DFO-WCHG: Department of Fisheries Oceans Canada Haida Gwaii 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 West Coast Haida Gwaii (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 West Coast Haida Gwaii 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
###############################################################################
#-----------------------------------------------------------------------------#
#### LOAD LIBRARIES AND FUNCTIONS ####
#-----------------------------------------------------------------------------#
```

```r
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)

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 West Coast Haida Gwaii 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 ####
#------------------------------------------------------------------------------#

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

WCHG_effort <- read_csv(
  "https://github.com/pinskylab/OceanAdapt/raw/master/data_raw/WCHG_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(),
```

```r
                              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 ####
#-----------------------------------------------------------------------------------#

WCHG <- left_join(WCHG_catch, WCHG_effort, by = c("Trip.identifier", "Set.number",
                                                  "Survey.Year"))


WCHG <- WCHG %>%
  # 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
  )

WCHG <- WCHG %>%
  rename(
    latitude = Start.latitude,
    longitude = Start.longitude,
```

```r
    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
  ) %>%
  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 <- WCHG %>%
  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
WCHG <- WCHG %>%
  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()

WCHG <- WCHG %>%
# add survey column etc.
  mutate(survey = "DFO-WCHG",
         source = "DFO",
         timestamp = mdy("08/21/2020"),
         country = "Canada",
         continent = "n_america",
         stat_rec = NA,
```

```r
          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
  ) %>%
  group_by(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, verbatim_aphia_id) %>%
  #this step sums over matching haul_ids and species
  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)) %>%

    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, 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_WCHG <- WCHG %>%
  group_by(haul_id, verbatim_name) %>%
  mutate(count = n())

#none!

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

unique_name_match
#empty

#The following are all duplicated if we don't sum over abundance and wgt (added above)
#1 SEBASTES REEDI
#2 SCYPHOZOA
#3 ACTINIARIA
#4 ZOROASTER EVERMANI
#5 CAREPROCTUS MELANURUS
```

```r
#6 GLYPTOCEPHALUS ZACHIRUS
#7 PANDALUS PLATYCEROS
#8 SEBASTES DIPLOPROA
#9 XENERETMUS LEIOPS
#10 BATHYRAJA INTERRUPTA
#11 MYCTOPHIDAE
#12 PRIMNOA
#13 ATHERESTHES STOMIAS
#14 ALLOCENTROTUS FRAGILIS
#15 CORYPHAENOIDES CINEREUS
#16 SEBASTES ALUTUS
#17 SEBASTES ALEUTIANUS/MELANOSTICTUS COMPLEX
#18 CYANEA CAPILLATA


#-------------------------------------------------------------------------------#
#### 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
wchg_survey_code <- "DFO-WCHG"

WCHG <- WCHG %>%
  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(WCHG$taxa2),
                         input_survey = wchg_survey_code, save = F, output=NA,
                         fishbase=T)
#takes  2.5 minutes

#This leaves out the following species, which are all inverts

#Nearchaster variabilis
#Cheiraster dawsoni
#Pandalopsis
#Nearchaster aciculosus


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

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

```r
clean_wchg <- left_join(WCHG, 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_wchg <- clean_wchg %>%
  group_by(haul_id, accepted_name) %>%
  mutate(count = n())

#none!

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

unique_name_match
#not empty

#one duplicate kept
#taxonomic cleaning
#verbatim_name                              accepted_name
#SEBASTES                                   Sebastes
#SEBASTES ALEUTIANUS/MELANOSTICTUS COMPLEX Sebastes
# -----------------------------------------------------------------------------#
#### SAVE DATABASE IN GOOGLE DRIVE ####
# -----------------------------------------------------------------------------#

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




# -----------------------------------------------------------------------------#
#### FAGS ####
# -----------------------------------------------------------------------------#
#install required packages that are not already installed
required_packages <- c("data.table",
```

```r
                          "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 colum)
regions <- levels(as.factor(clean_wchg$survey))

#run flag_spp function in a loop
for (r in regions) {
  flag_spp(clean_wchg, 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_wchg, 7)

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

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


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

```r
  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])
```

```r
  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 = NA_character_)

  rm(hex_res7_0, hex_res7_2, hex_res8_0, hex_res8_2)
 }
}


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

1. Overview of the survey data table

survey	source	timestamp	haul_id	country	sub_area
DFO-WCHG	DFO	2020-08-21	62066-001	Canada	NA
DFO-WCHG	DFO	2020-08-21	62066-001	Canada	NA
DFO-WCHG	DFO	2020-08-21	62066-001	Canada	NA
DFO-WCHG	DFO	2020-08-21	62066-001	Canada	NA
DFO-WCHG	DFO	2020-08-21	62066-001	Canada	NA

continent	stat_rec	station	stratum	year	month	day	quarter	season
n_america	NA	NA	NA	2006	8	30	NA	NA
n_america	NA	NA	NA	2006	8	30	NA	NA
n_america	NA	NA	NA	2006	8	30	NA	NA
n_america	NA	NA	NA	2006	8	30	NA	NA
n_america	NA	NA	NA	2006	8	30	NA	NA

latitude	longitude	haul_dur	area_swept	gear	depth	sbt	sst
52.98417	-132.5998	0.6166667	0.16075	NA	1067.5	NA	NA
52.98417	-132.5998	0.6166667	0.16075	NA	1067.5	NA	NA
52.98417	-132.5998	0.6166667	0.16075	NA	1067.5	NA	NA
52.98417	-132.5998	0.6166667	0.16075	NA	1067.5	NA	NA
52.98417	-132.5998	0.6166667	0.16075	NA	1067.5	NA	NA

