

WCANN: West Coast Annual survey data processing summary

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

January, 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 West Coast US Annual bottom trawl survey provided by Aimee Keller, Fisheries Research Surveys Supervisor, NOAA, NMFS, NWFSC, FRAM and John Buchanan Fisheries Biologist, Groundfish Ecology Program, Northwest Fisheries Science Center. It contains data from 2003 and up to 2018. Before 2003, a similar region is sampled in the West Coast US Triennial Slope and Shelf Survey (WCTRI).

Data cleaning in R

```
#####
##### R code to clean trawl survey West Coast US Annual Survey (WCANN)
#####
##### Public data Ocean Adapt
#####
##### Contacts: Aimee Keller smartt@dnr.sc.gov, Fisheries Research Surveys Supervisor
##### NOAA, NMFS, NWFSC, FRAM
#####
##### John Buchanan john.buchanan@noaa.gov Fisheries Biologist
##### Groundfish Ecology Program, Northwest Fisheries Science Center
#####
##### Coding: Michelle Stuart, Dan Forrest, Zoë Kitchel November 2021
#####

#-----#
##### LOAD LIBRARIES AND FUNCTIONS #####
#-----#
```

```

library(rfishbase) #needs R 4.0 or more recent
library(tidyverse)
library(lubridate)
library(googledrive)
library(taxize) # for getting correct species names
library(magrittr) # for names wrangling
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 US Annual Survey can be best accessed using the public
#Pinsky Lab Ocean Adapt Git Hub Repository.
#Contact malin.pinsky@rutgers.edu for questions or help accessing

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


temp <- tempfile()
download.file(
  "https://github.com/pinskylab/OceanAdapt/raw/master/data_raw/wcann_catch.csv.zip", temp)

wcann_catch <- read_csv(unz(temp, "wcann_catch.csv"), col_types = cols(
  catch_id = col_integer(),
  common_name = col_character(),
  cpue_kg_per_ha_der = col_double(),
  cpue_numbers_per_ha_der = col_double(),
  date_yyyyymmdd = col_integer(),
  depth_m = col_double(),
  latitude_dd = col_double(),
  longitude_dd = col_double(),
  pacfin_spid = col_character(),
  partition = col_character(),
  performance = col_character(),
  program = col_character(),
  project = col_character(),
  sampling_end_hhmmss = col_character(),
  sampling_start_hhmmss = col_character(),
  scientific_name = col_character(),
  station_code = col_double(),
  subsample_count = col_integer(),
  subsample_wt_kg = col_double(),
  total_catch_numbers = col_integer(),
  total_catch_wt_kg = col_double(),
  tow_end_timestamp = col_datetime(format = ""),
  tow_start_timestamp = col_datetime(format = ""),
  trawl_id = col_double(),

```

```

vessel = col_character(),
vessel_id = col_integer(),
year = col_integer(),
year_stn_invalid = col_integer()
))

wcann_haul <- read_csv(
  "https://github.com/pinskylab/OceanAdapt/raw/master/data_raw/wcann_haul.csv",
  col_types = cols(
area_swept_ha_der = col_double(),
date_yyyyymmdd = col_integer(),
depth_hi_prec_m = col_double(),
invertebrate_weight_kg = col_double(),
latitude_hi_prec_dd = col_double(),
longitude_hi_prec_dd = col_double(),
mean_seafloor_dep_position_type = col_character(),
midtow_position_type = col_character(),
nonspecific_organics_weight_kg = col_double(),
performance = col_character(),
program = col_character(),
project = col_character(),
sample_duration_hr_der = col_double(),
sampling_end_hhmmss = col_character(),
sampling_start_hhmmss = col_character(),
station_code = col_double(),
tow_end_timestamp = col_datetime(format = ""),
tow_start_timestamp = col_datetime(format = ""),
trawl_id = col_double(),
vertebrate_weight_kg = col_double(),
vessel = col_character(),
vessel_id = col_integer(),
year = col_integer(),
year_stn_invalid = col_integer()
))

# It is ok to get warning message that missing column names filled in: 'X1' [1].
```

```

##### REFORMAT AND MERGE DATA FILES #####
#####
```

```

wcann <- left_join(wcann_haul, wcann_catch, by = c(
  "trawl_id", "year", "date_yyyyymmdd", "station_code",
  "performance", "program", "project", "sampling_end_hhmmss",
  "sampling_start_hhmmss", "tow_end_timestamp", "tow_start_timestamp",
  "vessel", "vessel_id", "year_stn_invalid"))
wcann <- wcann %>%
  mutate(
    # create haul_id
    haul_id = trawl_id,
    # Add "strata" (define by lat, long and depth bands) where needed # no need to use
    # lon grids on west coast (so narrow)
```

```

stratum = paste(floor(latitude_dd)+0.5, floor(depth_m/100)*100 + 50, sep= "-"),
# adjust for tow area # kg per km2 (hectare/100 = km2)
area_swept = (area_swept_ha_der/100), #km^2
wgt_cpue = total_catch_wt_kg/area_swept,
num_cpue = total_catch_numbers/area_swept,
#note that sample duration is already in hours
wgt_h = total_catch_wt_kg/sample_duration_hr_der,
#note that sample duration is already in hours
num_h = total_catch_numbers/sample_duration_hr_der,
date = ymd(date_yyyymmdd),
month = month(date),
day = day(date),
quarter = case_when(month %in% c(1,2,3) ~ 1,
                     month %in% c(4,5,6) ~ 2,
                     month %in% c(7,8,9) ~ 3,
                     month %in% c(10,11,12) ~ 4),
season = NA_character_,
)

wcann <- wcann %>%
  rename(latitude = latitude_dd,
         longitude = longitude_dd,
         depth = depth_m,
         wgt = total_catch_wt_kg,
         num = total_catch_numbers,
         haul_dur = sample_duration_hr_der,
         spp = scientific_name,
         station = station_code) %>%
  # remove non-fish
  filter(!grepl("Egg", partition),
         !grepl("crushed", spp),
         #remove non satisfactory tows where target speed was not maintained
         performance == "Satisfactory"
  ) %>%
  # adjust spp names
  mutate(
    spp = ifelse(grepl("Lepidopsetta", spp), "Lepidopsetta sp.", spp),
    spp = ifelse(grepl("Bathyraja", spp), 'Bathyraja sp.', spp)
  ) %>%
  # add survey column and fill missing columns
  mutate(survey = "WCANN",
         source = "NOAA",
         timestamp = mdy("04/07/2021"),
         sbt = NA,
         sst = NA,
         country = "United States",
         continent = "n_america",
         sub_area = NA,
         stat_rec = NA,
         verbatim_name = spp,
         gear = 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,
num, num_h, num_cpue, wgt, wgt_h, wgt_cpue, verbatim_name)
#many rows with missing num_h, num_cpue, wgt_h, and wgt_cpue values
#due to missing haul_dur

#sum duplicates
wcann <- wcann %>%
  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()

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

#none!

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

unique_name_match
#empty

#now, I will sum over these duplicated verbatim names
#Porifera
#Bathyraja sp.
#Merluccius productus
#Nudibranchia
#Strongylocentrotus
#Pagurus
#Pennatulacea
#Ceramaster
#Neptunea
#Rossellinae
#Gorgonacea
#Colus
#Munidopsis
#Sebastes sp. (aleutianus / melanostictus)

