[norw_dat$NoMeas==(-1),]$NoMeas <- NA

```

```

#####
##### COMPUTE MISSING SWEEPED AREAS
#####

# reshape format with length measurements and delete 0
# a simple pivot_longer does not work because the vector is too long
# with lists and rbindlist function!
lengths <- colnames(norw_dat)[29:69]
norw_dat_list <- list()
for(i in 1:length(lengths)){
  xx <- norw_dat %>%
    select(names(norw_dat)[1:28], names(norw_dat)[70:73], lengths[i]) %>%
    mutate(NumLen = get(lengths[i]),
          Length = lengths[i]) %>%
    select(-lengths[i]) %>%
    filter(NumLen!="NA")
  norw_dat_list[[length(norw_dat_list)+1]] <- xx
  rm(xx)
}
norw_dat <- rbindlist(norw_dat_list)

# Estimate missing swept areas
norw_dat <- norw_dat %>%
  mutate(Area.swept = DoorSpread*Distance) %>%
  filter(!is.na(HaulDur))

nor <- norw_dat %>%
  select(HaulID, Year, Area.swept, HaulDur, Gear, Depth, Distance) %>%
  distinct()

par(mfrow=c(1,2))
plot(Area.swept ~ HaulDur, data=nor)
plot(Area.swept ~ Depth, data=nor)

nor$Dur2 <- (nor$HaulDur-mean(nor$HaulDur))^2
lm0 <- lm(Area.swept ~ HaulDur + Dur2, data=nor) # 68% of data variability explained

pred0 <- predict(lm0, newdata=nor, interval='confidence', level=0.95)
nor <- cbind(nor,pred0)
nor[is.na(nor$Area.swept),]$Area.swept <- nor[is.na(nor$Area.swept),]$fit

nor <- nor %>%
  select(HaulID, Area.swept) %>%
  dplyr::rename(Area2=Area.swept) %>%
  filter(Area2>=0)

nor2 <- left_join(norw_dat, nor, by='HaulID')
nor2 <- nor2 %>%
  mutate(Area.swept = coalesce(Area.swept,Area2))
norw_dat <- nor2
rm(nor, nor2, lm0, pred0, norw_dat_list, lengths)

```

```

#####
##### CHANGE FORMAT FOR FISHGLOB
#####

# Continue cleaning
norw_dat <- norw_dat %>%
  mutate(quarter = ceiling(as.numeric(Month)/3),
         num_cpue = NoMeas/Area.swept, # nbr / km2
         wgt_cpue = Weight/Area.swept, # kg / km2
         num_h = NoMeas*60/HaulDur2, # nbr / hour
         wgt_h = Weight*60/HaulDur2, # kg / h
         num = NoMeas,
         wgt = Weight,
         survey = 'Nor-BTS',
         season = 'NA',
         sbt=NA,
         sst=NA,
         haul_dur = HaulDur2/60,
         country = "norway",
         continent = "europe",
         stat_rec = NA_character_,
         station = NA_character_,
         stratum = NA_character_,
         day = NA,
         aphia_id = NA_character_,
         sub_area = NA_character_,
         sub_factor_ab = NoMeas/Sum, # compute sub factor
         num_len = NumLen*sub_factor_ab # raise abundance at length to total haul
       ) %%
  rename(haul_id = HaulID,
         year = Year,
         month = Month,
         latitude = ShootLat,
         longitude = ShootLong,
         gear = Gear,
         depth = Depth,
         area_swept = Area.swept,
         scientific_name= ScientificName) %>%
  filter(#haul_dur>(14/60),
         haul_dur<(120/60),
         !is.na(haul_dur)) %>%
  select(-Area2, -HaulDur2, -LengthMethod, -Max, -Min, -Interv, -MeasureType,
         -MeasureType2)

#####
##### CLEAN SPECIES NAMES
#####

# Set Survey code
survey_code <- "NOR-BTS"

norw_dat <- norw_dat %>%

```

```

mutate(
  scientific_name = str_to_sentence(str_to_lower(scientific_name)),
  scientific_name = cleanspl(scientific_name),
  scientific_name = str_trim(scientific_name, side = "right"),
  scientific_name = gsub(" | ", "", scientific_name)
) %%
filter(scientific_name!="")

norw_names <- norw_dat %>%
  group_by(scientific_name) %>%
  summarize(n=n()) %>%
  pull(scientific_name)

# There are too many so let's remove fishsealife first
norw_fish_names <- rfishbase::species(norw_names, server = "sealifebase") %>%
  dplyr::select(SpecCode,
                Species) %>%
  filter(is.na(SpecCode)) %>%
  pull(Species) %>%
  unique()

clean_auto <- clean_taxa(norw_fish_names, input_survey = survey_code, save = F,
                           fishbase = T)

#### Clean manual
missing_taxa <- setdiff(norw_fish_names, clean_auto$query)
missing_taxa[missing_taxa=="Notoscopelus elongatusi"] <- "Notoscopelus elongatus"

alphaid <- get_wormsid(missing_taxa)
alphaid <- tibble(taxa = missing_taxa,
                  worms_id = alphaid[1:length(missing_taxa)])

clean_manual <- clean_taxa(alphaid$worms_id, input_survey = survey_code,
                           save = F, fishbase = T)

#-----
##### INTEGRATE CLEAN TAXA in NOR-BTS survey data #####
#-----


clean_taxon <- rbind(clean_auto, clean_manual) %>%
  select(-survey) %%
  filter(rank %in% c("Species", "Genus", "Family", "Subspecies"))

clean_norw <- left_join(norw_dat, clean_taxon, by=c("scientific_name" = "query")) %>%
  filter(!is.na(taxa)) %>%
  rename(verbatim_name = scientific_name,
        accepted_name = taxa,
        SpecCode = SpecCode.y) %>%
  mutate(verbatim_aphia_id = NA)

```

```

#-
##### GET LENGTH-WEIGHTS RELATIONSHIPS #####
#-

#coeffs_nor <- get_coeffs(clean_taxon, survey = survey_code, save=T)

#-
##### RE-CALCULATE WEIGHTS BASED ON CLEANED SPP NAMES #####
#-

# open length-weight relationship coeff file
datalw <- read.csv("length_weight/length.weight_NOR-BTS.csv") %>%
  select(taxa, a, b) %>%
  distinct()