num	num_cpue	num_cpua	wgt	wgt_cpue	wgt_cpua	verbatim_name
0	0.000000	0.00000	27.28	44.2378378	169.7045101	ALBATROSSIA PECTORALIS
0	0.000000	0.00000	53.58	86.8864865	333.3125972	ANOPLOPOMA FIMBRIA
0	0.000000	0.00000	3.04	4.9297297	18.9113530	ANTIMORA MICROLEPIS
3	4.864865	18.66252	0.12	0.1945946	0.7465008	BATHYLAGIDAE
2	3.243243	12.44168	1.36	2.2054054	8.4603421	BATHYRAJA TRACHURA

verbatim_aphia_id	accepted_name	aphia_id	SpecCode	kingdom
NA	Albatrossia pectoralis	236135	8435	Animalia
NA	Anoplopoma fimbria	159463	512	Animalia
NA	Antimora microlepis	272460	2006	Animalia
NA	Bathylagidae	125509	NA	Animalia
NA	Bathyraja trachura	271538	2571	Animalia

| phylum	class	order	family	genus	rank	survey_unit
Chordata	Teleostei	Gadiformes	Macrouridae	Albatrossia	Species	DFO-WCHG
Chordata	Teleostei	Perciformes	Anoplopomatidae	Anoplopoma	Species	DFO-WCHG
Chordata	Teleostei	Gadiformes	Moridae	Antimora	Species	DFO-WCHG
Chordata	Teleostei	Argentiniformes	Bathylagidae	NA	Family	DFO-WCHG
Chordata	Elasmobranchii	Rajiformes	Arhynchobatidae	Bathyraja	Species	DFO-WCHG

2. Summary of sampling intensity

Number of hauls per year performed during the survey after data processing.
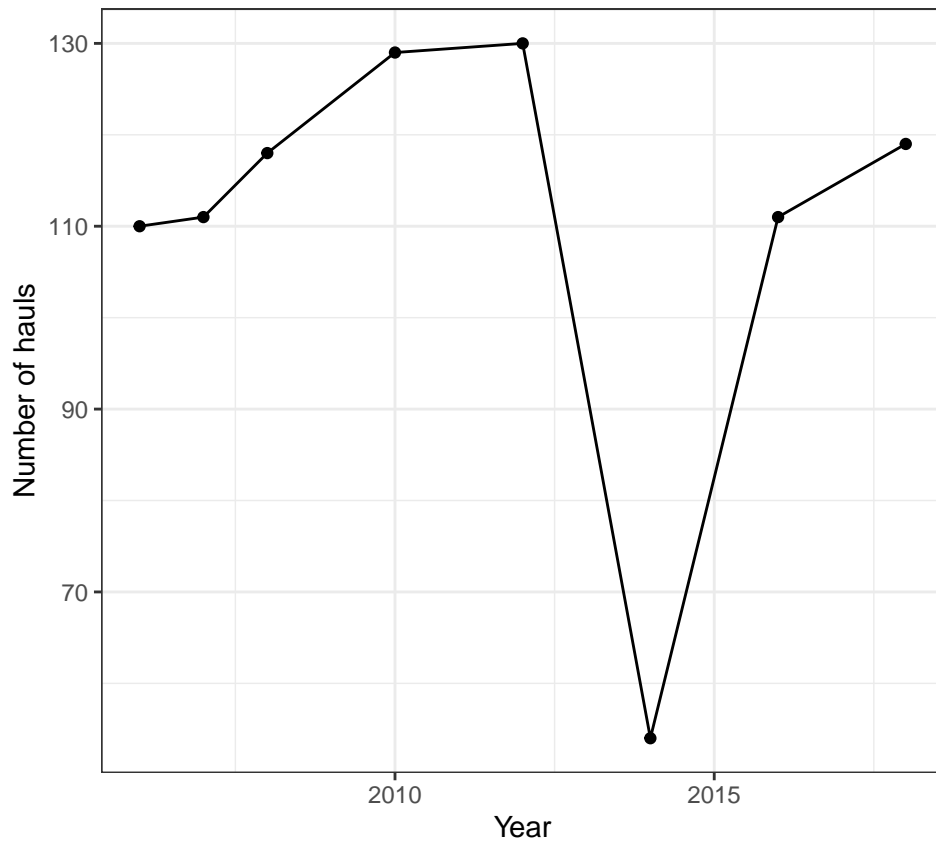
3. Summary of sampling variables from the survey

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)
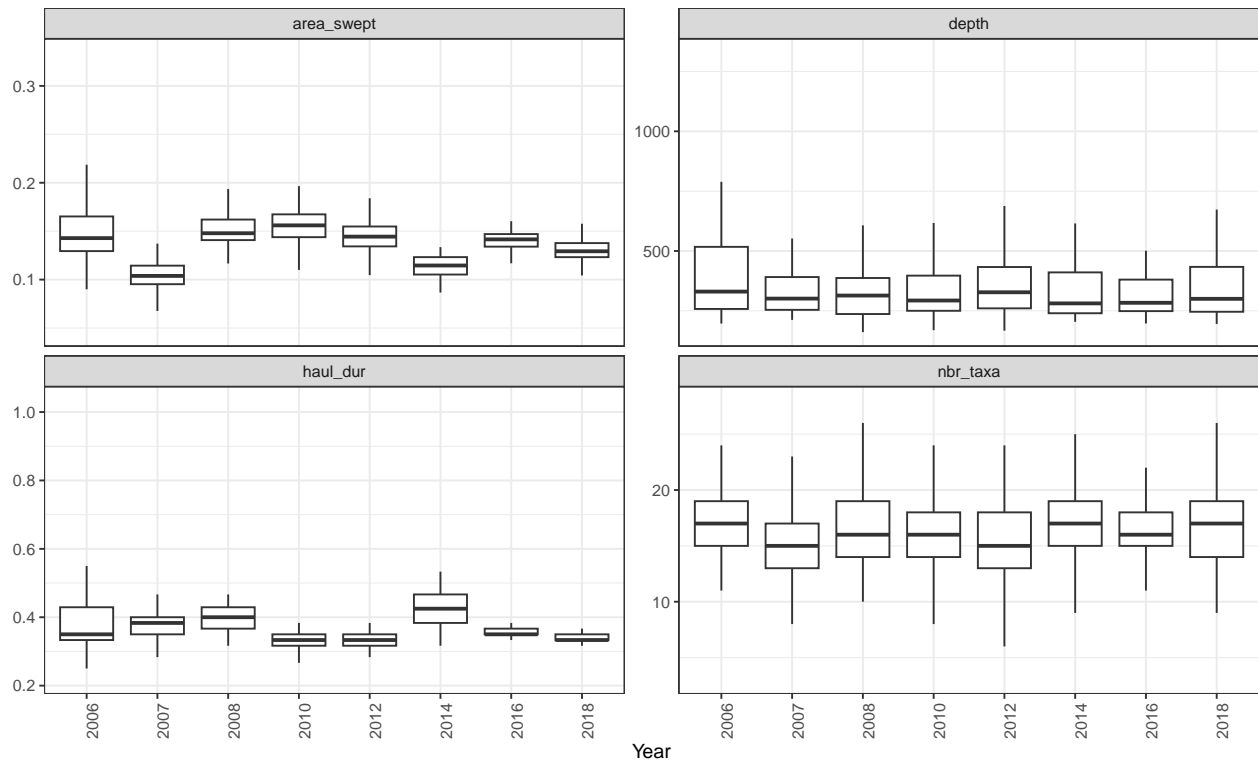
4. Summary of biological variables

