

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

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

November, 2023

## Contents

General info . . . . .	1
Data cleaning in R . . . . .	1
1. Overview of the survey data table . . . . .	11
2. Summary of sampling intensity . . . . .	12
3. Summary of sampling variables from the survey . . . . .	13
4. Summary of biological variables . . . . .	14
5. Extreme values . . . . .	15
6. Summary of variables against swept area . . . . .	16
7. Abundance or Weight trends of the six most abundant species . . . . .	17
8. Distribution mapping . . . . .	18
9. Taxonomic flagging . . . . .	19
10. Spatio-temporal standardization . . . . .	20
a. Standardization method 1 . . . . .	20
b. Standardization method 2 . . . . .	23
c. Standardization summary . . . . .	23

## General info

This document presents the cleaning code and summary of the Strait of Georgia Survey (Department of Fisheries Oceans Canada) bottom trawl survey provided by Maria Cornthwaite. It contains data from 2012 and up to 2015.

## Data cleaning in R

```
#####
#### R code to clean trawl survey for the DFO Strait of Georgia Survey
#### NB: this is only 2 years of data
#### 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
#####
#-----#
#### LOAD LIBRARIES AND FUNCTIONS ####
#-----#
```

```

library(tidyverse)
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 Strait of Georgia 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 ####
#-----#

SOG_catch <- read_csv(
  "https://github.com/pinskylab/OceanAdapt/raw/master/data_raw/SOG_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()
  ))

SOG_effort <- read_csv(
  "https://github.com/pinskylab/OceanAdapt/raw/master/data_raw/SOG_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 ####
#-----#

SOG <- left_join(SOG_catch, SOG_effort, by = c("Trip.identifier", "Set.number",
                                             "Survey.Year"))

SOG <- SOG %>%
  # 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
  )
SOG <- SOG %>%
  rename(
    latitude = Start.latitude,
    longitude = Start.longitude,
    depth = Bottom.depth..m.,
    verbatim_name = Scientific.name,
    num = Catch.count..pieces.,
    wgt = Catch.weight..kg.
  ) %>%
  mutate(
    date = as.Date(Set.date),
    year = Survey.Year,
    haul_dur = Tow.duration..min./60
  ) %>%
  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("Bathyrāja", verbatim_name),
                                'Bathyrāja 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 <- SOG %>%
  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
#i.e.
#HEXACTINELLIDA, GLASS SPONGES; WILLEMOES'S WHITE SEA PEN; CRANGONS
SOG <- SOG %>%
  group_by(haul_id, year, latitude, longitude, depth, verbatim_name, area_swept,
           num, wgt, wgt_cpue, wgt_h, num_cpue, num_h, 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)) %>%
  ungroup()

```

```

SOG <- SOG %>%
# add survey column
  mutate(survey = "DFO-SOG",
         source = "DFO",
         timestamp = mdy("08/21/2020"),
         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, source, timestamp, 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_SOG <- SOG %>%
  group_by(haul_id, verbatim_name) %>%
  mutate(count = n())

#none!

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

unique_name_match
#check if empty

#sum duplicates
SOG <- SOG %>%
  group_by(survey,
           source,timestamp,
           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) %>%
  summarise(num = sum(num, 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),
    wgt_h = sum(wgt_h, na.rm = T),
    wgt_cpue = sum(wgt_cpue, na.rm = T)) %>% ungroup()

#duplicated before fix above

#verbatim_name
#MERLUCCIOUS PRODUCTUS
#RAJA RHINA
#PLATICHTHYS STELLATUS
#PORICHTHYS NOTATUS
#GLYPTOCEPHALUS ZACHIRUS
#EUALUS
#APHRODITA
#BRISASTER LATIFRONS
#ECHIURA

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

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

### Automatic cleaning
# Set Survey code
sog_survey_code <- "DFO-SOG"

SOG <- SOG %>%
  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(SOG$taxa2), input_survey = sog_survey_code,
                          save = F, output=NA, fishbase=T)

#This fails to ID 0 species

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

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

clean_sog <- left_join(SOG, 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) %>%
mutate(verbatim_aphia_id = NA,
       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`)

#check for duplicates
count_clean_sog <- clean_sog %>%
  group_by(haul_id, accepted_name) %>%
  mutate(count = n())

#none!

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

unique_name_match
#check if empty

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

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

# -----#
#### 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_sog$survey))

#run flag_spp function in a loop
for (r in regions) {
  flag_spp(clean_sog, 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_sog, 7)

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

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

#-----#
#### ADD STRANDARDIZATION FLAGS ####
#-----#
surveys <- sort(unique(clean_sog$survey))
survey_units <- sort(unique(clean_sog$survey_unit))
survey_std <- clean_sog %>%
  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)){

  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)
# }
}

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

```

## 1. Overview of the survey data table

survey	source	timestamp	haul_id	country	sub_area
DFO-SOG	DFO	2020-08-21	72189-002	Canada	NA
DFO-SOG	DFO	2020-08-21	72189-002	Canada	NA
DFO-SOG	DFO	2020-08-21	72189-002	Canada	NA
DFO-SOG	DFO	2020-08-21	72189-002	Canada	NA
DFO-SOG	DFO	2020-08-21	72189-002	Canada	NA

continent	stat_rec	station	stratum	year	month	day	quarter	season
n_america	NA	NA	NA	2012	3	14	NA	NA
n_america	NA	NA	NA	2012	3	14	NA	NA
n_america	NA	NA	NA	2012	3	14	NA	NA
n_america	NA	NA	NA	2012	3	14	NA	NA
n_america	NA	NA	NA	2012	3	14	NA	NA

latitude	longitude	haul_dur	area_swept	gear	depth	sbt	sst
49.21642	-123.6979	0.3666667	0.108046	NA	387	NA	NA
49.21642	-123.6979	0.3666667	0.108046	NA	387	NA	NA
49.21642	-123.6979	0.3666667	0.108046	NA	387	NA	NA
49.21642	-123.6979	0.3666667	0.108046	NA	387	NA	NA
49.21642	-123.6979	0.3666667	0.108046	NA	387	NA	NA