# add length-weight relationships to norw_dat
length_now <- sort(unique(clean_norw$Length))
clean_norw <- left_join(clean_norw, datalw, by=c("accepted_name" = "taxa")) %>%
  mutate(length = case_when(Length==length_now[1] ~ 4.5,
                            Length==length_now[2] ~ 12,
                            Length==length_now[3] ~ 17,
                            Length==length_now[4] ~ 22,
                            Length==length_now[5] ~ 27,
                            Length==length_now[6] ~ 32,
                            Length==length_now[7] ~ 37,
                            Length==length_now[8] ~ 42,
                            Length==length_now[9] ~ 47,
                            Length==length_now[10] ~ 52,
                            Length==length_now[11] ~ 57,
                            Length==length_now[12] ~ 62,
                            Length==length_now[13] ~ 67,
                            Length==length_now[14] ~ 72,
                            Length==length_now[15] ~ 77,
                            Length==length_now[16] ~ 82,
                            Length==length_now[17] ~ 87,
                            Length==length_now[18] ~ 92,
                            Length==length_now[19] ~ 97,
                            Length==length_now[20] ~ 102,
                            Length==length_now[21] ~ 107,
                            Length==length_now[22] ~ 112,
                            Length==length_now[23] ~ 117,
                            Length==length_now[24] ~ 122,
                            Length==length_now[25] ~ 127,
                            Length==length_now[26] ~ 132,
                            Length==length_now[27] ~ 137,
                            Length==length_now[28] ~ 142,
                            Length==length_now[29] ~ 147,
                            Length==length_now[30] ~ 152,
                            Length==length_now[31] ~ 157,
                            Length==length_now[32] ~ 162,
                            Length==length_now[33] ~ 167,
                            Length==length_now[34] ~ 172,

```

```

        Length==length_now[35] ~ 177,
        Length==length_now[36] ~ 182,
        Length==length_now[37] ~ 200,
    ),
    wgt_len = num_len*a*(length^b)/1000,
    wgt_len_h = wgt_len/haul_dur*60,
    wgt_len_cpue = wgt_len/area_swept,
    num_len_h = num_len/haul_dur*60,
    num_len_cpue = num_len/area_swept) %>%
# aggregate values per length
group_by(survey, haul_id, country, sub_area, continent, stat_rec, station, stratum,
year, month, day, quarter, season, latitude, longitude, haul_dur,
area_swept, gear, depth, sbt, sst, num, num_h, num_cpue,
wgt, wgt_h, wgt_cpue, verbatim_name, verbatim_aphia_id, accepted_name,
aphia_id, SpecCode, kingdom, phylum, class, order, family, genus, rank, Sex) %>%
summarise_at(.vars = c("num_len", "num_len_h", "num_len_cpue",
                       "wgt_len", "wgt_len_h", "wgt_len_cpue"),
             .funs = function(x) sum(x, na.rm=T)) %>%
# aggregate values per sex
group_by(survey, haul_id, country, sub_area, continent, stat_rec, station, stratum,
year, month, day, quarter, season, latitude, longitude, haul_dur,
area_swept, gear, depth, sbt, sst,
verbatim_name, verbatim_aphia_id, accepted_name,
aphia_id, SpecCode, kingdom, phylum, class, order, family, genus, rank, Sex) %>%
summarise_at(.vars = c("num_len", "num_len_h", "num_len_cpue",
                       "wgt_len", "wgt_len_h", "wgt_len_cpue",
                       "num", "num_h", "num_cpue",
                       "wgt", "wgt_h", "wgt_cpue"),
             .funs = function(x) sum(x, na.rm=T))

# check relationships between total weight and re-calculated weights
ggplot(clean_norw) + geom_point(aes(x = num, y = num_len)) +
  geom_abline(intercept = 0, slope = 1, color="red",
              linetype="dashed", size=1.5) # looks great

ggplot(clean_norw) + geom_point(aes(x = wgt, y = wgt_len)) +
  geom_abline(intercept = 0, slope = 1, color="red",
              linetype="dashed", size=1.5)

# cannot be linear because wgt is missing a lot of values
clean_norw <- clean_norw %>%
  select(-num, -num_h, -num_cpue, -wgt, -wgt_h, -wgt_cpue) %>%
  rename(num = num_len,
         num_h = num_len_h,
         num_cpue = num_len_cpue,
         wgt = wgt_len,
         wgt_h = wgt_len_h,
         wgt_cpue = wgt_len_cpue) %>%
  mutate(source = "IMR",
         timestamp = "2018",
         num_cpua = num_cpue,
         num_cpue = num_h,

```

```

wgt_cpua = wgt_cpue,
wgt_cpue = wgt_h,
survey_unit = ifelse(survey %in% c("BITS", "NS-IBTS", "SWC-IBTS"),
                     paste0(survey, "-", quarter), survey),
survey_unit = ifelse(survey %in% c("NEUS", "SEUS", "SCS", "GMEX"),
                     paste0(survey, "-", season), survey_unit)) %>%
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 = clean_norw, survey = survey_code, overwrite = T)

# -----
##### FAGS #####
# -----

#install required packages that are not already installed
required_packages <- c("data.table",
                      "devtools",
                      "dgridR",
                      "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(clean_norw$survey))

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

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

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

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

#-----#
#### ADD STANDARDIZATION FLAGS ####
#-----#

surveys <- sort(unique(clean_norw$survey))
survey_units <- sort(unique(clean_norw$survey_unit))
survey_std <- clean_norw %>%
  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")){
    xx <- data.frame(read_delim(paste0("outputs/Flags/taxonomic_flagging/",
                                         surveys[i], "_flagsppl.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)
}