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
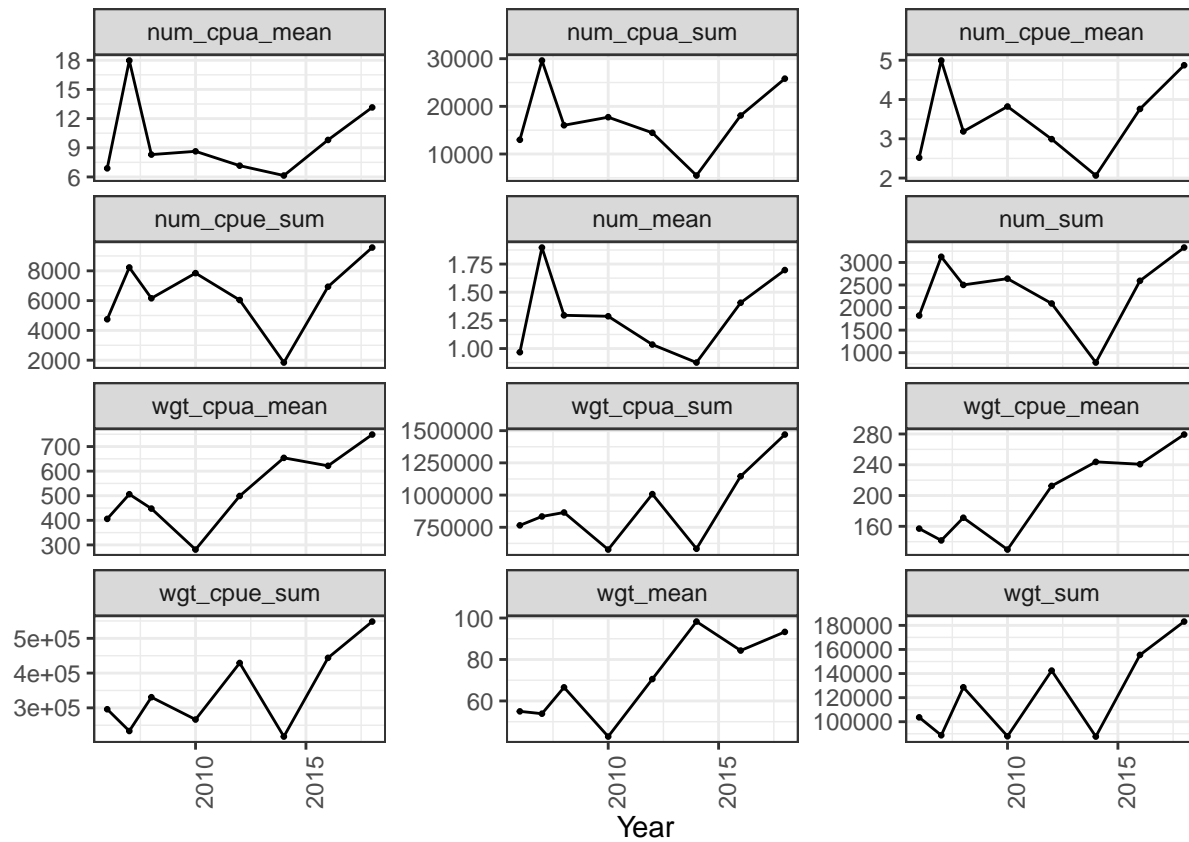
5. Extreme values

