AI: Aleutian Islands US survey data processing summary

fishglob, Aurore A. Maureaud, Juliano 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 summary of the Aleutian Island bottom trawl survey provided by Stan Kotwicki and Jim Thorson It contains data from 1983-1997 (triennial) and 2000-2020 (biennial; 2008 cancelled).

Data cleaning in R

```
################################################################################
#### R code to clean trawl survey Aleutian Islands
#### Public data Ocean Adapt
#### Contacts: Stan Kotwicki    stan.kotwicki@noaa.gov  Program Manager,
####      1 Groundfish Assessment Program, NOAA AFSC
####          Jim Thorson   james.thorson@noaa.gov  Program Leader,
####     Habitat and Ecological Processes Research, NOAA AFSC
#### Coding: Michelle Stuart, Dan Forrest, Zoë Kitchel November 2021
################################################################################
#Alaska Fisheries Science Center - NOAA
#https://www.afsc.noaa.gov/RACE/groundfish/survey_data/
#metadata_template.php?fname=RACEweb.xml
#This NOAA center provides data for the Aleutian Islands, Eastern Bering Sea,
#and Gulf of Alaska.  (source)
#Files provided by the Alaska Fisheries Science Center

#--------------------------------------------------------------------------------------#
```

```r
#### 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(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 Aleutian Islands can be accessed using the public Pinsky
#Lab OceanAdapt Git Hub Repository.
#Files obtained from data providers Mar 1, 2021  (timestamp)
#Contact malin.pinsky@rutgers.edu for questions or help accessing

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

## Special fix
#there is a comment that contains a comma in the 2014-2018 file that
#causes the delimiters to read incorrectly.  Fix that here::here:
aiURL <- "https://github.com/pinskylab/OceanAdapt/raw/master/data_raw/ai2014_2018.csv"

temp <- readLines(aiURL)
# replace the string that causes the problem
temp_fixed  <- gsub(pattern = "Stone et al., 2011", replace = "Stone et al. 2011", x = temp)
writeLines(temp_fixed, "cleaning_codes/ai2014_2018.txt") #save as text file
# read the result in as a csv
temp_csv <- read_csv(file = "cleaning_codes/ai2014_2018.txt", col_names = T)

#delete this file we temporarily made
file.remove("cleaning_codes/ai2014_2018.txt")
## End special fix

ai83_00 <- "https://github.com/pinskylab/OceanAdapt/raw/master/data_raw/ai1983_2000.csv"
ai02_12 <- "https://github.com/pinskylab/OceanAdapt/raw/master/data_raw/ai2002_2012.csv"

#make list of csv files from OceanAdapt GitHub
files <- as.list(c(ai83_00, ai02_12))

ai_data <- files %>%
  # read in all of the csv's in the files list
  map_dfr(read_csv) %>% #applies function to each element of list
  # add in the data fixed above
  rbind(temp_csv) %>%
  # remove any data rows that have headers as data rows
  filter(LATITUDE != "LATITUDE", !is.na(LATITUDE)) %>%
  mutate(stratum = as.integer(STRATUM)) %>%
  # remove any extra white space from around spp and common names
  mutate(COMMON = str_trim(COMMON),
         SCIENTIFIC = str_trim(SCIENTIFIC))

# The warning of 13 parsing failures is pointing to a row in the middle
#of the data set that contains headers instead of the numbers expected,
#this row is removed by the filter above.

aistrat <- "https://github.com/pinskylab/OceanAdapt/raw/master/data_raw/ai_strata.csv"

ai_strata <- read_csv(aistrat, col_types = cols(NPFMCArea = col_character(),
                                                SubareaDescription = col_character(),
                                                StratumCode = col_integer(),
                                                DepthIntervalm = col_character(),
                                                Areakm2 = col_integer()
))  %>%
  mutate(stratum = StratumCode)


ai <- left_join(ai_data, ai_strata, by = "stratum")

# are there any strata in the data that are not in the strata file?
stopifnot(nrow(filter(ai, is.na(Areakm2))) == 0)

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

ai <- ai %>%
  mutate(
    # Create a unique haul_id
    haul_id = paste(formatC(VESSEL, width=3, flag=0), CRUISE,
                    formatC(HAUL, width=3, flag=0), LONGITUDE, LATITUDE, sep=''),
    numcpue = ifelse(NUMCPUE < -9000,NA, NUMCPUE),
    sbt = ifelse(BOT_TEMP < -9000, NA,BOT_TEMP),
    sst = ifelse(SURF_TEMP < -9000, NA, SURF_TEMP)) %>% #get rid of any use of
                   #-9999 as a no data marker
  rename(year = YEAR,
         latitude = LATITUDE,
         longitude = LONGITUDE,
         depth = BOT_DEPTH,
         spp = SCIENTIFIC,
         station = STATION,
         num_cpue.raw = numcpue, #units = number/hectare
         wgt_cpue.raw = WTCPUE #units = kg/hectare (1 hectare = 0.01 km^2)
  ) %>%
