DFO-HS: Department of Fisheries and Oceans Canada Hecate Strait survey data processing summary

fishglob, Aurore A. Maureaud, Juliano Palacios Abrantes, Zoë Kitchel, Dan Forrest, & Michelle Stuart

November, 2023

Contents

General info

This document presents the cleaning code and summary of the Hecate Strait Survey (Department of Fisheries Oceans Canada) bottom trawl survey provided by Shelee Hamilton, and Maria Cornthwaite. It contains data from 2005 and up to 2019.

Data cleaning in R

```
##############################################################################
#### R code to clean trawl survey for the DFO Hecate Strait Survey
#### Public data Ocean Adapt
#### Contacts: Shelee Hamilton  Shelee.Hamilton@dfo-mpo.gc.ca   Head,
####     Fishery & Assessment Data Section, Science Branch, DFO Canada
####           Maria Cornthwaite    Maria.Cornthwaite@dfo-mpo.gc.ca Program Head,
####            Groundfish Data Unit, Science Branch, DFO Canada
#### Coding: Dan Forrest, Zoë Kitchel November 2021 + October 2023
##############################################################################
#-----------------------------------------------------------------------------#
#### LOAD LIBRARIES AND FUNCTIONS ####
#-----------------------------------------------------------------------------#

library(tidyverse)
```

```r
library(lubridate)
library(googledrive)
library(taxize) # for getting correct species names
library(magrittr) # for names wrangling
library(readr)
library(dplyr)
library(PBSmapping)
library(readxl)
library(here)

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")
fishglob_data_columns <- read_excel("standard_formats/fishglob_data_columns.xlsx")

#Data for the Hecate Strait Survey can be best accessed using the Pinsky
#Lab Ocean Adapt Git Hub Repository.
#Contact malin.pinsky@rutgers.edu for questions or help accessing

#-------------------------------------------------------------------------------------------#
#### PULL IN AND EDIT RAW DATA FILES ####
#-------------------------------------------------------------------------------------------#


HS_catch <- read_csv(
  "https://github.com/pinskylab/OceanAdapt/raw/master/data_raw/HS_catch.csv",
                     col_types = cols(
  Survey.Year = col_integer(),
  Trip.identifier = col_integer(),
  Set.number = col_integer(),
  ITIS.TSN = col_integer(),
  Species.code = col_character(),
  Scientific.name = col_character(),
  English.common.name = col_character(),
  French.common.name = col_character(),
  LSID = col_character(),
  Catch.weight..kg. = col_double(),
  Catch.count..pieces. = col_integer()
))

HS_effort <- read_csv(
  "https://github.com/pinskylab/OceanAdapt/raw/master/data_raw/HS_effort.csv",
                      col_types =
                         cols(
                           Survey.Year = col_integer(),
                           Trip.identifier = col_integer(),
                           Vessel.name = col_character(),
                           Trip.start.date = col_character(),
                           Trip.end.date = col_character(),
                           GMA = col_character(),
                           PFMA = col_character(),
```

```
                            Set.number = col_integer(),
                            Set.date = col_character(),
                            Start.latitude = col_double(),
                            Start.longitude = col_double(),
                            End.latitude = col_double(),
                            End.longitude = col_double(),
                            Bottom.depth..m. = col_double(),
                            Tow.duration..min. = col_integer(),
                            Distance.towed..m. = col_double(),
                            Vessel.speed..m.min. = col_double(),
                            Trawl.door.spread..m. = col_double(),
                            Trawl.mouth.opening.height..m. = col_double()
                          )) %>%
  select(Trip.identifier, Set.number,Survey.Year,Set.date, Trip.start.date,Trip.end.date,
         GMA, PFMA,Set.date, Start.latitude,Start.longitude, End.latitude, End.longitude,
         Bottom.depth..m., Tow.duration..min.,Distance.towed..m., Trawl.door.spread..m.,
         Trawl.mouth.opening.height..m. )

#-------------------------------------------------------------------------------------------#
#### REFORMAT AND MERGE DATA FILES ####
#-------------------------------------------------------------------------------------------#

HS <- left_join(HS_catch, HS_effort, by = c(
                             "Trip.identifier", "Set.number","Survey.Year"))



HS <- HS %>%
  # Create a unique haul_id
  mutate(
    haul_id = paste(formatC(Trip.identifier, width=3, flag=0),
                    formatC(Set.number, width=3, flag=0), sep= "-"),
    # Add "strata" (define by lat, lon and depth bands) where needed # degree bins
    # 100 m bins # no need to use lon grids on west coast (so narrow)
    stratum = paste(floor(Start.latitude), floor(Start.longitude),
                    floor(Bottom.depth..m./100)*100, sep= "-"),
    # catch weight (kg.) per tow/
    #                 (distance towed in m * trawl door spread m) * (1000000m^2/1km^2)
    wgt_cpue = Catch.weight..kg./(Distance.towed..m.*Trawl.door.spread..m.) *1000000,
    # catch weight (kg.) per tow/
    #                 time of tow in minutes*60 minutes/hour
    wgt_h = Catch.weight..kg./Tow.duration..min.*60,
    # catch abundance per tow/
    #                 (distance towed in m * trawl door spread m) * (1000000m^2/1km^2)
    num_cpue = Catch.count..pieces./(Distance.towed..m.*Trawl.door.spread..m.) *1000000,
    # catch weight (kg.) per tow/
    #                 time of tow in minutes*60 minutes/hour
    num_h = Catch.count..pieces./Tow.duration..min.*60,

    area_swept = (Distance.towed..m.*Trawl.door.spread..m.)/1000000 #(1km^2/1000000m^2)
  )

HS <- HS %>%
```

```
  rename(
    latitude = Start.latitude,
    longitude = Start.longitude,
    depth = Bottom.depth..m.,
    verbatim_name = Scientific.name,
    year = Survey.Year,
    num = Catch.count..pieces.,
    wgt = Catch.weight..kg.
    ) %>%
    mutate(
    date = as.Date(Set.date),
    haul_dur = Tow.duration..min./60 #convert minutes to hours
  ) %>%
  filter(
    verbatim_name != "" &
      !grepl("egg", verbatim_name)
  ) %>%
  # adjust verbatim_name names
  mutate(verbatim_name = ifelse(grepl("Lepidopsetta", verbatim_name),
                                "Lepidopsetta sp.", verbatim_name),
         verbatim_name = ifelse(grepl("Bathyraja", verbatim_name),
                                'Bathyraja sp.', verbatim_name),
         verbatim_name = ifelse(grepl("Squalus", verbatim_name),
                                'Squalus suckleyi', verbatim_name))


# Does the spp column contain any eggs or non-organism notes?
#As of fall 2021, nothing stuck out as needing to be removed
test <- HS %>%
  select(verbatim_name) %>%
  filter(!is.na(verbatim_name)) %>%
  distinct() %>%
  mutate(verbatim_name = as.factor(verbatim_name)) %>%
  filter(grepl("egg", verbatim_name) & grepl("", verbatim_name))
stopifnot(nrow(test)==0)


# combine the wtcpue for each species by haul which is necessary
#because sometimes there are multiple observations for a single genus or family
#this removes duplicates
#i.e.
#HEXACTINELLIDA, GLASS SPONGES; WILLEMOES'S WHITE SEA PEN; CRANGONS
HS <- HS %>%
  group_by(haul_id,year, latitude, longitude, depth, verbatim_name, area_swept,
           date, haul_dur) %>%
  summarise(wgt_cpue = sum(wgt_cpue, na.rm = T), wgt_h = sum(wgt_h, na.rm = T),
            num_h = sum(num_h, na.rm = T), num_cpue = sum(num_cpue, na.rm = T),
            wgt = sum(wgt, na.rm = T), num = sum(num, na.rm = T)) %>%
  ungroup()

HS <- HS %>%
# add survey column
  mutate(survey = "DFO-HS",
```

```
          country = "Canada",
          continent = "n_america",
          stat_rec = NA,
          verbatim_aphia_id = NA,
          aphia_id = NA,
          sub_area = NA,
          station = NA,
          stratum = NA,
          month = lubridate::month(date),
          day = lubridate::day(date),
          season = NA,
          quarter = NA,
          gear = NA,
          sbt = NA,
          sst = NA
  ) %>%
    select(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, num, num_h, num_cpue,
           wgt, wgt_h, wgt_cpue, verbatim_name, verbatim_aphia_id)

#check for duplicates, should not be any with more than 1 obs
#check for duplicates
count_HS <- HS %>%
  group_by(haul_id, verbatim_name) %>%
  mutate(count = n())

#none!

#which ones are duplicated?
unique_name_match <- count_HS %>%
  group_by(verbatim_name) %>%
  filter(count>1) %>%
  distinct(verbatim_name)

unique_name_match
#empty

#-------------------------------------------------------------------------------------------#
#### INTEGRATE CLEAN TAXA FROM TAXA ANALYSIS ####
#-------------------------------------------------------------------------------------------#

# Get WoRM's id for sourcing
wrm <- gnr_datasources() %>%
  filter(title == "World Register of Marine Species") %>%
  pull(id)

### Automatic cleaning
# Set Survey code
hs_survey_code <- "DFO-HS"

HS <- HS %>%
  mutate(
```

```
    taxa2 = str_squish(verbatim_name),
    taxa2 = str_remove_all(taxa2," spp.| sp.| spp| sp|NO "),
    taxa2 = str_to_sentence(str_to_lower(taxa2))
  )

# Get clean taxa
clean_auto <- clean_taxa(unique(HS$taxa2), input_survey = hs_survey_code,
                         save = F, output=NA, fishbase=T)

#This leaves out the following species, which are all inverts
#Cancer branneri
#Cancer gracilis
#Cheiraster dawsoni
#Pandalopsis

#-------------------------------------------------------------------------------------------#
#### INTEGRATE CLEAN TAXA in DFO-HS survey data ####
#-------------------------------------------------------------------------------------------#


correct_taxa <- clean_auto %>%
  select(-survey)

clean_hs <- left_join(HS, correct_taxa, by=c("taxa2"="query")) %>%
  filter(!is.na(taxa)) %>% # query does not indicate taxa entry that were
  #removed in the cleaning procedure
  # so all NA taxa have to be removed from the surveys because: non-existing,
  #non marine or non fish
  rename(accepted_name = taxa,
         aphia_id = worms_id,
         num_cpua = num_cpue,
         num_cpue = num_h,
         wgt_cpua = wgt_cpue,
         wgt_cpue = wgt_h) %>%
  mutate(verbatim_aphia_id = NA,
         source = "DFO",
         timestamp = "09/2020",
         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`)

#check for duplicates
count_clean_hs <- clean_hs %>%
  group_by(haul_id, verbatim_name, accepted_name) %>%
  mutate(count = n())

#none!

#which ones are duplicated?
unique_name_match <- count_clean_hs %>%
  group_by(verbatim_name, accepted_name) %>%
```

```
  filter(count>1) %>%
  distinct(verbatim_name, accepted_name)

unique_name_match
#empty


# ------------------------------------------------------------------------------------------#
#### SAVE DATABASE IN GOOGLE DRIVE ####
# ------------------------------------------------------------------------------------------#

# Just run this routine should be good for all
write_clean_data(data = clean_hs, survey = "HS", overwrite = T, rdata=TRUE)



# ------------------------------------------------------------------------------------------#
#### FAGS ####
# ------------------------------------------------------------------------------------------#
#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(clean_hs$survey))

#run flag_spp function in a loop
```

```r
for (r in regions) {
  flag_spp(clean_hs, 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_hs, 7)

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

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


#-----------------------------------------------------------------------------------------------#
#### ADD STRANDARDIZATION FLAGS ####
#-----------------------------------------------------------------------------------------------#
surveys <- sort(unique(clean_hs$survey))
survey_units <- sort(unique(clean_hs$survey_unit))
survey_std <- clean_hs %>%
  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("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 = "HS_std",
                 overwrite = T, rdata=TRUE)
```