# Just run this routine should be good for all
write_clean_data(data = survey_std, survey = "NOR-BTS_std",
                 overwrite = T, rdata=TRUE)

```

1. Overview of the survey data table

survey	source	timestamp	haul_id		country	sub_area
Nor-BTS	IMR	2018	NOR-BTS 1982 9 3271 18.567 75.5 119 1300		norway	NA
Nor-BTS	IMR	2018	NOR-BTS 1982 9 3271 18.567 75.5 119 1300		norway	NA
Nor-BTS	IMR	2018	NOR-BTS 1982 9 3271 18.567 75.5 119 1300		norway	NA
Nor-BTS	IMR	2018	NOR-BTS 1982 9 3271 18.567 75.5 119 1300		norway	NA
Nor-BTS	IMR	2018	NOR-BTS 1982 9 3271 18.567 75.5 119 1300		norway	NA

continent	stat_rec	station	stratum	year	month	day	quarter	season
europe	NA	NA	NA	1982	9	NA	3	NA
europe	NA	NA	NA	1982	9	NA	3	NA
europe	NA	NA	NA	1982	9	NA	3	NA
europe	NA	NA	NA	1982	9	NA	3	NA
europe	NA	NA	NA	1982	9	NA	3	NA

latitude	longitude	haul_dur	area_swept	gear	depth	sbt	sst
75.5	18.567	0.5	0.120251	3271	119	NA	NA