```

```

#Buccinum
#Ophiacantha
#Glyptocephalus zachirus
#Oncorhynchus tshawytscha
#Antipatharia
#Urticina
#Stomphia
#Hormathiidae
#Halipteris
#Molpadiia intermedia
#Sebastes sp. (miniatus / crocotulus)
#Hexactinosida
#Suberites

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


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

### Automatic cleaning
# Set Survey code
wcann_survey_code <- "WCANN"

wcann <- wcann %>%
  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(wcann$taxa2), input_survey = wcann_survey_code,
                           fishbase=T)
# takes 4.5 mins

#This cuts out the following species, one should be added

#1 Nearchaster aciculosus
#2 Cheiraster dawsoni
#3 Crangon communis
#4 Cancer gracilis
#5 Cancer anthonyi
#6 Cancer branneri
#7 Cyclopterinae (fish, but only to genus)

cyclop <- c("Cyclopterinae", NA, NA, "Cyclopterinae", "Animalia", "Chordata", "Actinopteri",
          "Scorpaeniformes", "Cyclopteridae", "NA", "Family", "WCANN")

clean_auto.missing <- rbind(clean_auto, cyclop)

```

```

#-----#
##### INTEGRATE CLEAN TAXA in WCANN survey data #####
#-----#


clean_taxa <- clean_auto.missing %>%
  select(-survey)

clean_wcann <- left_join(wcann, clean_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_wcann <- clean_wcann %>%
  group_by(haul_id, accepted_name) %>%
  mutate(count = n())

#none!

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

unique_name_match
#not empty

#a few duplicates are maintained with different verbatim name
#and the same accepted names. Data users should decide if they want to sum over.
#currently, these are independent observations

#Sebastes or not
#verbatim name          accepted name
#Sebastes                Sebastes
#Sebastes sp. (miniatus / crocotulus)    Sebastes
#Sebastes sp. (aleutianus / melanostictus) Sebastes
# -----#

```

```

##### SAVE DATABASE IN GOOGLE DRIVE #####
# -----#
# Just run this routine should be good for all
write_clean_data(data = clean_wcann, survey = "WCANN", overwrite = T)

# -----#
##### FAGS #####
# -----#
#install required packages that are not already installed
required_packages <- c("data.table",
                      "devtools",
                      "dgridR",
                      "dplyr",
                      "fields",
                      "forcats",
                      "ggplot2",
                      "here",
                      "magrittr",
                      "maps",
                      "maptools",
                      "raster",
                      "rcompendium",
                      "readr",
                      "remotes",
                      "rrtools",
                      "sf",
                      "sp",
                      "tidyverse",
                      "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_wcann$survey))

#run flag_spp function in a loop
for (r in regions) {
  flag_spp(clean_wcann, 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_wcann, 7)

```

```

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

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

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

surveys <- sort(unique(clean_wcann$survey))
survey_units <- sort(unique(clean_wcann$survey_unit))
survey_std <- clean_wcann %>%
  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], "_flagsp.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 = "WCANN_std",
                 overwrite = T, rdata=TRUE)

```

1. Overview of the survey data table

survey	source	timestamp	haul_id	country	sub_area
WCANN	NOAA	2021-04-07	2.00303e+11	United States	NA
WCANN	NOAA	2021-04-07	2.00303e+11	United States	NA
WCANN	NOAA	2021-04-07	2.00303e+11	United States	NA
WCANN	NOAA	2021-04-07	2.00303e+11	United States	NA
WCANN	NOAA	2021-04-07	2.00303e+11	United States	NA

continent	stat_rec	station	stratum	year	month	day	quarter	season
n_america	NA	7277	46.5-550	2003	8	31	3	NA
n_america	NA	7277	46.5-550	2003	8	31	3	NA
n_america	NA	7277	46.5-550	2003	8	31	3	NA
n_america	NA	7277	46.5-550	2003	8	31	3	NA
n_america	NA	7277	46.5-550	2003	8	31	3	NA

latitude	longitude	haul_dur	area_swept	gear	depth	sbt	sst
46.30417	-124.725	0.294722	0.0165671	NA	527.5	NA	NA
46.30417	-124.725	0.294722	0.0165671	NA	527.5	NA	NA
46.30417	-124.725	0.294722	0.0165671	NA	527.5	NA	NA
46.30417	-124.725	0.294722	0.0165671	NA	527.5	NA	NA
46.30417	-124.725	0.294722	0.0165671	NA	527.5	NA	NA