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

Year
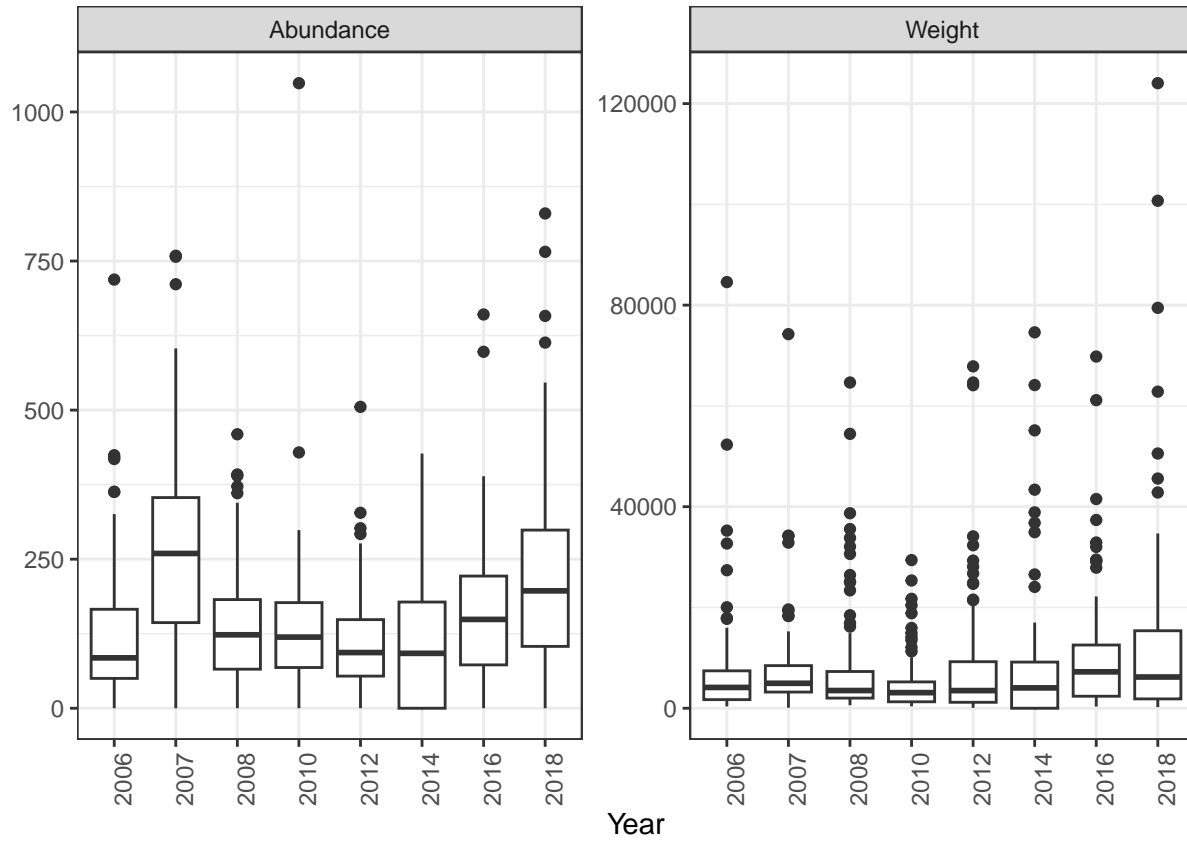
6. Summary of variables against swept area

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 $\frac{kg}{km^2}$
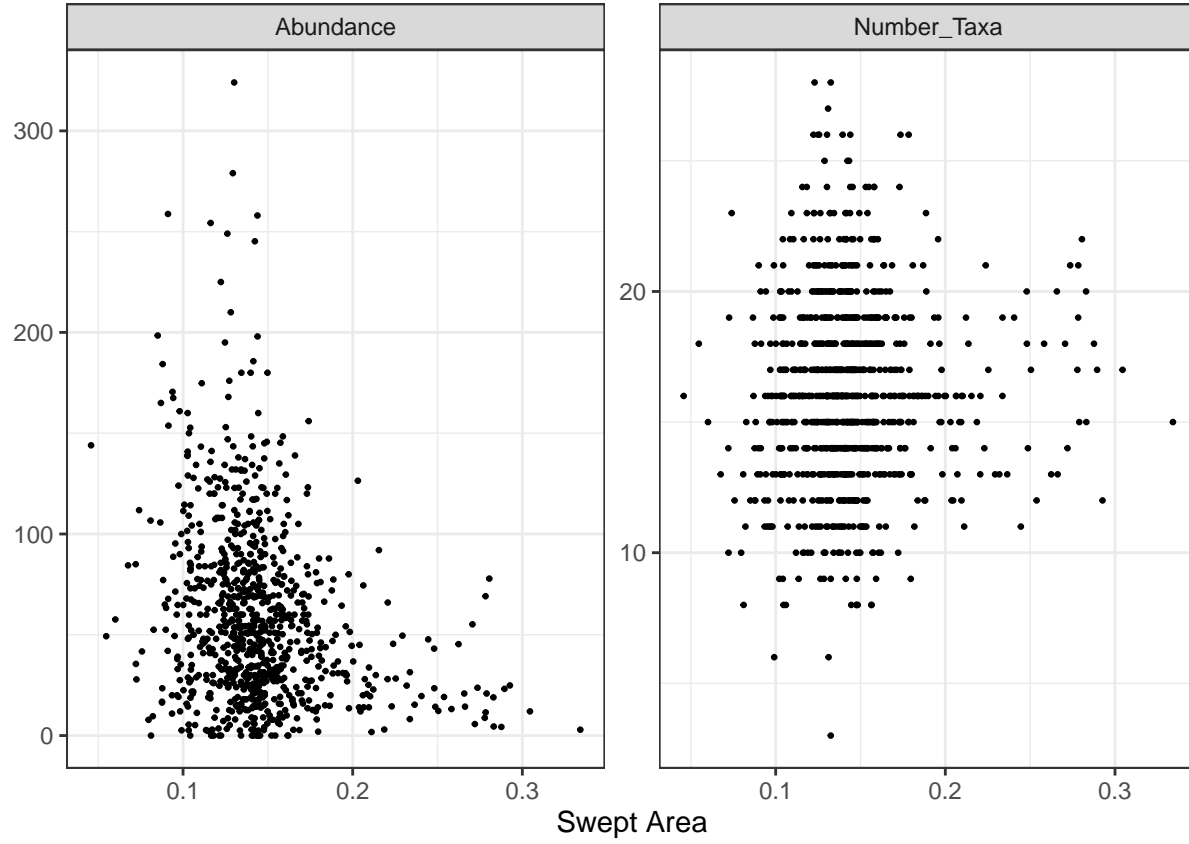
7. Abundance or Weight trends of the six most abundant species
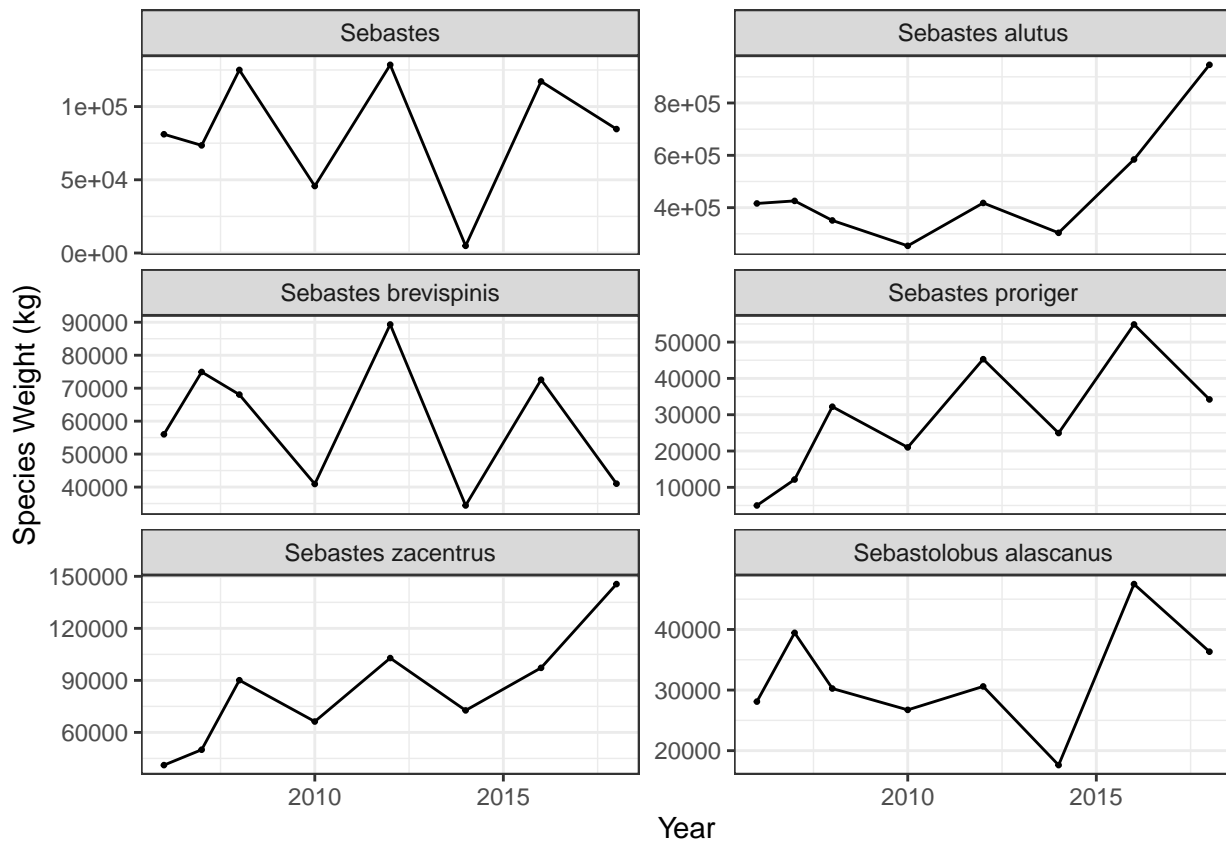
8. Distribution mapping