1. Overview of the survey data table

survey	source	timestamp	haul_id	country	sub_area
DFO-HS	DFO	09/2020	59140-001	Canada	NA
DFO-HS	DFO	09/2020	59140-001	Canada	NA
DFO-HS	DFO	09/2020	59140-001	Canada	NA
DFO-HS	DFO	09/2020	59140-001	Canada	NA
DFO-HS	DFO	09/2020	59140-001	Canada	NA

continent	stat_rec	station	stratum	year	month	day	quarter	season
n_america	NA	NA	NA	2005	5	27	NA	NA
n_america	NA	NA	NA	2005	5	27	NA	NA
n_america	NA	NA	NA	2005	5	27	NA	NA
n_america	NA	NA	NA	2005	5	27	NA	NA
n_america	NA	NA	NA	2005	5	27	NA	NA

latitude	longitude	haul_dur	area_swept	gear	depth	sbt	sst
52.68587	-130.6156	0.3666667	0.133224	NA	137	NA	NA
52.68587	-130.6156	0.3666667	0.133224	NA	137	NA	NA
52.68587	-130.6156	0.3666667	0.133224	NA	137	NA	NA
52.68587	-130.6156	0.3666667	0.133224	NA	137	NA	NA
52.68587	-130.6156	0.3666667	0.133224	NA	137	NA	NA