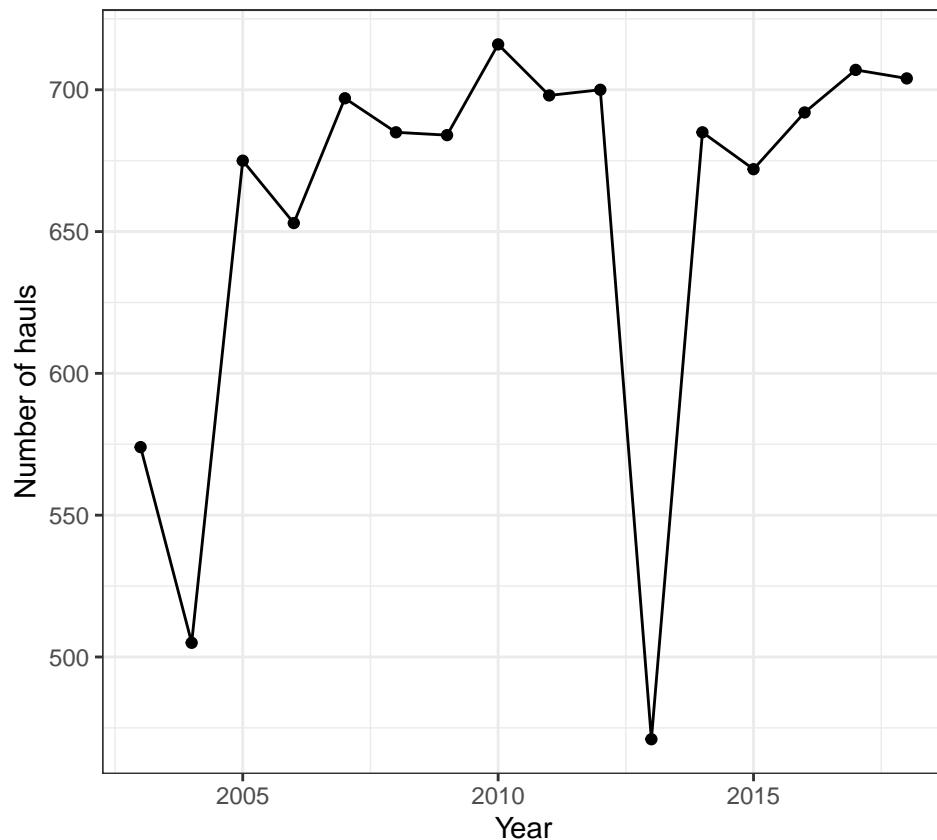
num	num_cpue	num_cputa	wgt	wgt_cpue	wgt_cputa	verbatim_name
8	27.144224	482.88335	10.800	36.6447025	651.8925282	Anoplopoma fimbria
20	67.860560	1207.20839	2.300	7.8039644	138.8289643	Antimora microlepis
15	50.895420	905.40629	4.050	13.7417634	244.4596981	Aristurus brunneus
2	6.786056	120.72084	0.005	0.0169651	0.3018021	Bathyagonus nigripinnis
1	3.393028	60.36042	0.900	3.0537252	54.3243774	Bathyraja sp.

verbatim_aphia_id	accepted_name	aphia_id	SpecCode	kingdom
NA	Anoplopoma fimbria	159463	512	Animalia
NA	Antimora microlepis	272460	2006	Animalia
NA	Aristurus brunneus	158512	763	Animalia
NA	Bathyagonus nigripinnis	254505	4161	Animalia
NA	Bathyraja	105761	NA	Animalia

phylum	class	order	family	genus	rank	survey_unit
Chordata	Teleostei	Perciformes	Anoplopomatidae	Anoplopoma	Species	WCANN
Chordata	Teleostei	Gadiformes	Moridae	Antimora	Species	WCANN
Chordata	Elasmobranchii	Carcharhiniformes	Pentanchidae	Apristurus	Species	WCANN
Chordata	Teleostei	Perciformes	Agonidae	Bathyagonus	Species	WCANN
Chordata	Elasmobranchii	Rajiformes	Arhynchobatidae	Bathyraja	Genus	WCANN

2. Summary of sampling intensity

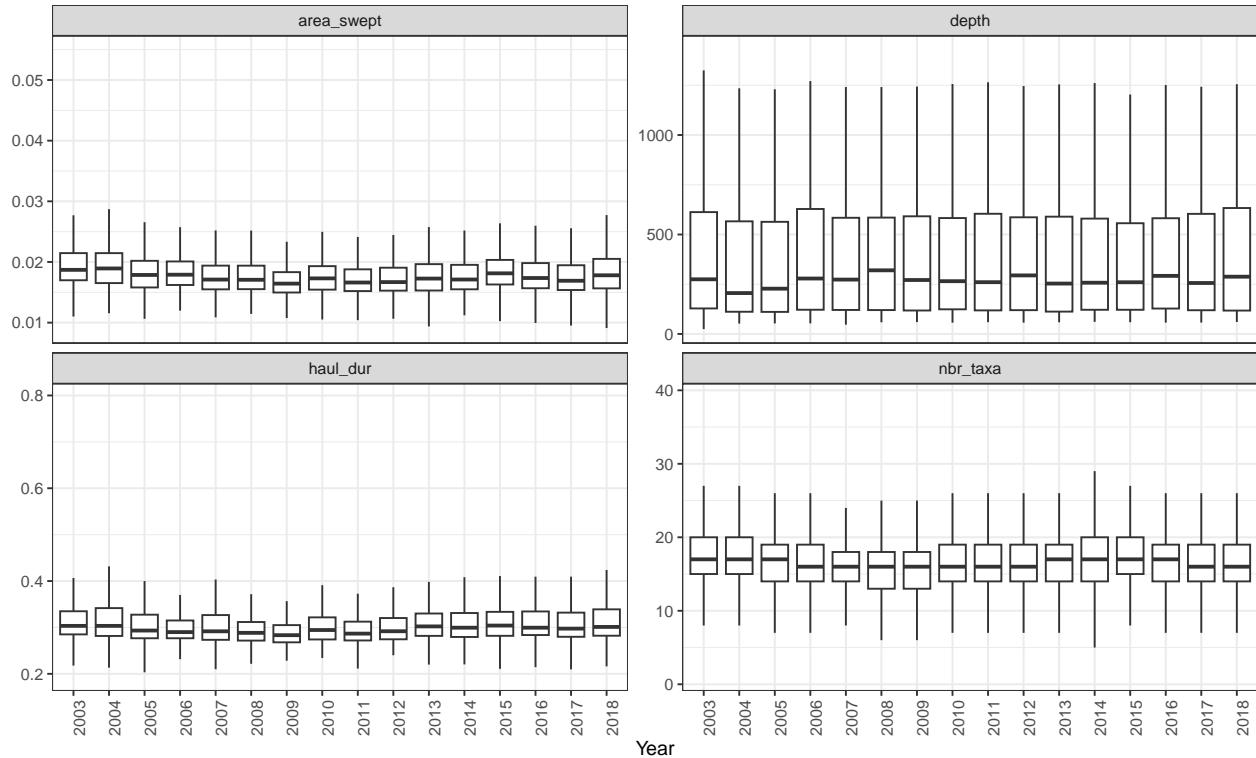
Number of hauls per year performed during the survey after data processing.