75.5	18.567	0.5	0.120251	3271	119	NA	NA
75.5	18.567	0.5	0.120251	3271	119	NA	NA
75.5	18.567	0.5	0.120251	3271	119	NA	NA
75.5	18.567	0.5	0.120251	3271	119	NA	NA

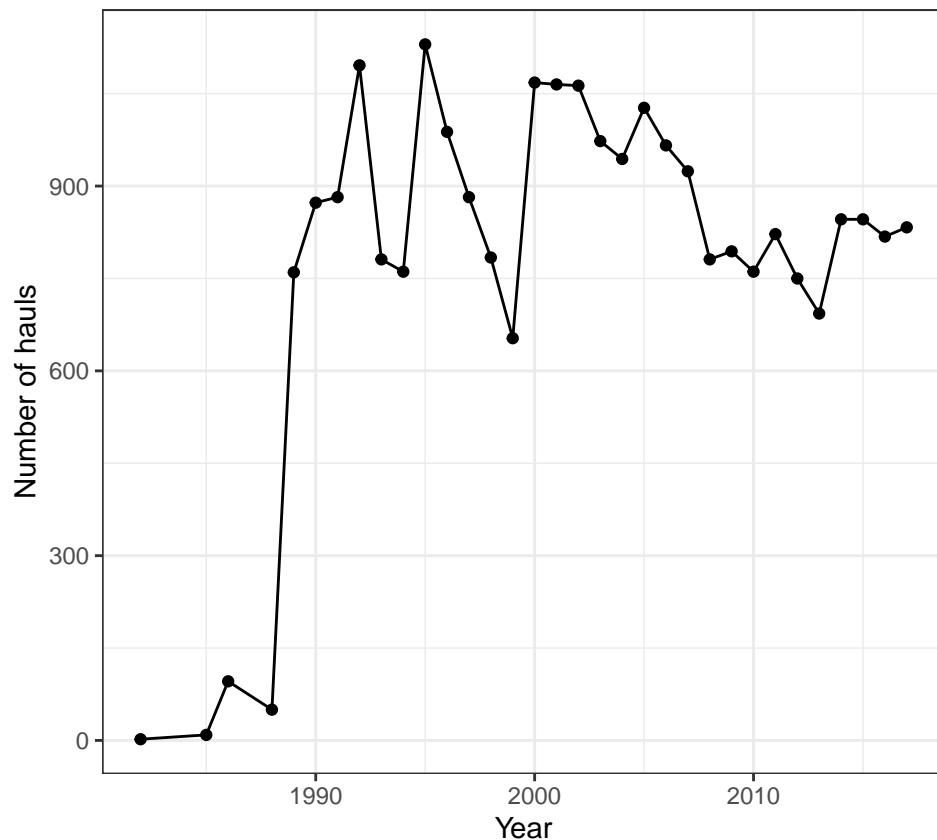
num	num_cpue	num_ccpuia	wgt	wgt_cpue	wgt_ccpuia	verbatim_name
3	360	24.947811	2.1252027	255.02432	17.6730519	Anarhichas lupus
1	120	8.315937	0.1824977	21.89973	1.5176397	Anarhichas minor
1	120	8.315937	0.0968518	11.62221	0.8054133	Eumicrotremus spinosus
585	70200	4864.823172	452.2851868	54274.22242	3761.1751404	Gadus morhua
1600	192000	13305.499273	5.8142799	697.71359	48.3511855	Hippoglossoides platessoides

verbatim_aphia_id	accepted_name	aphia_id	SpecCode	kingdom
NA	Anarhichas lupus	NA	2501	Animalia
NA	Anarhichas minor	NA	3811	Animalia
NA	Eumicrotremus spinosus	NA	4185	Animalia
NA	Gadus morhua	NA	69	Animalia
NA	Hippoglossoides platessoides	NA	4239	Animalia

phylum	class	order	family	genus	rank	survey_unit