  #convert date to month and day columns
  mutate(
    #convert date to month and day columns
    datetime = mdy_hm(DATETIME),
    month = month(datetime),
    day = day(datetime),
    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',
    #convert cpue which is currently per hectare to per km^2 by multiplying by 100
    wgt_cpue = 100*wgt_cpue.raw,
    num_cpue = 100*num_cpue.raw
  ) %>%
  # remove non-fish
  filter(
    spp != '' &
      !grepl("egg", spp)) %>%
  # adjust spp names
  mutate(
    #Manual taxa cleaning (happens later in other get.x.R scripts)
    spp = ifelse(grepl("Lepidopsetta", spp), "Lepidopsetta sp.", spp),
    spp = ifelse(grepl("Myoxocephalus", spp ) & !grepl("scorpius", spp),
                 "Myoxocephalus sp.", spp),
    spp = ifelse(grepl("Bathyraja", spp) & !grepl("panthera", spp), 'Bathyraja sp.', spp)
  ) %>%
  mutate(
    #convert cpue which is currently per hectare to per km^2 by multiplying by 100
  wgt_cpue = 100*wgt_cpue.raw,
num_cpue = 100*num_cpue.raw
) %>%
  # remove non-fish
  filter(
    spp != '' &
      !grepl("egg", spp)) %>%
  # adjust spp names
  mutate(
    #Manual taxa cleaning (happens later in other get.x.R scripts)
    spp = ifelse(grepl("Lepidopsetta", spp), "Lepidopsetta sp.", spp),
    spp = ifelse(grepl("Myoxocephalus", spp ) & !grepl("scorpius", spp),
                 "Myoxocephalus sp.", spp),
    spp = ifelse(grepl("Bathyraja", spp) & !grepl("panthera", spp), 'Bathyraja sp.', spp)
  ) %>%
  #finalize columns
  mutate(survey = "AI",
       source = "NOAA",
       timestamp = mdy("3/1/2021"),
         country = "United States",
         sub_area = NA,
         continent = "n_america",
         stat_rec = NA,
         verbatim_name = spp,
         haul_dur = NA,
         gear = "Poly Nor'Eastern trawl",
         num = NA,
         num_h = NA,
         wgt = NA,
         wgt_h = NA,
         area_swept = NA
  ) %>%
  select(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,
         num, num_h, num_cpue, wgt, wgt_h, wgt_cpue, verbatim_name)

#sum duplicates
ai <- ai %>%
  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_ai <- ai %>%
  group_by(haul_id, verbatim_name) %>%
  mutate(count = n())

#none!

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

unique_name_match
#check if empty

#there are some duplicates, so we will add a sum above
#1 Bathyraja sp.
#2 Malacocottus zonurus
#3 Lepidopsetta sp.
#4 Myoxocephalus sp.
#5 Aphrocallistes vastus

#--------------------------------------------------------------------------------------#
#### 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
ai_survey_code <- "AI"

ai_taxa <- ai %>%
  select(verbatim_name) %>%
  mutate(
    taxa = str_squish(verbatim_name),
    taxa = str_remove_all(taxa," spp.| sp.| spp| sp|NO "),
    taxa = str_to_sentence(str_to_lower(taxa))
  ) %>%
  pull(taxa) %>%
  unique()

# Get clean taxa
clean_auto <- clean_taxa(ai_taxa, input_survey = ai_survey_code,
                         save = F, output=NA, fishbase=TRUE) # takes 9 mins

#Check those with no match from clean_taxa()
#Beringius beringii                                   no match
#Cheiraster dawsoni                                   no match
#Crangon communis                                     no match
#Pandalopsis                                          no match
#Scalpellum cornutum                                  no match
#Nearchaster variabilis                               no match
#Bathybuccinum clarki                                 no match
#Cancer branneri                                      no match
#Hippodiplosia                                        no match

#######clear, all invertebrates

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

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

clean_ai <- left_join(ai, clean_taxa, by=c("verbatim_name"="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,
         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_ai <- clean_ai %>%
  group_by(haul_id, accepted_name) %>%
  mutate(count = n())

#none!

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

unique_name_match
#check if empty


# -------------------------------------------------------------------------------------#
#### SAVE DATABASE ####
# -------------------------------------------------------------------------------------#

# Just run this routine should be good for all if you're synced to Google Drive
write_clean_data(data = clean_ai, survey = "AI", 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_ai$survey))

#run flag_spp function in a loop
for (r in regions) {
  flag_spp(clean_ai, 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_ai, 7)

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

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


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

