

NOR-BTS: Norwegian bottom trawl survey data processing summary

fishglob, Aurore A. Maureaud & Juliano Palacios Abrantes

November, 2023

Contents

| | |
|--|----|
| General info | 1 |
| Data cleaning in R | 1 |
| 1. Overview of the survey data table | 12 |
| 2. Summary of sampling intensity | 13 |
| 3. Summary of sampling variables from the survey | 14 |
| 4. Summary of biological variables | 15 |
| 5. Extreme values | 16 |
| 6. Summary of variables against swept area | 17 |
| 7. Abundance or Weight trends of the six most abundant species | 18 |
| 8. Distribution mapping | 19 |
| 9. Taxonomic flagging | 20 |
| 10. Spatio-temporal standardization | 21 |
| a. Standardization method 1 | 21 |
| b. Standardization method 2 | 24 |
| c. Standardization summary | 24 |

General info

This document presents the cleaning code and summary of the Norwegian Barents Sea bottom trawl survey made available by IMR. It contains data from **2004** and up to **2019**.

Data cleaning in R

```
#####  
#### R code to clean trawl survey Barents Sea  
#### Public data from IMR at NMDC "ecosystem survey fish diversity" datasets  
#### Coding: Aurore Maureaud, old code + changes in May 2021  
#### Changes in gear and hauld duration: October 2022 according to Laurene  
#### Update January 2023  
#### Changes of input survey data Aug 23. The Norwegian surveys within the FishGlob  
####     compiled data prior to Sep 23 SHOULD NOT BE USED for scientific analysis due  
####     to mistakes in old input survey data. The correct data to be used  
####     from IMR NMDC is called "Barents Sea ecosystem survey fish diversity data export"  
####     with one dataset per year. Only these surveys should be used.  
#####  
#### Updates  
####     Juliano Palacios  
####     Spetember 5, 2023
```

```

#### Update in response to Issue #23
# Note: I was not able to fix the code from the root as the data is not in the repo. See line 356 for a

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)
library(here)

# Get WoRM's id for sourcing
worm <- 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
## IMR ecosystem survey data obtained in NMDC nmhc.no
## The files are called "Barents Sea ecosystem survey fish diversity data export XX"
## The ecosystem survey is done by IMR, the version of the datasets can be changed if
## IMR find error in the code. Visit NMDC to get the latest version and credit IMR.
## Each year have a different file. In the downloaded dataset we use the file under
## output/report/ReportSpeciesCategoryCatch_Count and output/report/ReportSpeciesCategoryCatch_weight

# we create a vector list with the filenames that match with a .txt ending
files_ct = list.files('C:/Users/lpe116/OneDrive - UiT Office 365/FishGlob/IMR Surveys/Count/',
                      pattern="*.txt")
files_wg = list.files('C:/Users/lpe116/OneDrive - UiT Office 365/FishGlob/IMR Surveys/Weight/',
                      pattern="*.txt")

# Aurore's path

```

```

path_ct <- 'data/Publicly available/IMR Surveys/Count/'
path_wg <- 'data/Publicly available/IMR Surveys/Weight/'

files_ct = list.files(path_ct, pattern="*.txt")
files_wg = list.files(path_wg, pattern="*.txt")

## For now remove 2021 and 2022 problem in format

ct<-data.frame()

for (i in files_ct){
  paste(i)
  ct.yr<-read.delim(paste0(path_ct,i))
  ct.yr<-ct.yr %>% pivot_longer(cols = -c(Haul:samplequality), names_to="species", values_to = "Count")
  ct.yr<-ct.yr %>% filter(!Count==0)
  ct<-rbind(ct, ct.yr)
}

wg<-data.frame()

for (i in files_wg){
  paste(i)
  wg.yr<-read.delim(paste0(path_wg,i))
  wg.yr<-wg.yr %>% pivot_longer(cols = -c(Haul:samplequality), names_to="species", values_to = "Weight")
  wg.yr<-wg.yr %>% filter(!Weight==0)
  wg<-rbind(wg, wg.yr)
}

common_col_names <- intersect(names(ct), names(wg))
norw_dat = merge(ct, wg, by=common_col_names, all.x=T)

rm(files_ct, files_wg, ct, ct.yr, wg, wg.yr)

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

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

## Get the year
norw_dat$year<-format(as.Date(norw_dat$DateTime, format="%Y-%m-%d"),"%Y")
## Get the month
norw_dat$month<-format(as.Date(norw_dat$DateTime, format="%Y-%m-%d"),"%m")
## Get the day
norw_dat$day<-format(as.Date(norw_dat$DateTime, format="%Y-%m-%d"),"%d")

# Recalculate the haul duration from EffectiveTowDistance (in nm) knowing the ship go on average 3nm/h

norw_dat$HaulDur<-norw_dat$EffectiveTowDistance*60/3
# In the Barents Sea Ecosystem most hauls last 15mn

```

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

### All gears in the dataset are valid gear, all are shrimp trawls
# "3270" is a Campelen 1800 shrimp trawl with 22mm mesh size. Reketrål. Campelen 1800 ma 20 mm m/40 m s
# "3271" is like 3270 with strapping Reketrål. Campelen 1800 ma 20 mm m/40 m sveiper. Rockhopper gear, s

#####
#### REMOVE BAD QUALITY HAULS
#####
# Remove bad quality hauls and gears
norw_dat <- subset(norw_dat, norw_dat$gearcondition %in% c(1)) # very few hauls with gear.condition = 2

# Is there still empty species names and abundances?
check.sp <- subset(norw_dat, norw_dat$species=='') # ok
check.ab <- subset(norw_dat, is.na(norw_dat$Count)) # ok
check.wg <- subset(norw_dat, is.na(norw_dat$Weight)) # 54 observ. without weight

rm(check.sp, check.ab, check.wg)

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

# Transform distance nautical miles to km
# 1nm <-> 1.852km
norw_dat$Distance <- norw_dat$EffectiveTowDistance*1.852/1
norw_dat[norw_dat$Distance<0,]$Distance <- NA

# Trawl opening in the Barents Sea ecosystem survey is 25m (small variations could occur)
# Door spread is 50m (small variations could occur as well)
norw_dat$DoorSpread <- 0.05 #in km

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

# Estimate missing swept areas in km2
norw_dat <- norw_dat %>%
  mutate(Area.swept = DoorSpread*Distance) %>%
```

```

filter(!is.na(HaulDur)) # there is no missing haul duration

nor <- norw_dat %>%
  select(Haul, year, Area.swept, HaulDur, Gear, BottomDepth, Distance) %>%
  distinct()

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

# no need to estimate missing swept areas with this dataset

rm(nor)