Chordata	Teleostei	Perciformes	Anarhichadidae	Anarhichas	Species	Nor-BTS
Chordata	Teleostei	Perciformes	Anarhichadidae	Anarhichas	Species	Nor-BTS
Chordata	Teleostei	Perciformes	Cyclopteridae	Eumicrotremus	Species	Nor-BTS
Chordata	Teleostei	Gadiformes	Gadidae	Gadus	Species	Nor-BTS
Chordata	Teleostei	Pleuronectiformes	Pleuronectidae	Hippoglossoides	Species	Nor-BTS

2. Summary of sampling intensity

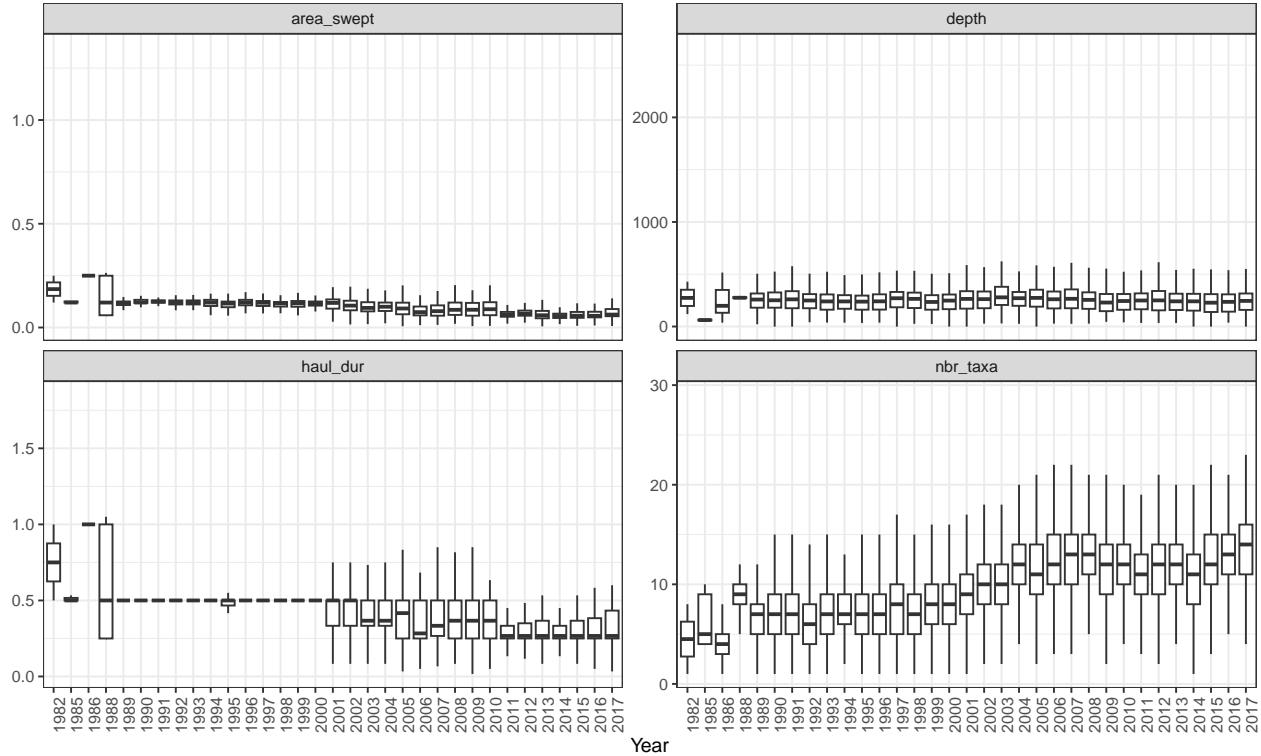
Number of hauls per year performed during the survey after data processing.



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:

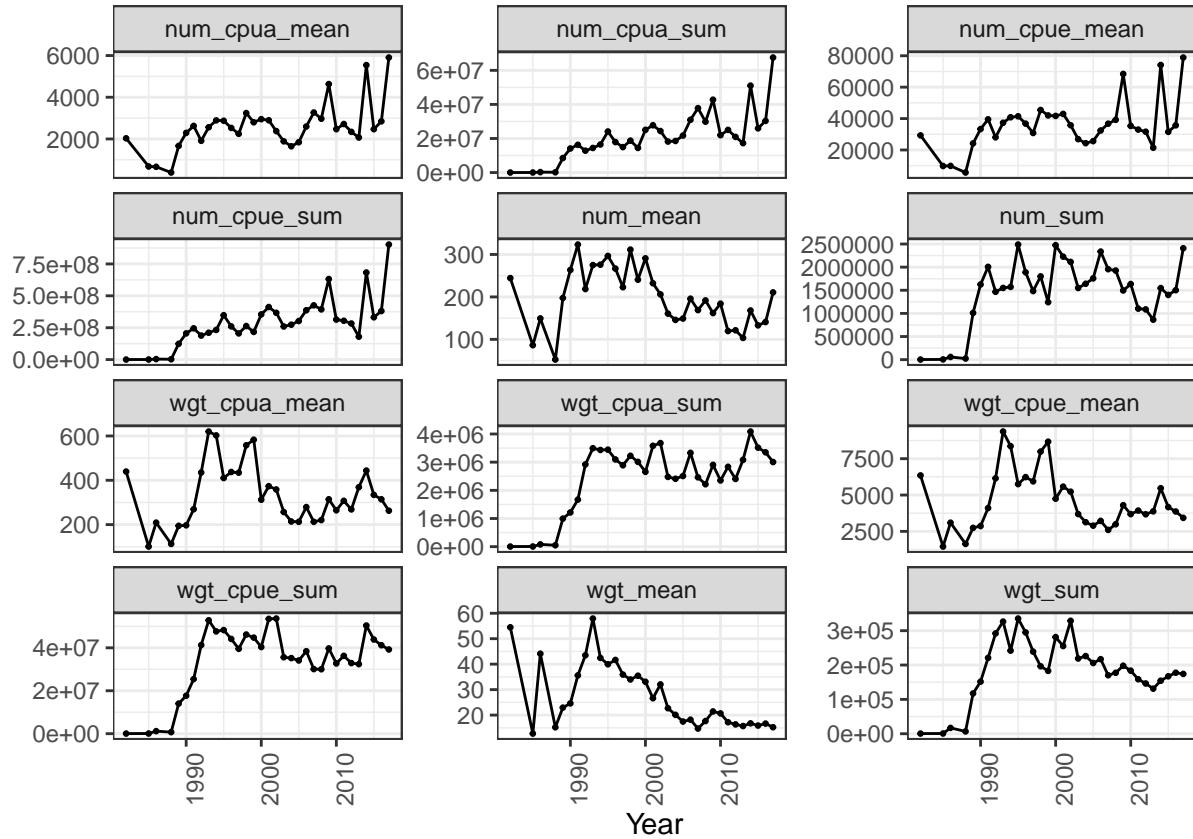
- *area_swept*, swept area by the bottom trawl gear km^2
- *depth*, sampling depth in m
- *haul_dur*, haul sampling duration *hours*
- *number of marine fish taxa*, taxa were cleaned following the last version of taxonomy from the World Register of Marine Species (<https://www.marinespecies.org/>, October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