```r
}


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

1. Overview of the survey data table

survey	source	timestamp	haul_id	country	sub_area
AI	NOAA	2021-03-01	001198304005-166.3183354.11833	United States	NA
AI	NOAA	2021-03-01	001198304005-166.3183354.11833	United States	NA
AI	NOAA	2021-03-01	001198304005-166.3183354.11833	United States	NA
AI	NOAA	2021-03-01	001198304005-166.3183354.11833	United States	NA
AI	NOAA	2021-03-01	001198304005-166.3183354.11833	United States	NA

continent	stat_rec	station	stratum	year	month	day	quarter	season
n_america	NA	316-72	721	1983	8	18	3	NA
n_america	NA	316-72	721	1983	8	18	3	NA
n_america	NA	316-72	721	1983	8	18	3	NA
n_america	NA	316-72	721	1983	8	18	3	NA
n_america	NA	316-72	721	1983	8	18	3	NA

latitude	longitude	haul_dur	area_swept	gear	depth	sbt	sst
54.11833	-166.3183	NA	NA	Poly Nor'Eastern trawl	99	4.7	8.5
54.11833	-166.3183	NA	NA	Poly Nor'Eastern trawl	99	4.7	8.5
54.11833	-166.3183	NA	NA	Poly Nor'Eastern trawl	99	4.7	8.5
54.11833	-166.3183	NA	NA	Poly Nor'Eastern trawl	99	4.7	8.5
54.11833	-166.3183	NA	NA	Poly Nor'Eastern trawl	99	4.7	8.5

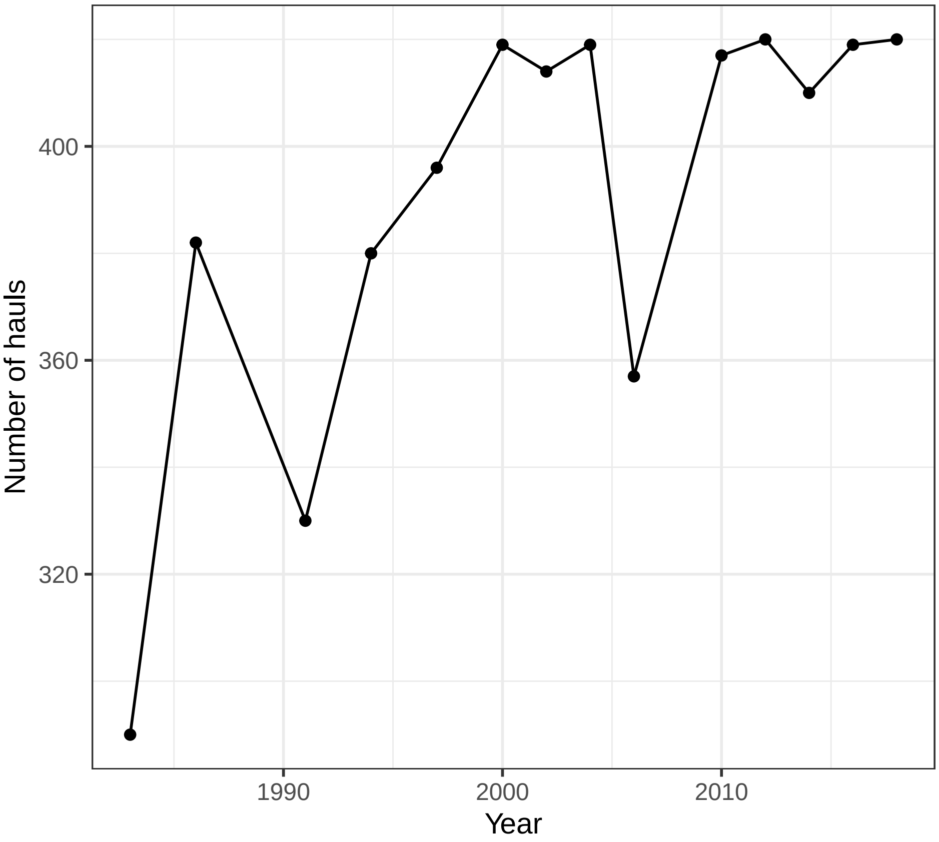
num	num_cpue	num_cpua	wgt	wgt_cpue	wgt_cpua	verbatim_name
0	0	952.96	0	0	1037.41	Anoplopoma fimbria
0	0	762.37	0	0	216.13	Atheresthes stomias
0	0	89577.97	0	0	19883.69	Clupea pallasii
0	0	163336.86	0	0	136332.96	Gadus chalcogrammus
0	0	8195.43	0	0	14264.39	Gadus macrocephalus

verbatim_aphia_id	accepted_name	aphia_id	SpecCode	kingdom
NA	Anoplopoma fimbria	159463	512	Animalia
NA	Atheresthes stomias	279792	517	Animalia
NA	Clupea pallasii	151159	NA	Animalia
NA	Gadus chalcogrammus	300735	318	Animalia
NA	Gadus macrocephalus	254538	308	Animalia

phylum	class	order	family	genus	rank	survey_unit
Chordata	Teleostei	Perciformes	Anoplopomatidae	Anoplopoma	Species	AI
Chordata	Teleostei	Pleuronectiformes	Pleuronectidae	Atheresthes	Species	AI
Chordata	Teleostei	Clupeiformes	Clupeidae	Clupea	Species	AI
Chordata	Teleostei	Gadiformes	Gadidae	Gadus	Species	AI
Chordata	Teleostei	Gadiformes	Gadidae	Gadus	Species	AI