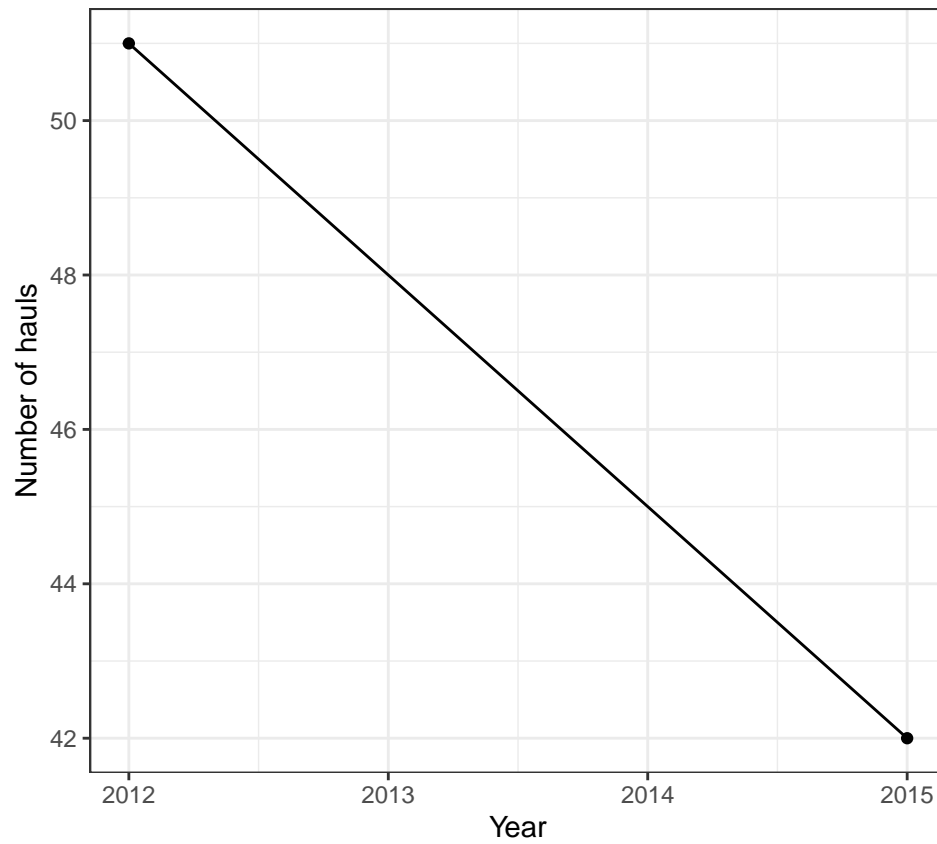
num	num_cpue	num_cpua	wgt	wgt_cpue	wgt_cpua	verbatim_name
0	0.000000	0.000000	0.60	1.6363636	5.5531903	APRISTURUS BRUNNEUS
1	2.727273	9.255317	0.50	1.3636364	4.6276586	ATHERESTHES STOMIAS
0	0.000000	0.000000	36.34	99.1090909	336.3382263	HYDROLAGUS COLLIEI
1	2.727273	9.255317	0.02	0.0545455	0.1851063	LYCODES DIAPTERUS
0	0.000000	0.000000	0.34	0.9272727	3.1468078	LYOPSETTA EXILIS

verbatim_aphia_id	accepted_name	aphia_id	SpecCode	kingdom
NA	Apristurus brunneus	158512	763	Animalia
NA	Atheresthes stomias	279792	517	Animalia
NA	Hydrolagus colliei	271406	2589	Animalia
NA	Lycodes diapterus	254597	3137	Animalia
NA	Lyopsetta exilis	281452	4246	Animalia

phylum	class	order	family	genus	rank	survey_unit
Chordata	Elasmobranchii	Carcharhiniformes	Pentanchidae	Apristurus	Species	DFO-SOG
Chordata	Teleostei	Pleuronectiformes	Pleuronectidae	Atheresthes	Species	DFO-SOG
Chordata	Holocephali	Chimaeriformes	Chimaeridae	Hydrolagus	Species	DFO-SOG
Chordata	Teleostei	Perciformes	Zoarcidae	Lycodes	Species	DFO-SOG
Chordata	Teleostei	Pleuronectiformes	Pleuronectidae	Lyopsetta	Species	DFO-SOG

## 2. Summary of sampling intensity

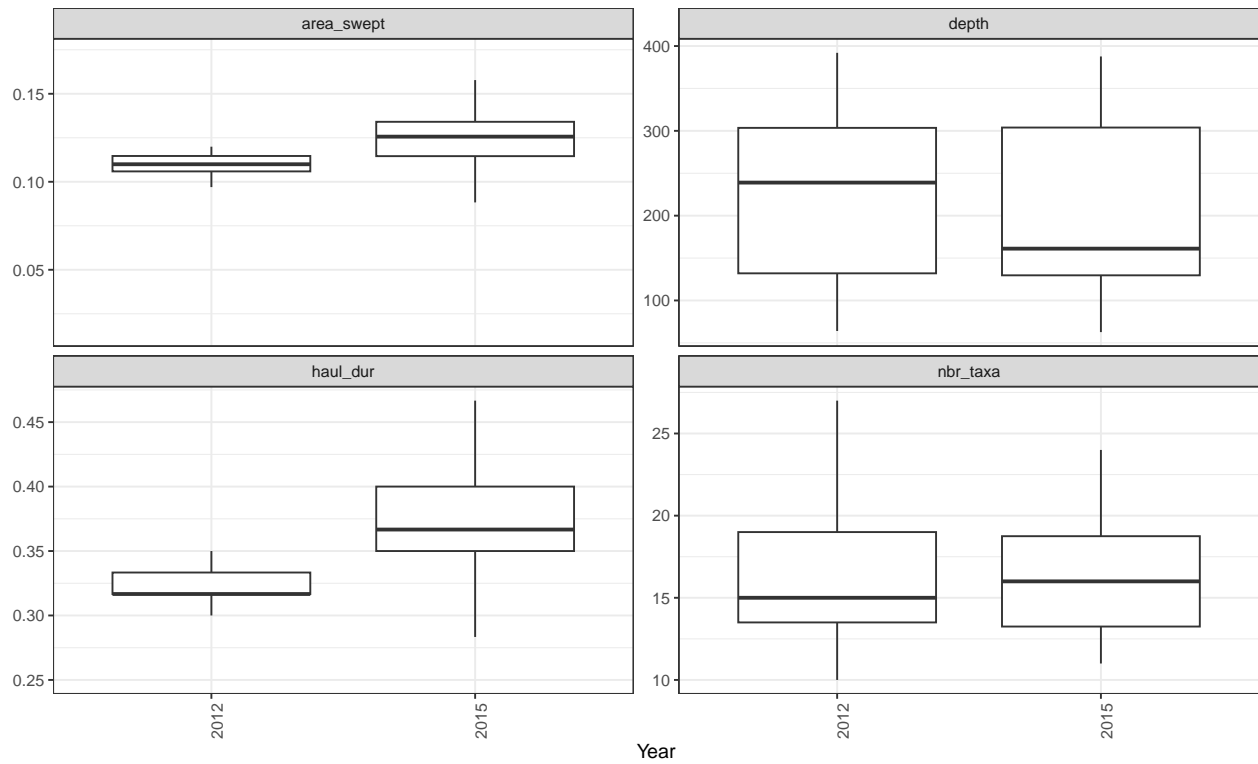
Number of hauls per year performed during the survey after data processing.