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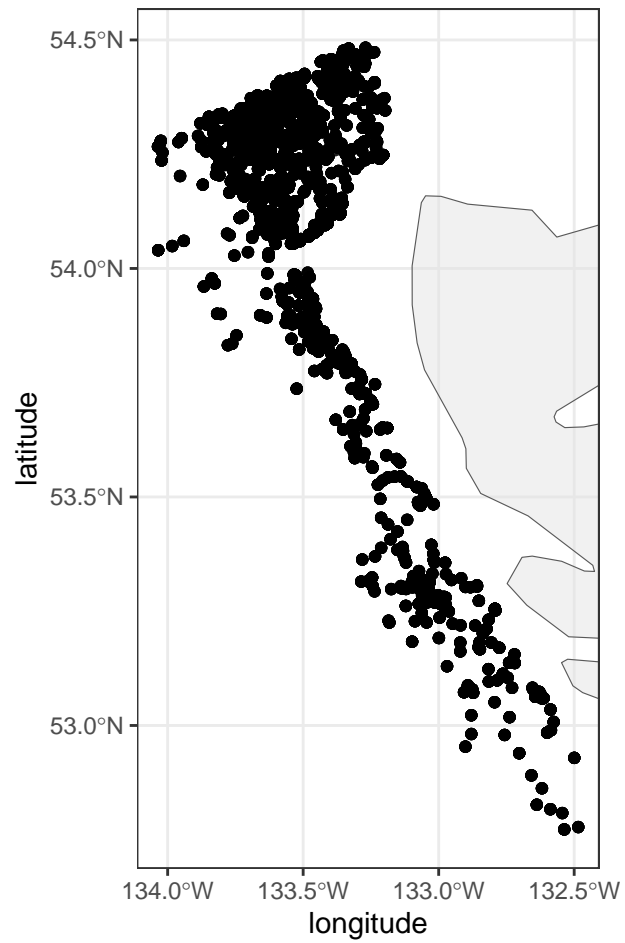
9. Taxonomic flagging