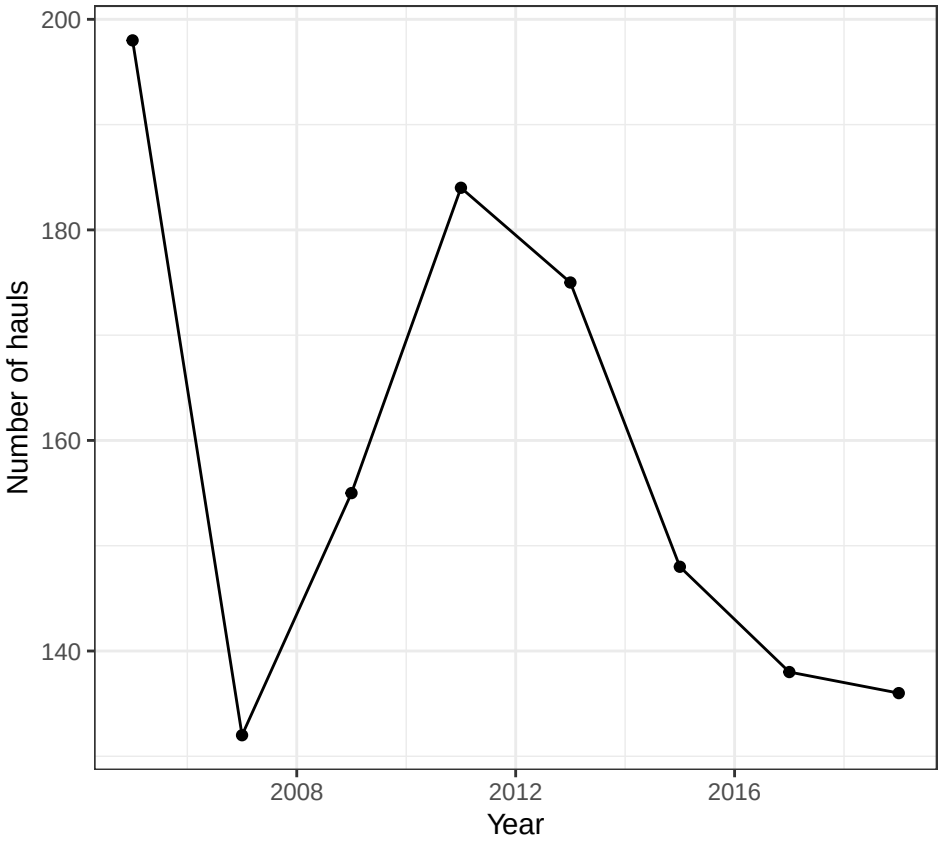
num	num_cpue	num_cpua	wgt	wgt_cpue	wgt_cpua	verbatim_name
0	0.00000	0.00000	2.40	6.545454	18.014772	ATHERESTHES STOMIAS
4	10.90909	30.02462	0.50	1.363636	3.753077	CLUPEA PALLASII
0	0.00000	0.00000	0.44	1.200000	3.302708	EOPSETTA JORDANI
0	0.00000	0.00000	2.30	6.272727	17.264157	GADUS CHALCOGRAMMUS
0	0.00000	0.00000	1.80	4.909091	13.511079	GADUS MACROCEPHALUS

verbatim_aphia_id	accepted_name	aphia_id	SpecCode	kingdom
NA	Atheresthes stomias	279792	517	Animalia
NA	Clupea pallasii	151159	NA	Animalia
NA	Eopsetta jordani	280690	4237	Animalia
NA	Gadus chalcogrammus	300735	318	Animalia
NA	Gadus macrocephalus	254538	308	Animalia

phylum	class	order	family	genus	rank	survey_unit
Chordata	Teleostei	Pleuronectiformes	Pleuronectidae	Atheresthes	Species	DFO-HS
Chordata	Teleostei	Clupeiformes	Clupeidae	Clupea	Species	DFO-HS
Chordata	Teleostei	Pleuronectiformes	Pleuronectidae	Eopsetta	Species	DFO-HS
Chordata	Teleostei	Gadiformes	Gadidae	Gadus	Species	DFO-HS
Chordata	Teleostei	Gadiformes	Gadidae	Gadus	Species	DFO-HS

2. Summary of sampling intensity

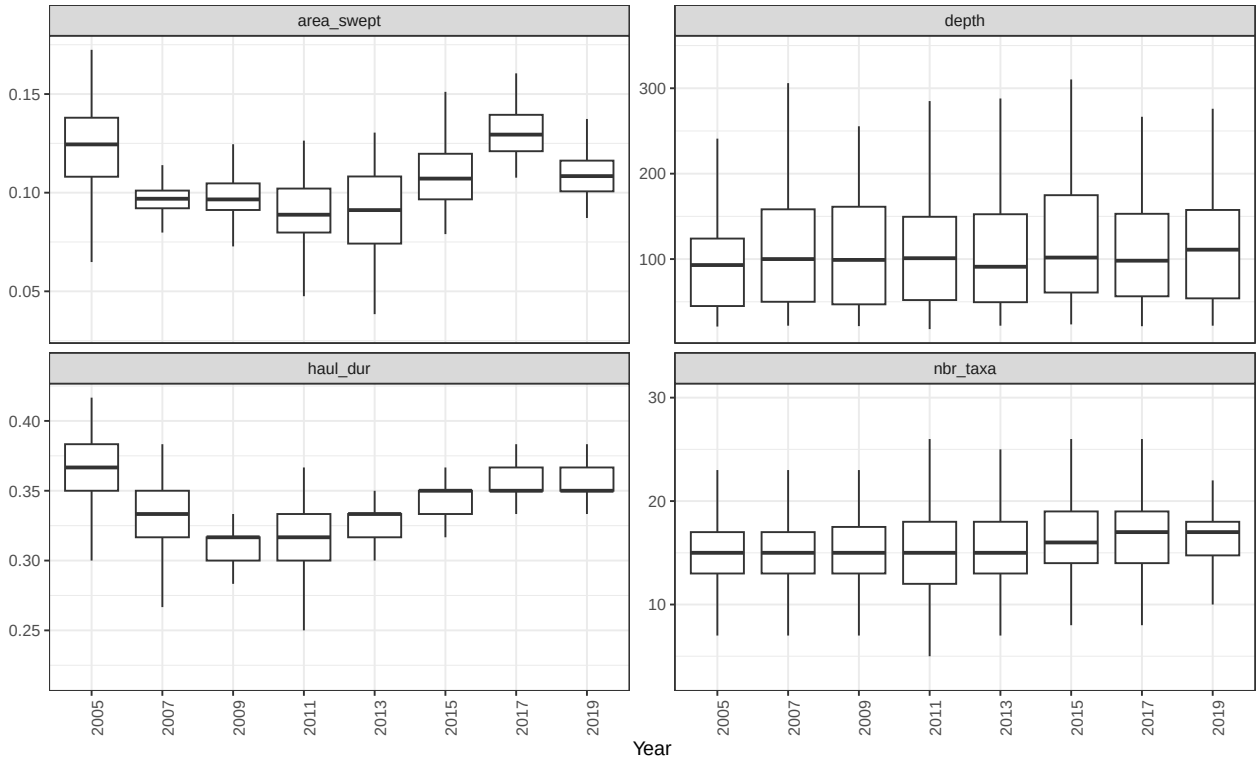
Number of hauls per year performed during the survey after data processing.