### 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:

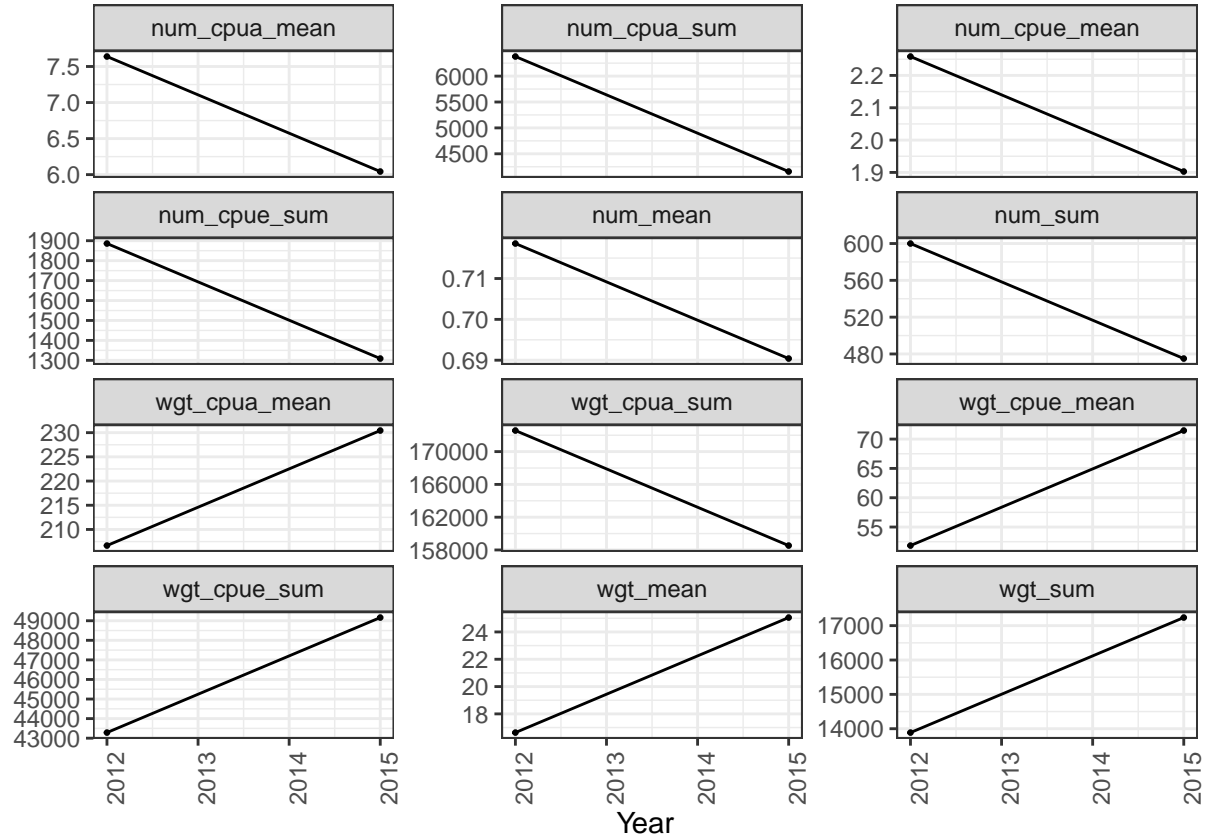
- *area\_swept*, swept area by the bottom trawl gear  $km^2$
- *depth*, sampling depth in *m*
- *haul\_dur*, haul sampling duration *hour*
- *number of marine fish taxa*, taxa were cleaned following the last version of taxonomy from the World Register of Marine Species (<https://www.marinespecies.org/>,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