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
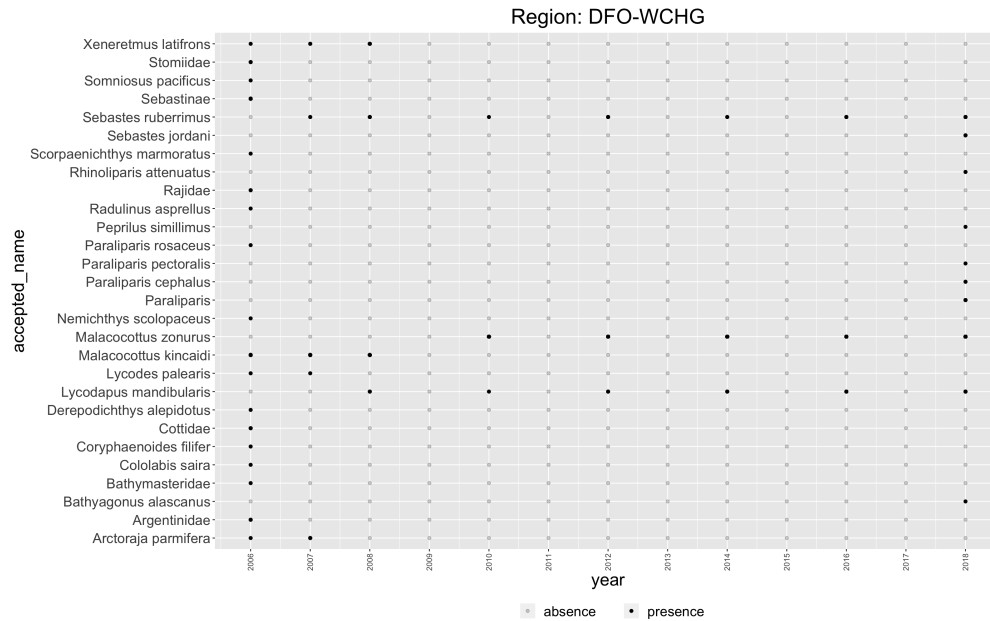
Statistics related to the taxonomic flagging outputs

Total number of species	177.0
Percentage of species flagged	15.8

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
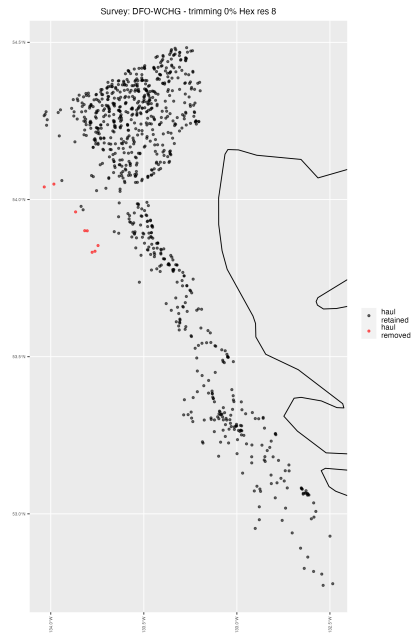It was run for hex resolution 7 and 8.

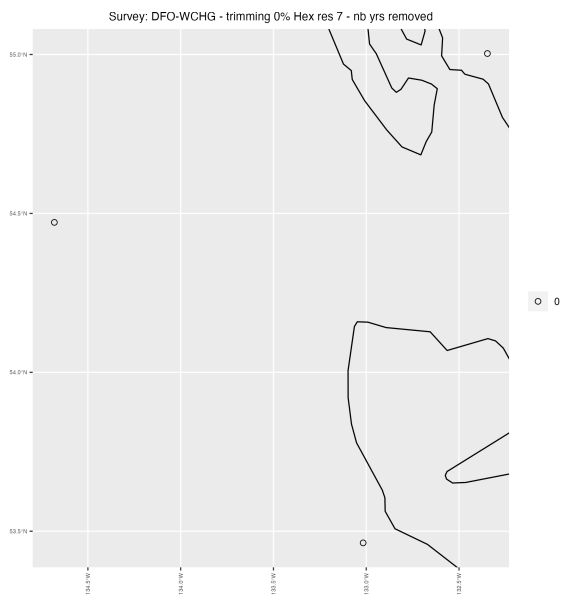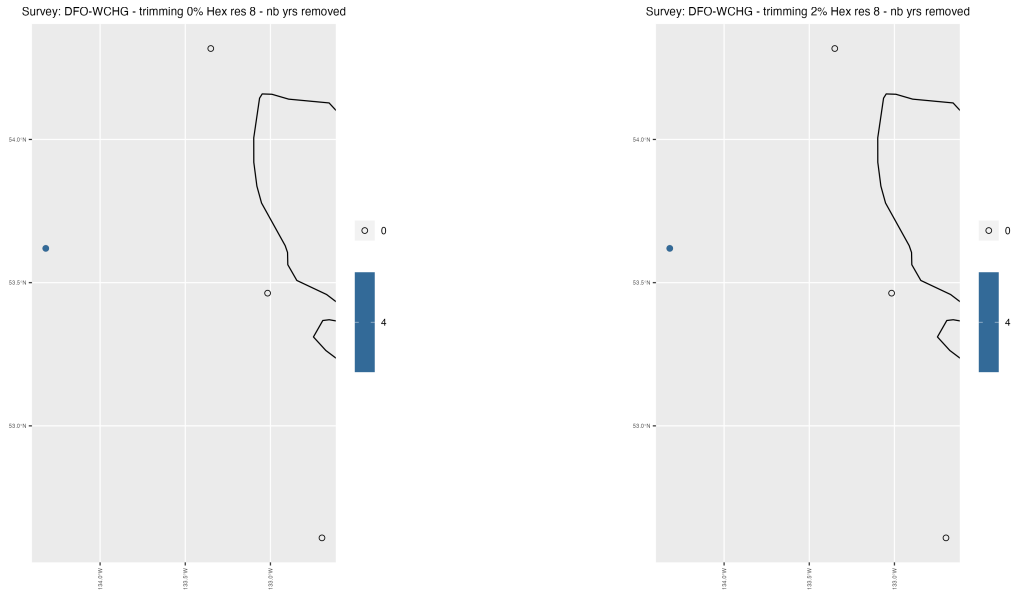Plot of number of cells x years with overlaid flagging options