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:

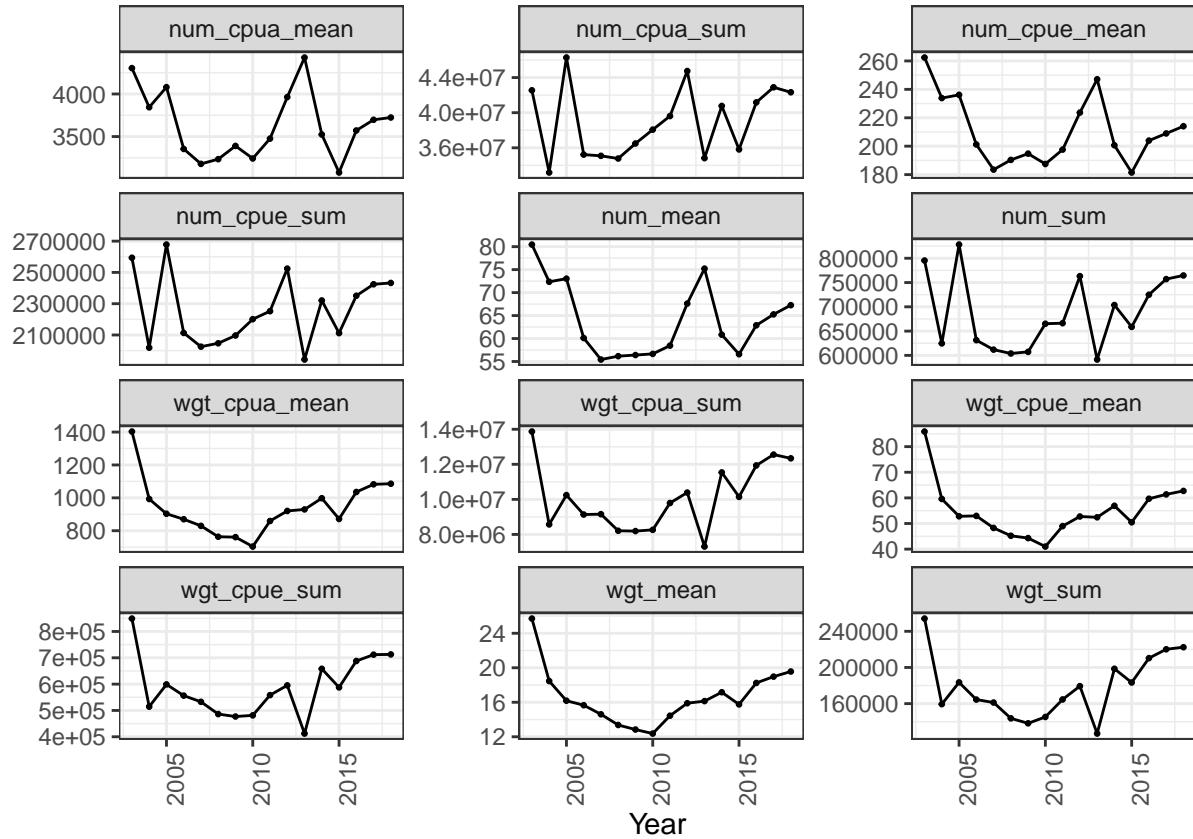
- *area_swept*, swept area by the bottom trawl gear km^2
- *depth*, sampling depth in m
- *haul_dur*, haul sampling duration *hour*
- *number of marine fish taxa*, taxa were cleaned following the last version of taxonomy from the World Register of Marine Species (<https://www.marinespecies.org/>, October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



4. Summary of biological variables

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

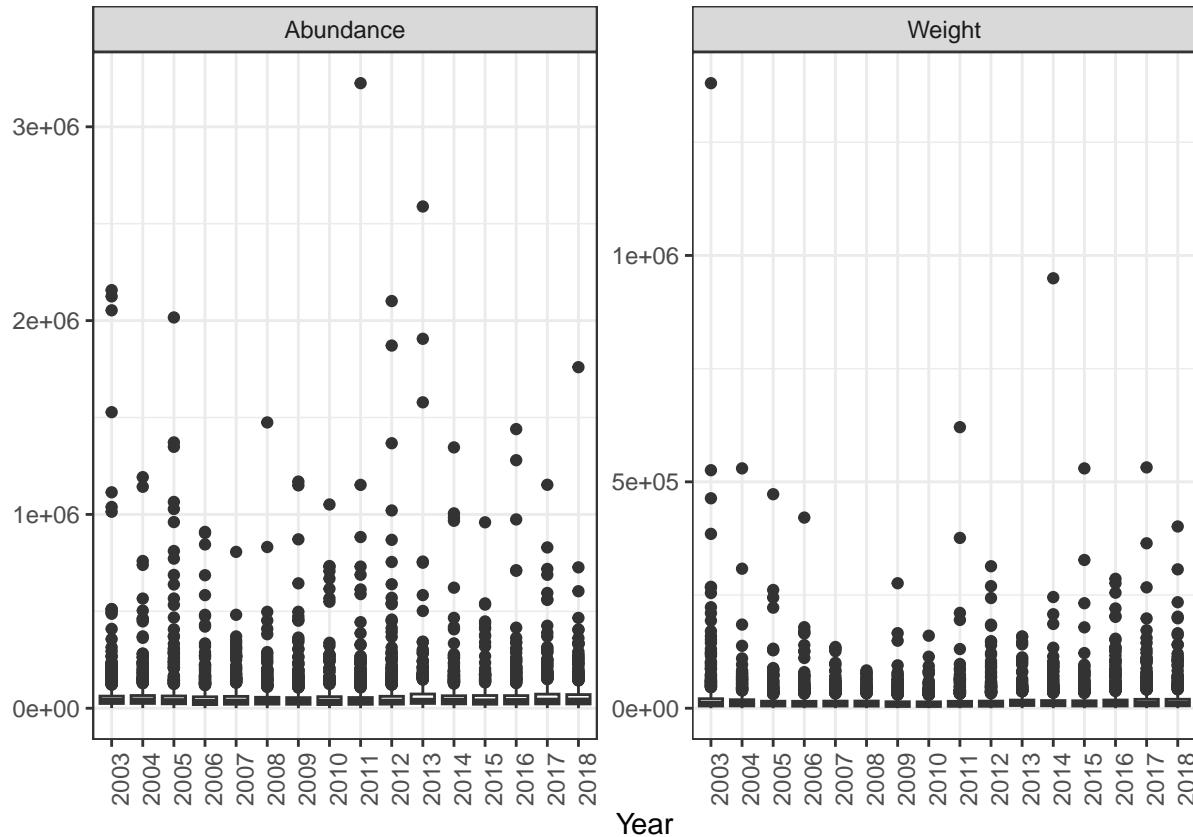
- num_cpua , number of individuals (abundance) in $\frac{individuals}{km^2}$
- num_cpue , number of individuals (abundance) in $\frac{individuals}{h}$
- num , number of individuals (abundance)
- wgt_cpua , weight in $\frac{kg}{km^2}$
- wgt_cpue , weight in $\frac{kg}{h}$
- wgt , weight in kg