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:

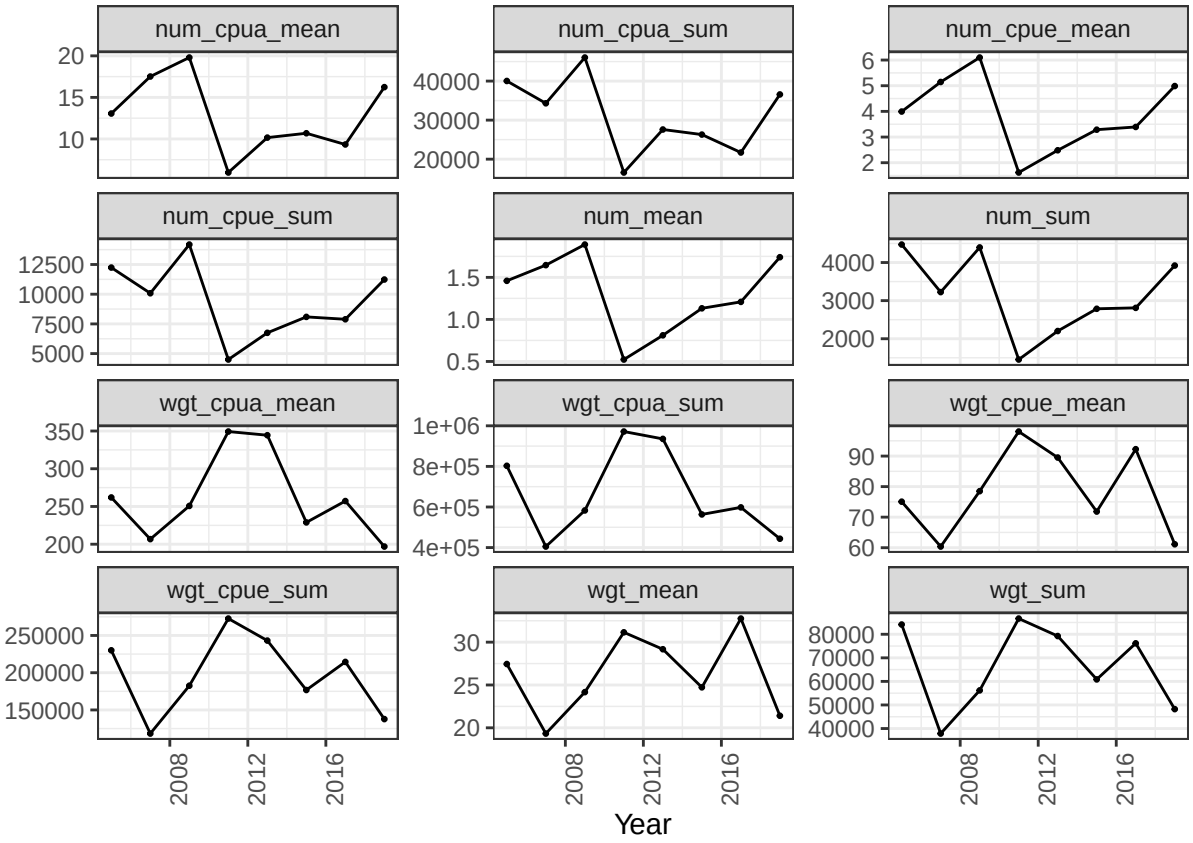
- *area_swept*, swept area by the bottom trawl gear km^2
- *depth*, sampling depth in m
- *haul_dur*, haul sampling duration *hour*
- *number of marine fish taxa*, taxa were cleaned following the last version of taxonomy from the World Register of Marine Species (https://www.marinespecies.org/, October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



4. Summary of biological variables

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

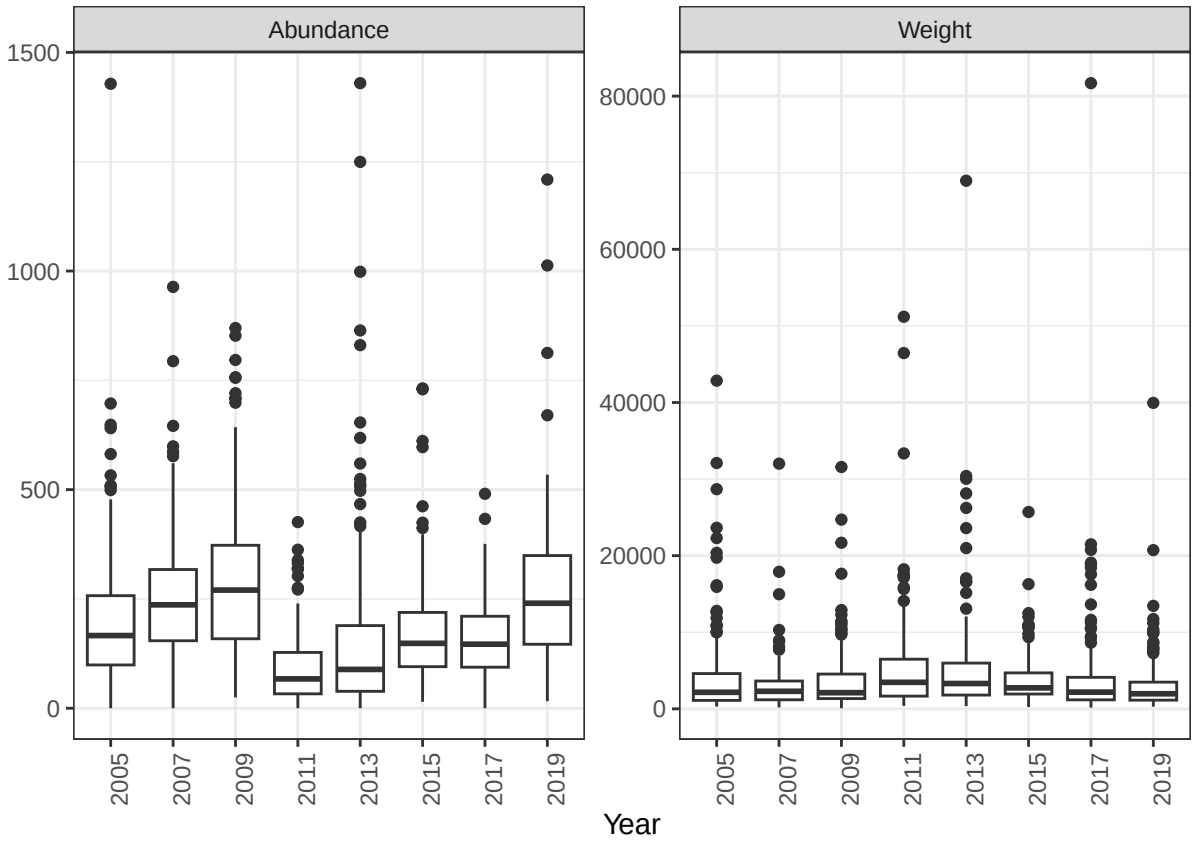
- *num_cpua*, number of individuals (abundance) in $\frac{individuals}{km^2}$
- *num_cpue*, number of individuals (abundance) in $\frac{individuals}{h}$
- *num*, number of individuals (abundance)
- *wgt_cpua*, weight in $\frac{kg}{km^2}$
- *wgt_cpue*, weight in $\frac{kg}{h}$
- *wgt*, weight in kg