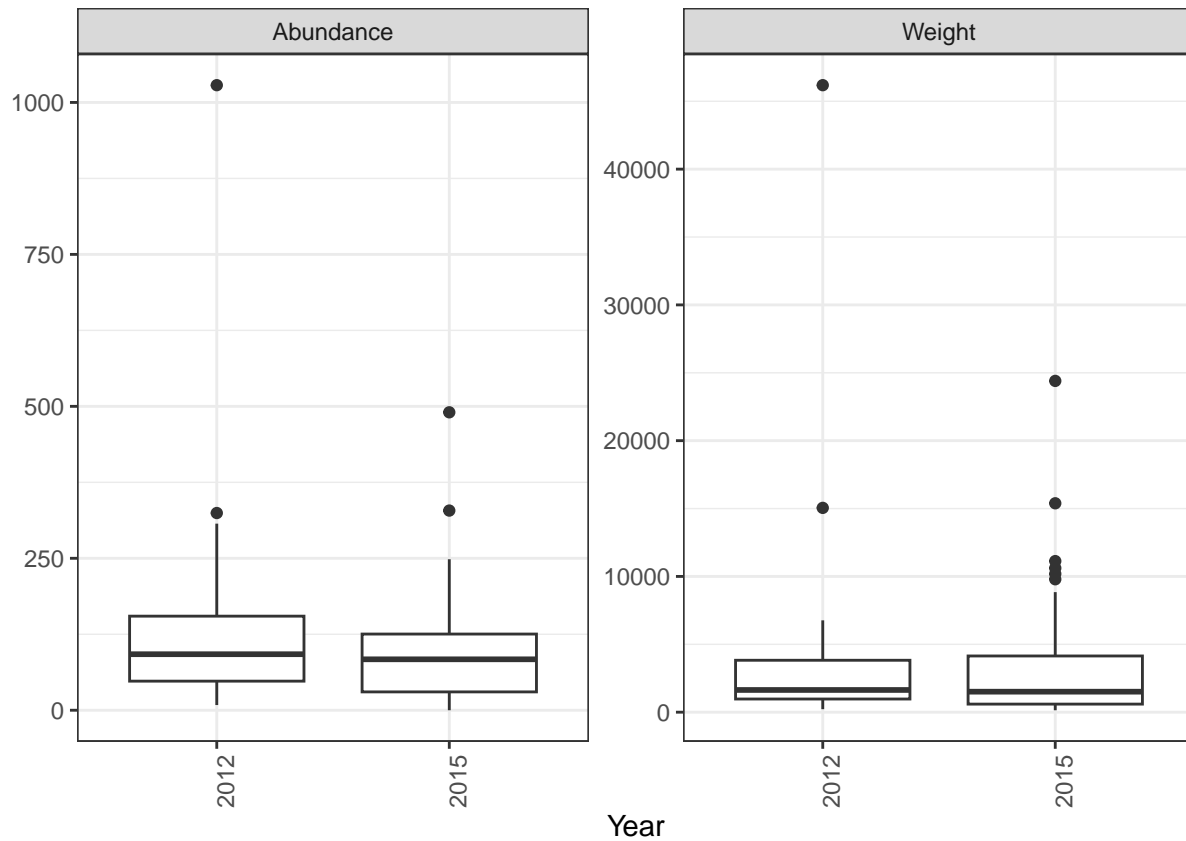
- *num\_cpua*, number of individuals (abundance) in  $\frac{\text{individuals}}{\text{km}^2}$
- *num\_cpue*, number of individuals (abundance) in  $\frac{\text{individuals}}{h}$
- *num*, number of individuals (abundance)
- *wgt\_cpua*, weight in  $\frac{kg}{\text{km}^2}$
- *wgt\_cpue*, weight in  $\frac{kg}{h}$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



## 5. Extreme values

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

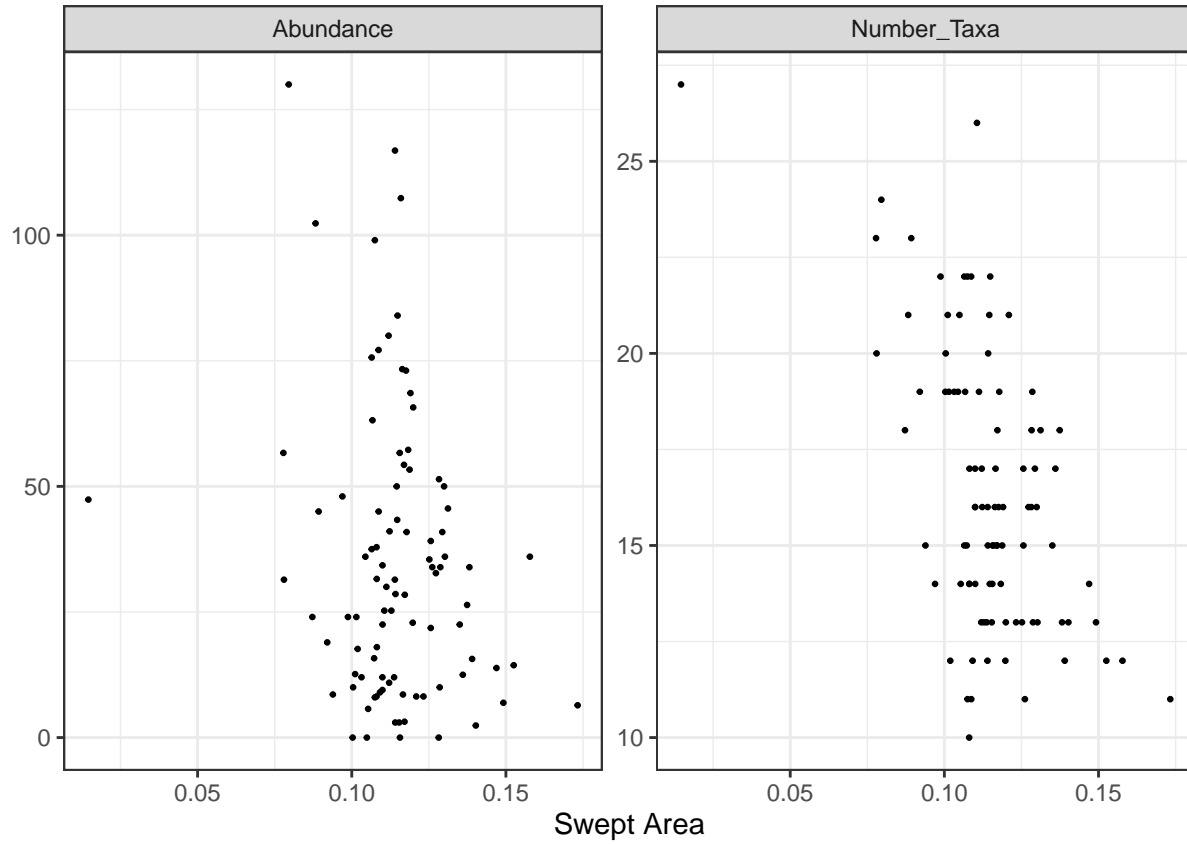
- *num\_cpue*, number of individuals (abundance) in  $\frac{\text{individuals}}{\text{km}^2}$
- *wgt\_cpue*, weight in  $\frac{\text{kg}}{\text{km}^2}$