2. Summary of sampling intensity

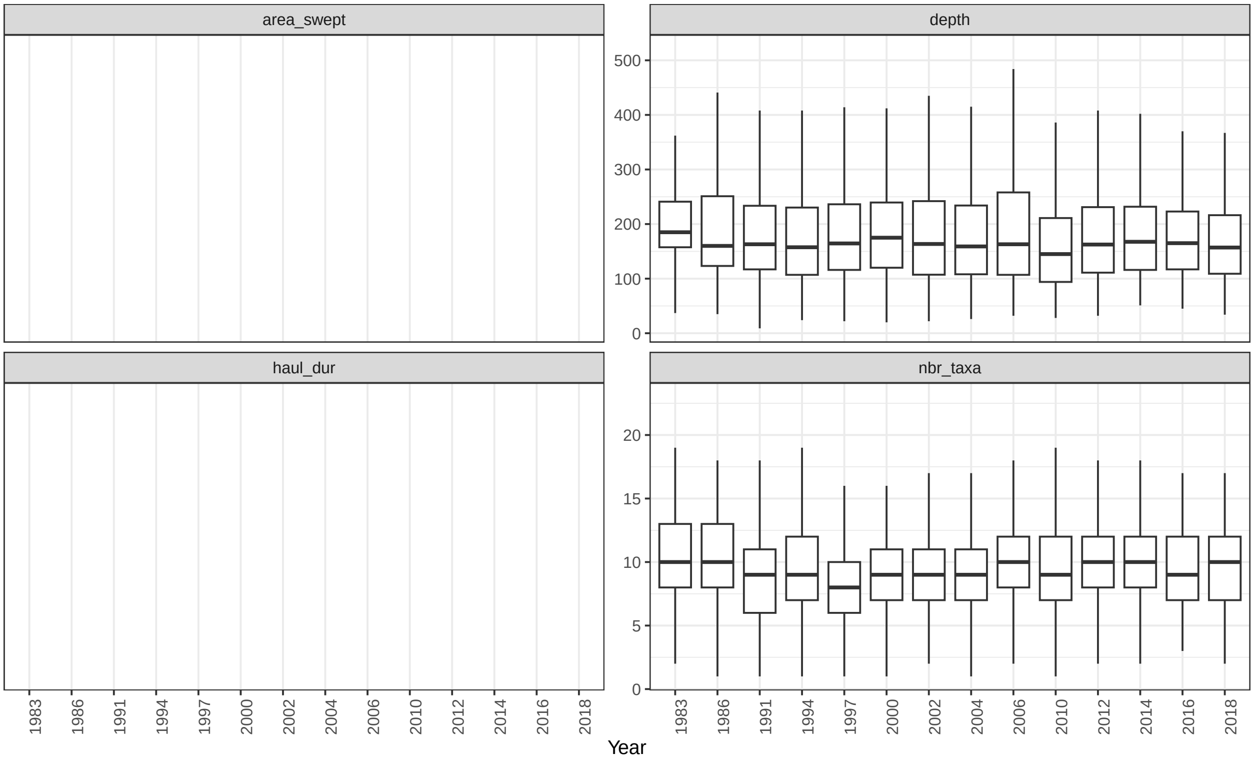
Number of hauls per year performed during the survey after data processing.



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:

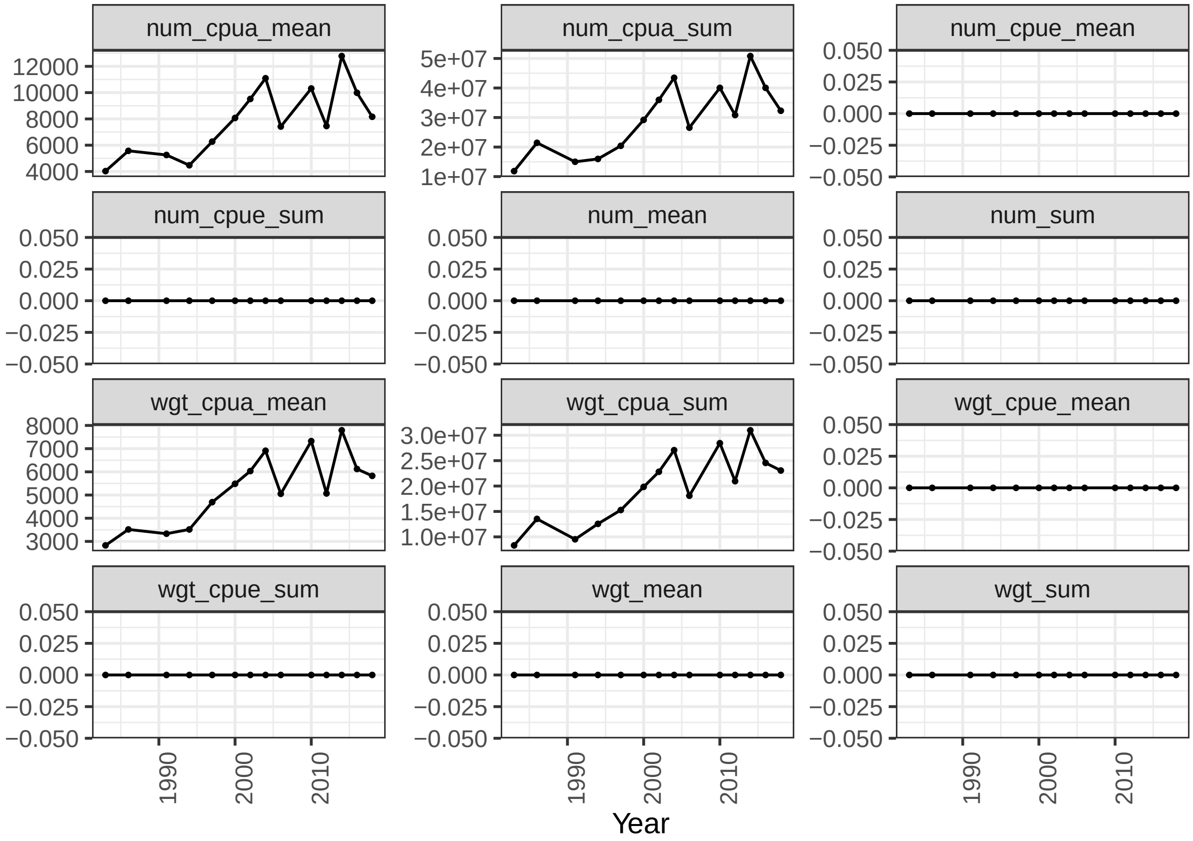
- *area_swept*, swept area by the bottom trawl gear km^2
- *depth*, sampling depth in m
- *haul_dur*, haul sampling duration *hour*
- *number of marine fish taxa*, taxa were cleaned following the last version of taxonomy from the World Register of Marine Species (https://www.marinespecies.org/, October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