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

# Continue cleaning
norw_dat <- norw_dat %>%
  mutate(quarter = ceiling(as.numeric(month)/3),
         num_cpue = Count/Area.swept, # nbr / km2
         wgt_cpue = Weight/Area.swept, # kg / km2
         num_h = Count*60/HaulDur, # nbr / hour
         wgt_h = Weight*60/HaulDur, # kg / h
         survey = 'Nor-BTS',
         season = 'Summer',
         sbt=NA,
         sst=NA,
         country = "norway",
         continent = "europe",
         stat_rec = NA_character_,
         stratum = NA_character_,
         aphia_id = NA_character_,
         sub_area = NA_character_,
         sub_factor_ab = NA
         ) %>%
  rename(haul_id = Haul,
         haul_dur = HaulDur,
         station = StationKey,
         latitude = Latitude,
         longitude = Longitude,
         depth = BottomDepth,
         num = Count,
         wgt = Weight,
         scientific_name = species) %>%
  filter(haul_dur<(60), # Keep only hauls shorter than 60mn/1h
         !is.na(haul_dur)) %>%
  select(-CruiseKey,-HaulKey, -Cruise, -Platform, -Station, -CatchPlatform, -stationtype)

#####

```

```

#### 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),
    scientific_name = gsub(".", " ", norw_dat$scientific_name, fixed=TRUE)
  ) %>%
  filter(scientific_name!="")

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

# Cleaning with Fishbase if needed
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) #Getting this error message -> Error: Result 1 must be a single
                                     # It happens in the step "Get fishbase id"

#clean_auto<-output_df #done manually meanwhile

### Clean manual
missing_taxa <- setdiff(norw_fish_names, clean_auto$query) # they're all inverts

#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 <- clean_auto %>%
  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)) %>%
  dplyr::rename(verbatim_name = scientific_name,
               accepted_name = taxa) %>%
  mutate(verbatim_aphia_id = NA)

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

## No Length data
#coeffs_nor <- get_coeffs(clean_taxon, survey = survey_code, save=T)

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

# No Length data

### Format dataset (same columns than FishGlob data)

clean_norw <- clean_norw %>%
  select(-DateTime, -serialnumber, -TowDistance, -EffectiveTowDistance, -MinHaulDepth, -MaxHaulDepth, -
        -HorizontalNetOpening, -TrawlDoorSpread, -gearcondition, -samplequality, -Distance, -DoorSpread)
  dplyr::rename(
    area_swept=Area.swept,
    gear=Gear
  ) %>%
  mutate (source = "IMR",
          timestamp = "2023",
          num_cpua = num_cpue,
          num_cpue = num_h,
          wgt_cpua = wgt_cpue,
          wgt_cpue = wgt_h,
          year = as.integer(year),
          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)

```

```

##### JEPa #####
# Quick FIX for issue #23 #

# Load data
# load("~/GitHub/FishGlob_data/outputs/Cleaned_data/NOR-BTS_clean.RData")
#
# # Get missing data
# missing_aphia_id <- data %>% filter(is.na(aphia_id)) %>% pull(accepted_name) %>% unique()
#
# # Get ID for missing data
# fix_missing_ids <- clean_taxa(missing_aphia_id) %>%
#   select(accepted_name = query,
#          worms_id_new = worms_id)
#
# # double check they are all there
# fix_missing_ids %>% filter(is.na(worms_id_new)) # checked!
#
# # Joint data with new values
# fixed_ids_df <-
#   data %>% filter(is.na(aphia_id)) %>% #filter data missing id
#   left_join(fix_missing_ids) %>% # include new ids
#   mutate(aphia_id = as.character(worms_id_new)) %>%
#   select(-worms_id_new) %>%
#   bind_rows(data) %>% # incorporate all data
#   filter(!is.na(aphia_id)) # remove blank-id duplicates
#
# # Double check no more na's
# fixed_ids_df %>% filter(is.na(aphia_id)) #check!
#
# # double check all hauls are there
# anti_join(data %>% select(haul_id), fixed_ids_df %>% select(haul_id)) #check!
#
# # Save new data
# write_clean_data(data = fixed_ids_df, survey = "Nor-BTS", overwrite = T)

# ----- end fix ----- #

# -----#
#### 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",
      "ggnewscale")

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 STRANDARDIZATION 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], "_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("DF0-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 | 2023 | 2004210/4/2004/1019/10-412-2303 | norway | NA |
| Nor-BTS | IMR | 2023 | 2004210/4/2004/1019/10-412-2303 | norway | NA |
| Nor-BTS | IMR | 2023 | 2004210/4/2004/1019/10-412-2303 | norway | NA |
| Nor-BTS | IMR | 2023 | 2004210/4/2004/1019/10-412-2303 | norway | NA |
| Nor-BTS | IMR | 2023 | 2004210/4/2004/1019/10-412-2303 | norway | NA |

| continent | stat_rec | station | stratum | year | month | day | quarter | season |
|-----------|----------|---------|---------|------|-------|-----|---------|--------|
| europe | NA | 412 | NA | 2004 | 08 | 07 | 3 | Summer |
| europe | NA | 412 | NA | 2004 | 08 | 07 | 3 | Summer |
| europe | NA | 412 | NA | 2004 | 08 | 07 | 3 | Summer |
| europe | NA | 412 | NA | 2004 | 08 | 07 | 3 | Summer |
| europe | NA | 412 | NA | 2004 | 08 | 07 | 3 | Summer |

| latitude | longitude | haul_dur | area_swept | gear | depth | sbt | sst |
|----------|-----------|----------|------------|------|-------|-----|-----|
| 70.04833 | 17.27 | 22 | 0.10186 | 3271 | 120 | NA | NA |
| 70.04833 | 17.27 | 22 | 0.10186 | 3271 | 120 | NA | NA |
| 70.04833 | 17.27 | 22 | 0.10186 | 3271 | 120 | NA | NA |
| 70.04833 | 17.27 | 22 | 0.10186 | 3271 | 120 | NA | NA |
| 70.04833 | 17.27 | 22 | 0.10186 | 3271 | 120 | NA | NA |