4. Summary of biological variables

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

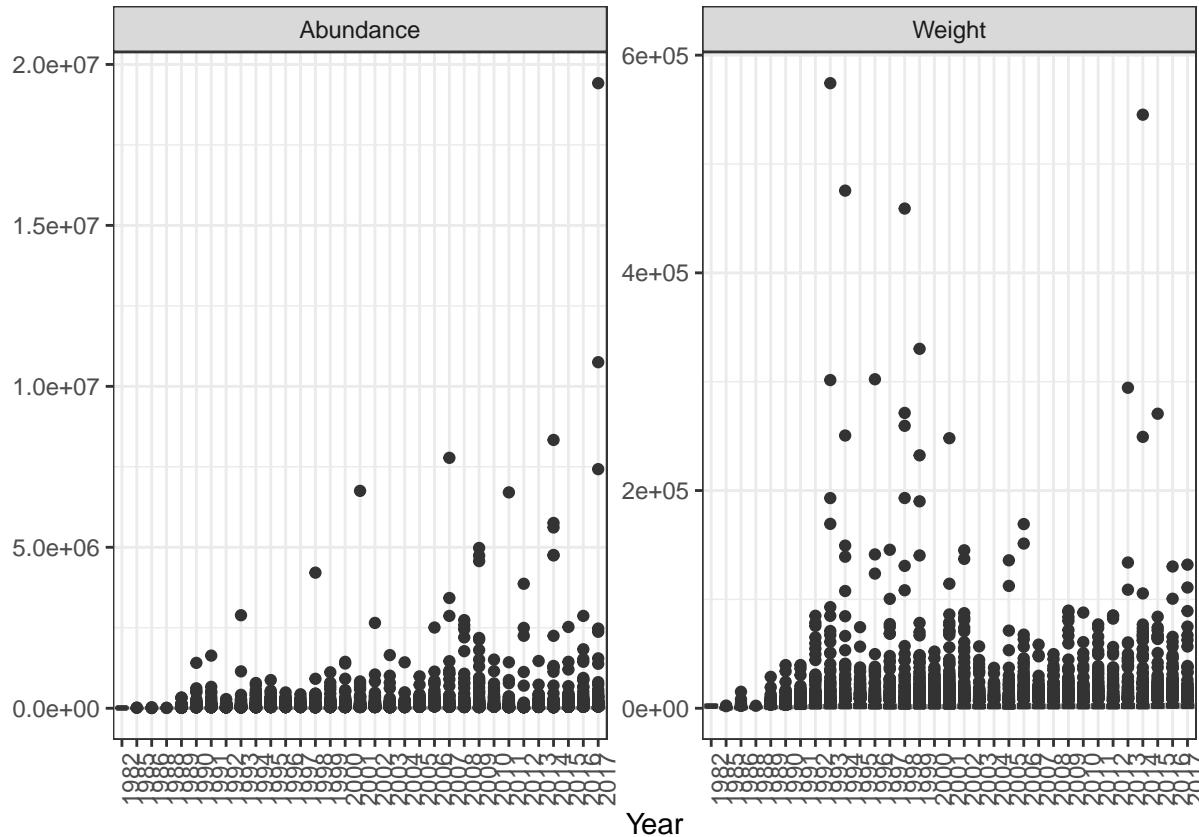
- num_cpua , number of individuals (abundance) in $\frac{individuals}{km^2}$
- num_cpue , number of individuals (abundance) in $\frac{individuals}{h}$
- num , number of individuals (abundance)
- wgt_cpua , weight in $\frac{kg}{km^2}$
- wgt_cpue , weight in $\frac{kg}{h}$
- wgt , weight in kg



5. Extreme values

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

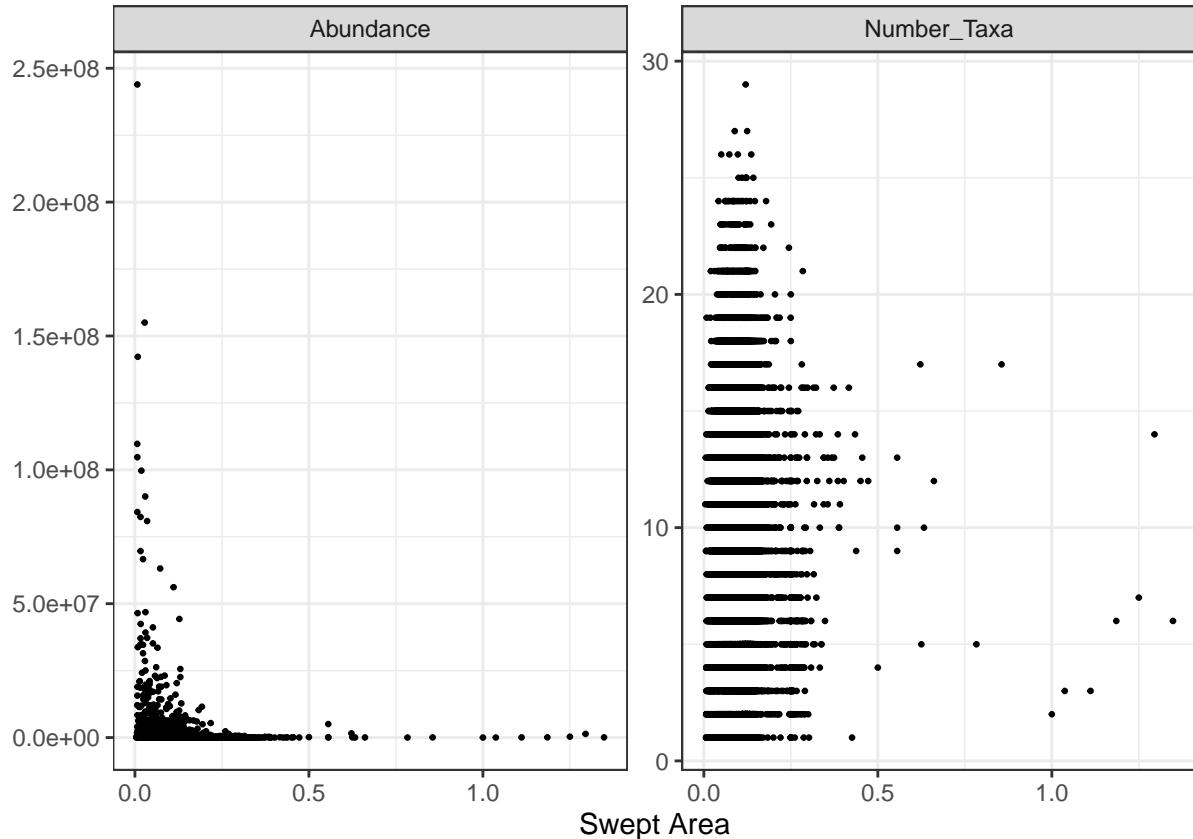
- num_cpue , number of individuals (abundance) in $\frac{individuals}{km^2}$
- wgt_cpue , weight in $\frac{kg}{km^2}$



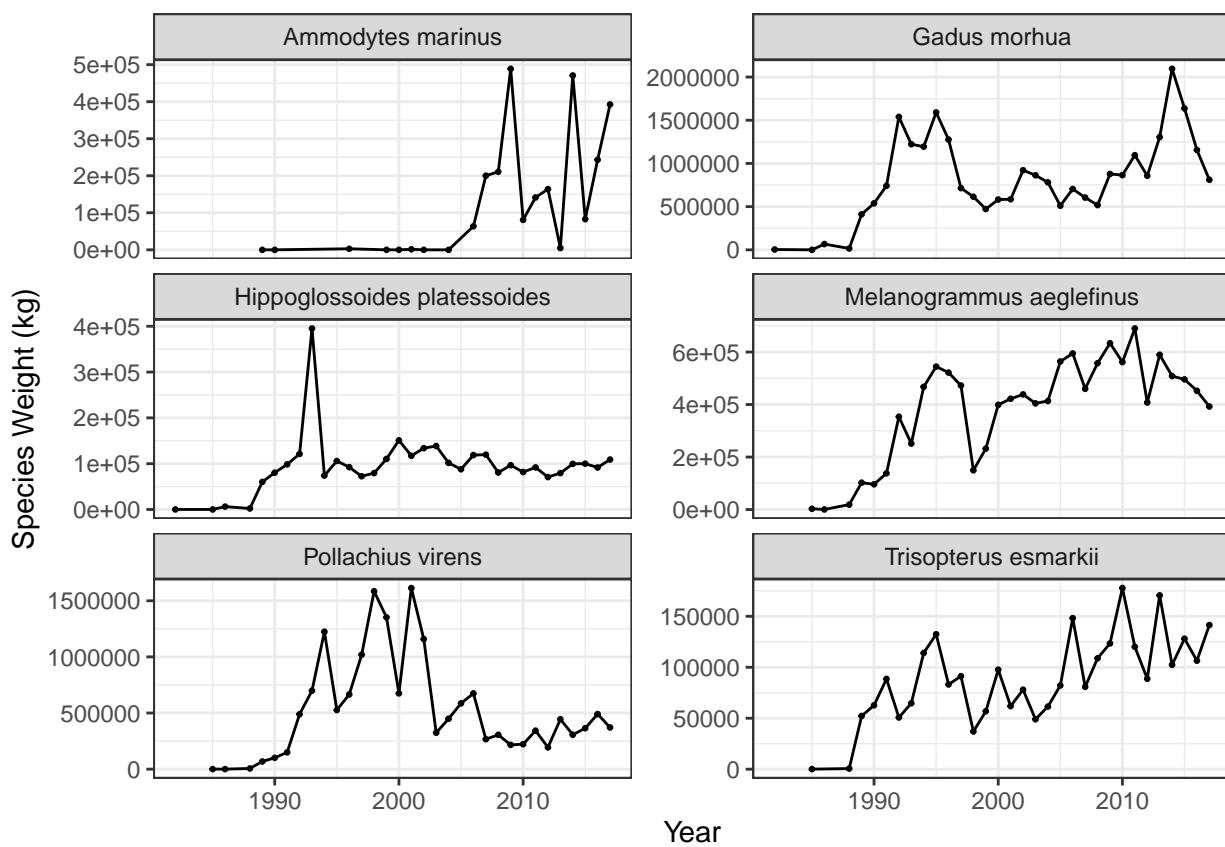
6. Summary of variables against swept area

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_cpua , number of individuals (abundance) in $\frac{individuals}{km^2}$