4. Summary of biological variables

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

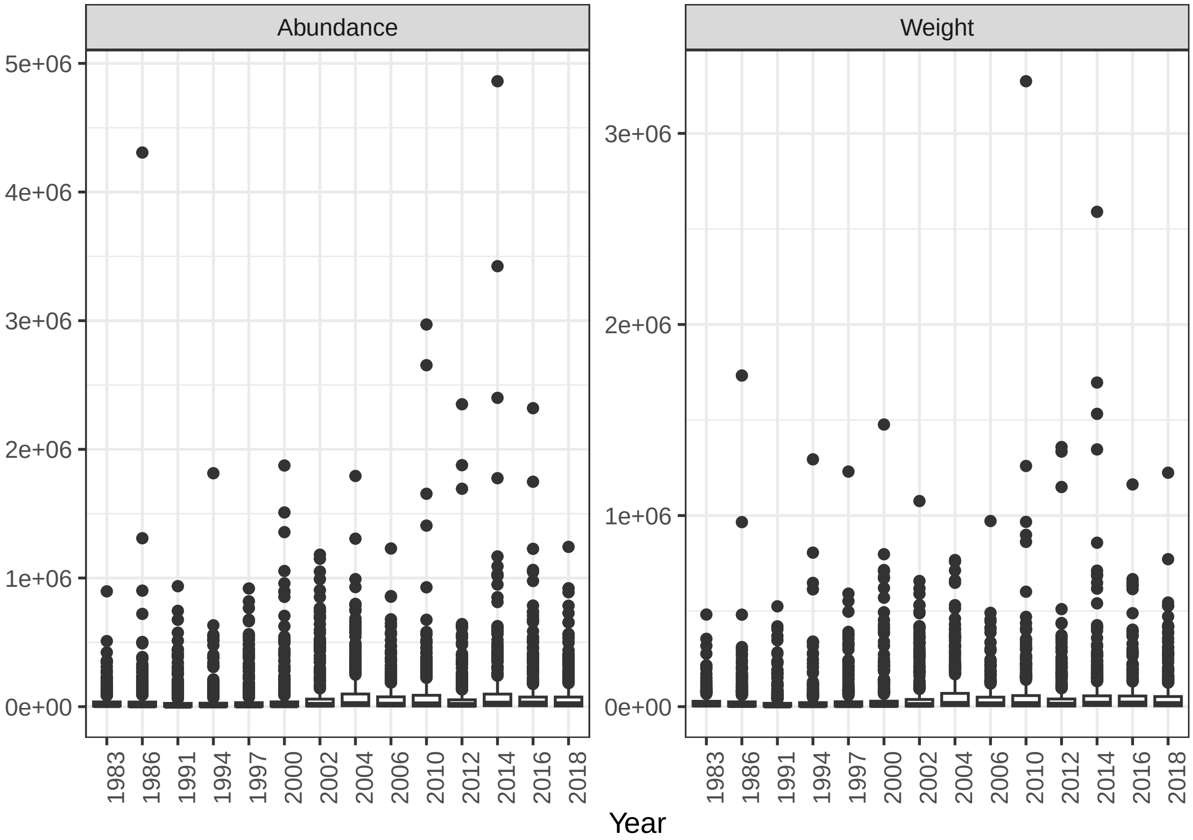
- *num_cpua*, number of individuals (abundance) in $\frac{individuals}{km^2}$
- *num_cpue*, number of individuals (abundance) in $\frac{individuals}{h}$
- *num*, number of individuals (abundance)
- *wgt_cpua*, weight in $\frac{kg}{km^2}$
- *wgt_cpue*, weight in $\frac{kg}{h}$
- *wgt*, weight in kg