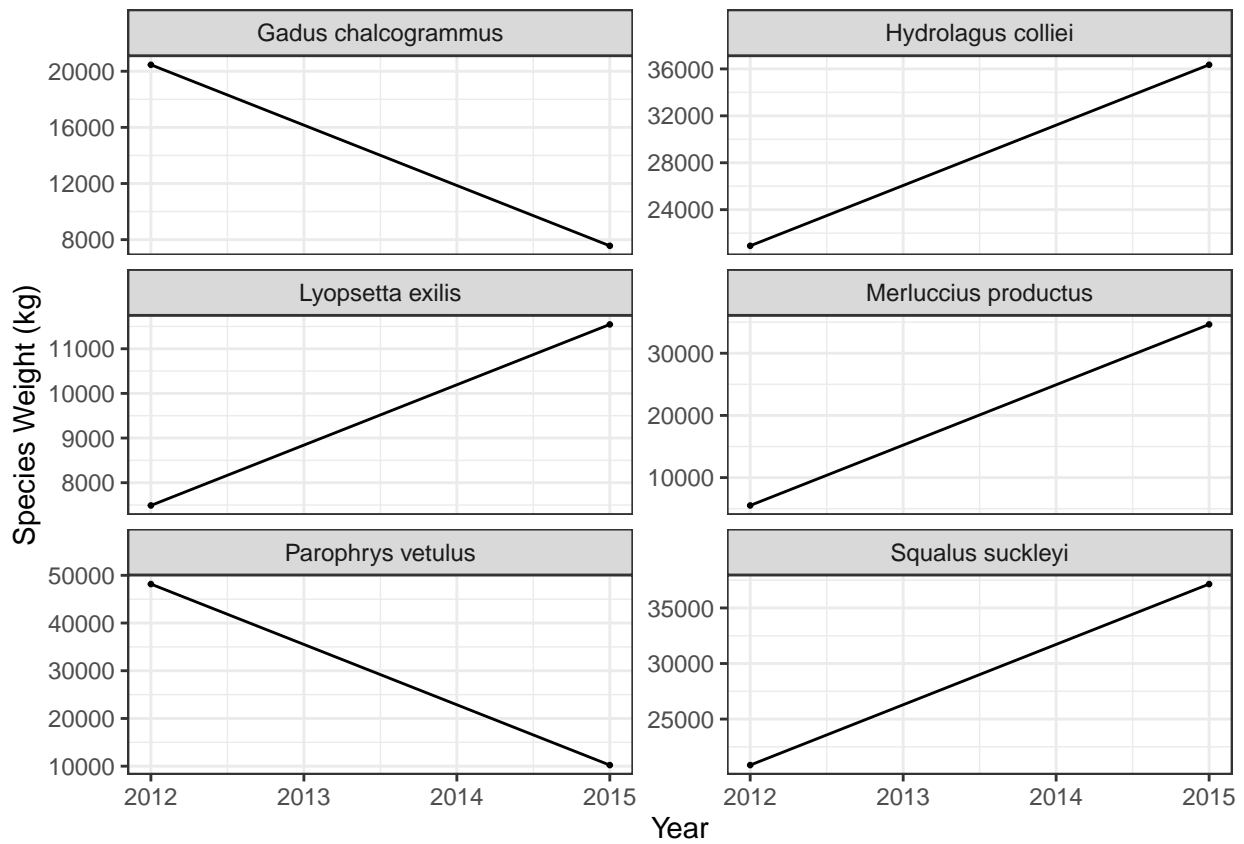
## 6. Summary of variables against swept area