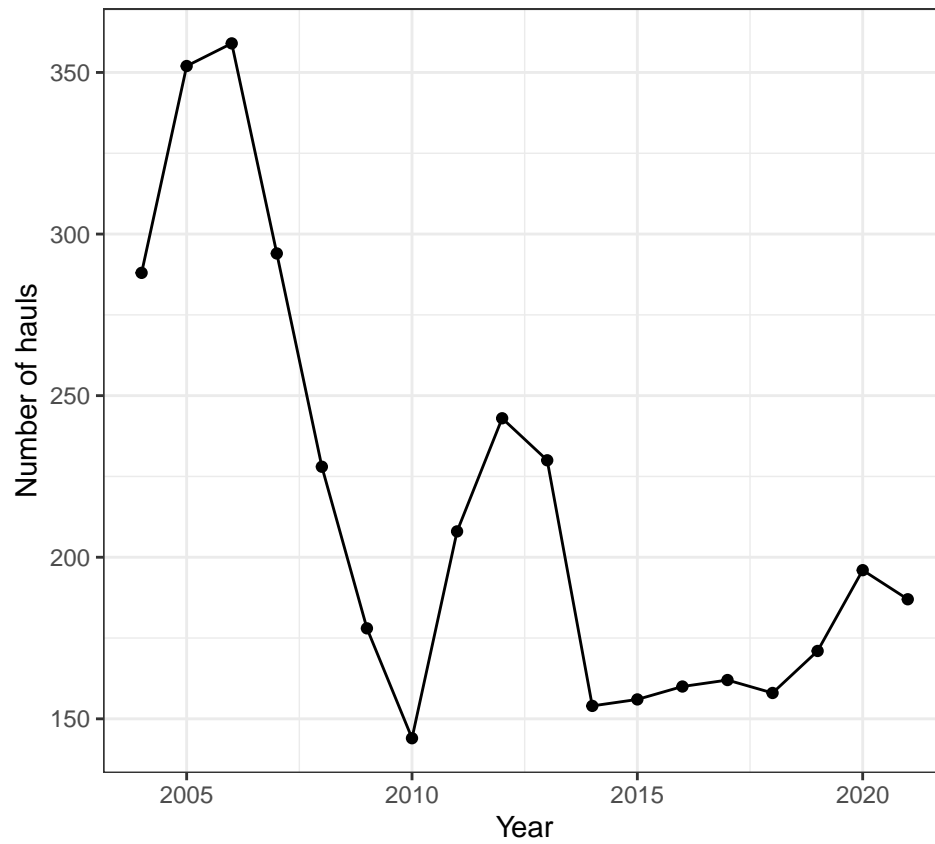
| num | num_cpue | num_cpua | wgt | wgt_cpue | wgt_cpua | verbatim_name |
|-----|----------|-----------|--------|------------|------------|------------------------------|
| 5 | 13.63636 | 49.08698 | 3.930 | 10.7181818 | 38.582368 | Anarhichas lupus |
| 29 | 79.09091 | 284.70450 | 0.359 | 0.9790909 | 3.524445 | Argentina silus |
| 6 | 16.36364 | 58.90438 | 2.640 | 7.2000000 | 25.917927 | Brosme brosme |
| 11 | 30.00000 | 107.99136 | 29.110 | 79.3909091 | 285.784410 | Gadus morhua |
| 5 | 13.63636 | 49.08698 | 0.196 | 0.5345455 | 1.924210 | Hippoglossoides platessoides |

| verbatim_aphia_id | accepted_name | aphia_id | SpecCode | kingdom |
|-------------------|------------------------------|----------|----------|----------|
| NA | Anarhichas lupus | NA | 2501 | Animalia |
| NA | Argentina silus | NA | 2700 | Animalia |
| NA | Brosme brosme | NA | 51 | 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 | Argentiniformes | Argentinidae | Argentina | Species | Nor-BTS |
| Chordata | Teleostei | Gadiformes | Lotidae | Brosme | Species | Nor-BTS |
| Chordata | Teleostei | Gadiformes | Gadidae | Gadus | Species | Nor-BTS |
| Chordata | Teleostei | Pleuronectiformes | Pleuronectidae | Hippoglossoides | Species | Nor-BTS |

2. Summary of sampling intensity

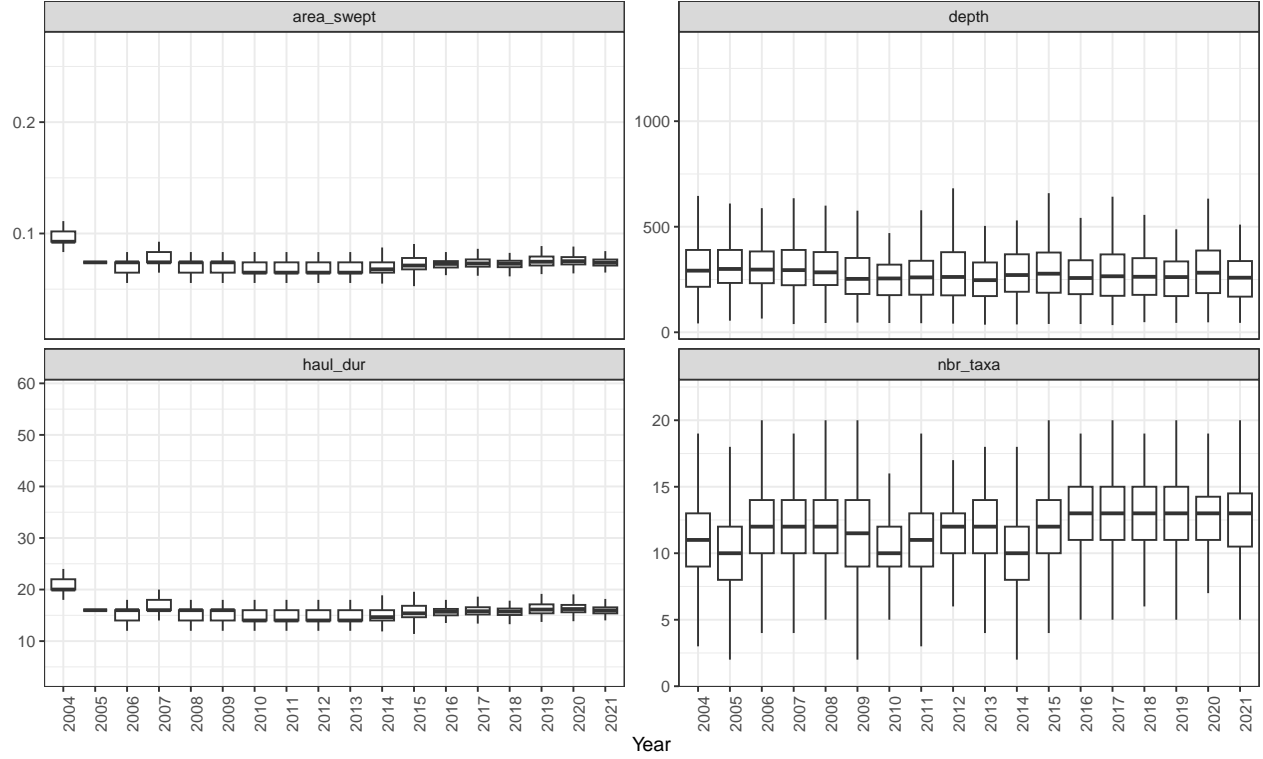
Number of hauls per year performed during the survey after data processing.



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:

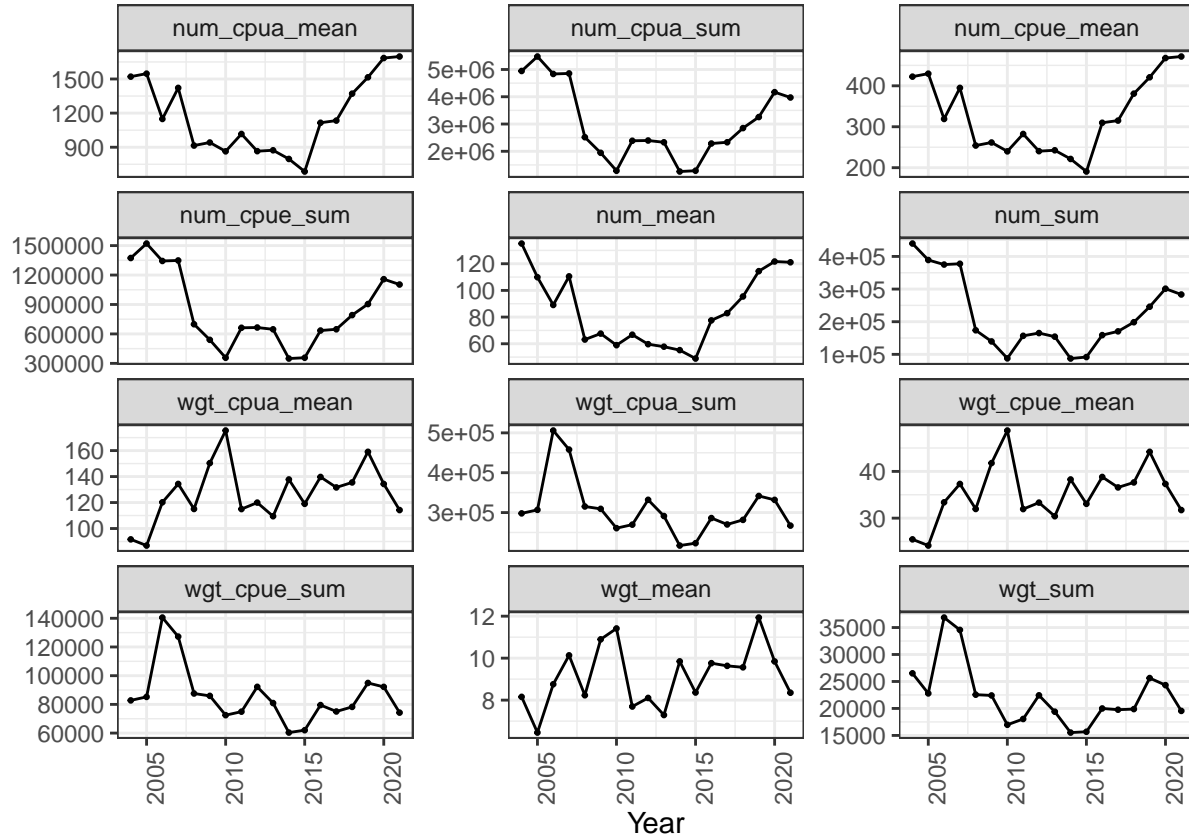
- *area_swept*, swept area by the bottom trawl gear km^2
- *depth*, sampling depth in *m*
- *haul_dur*, haul sampling duration *hours*
- *number of marine fish taxa*, taxa were cleaned following the last version of taxonomy from the World Register of Marine Species (<https://www.marinespecies.org/>,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

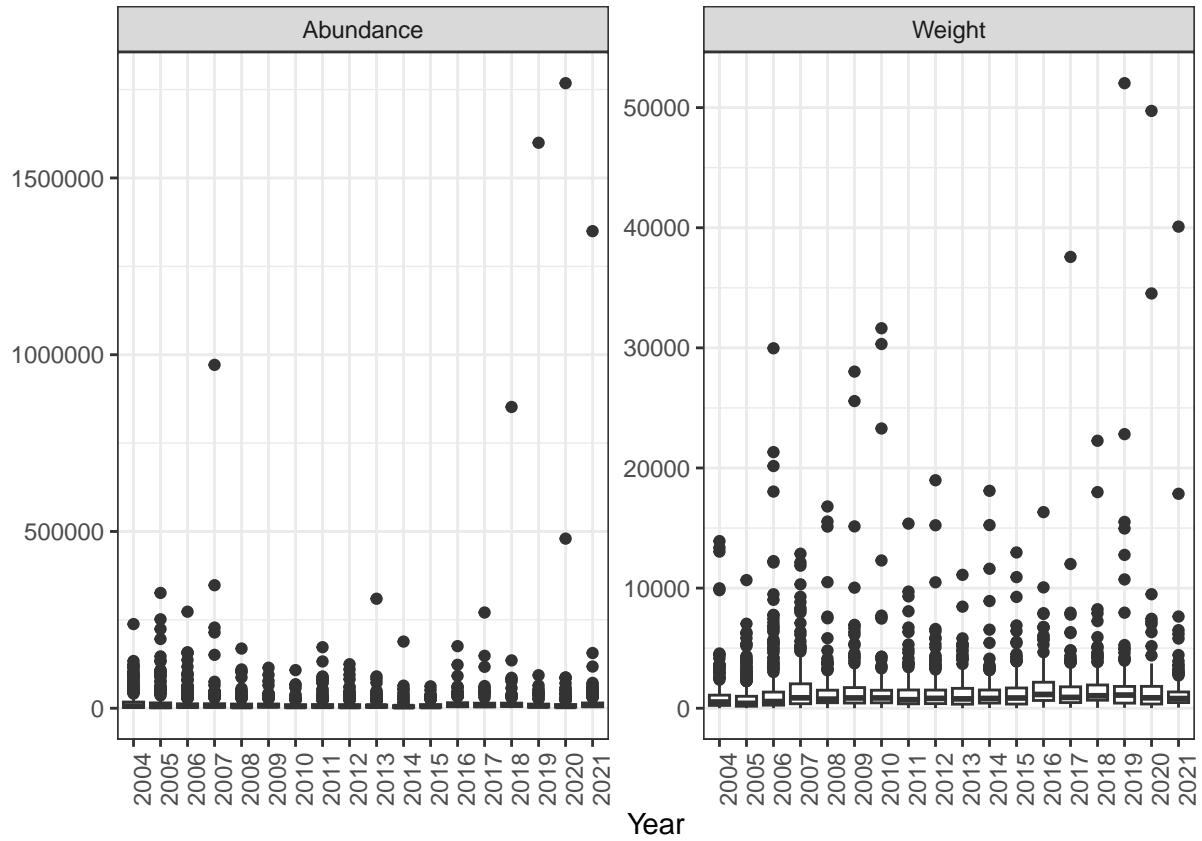
- *num_cpua*, number of individuals (abundance) in $\frac{\text{individuals}}{\text{km}^2}$
- *num_cpue*, number of individuals (abundance) in $\frac{\text{individuals}}{h}$
- *num*, number of individuals (abundance)
- *wgt_cpua*, weight in $\frac{kg}{\text{km}^2}$
- *wgt_cpue*, weight in $\frac{kg}{h}$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