5. Extreme values

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

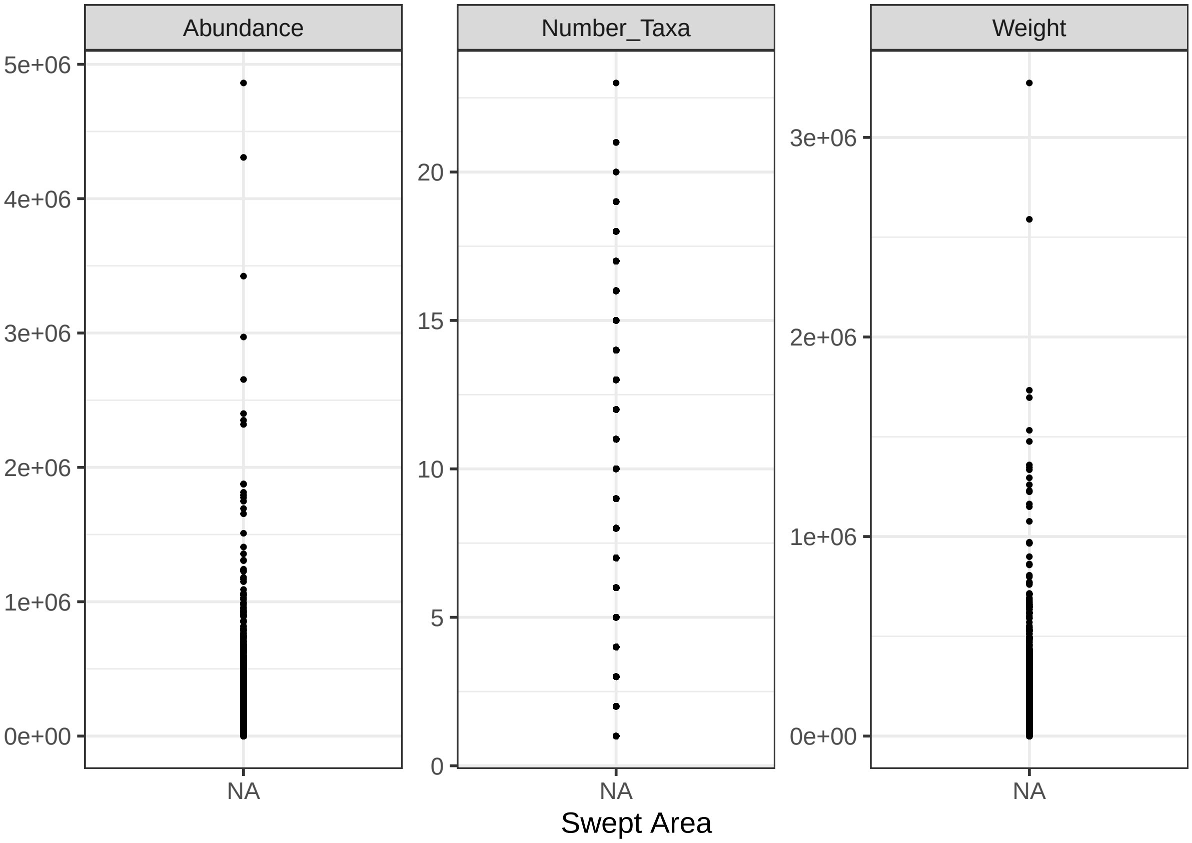
- *num_cpue*, number of individuals (abundance) in $\frac{individuals}{km^2}$
- *wgt_cpue*, weight in $\frac{kg}{km^2}$