- wgt_cpua , weight in $\frac{kg}{km^2}$

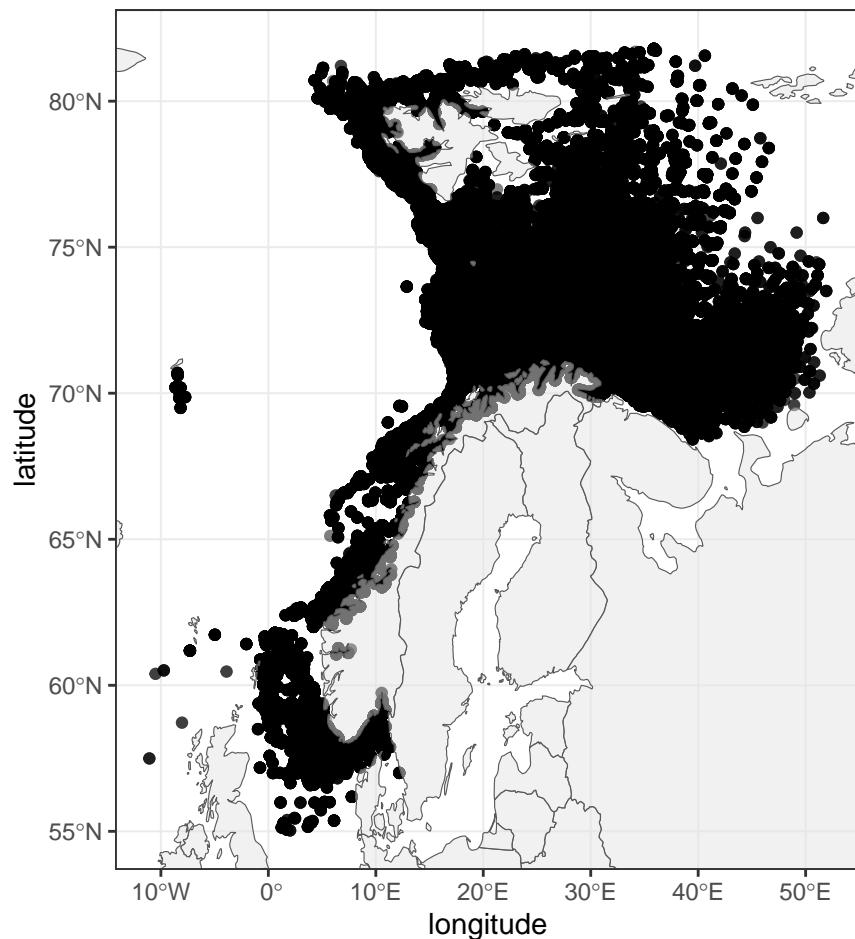


7. Abundance or Weight trends of the six most abundant species



8. Distribution mapping

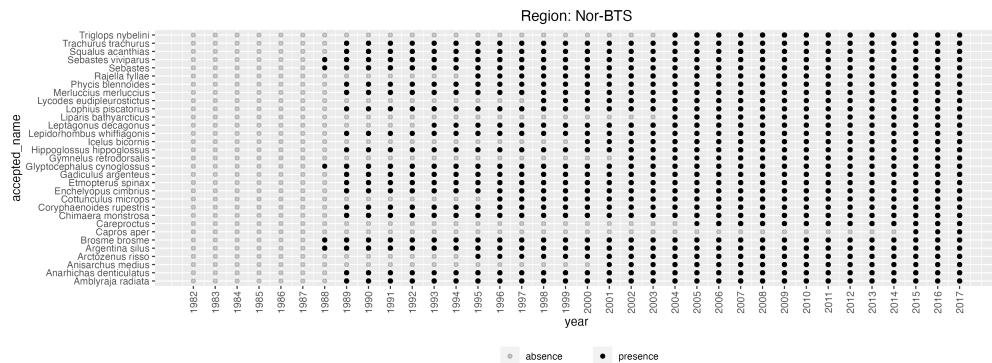
Map of the sampling distribution in space. Note that we only show one year per coordinate.



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

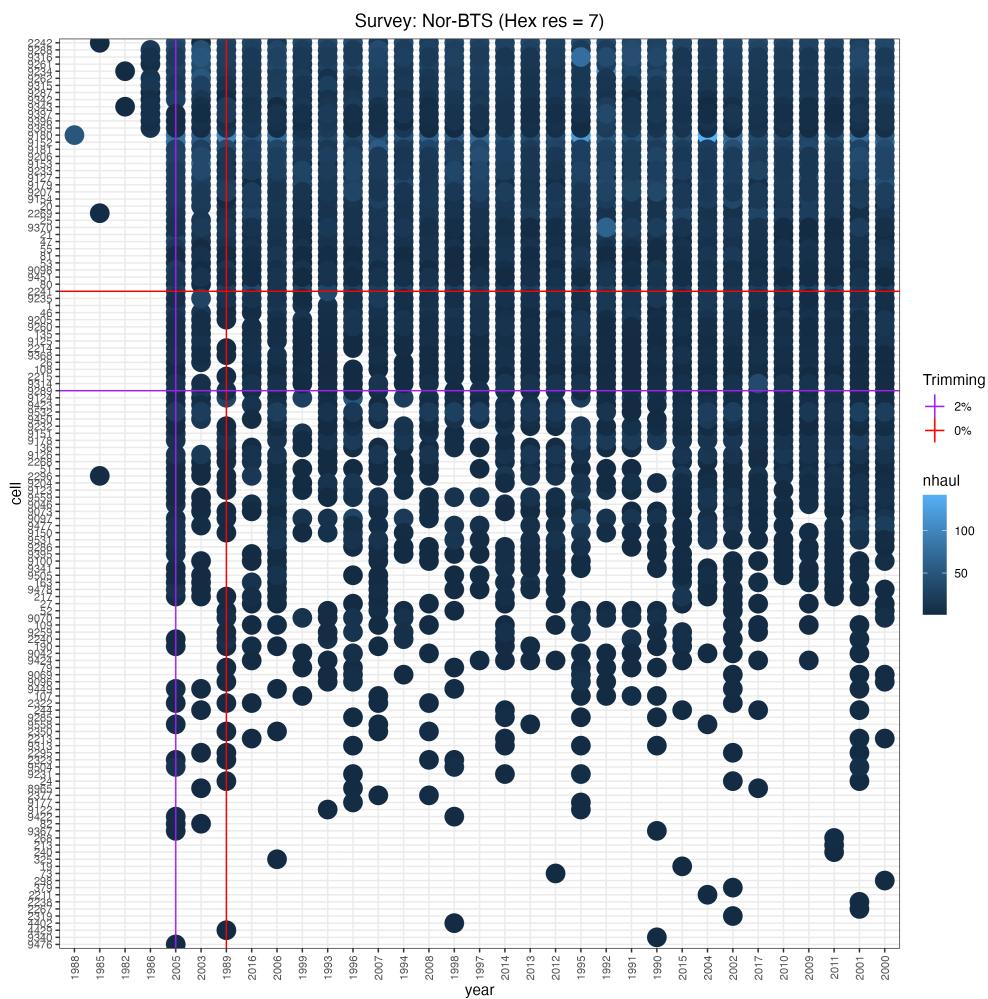
Total number of species	228.0
Percentage of species flagged	13.6

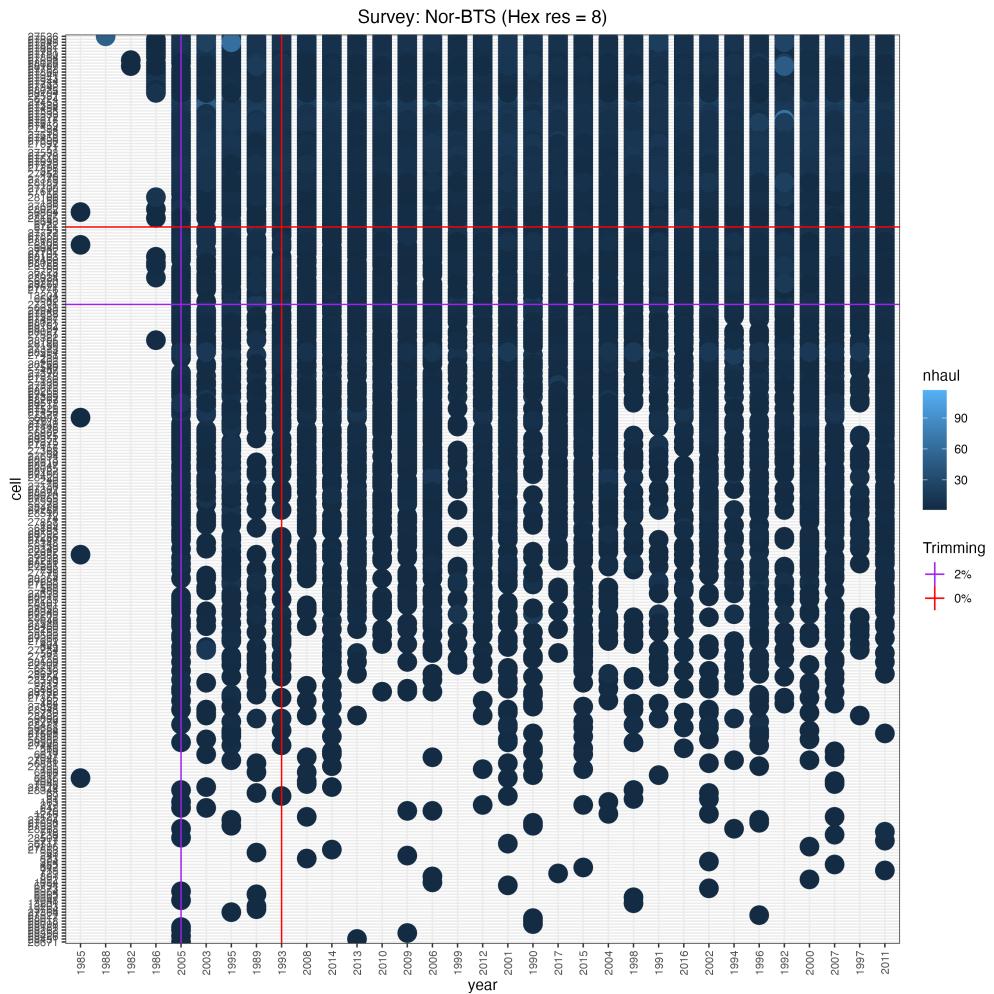
10. Spatio-temporal standardization

a. Standardization method 1

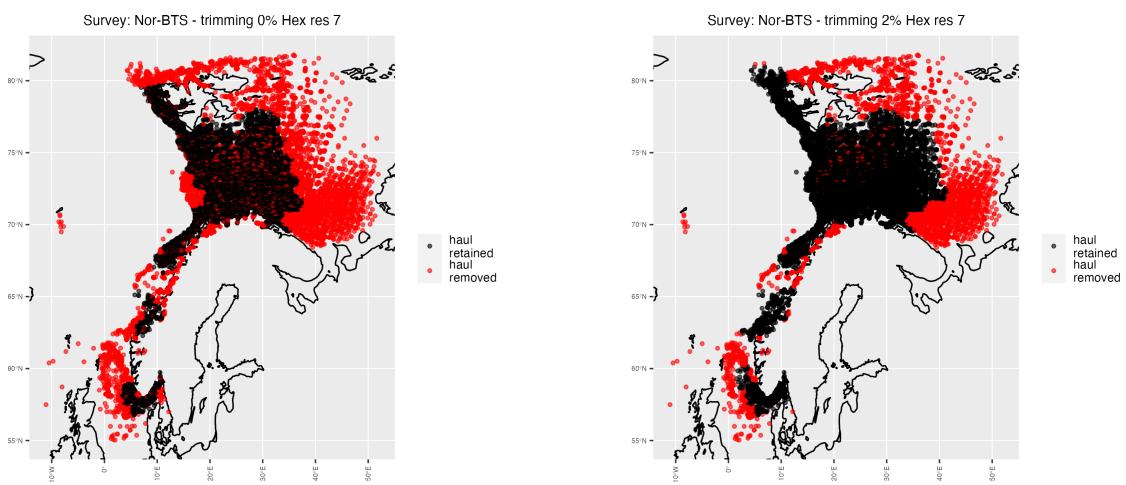
This standardization method was adapted from https://github.com/zoekitchel/trawl_spatial_turnover/blob/master/data_prep_code/species/explore_NorthSea_trimming.Rmd
It was run for hex resolution 7 and 8.

Plot of number of cells x years with overlaid flagging options