5. Extreme values

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

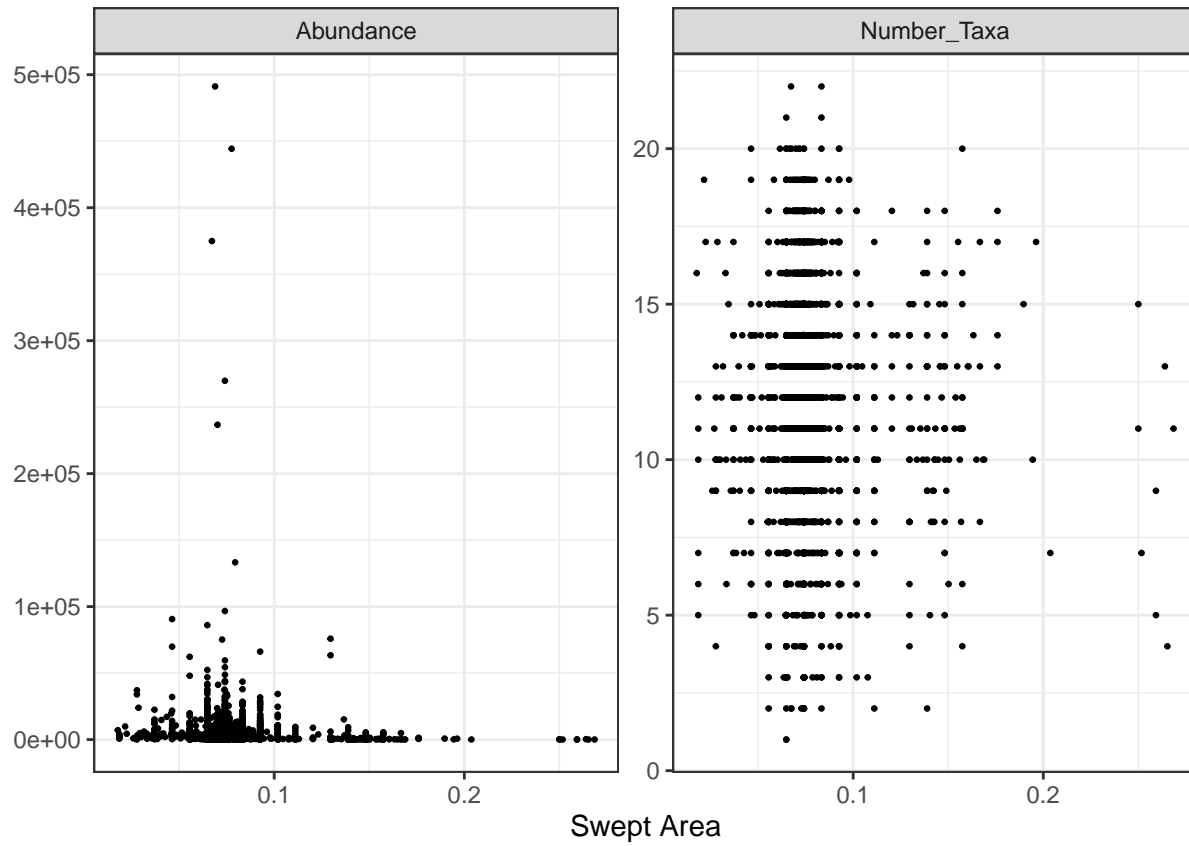
- *num_cpue*, number of individuals (abundance) in $\frac{\text{individuals}}{\text{km}^2}$
- *wgt_cpue*, weight in $\frac{\text{kg}}{\text{km}^2}$



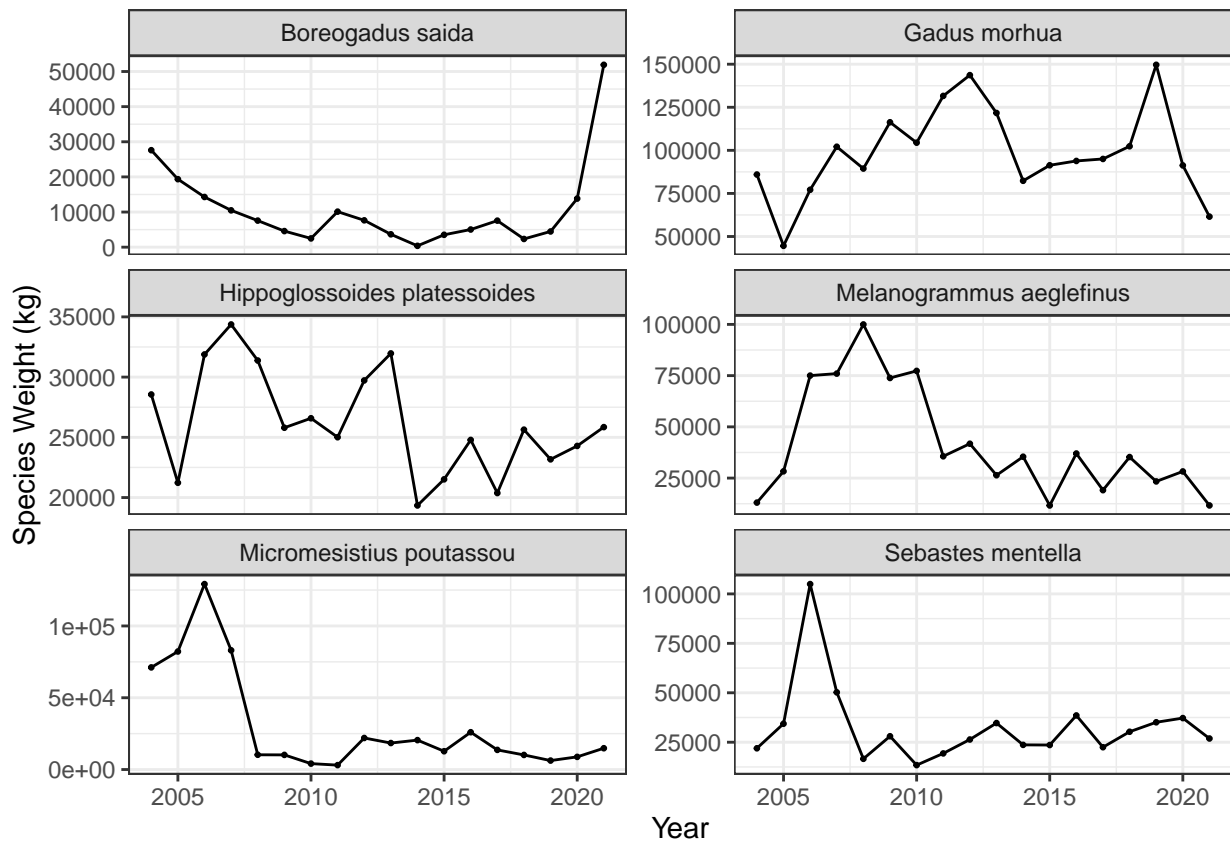
6. Summary of variables against swept area

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 $\frac{\text{individuals}}{\text{km}^2}$
- *wgt_cpua*, weight in $\frac{\text{kg}}{\text{km}^2}$