5. Extreme values

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

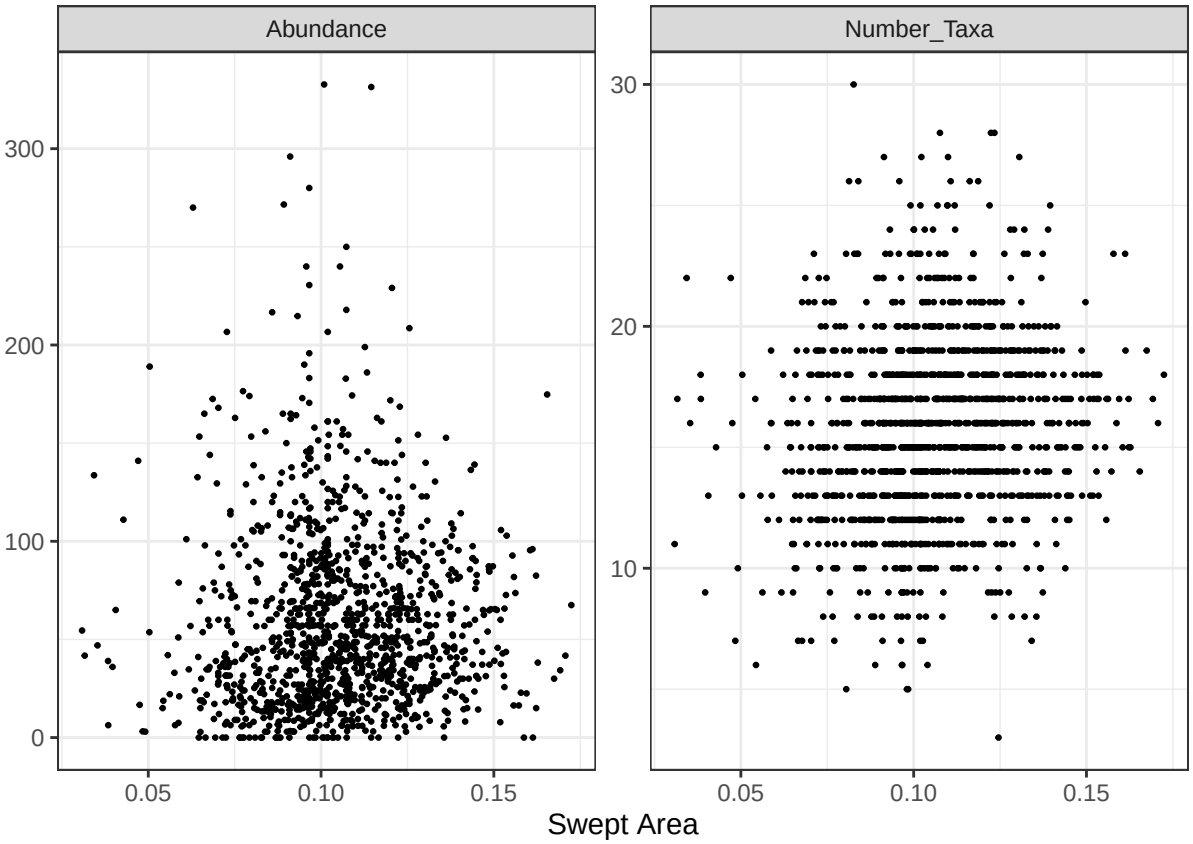
- *num_cpue*, number of individuals (abundance) in $\frac{individuals}{km^2}$
- *wgt_cpue*, weight in $\frac{kg}{km^2}$



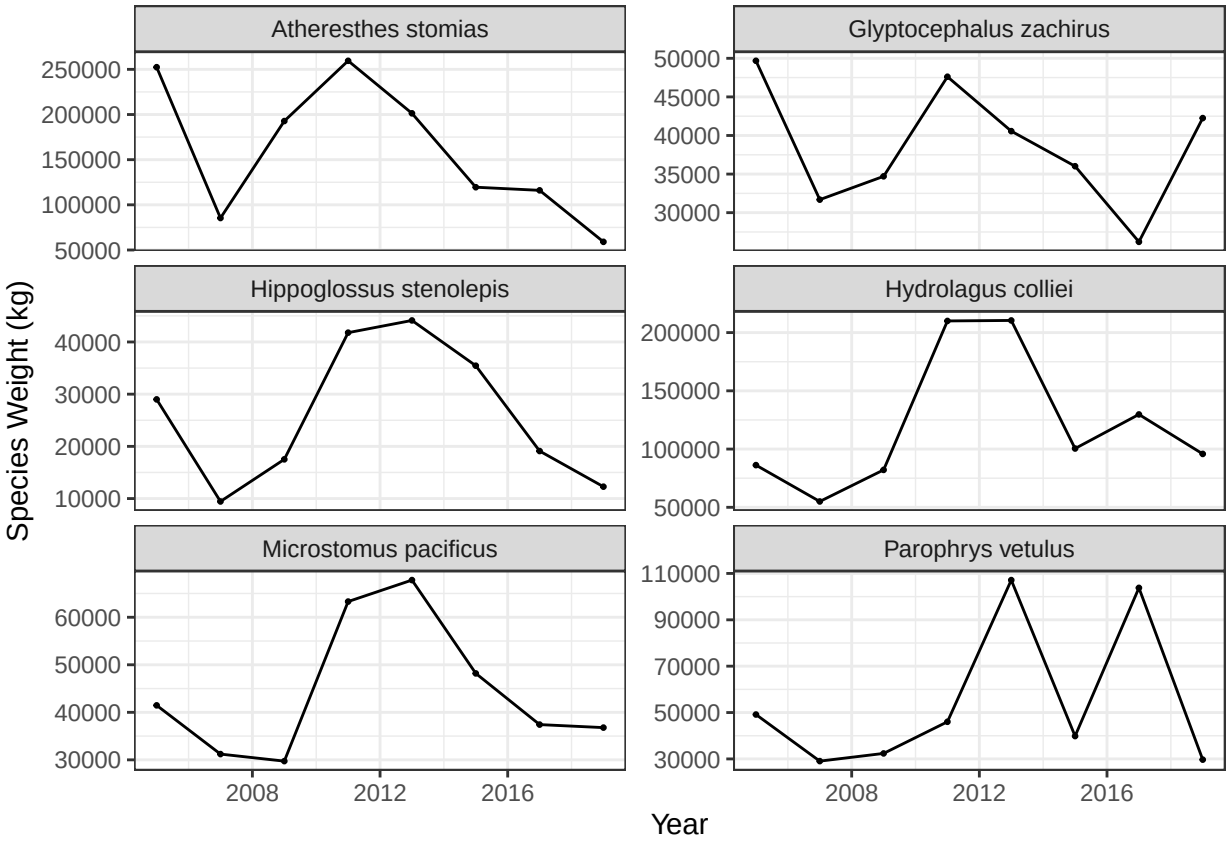
6. Summary of variables against swept area