Survey: DFO-WCHG (Hex res = 8)
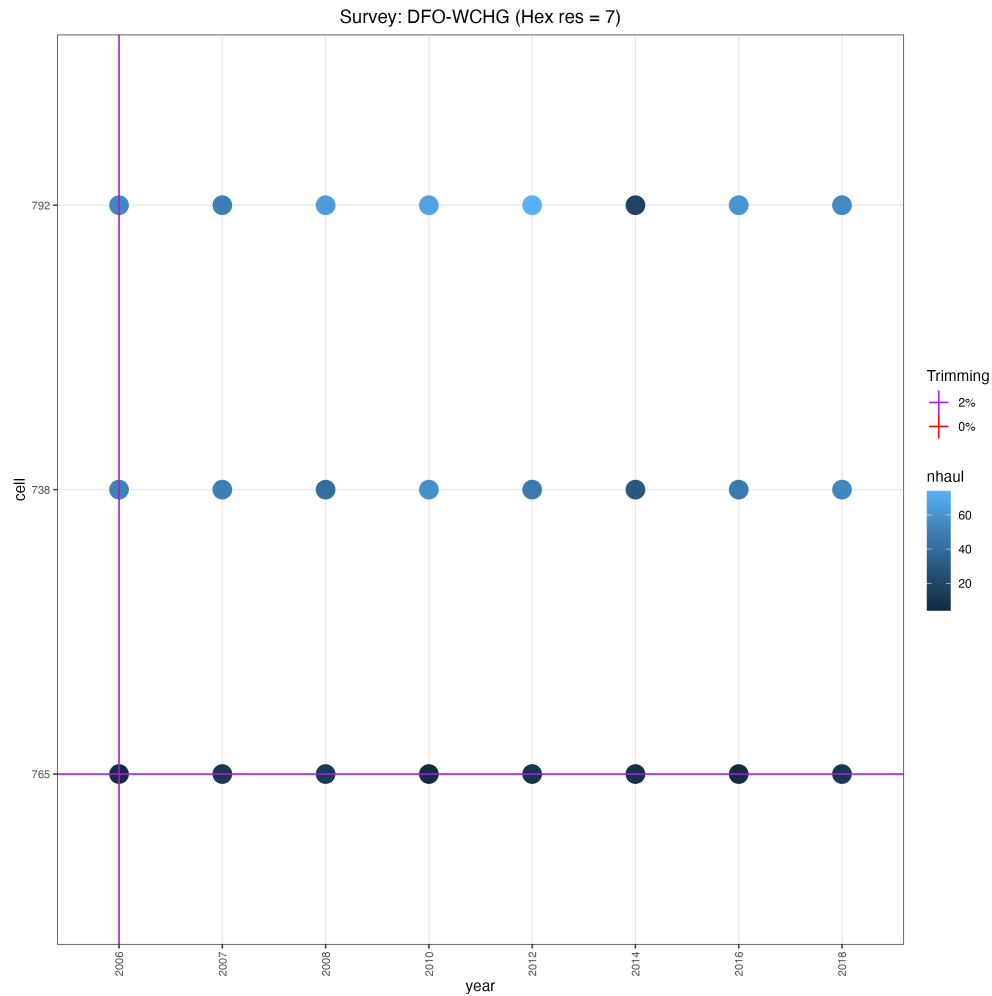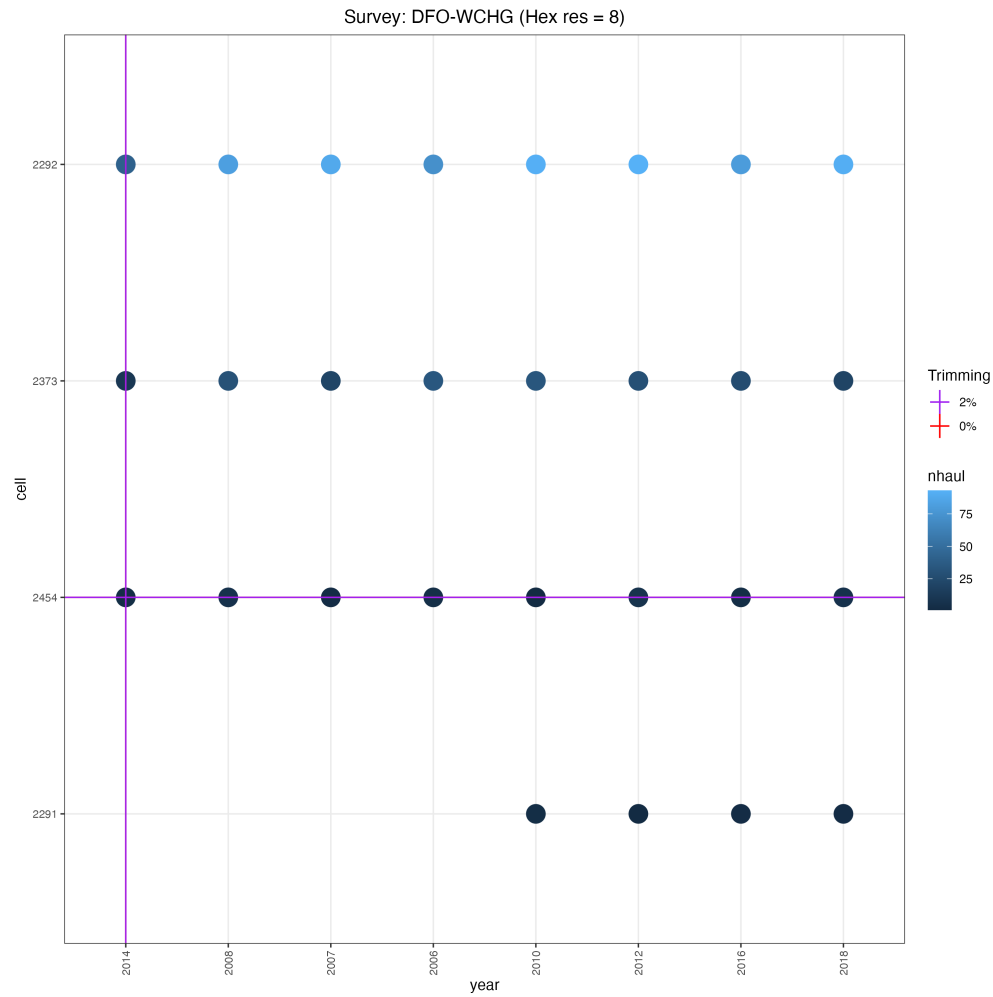
Map of hauls retained and removed per flagging method and threshold