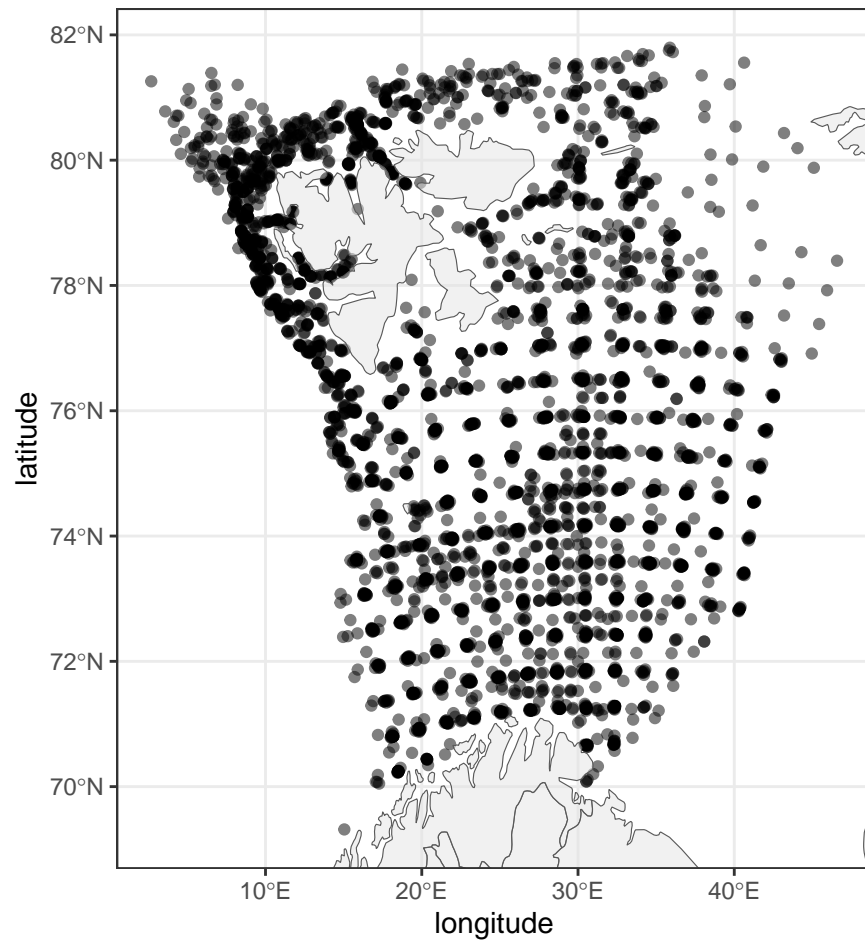


7. Abundance or Weight trends of the six most abundant species



8. Distribution mapping

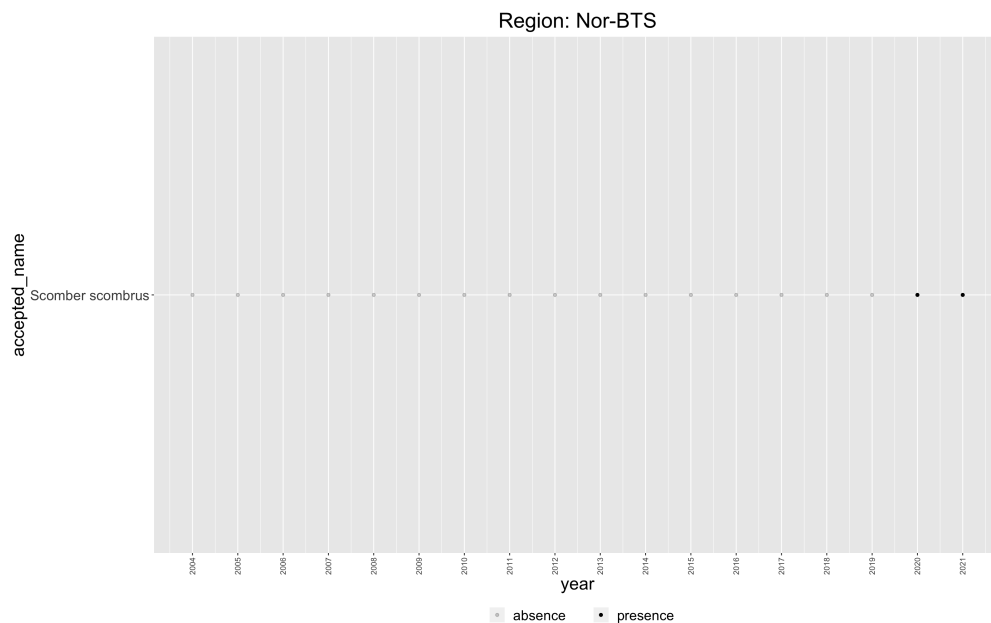
Map of the sampling distribution in space. Note that we only show one year per coordinate.



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

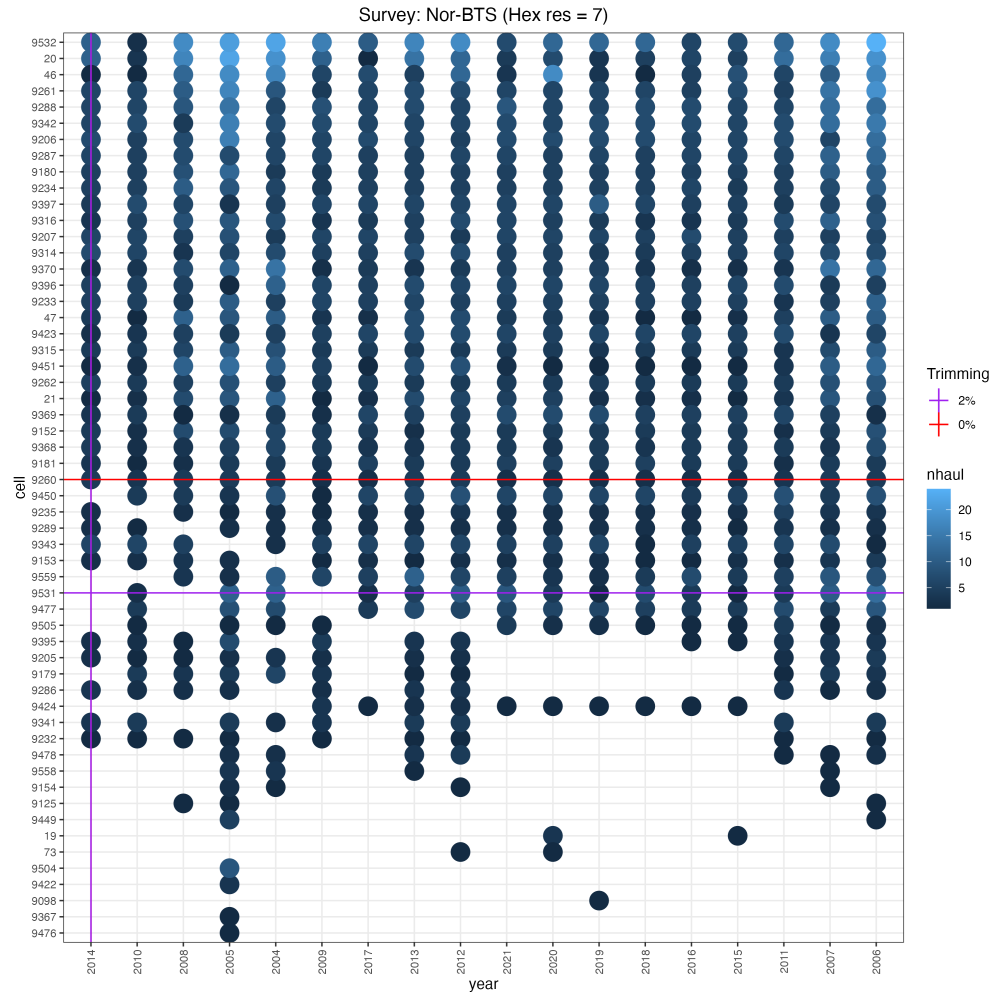
| | |
|-------------------------------|-------|
| Total number of species | 108.0 |
| Percentage of species flagged | 0.9 |