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

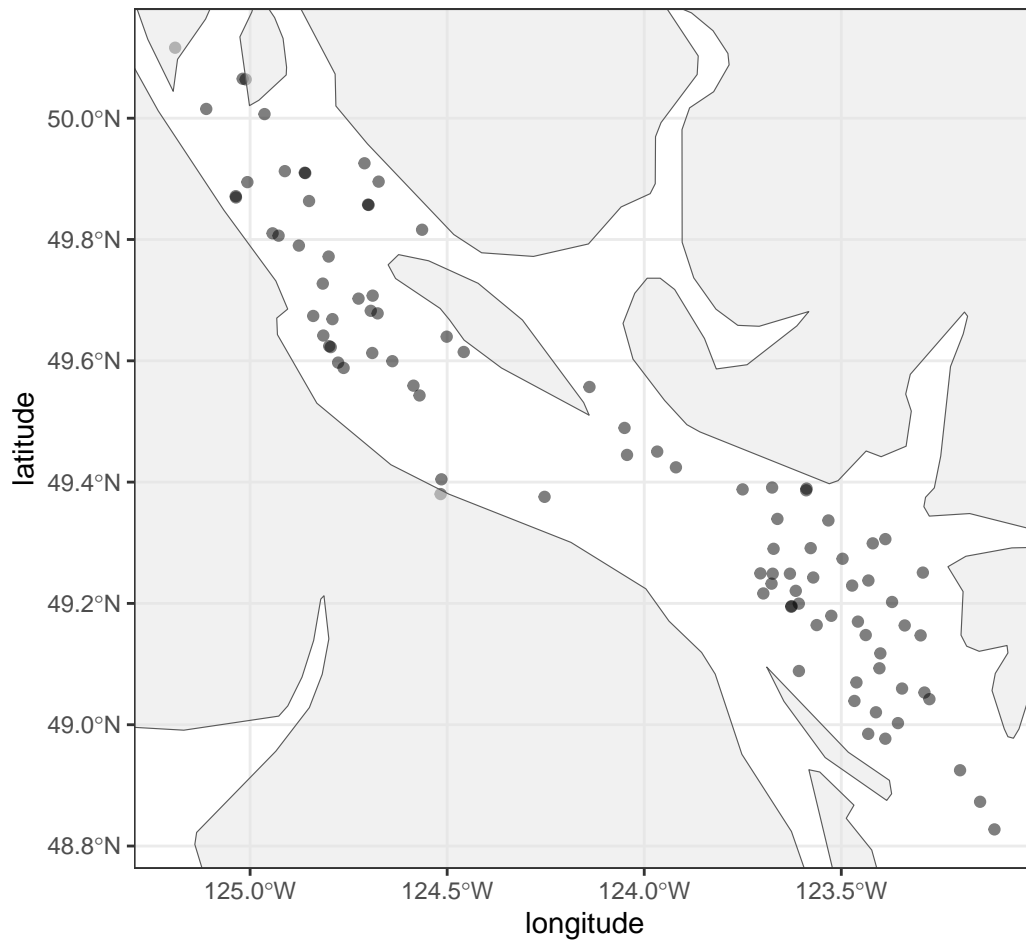


## 7. Abundance or Weight trends of the six most abundant species



## 8. Distribution mapping

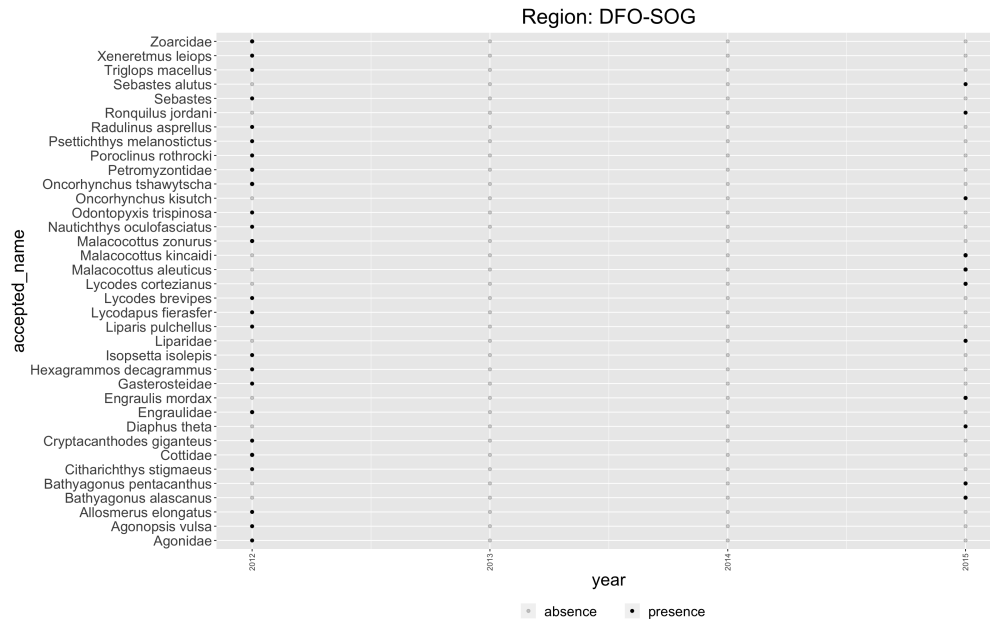
Map of the sampling distribution in space. Note that we only show one year per coordinate.



## 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

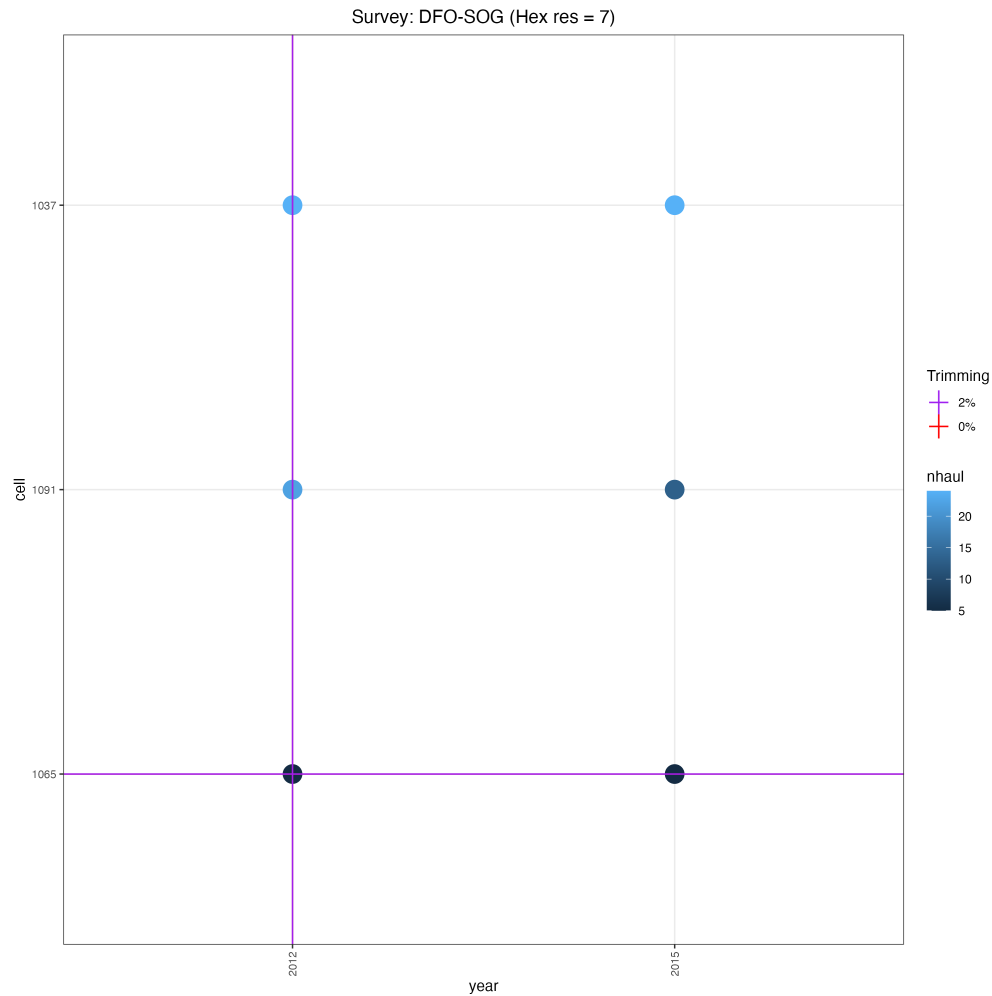
Total number of species	84.0
Percentage of species flagged	42.9