5. Extreme values

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

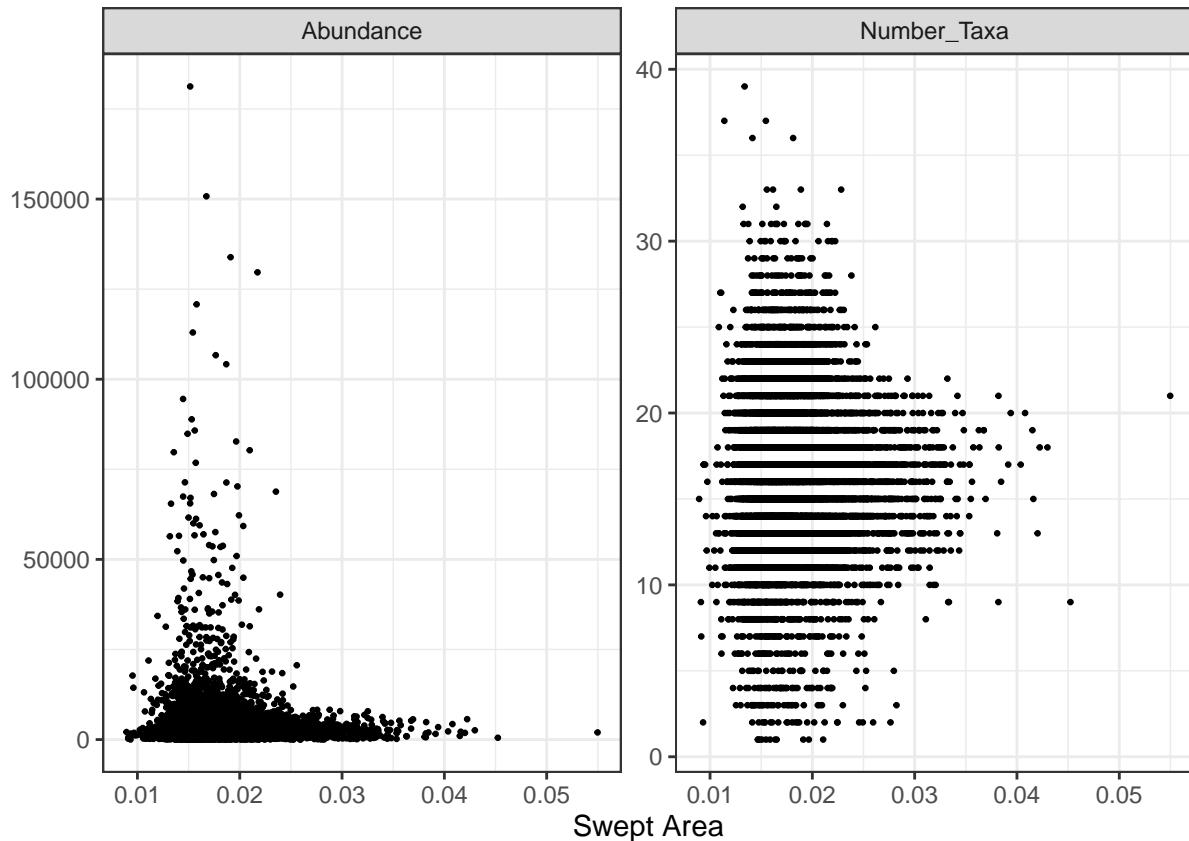
- num_cpue , number of individuals (abundance) in $\frac{individuals}{km^2}$
- wgt_cpue , weight in $\frac{kg}{km^2}$



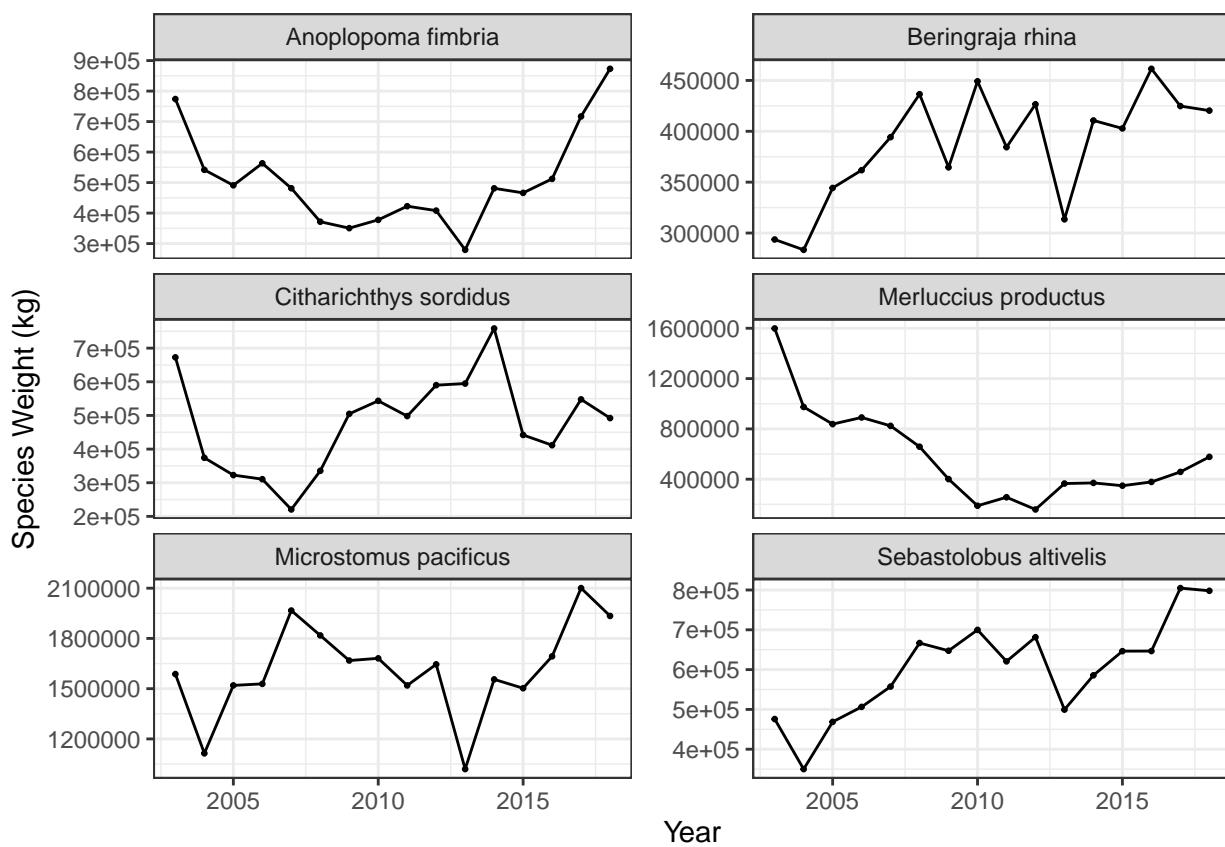
6. Summary of variables against swept area