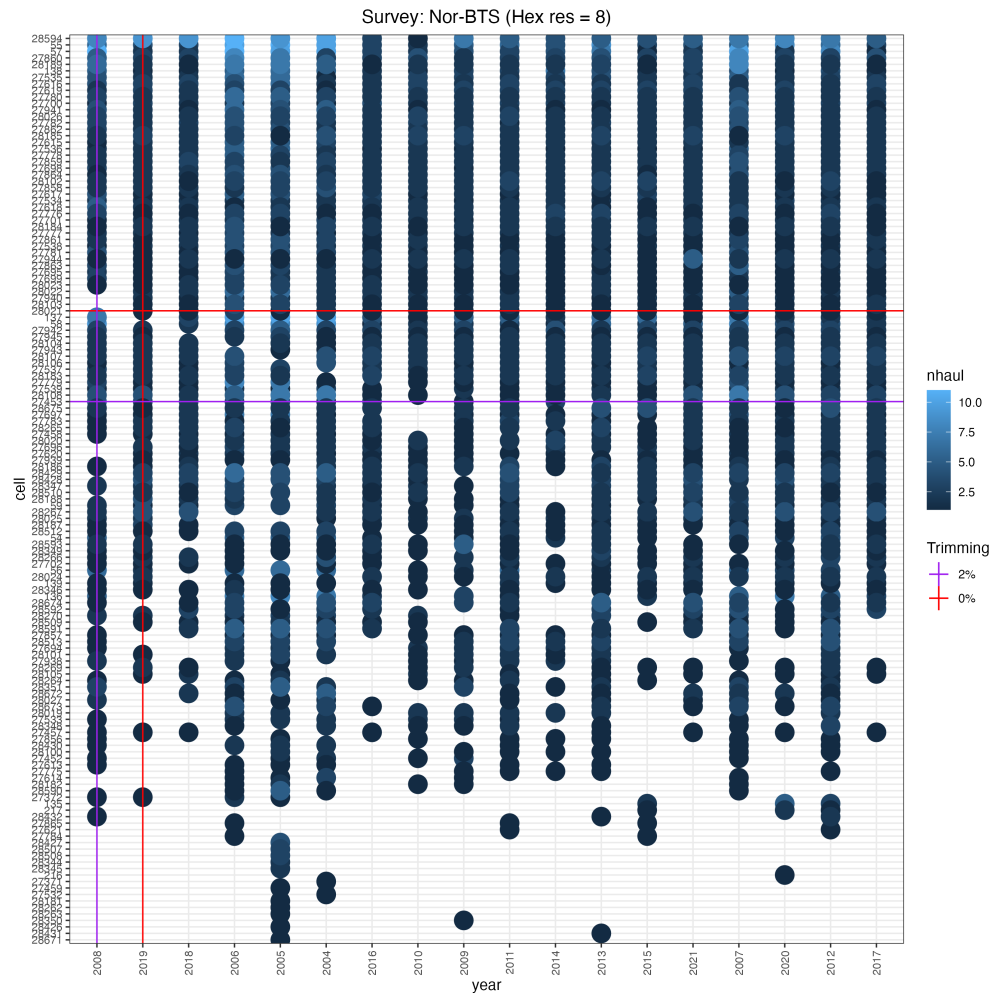
10. Spatio-temporal standardization

a. Standardization method 1

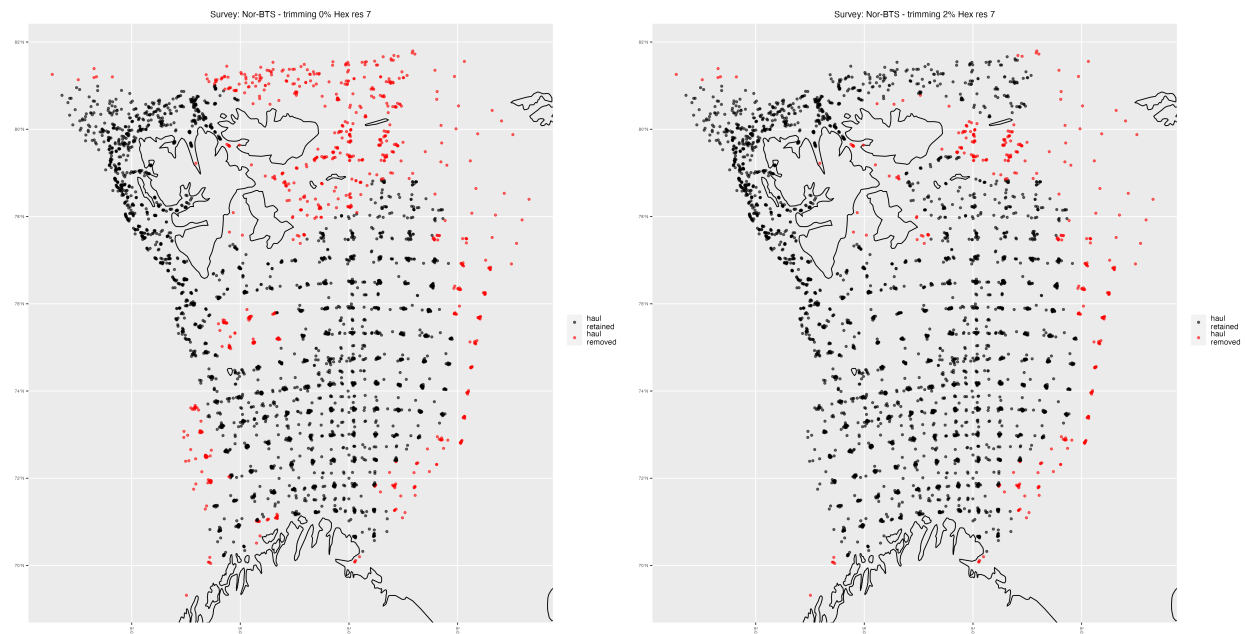
This standardization method was adapted from https://github.com/zoekitchel/trawl_spatial_turnover/blob/master/data_prep_code/species/explore_NorthSea_trimming.Rmd
It was run for hex resolution 7 and 8.