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 $\frac{kg}{km^2}$

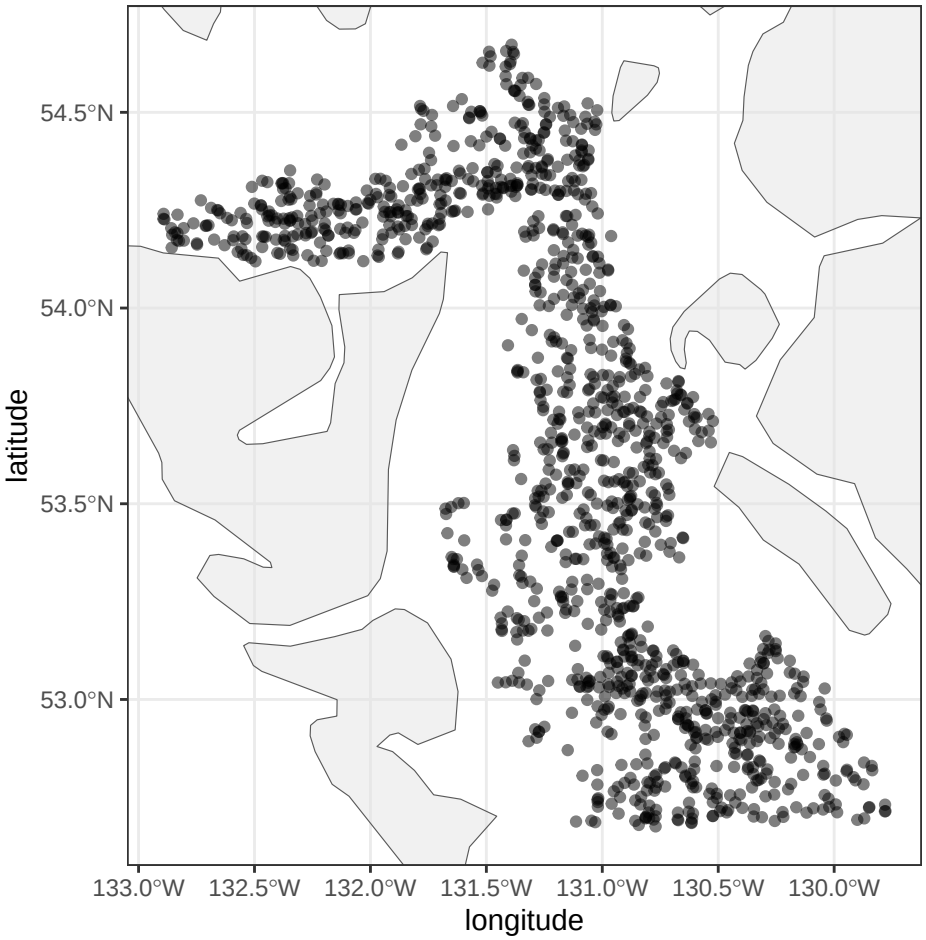


7. Abundance or Weight trends of the six most abundant species



8. Distribution mapping

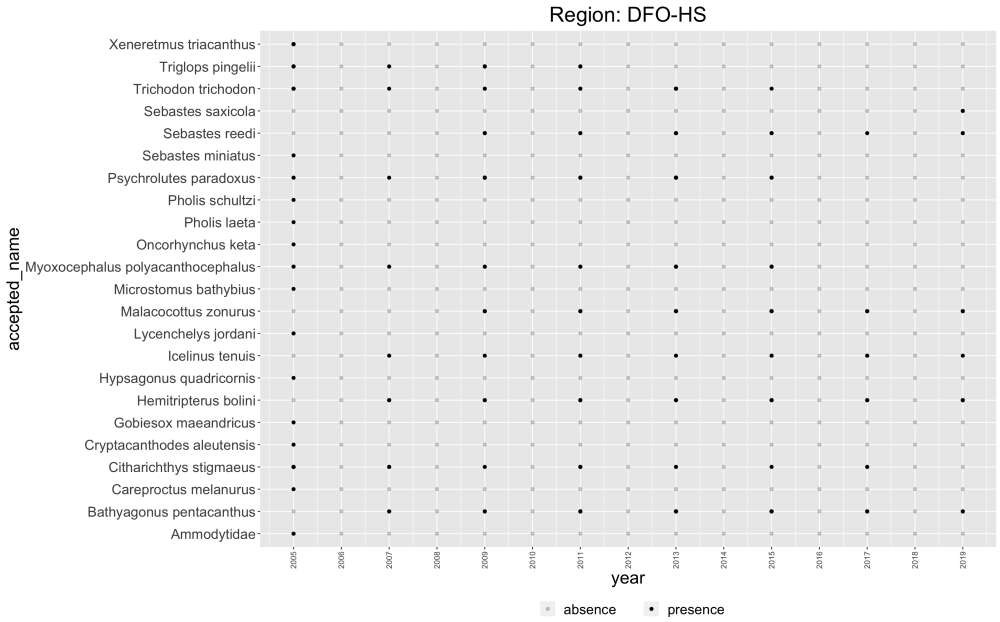
Map of the sampling distribution in space. Note that we only show one year per coordinate.



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

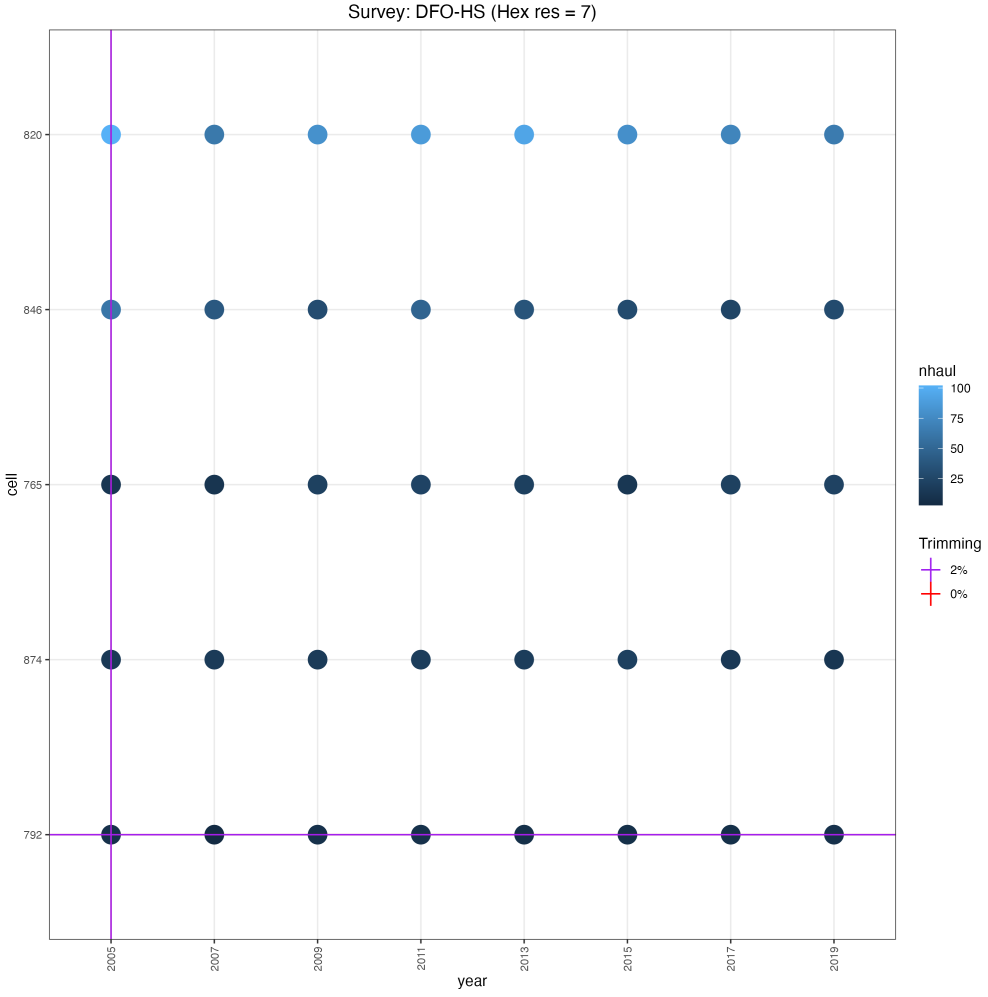
Total number of species	167.0
Percentage of species flagged	13.8