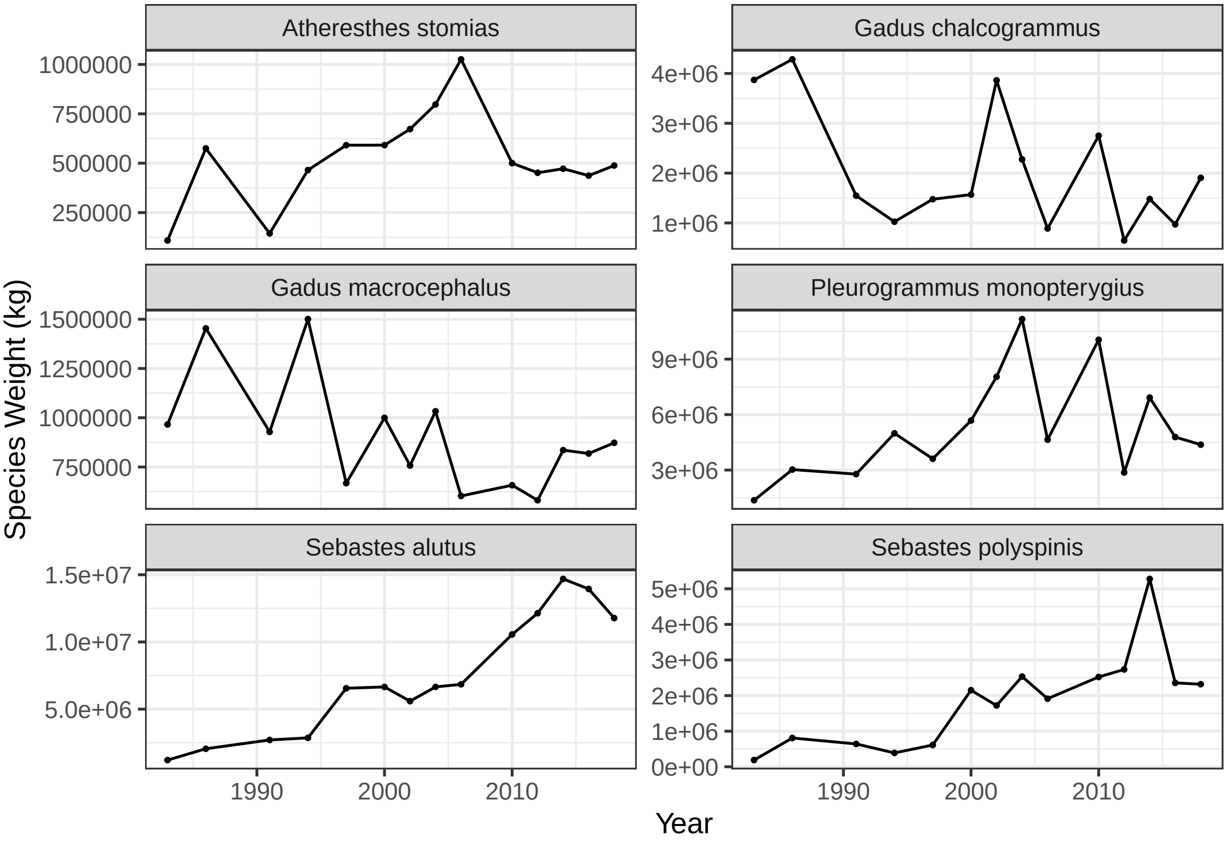
6. Summary of variables against swept area