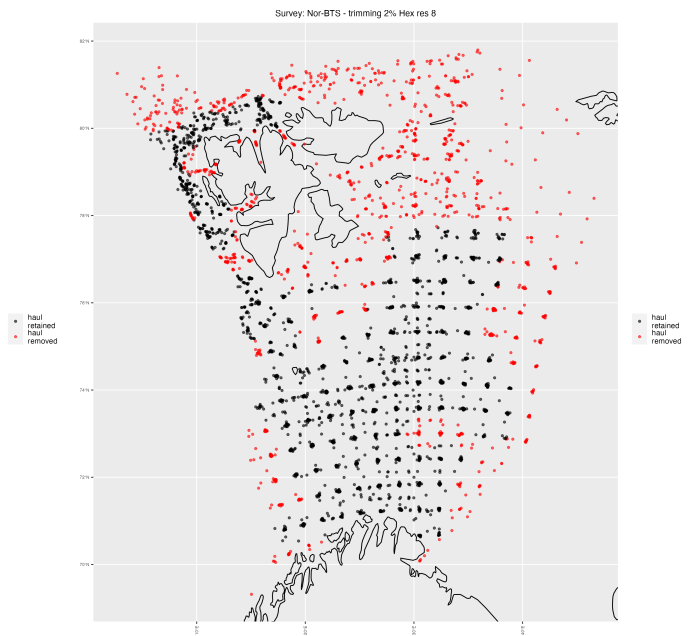
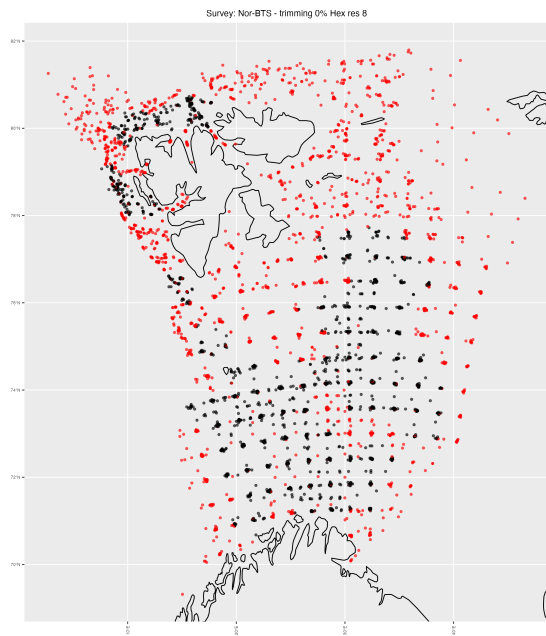
Plot of number of cells x years with overlaid flagging options



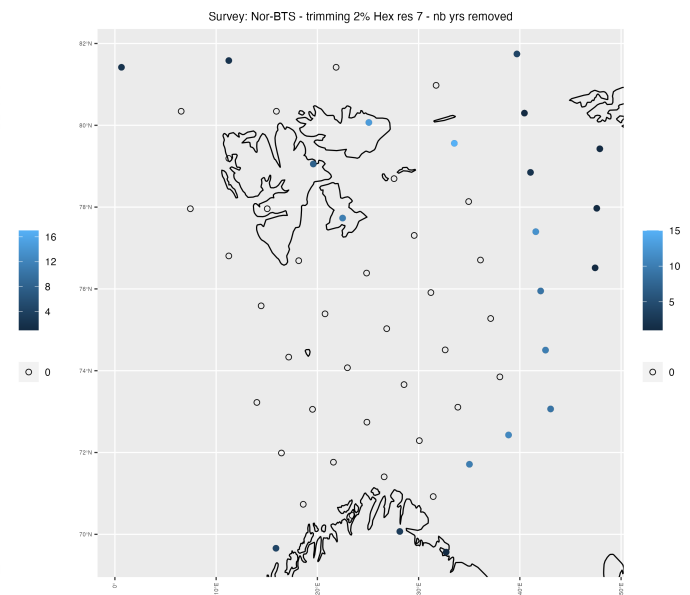
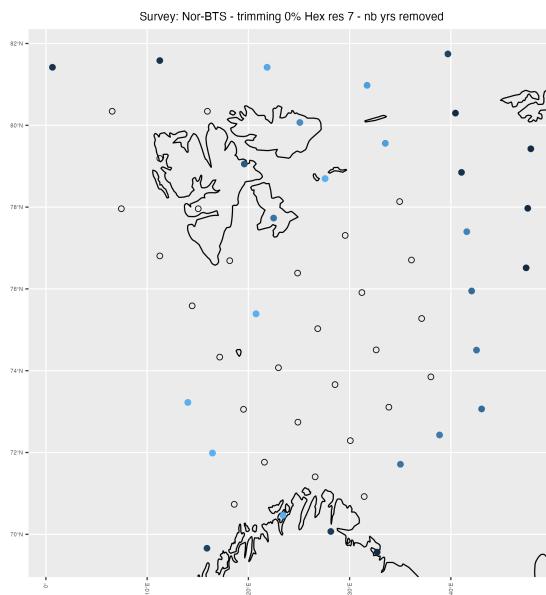


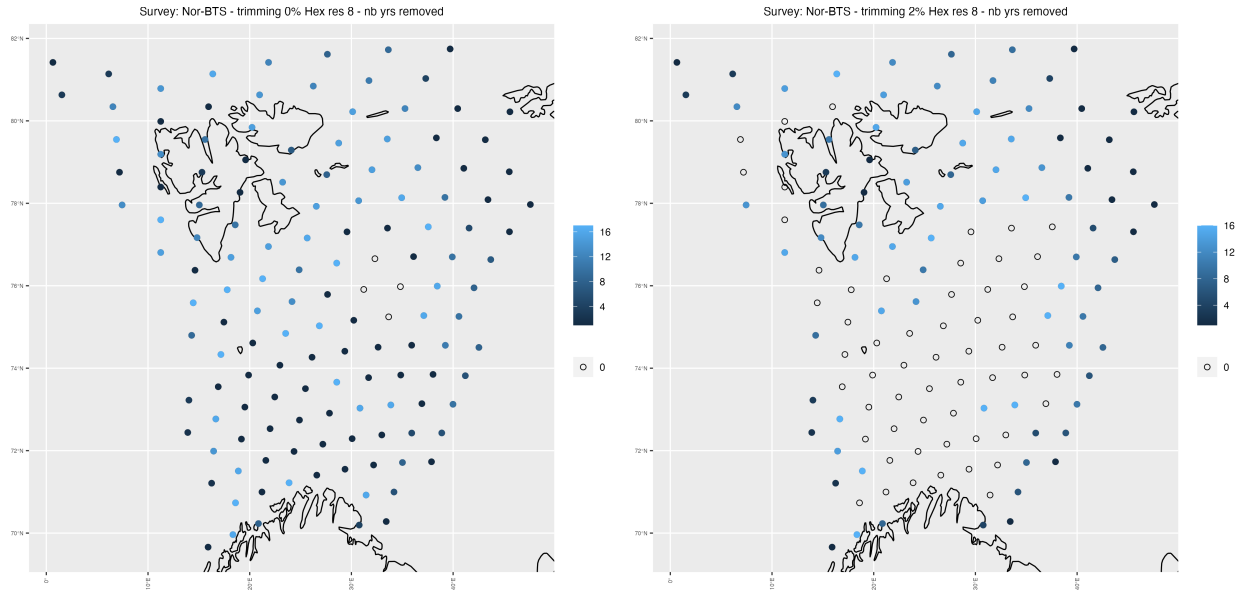
Map of hauls retained and removed per flagging method and threshold





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 | 797.0 | 332.0 | 2101.0 | 1371.0 | 772.0 |
| percentage of hauls removed | 20.6 | 8.6 | 54.3 | 35.4 | 1.7 |