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

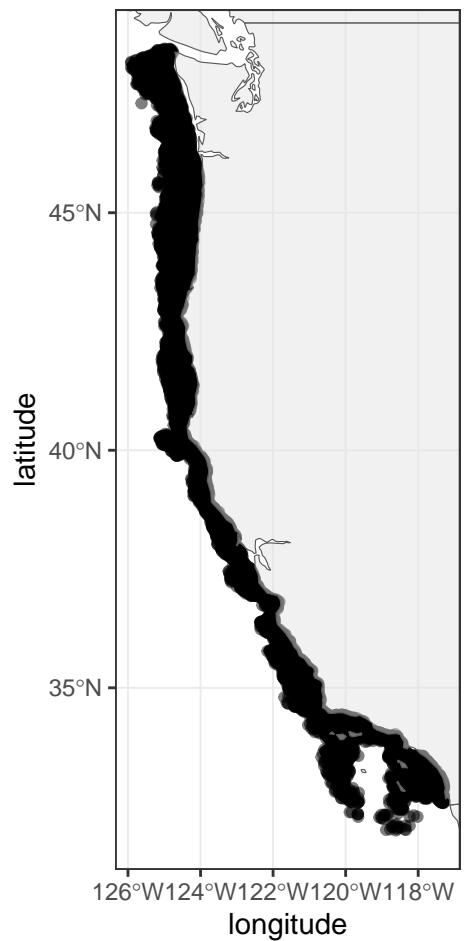


7. Abundance or Weight trends of the six most abundant species



8. Distribution mapping

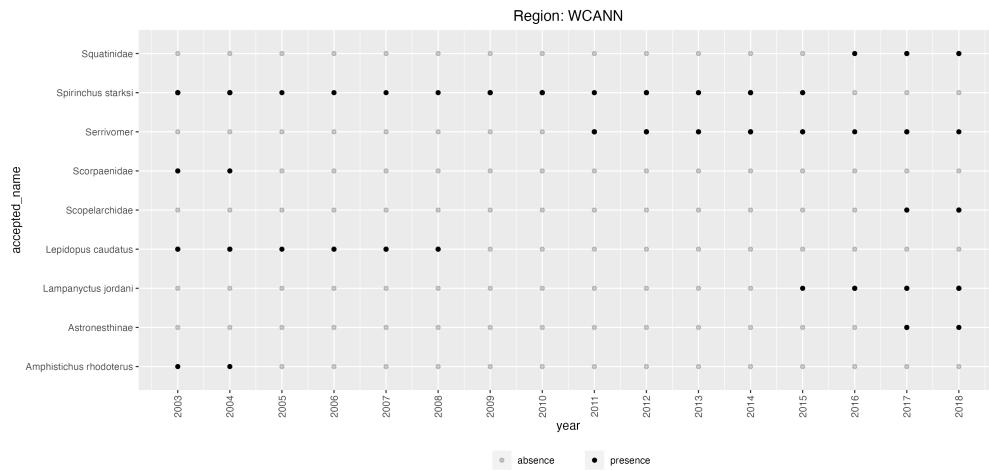
Map of the sampling distribution in space. Note that we only show one year per coordinate.



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

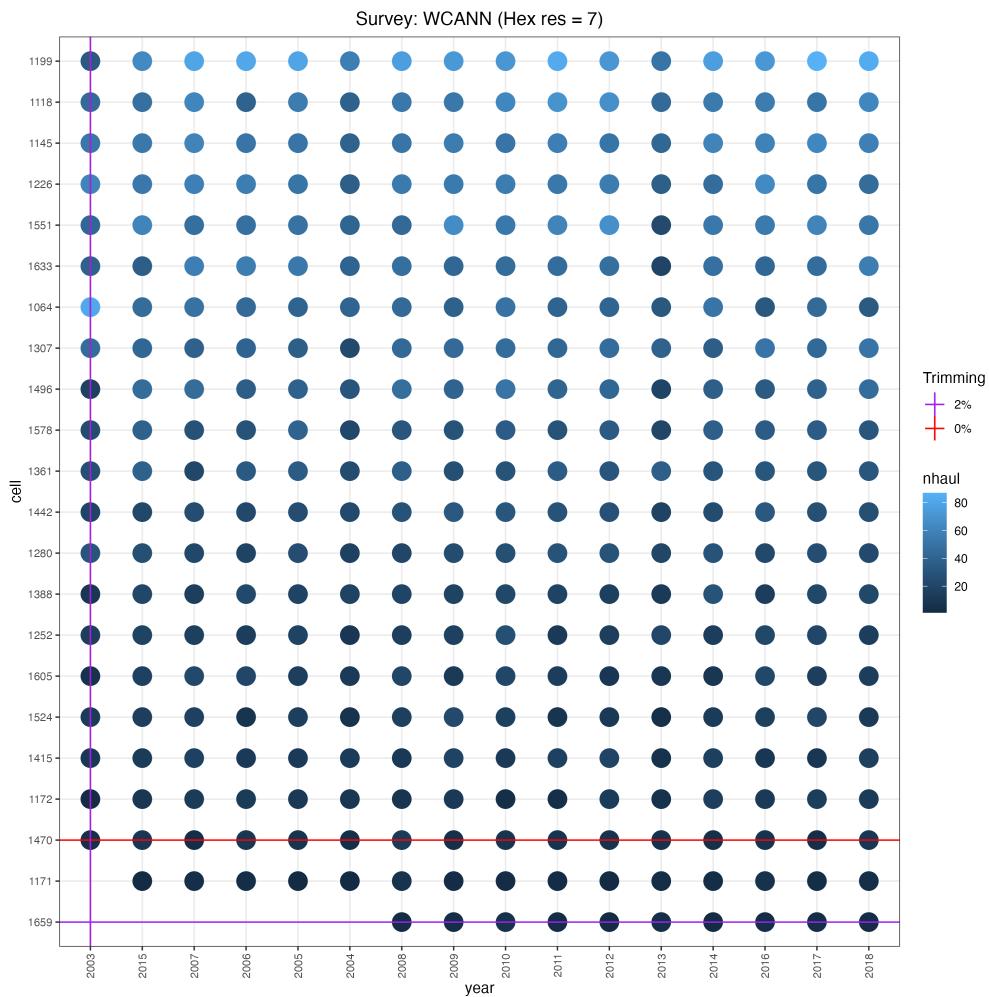
Total number of species	456
Percentage of species flagged	2