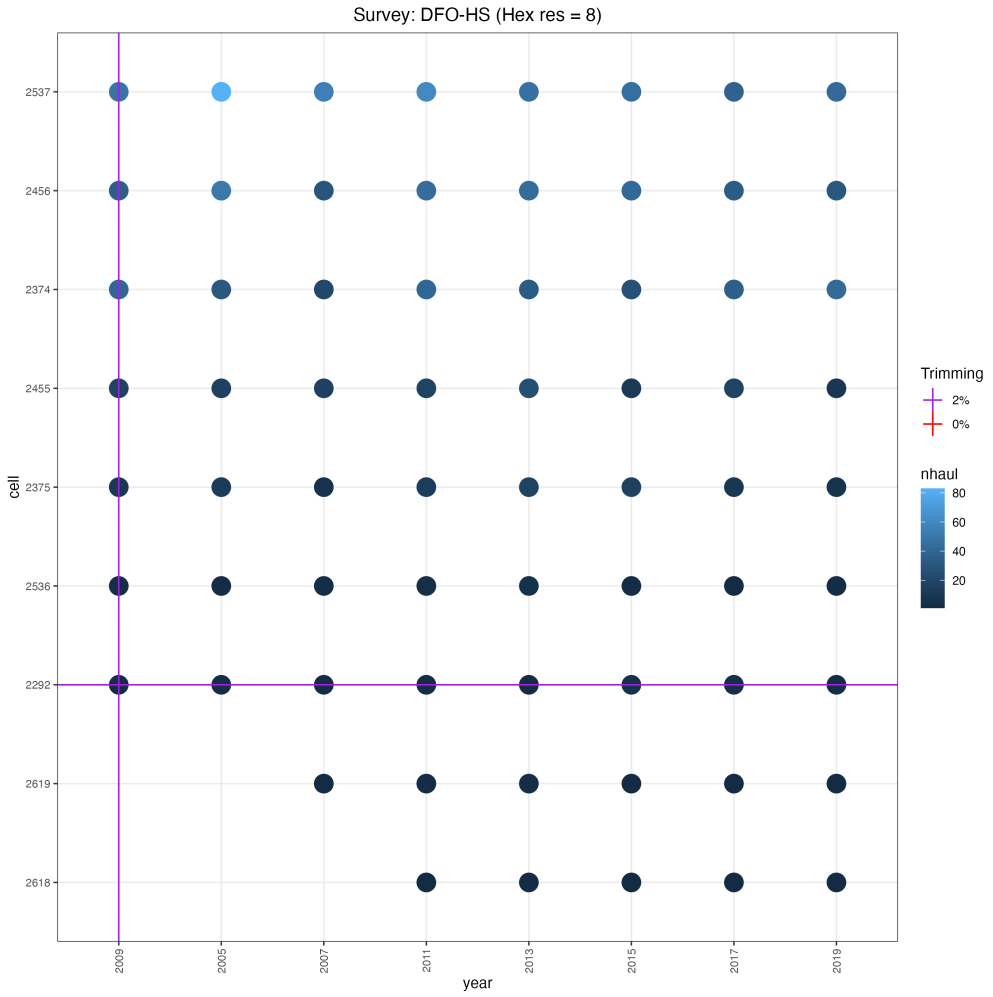
10. Spatio-temporal standardization

a. Standardization method 1

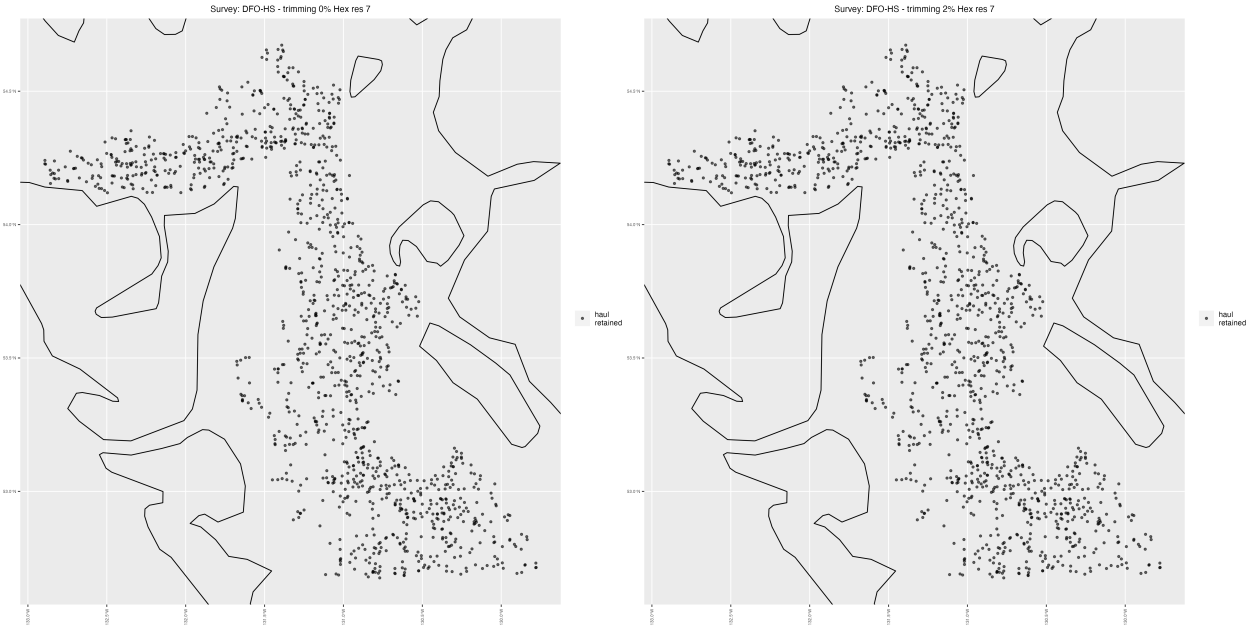
This standardization method was adapted from https://github.com/zoekitchel/trawl_spatial_turnover/blob/master/data_prep_code/species/explore_NorthSea_trimming.Rmd
It was run for hex resolution 7 and 8.