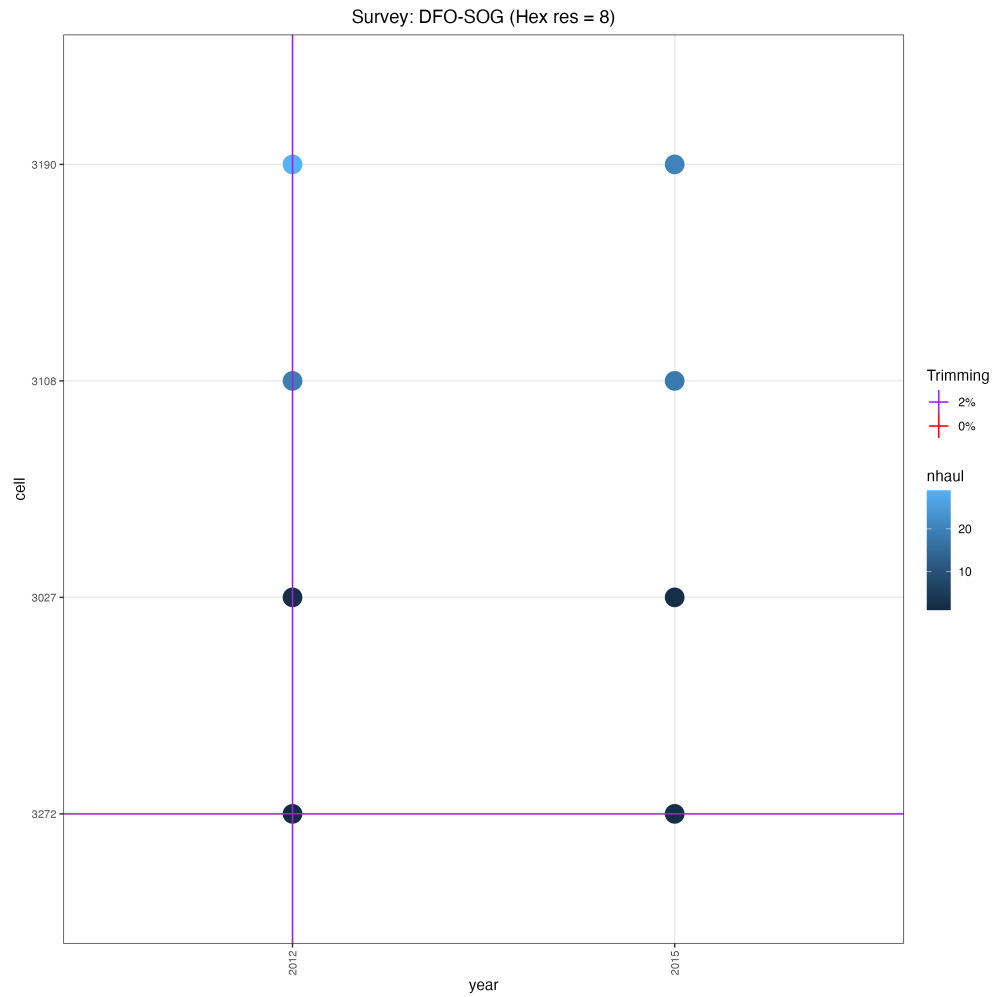
## 10. Spatio-temporal standardization

### a. Standardization method 1

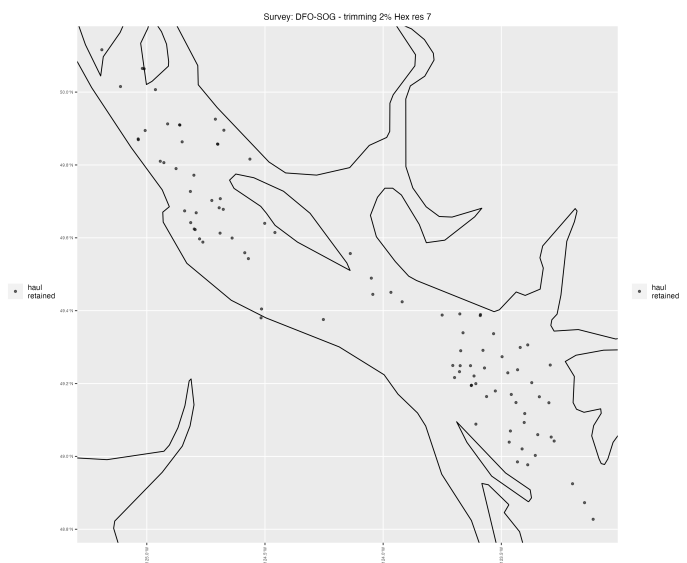
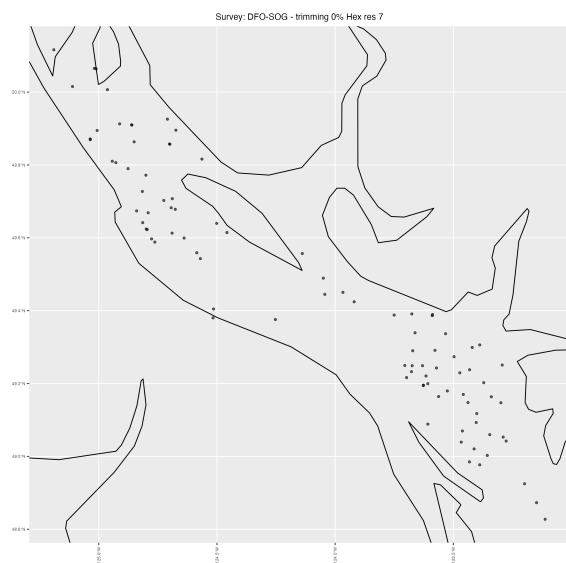
This standardization method was adapted from [https://github.com/zoekitchel/trawl\\_spatial\\_turnover/blob/master/data\\_prep\\_code/species/explore\\_NorthSea\\_trimming.Rmd](https://github.com/zoekitchel/trawl_spatial_turnover/blob/master/data_prep_code/species/explore_NorthSea_trimming.Rmd)  
It was run for hex resolution 7 and 8.

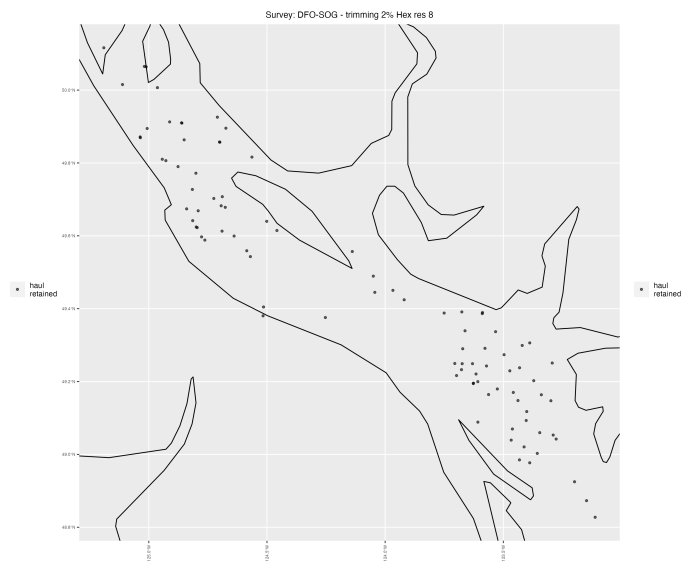
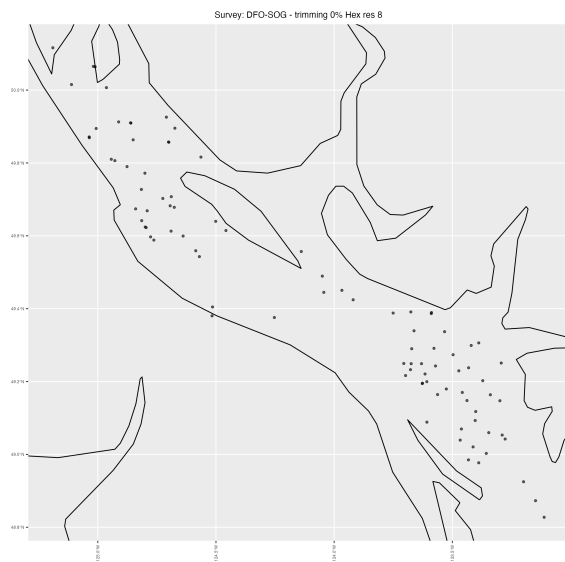
Plot of number of cells x years with overlaid flagging options