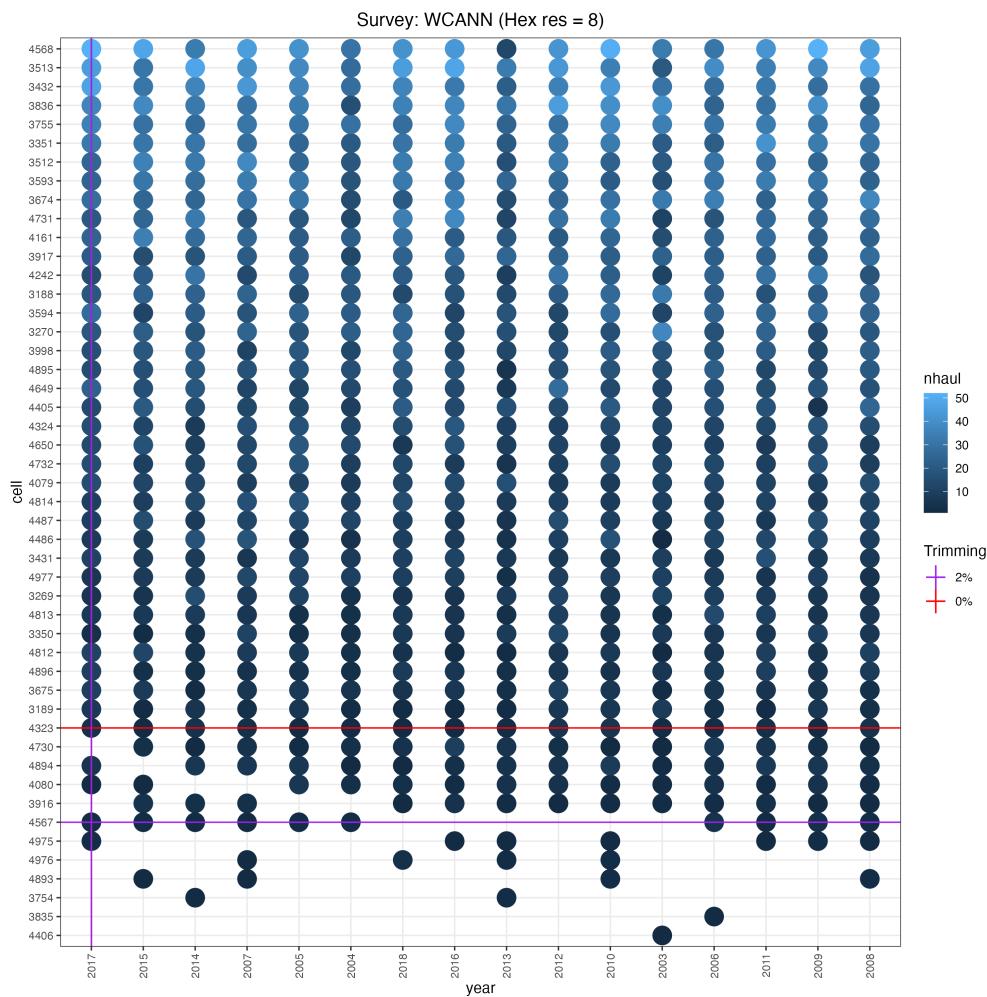
10. Spatio-temporal standardization

a. Standardization method 1

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

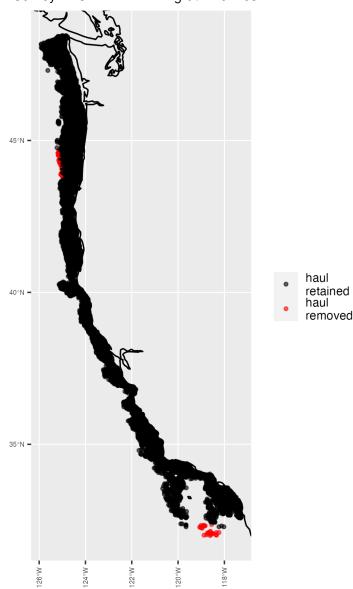
Plot of number of cells x years with overlaid flagging options