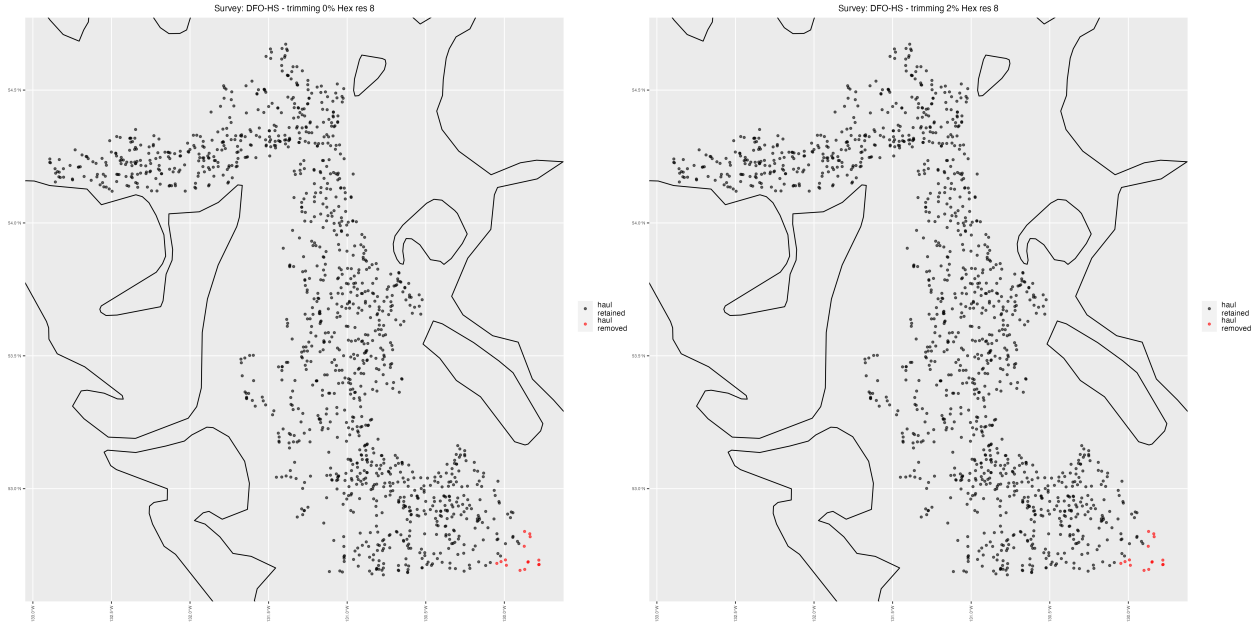
Plot of number of cells x years with overlaid flagging options



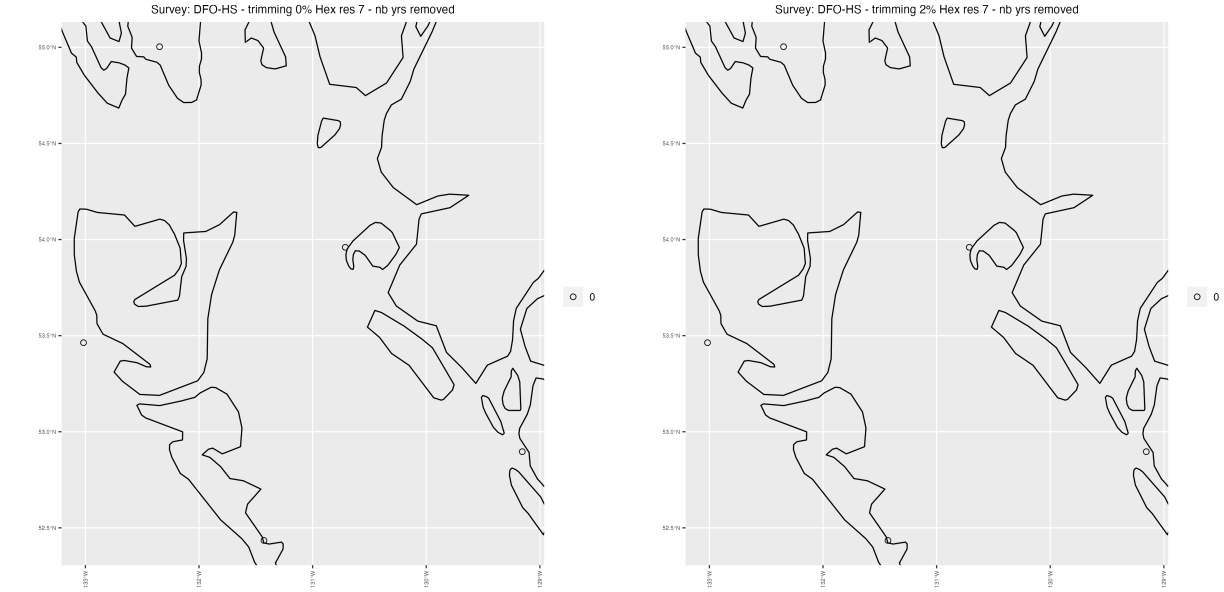


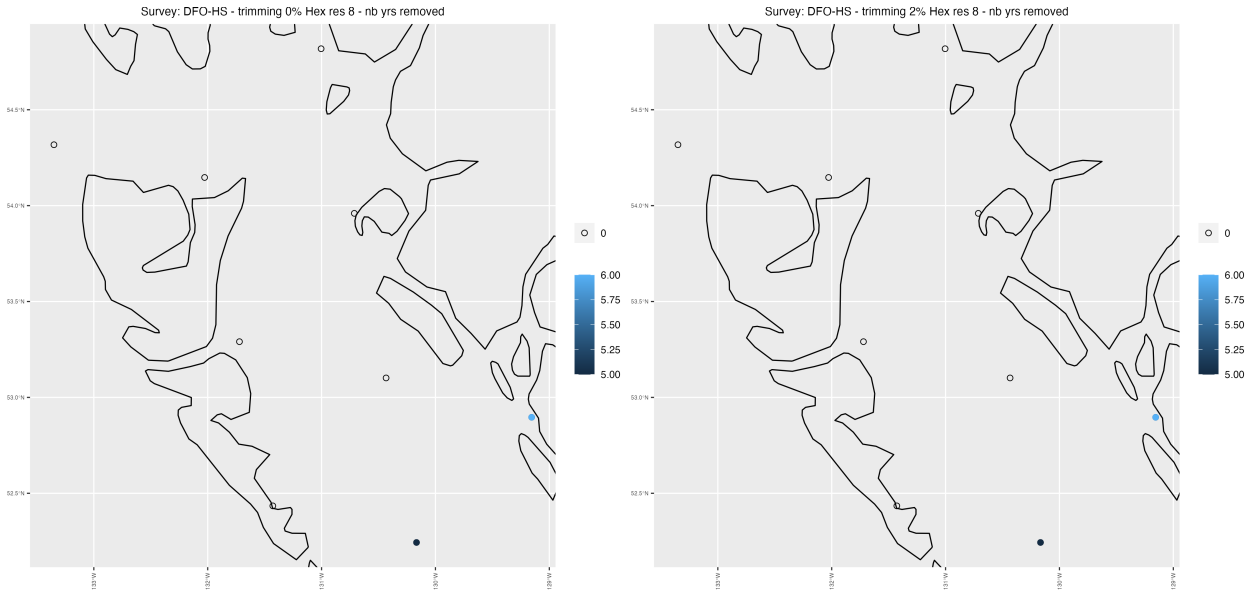
Map of hauls retained and removed per flagging method and threshold





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	0	0	15.0	15.0	620.0
percentage of hauls removed	0	0	1.2	1.2	3.1