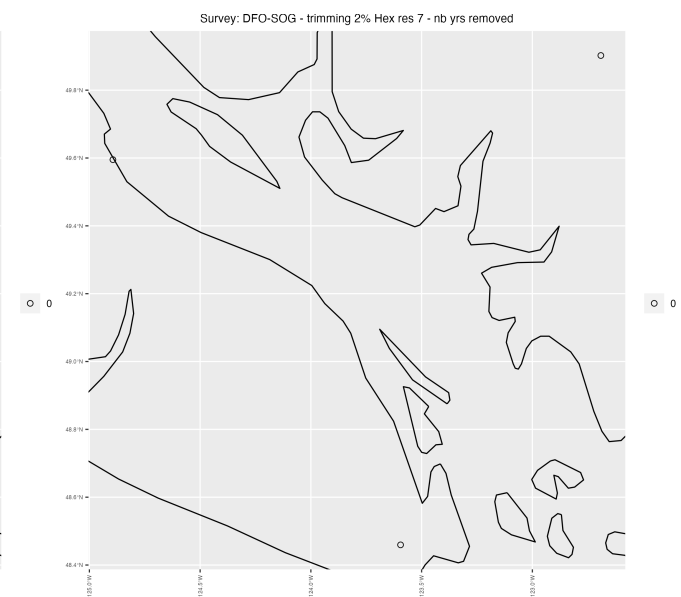
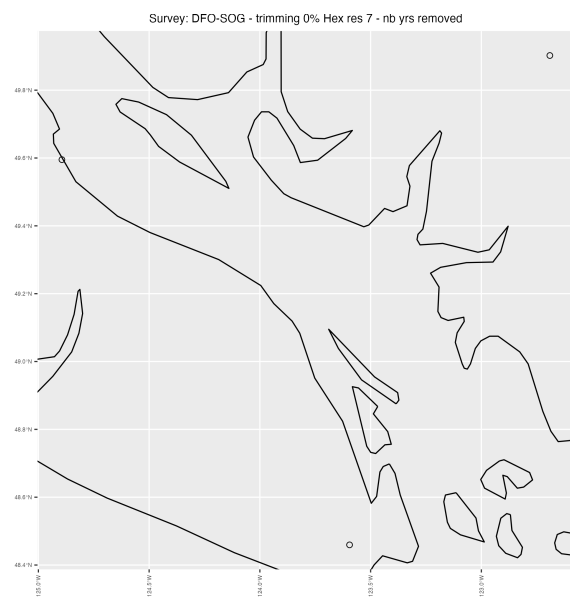


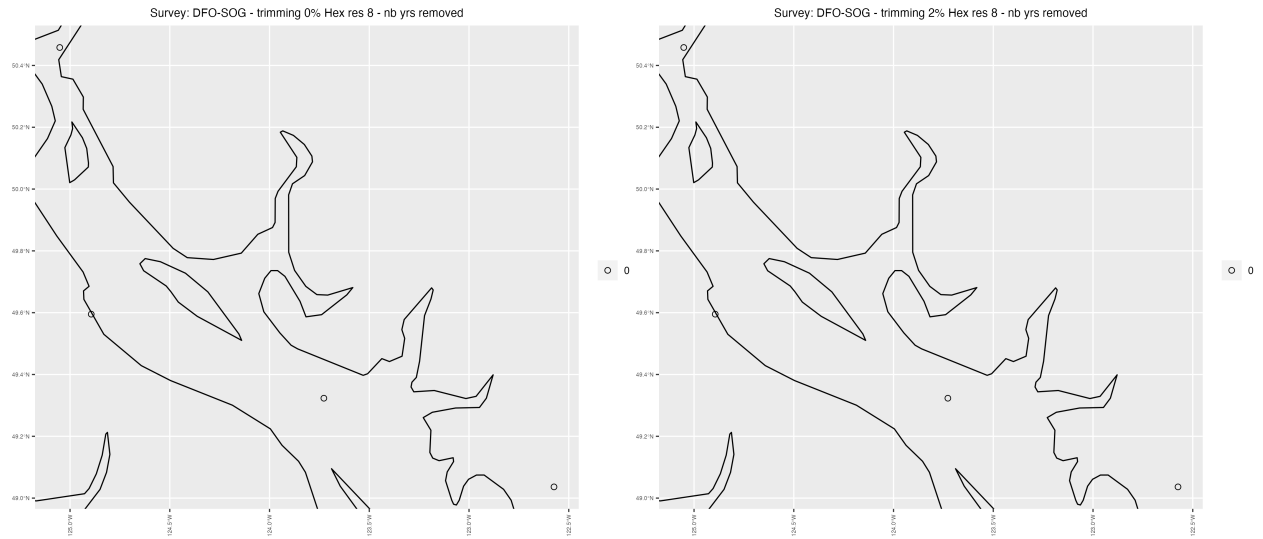
Map of hauls retained and removed per flagging method and threshold





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](https://github.com/Wubing-Xu/Range_size_winners_losers). This method could not be applied to this survey because it contains too few years.

## 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
number of hauls removed	0	0	0	0
percentage of hauls removed	0	0	0	0