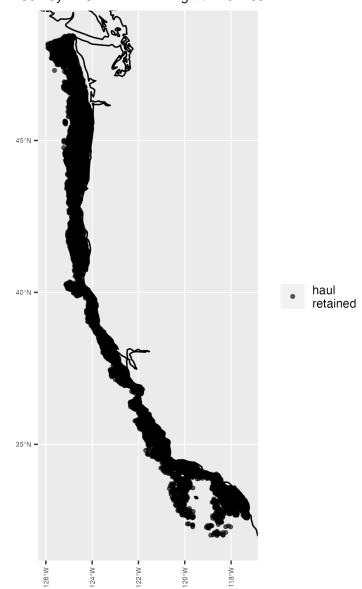


Map of hauls retained and removed per flagging method and threshold

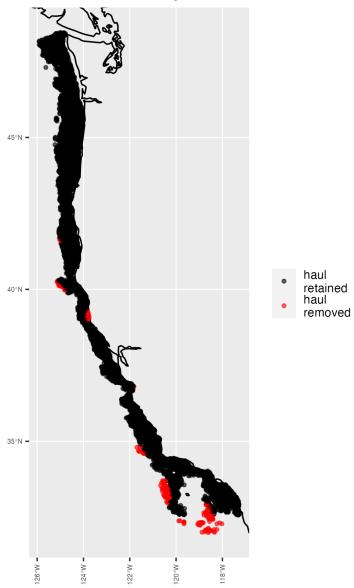
Survey: WCANN - trimming 0% Hex res 7



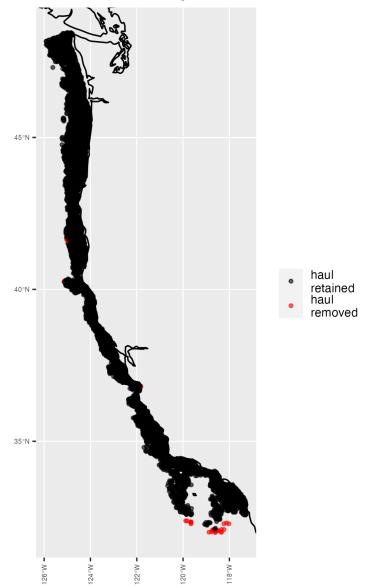
Survey: WCANN - trimming 2% Hex res 7



Survey: WCANN - trimming 0% Hex res 8

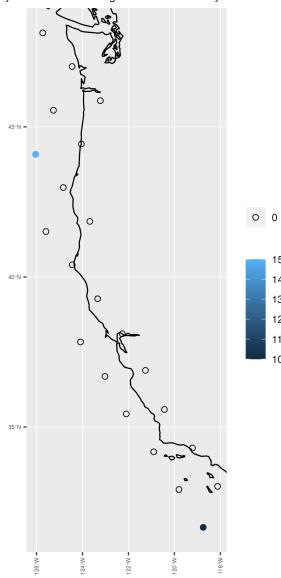


Survey: WCANN - trimming 2% Hex res 8

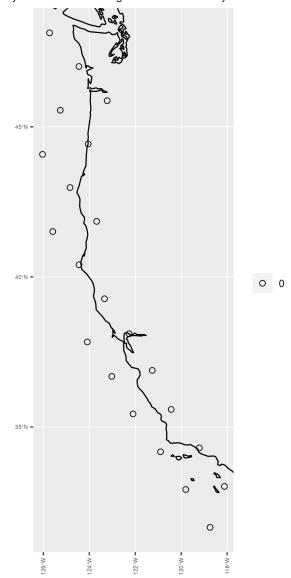


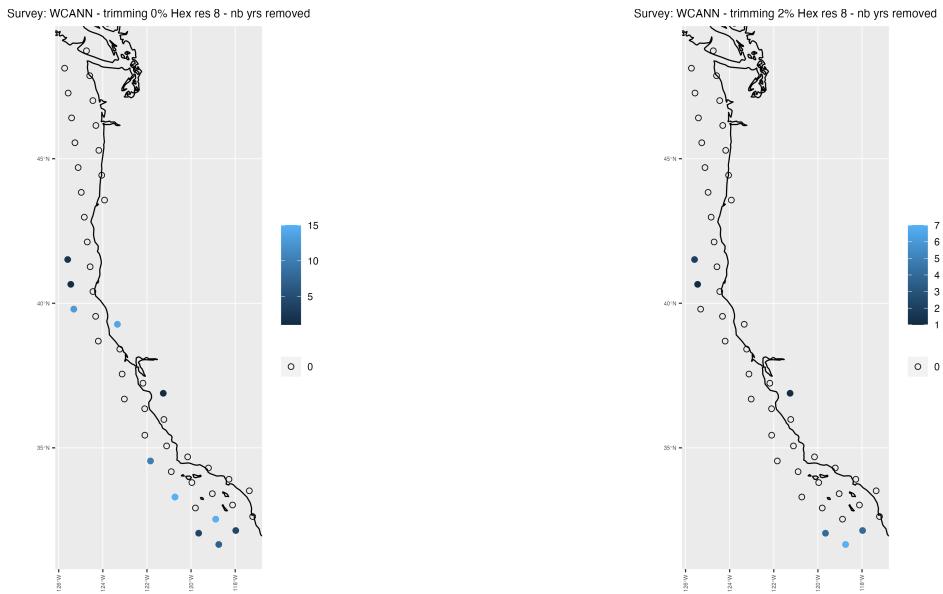
Map of numbers of years removed per grid cell and flagging method/threshold

Survey: WCANN - trimming 0% Hex res 7 - nb yrs removed



Survey: WCANN - trimming 2% Hex res 7 - nb yrs removed

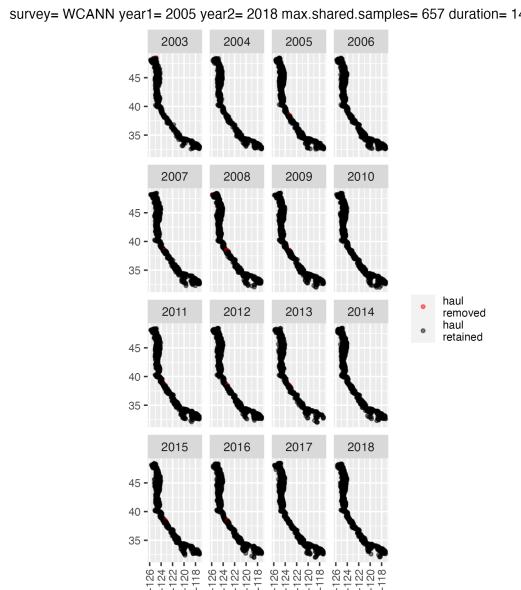




b. Standardization method 2

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

Map of hauls retained and removed



c. Standardization summary

Statistics of hauls removed for each standardization method

summary	grid cell 7, 0% threshold	grid cell 7, 2% threshold	grid cell 8, 0% threshold	grid cell 8, 2% threshold	method 2 (biotime)
number of hauls removed	70.0	0	236.0	28.0	526.0
percentage of hauls removed	0.7	0	2.2	0.3	0.3