Survey: DFO-WCHG - trimming 0% Hex res 7

Survey: DFO-WCHG - trimming 2% Hex res 7
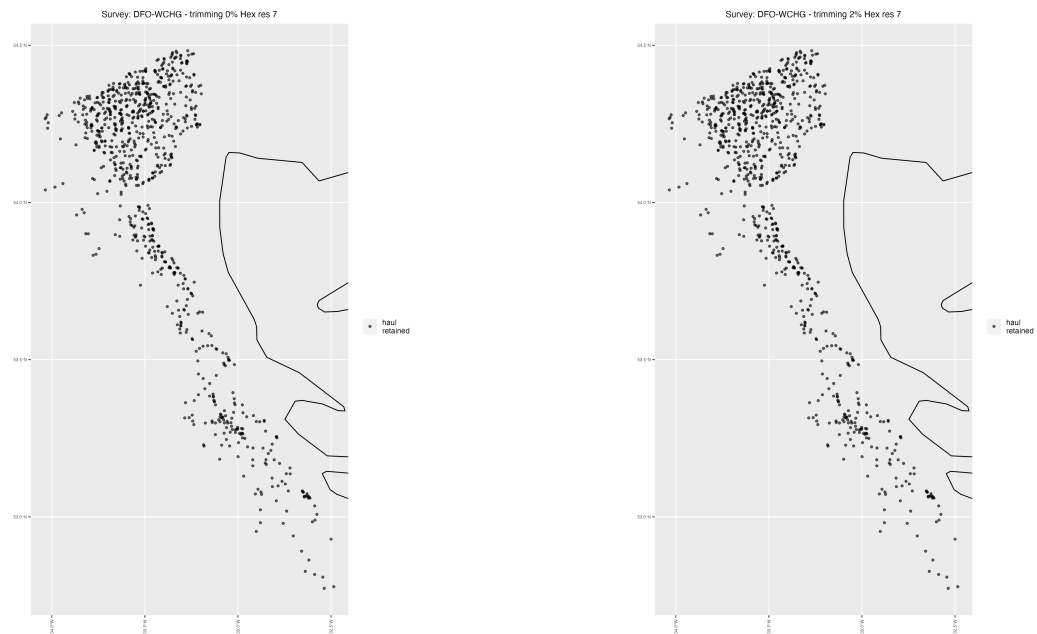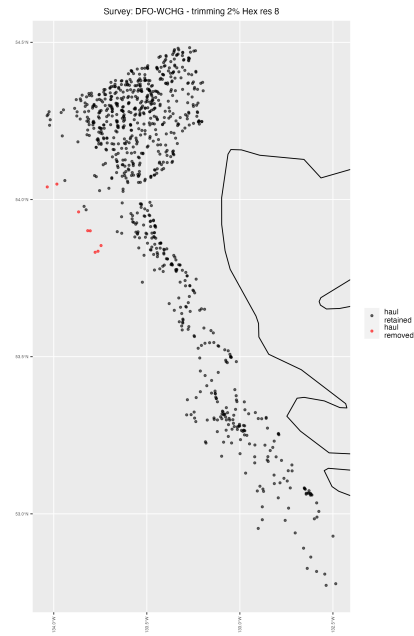
Map of numbers of years removed per grid cell and flagging method/threshold

Survey: DFO-WCHG - trimming 0% Hex res 8 - nb yrs removed

Survey: DFO-WCHG - trimming 2% Hex res 8 - nb yrs removed
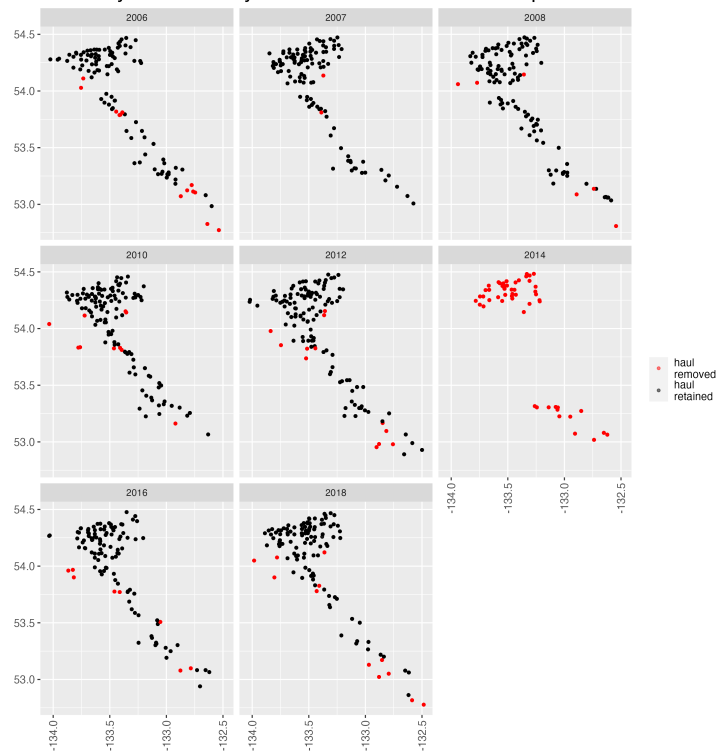
b. Standardization method 2

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

Map of hauls retained and removed

survey= DFO-WCHG year1= 2006 year2= 2018 max.shared.samples= 1479 duration= 13

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	8.0	8.0	1940.0
percentage of hauls removed	0	0	0.9	0.9	13.6