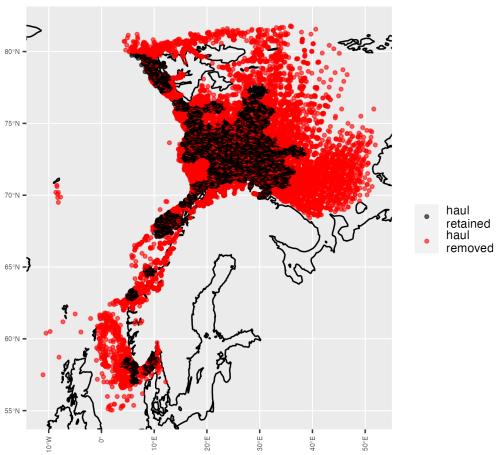




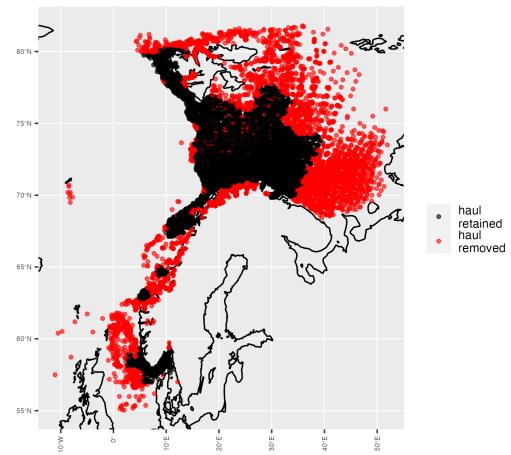
Map of hauls retained and removed per flagging method and threshold



Survey: Nor-BTS - trimming 0% Hex res 8

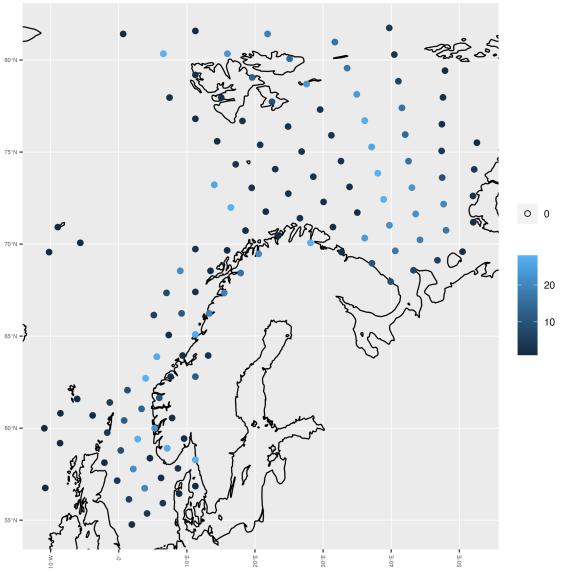


Survey: Nor-BTS - trimming 2% Hex res 8

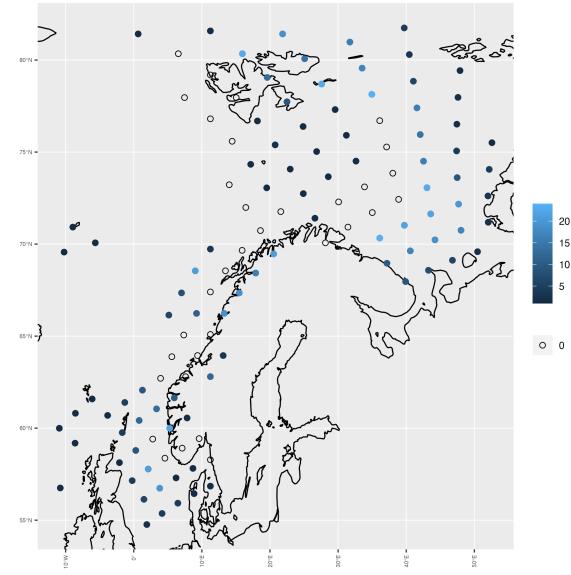


Map of numbers of years removed per grid cell and flagging method/threshold

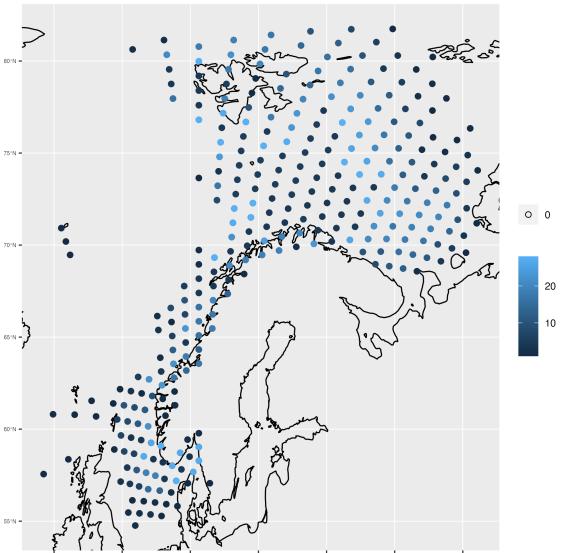
Survey: Nor-BTS - trimming 0% Hex res 7 - nb yrs removed



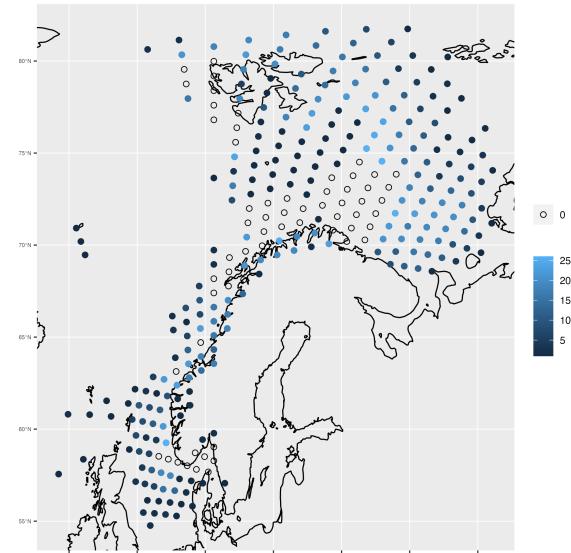
Survey: Nor-BTS - trimming 2% Hex res 7 - nb yrs removed



Survey: Nor-BTS - trimming 0% Hex res 8 - nb yrs removed



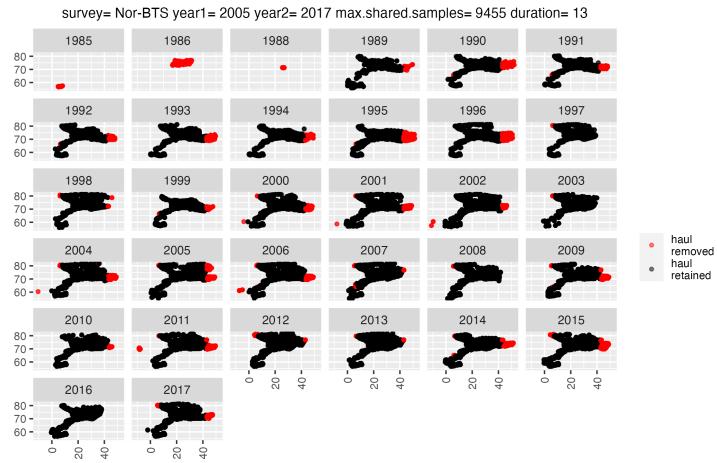
Survey: Nor-BTS - trimming 2% Hex res 8 - nb yrs removed



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	8322.0	4359.0	12452.0	7051.0	49813.0
percentage of hauls removed	32.4	16.9	48.4	27.4	1.8