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

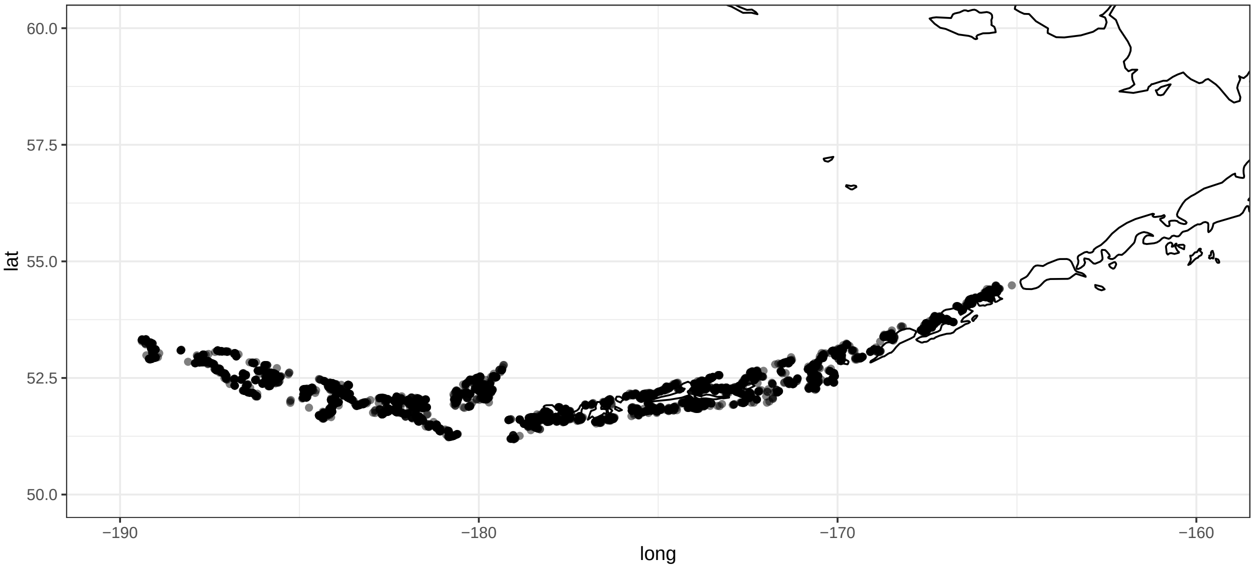


7. Abundance or Weight trends of the six most abundant species



8. Distribution mapping

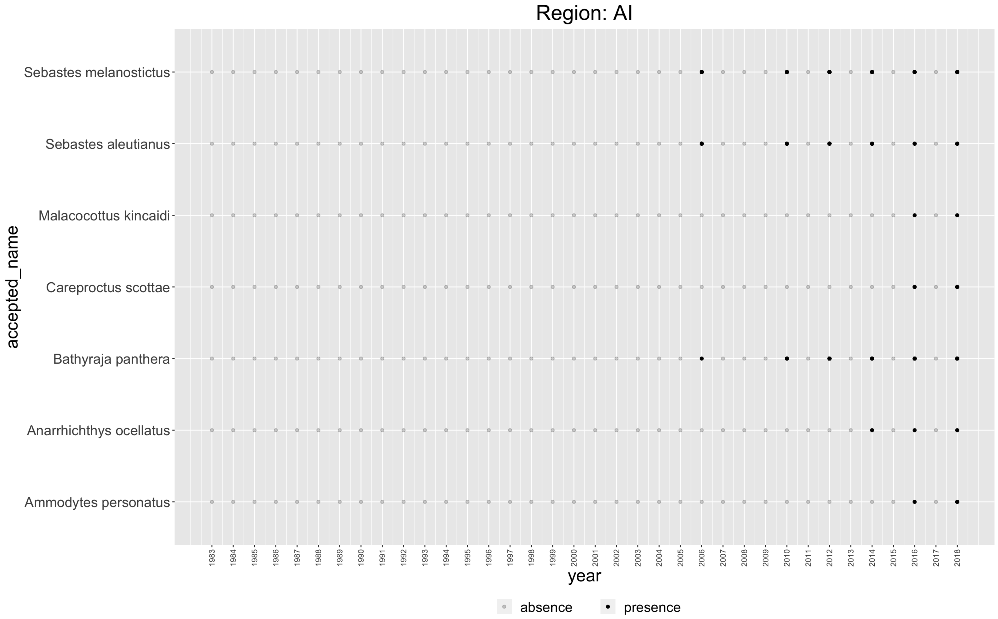
Map of the sampling distribution in space. Note that we only show one year per coordinate.



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

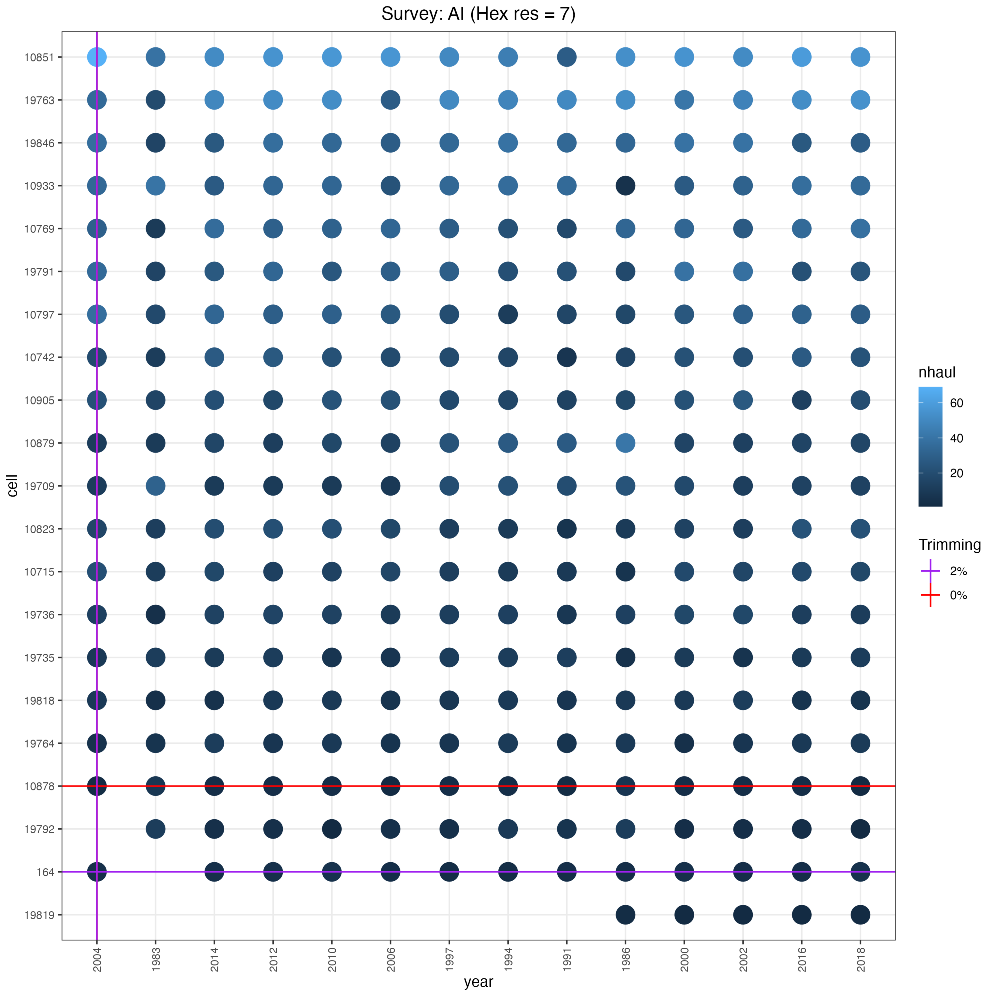
Total number of species	251.0
Percentage of species flagged	2.8

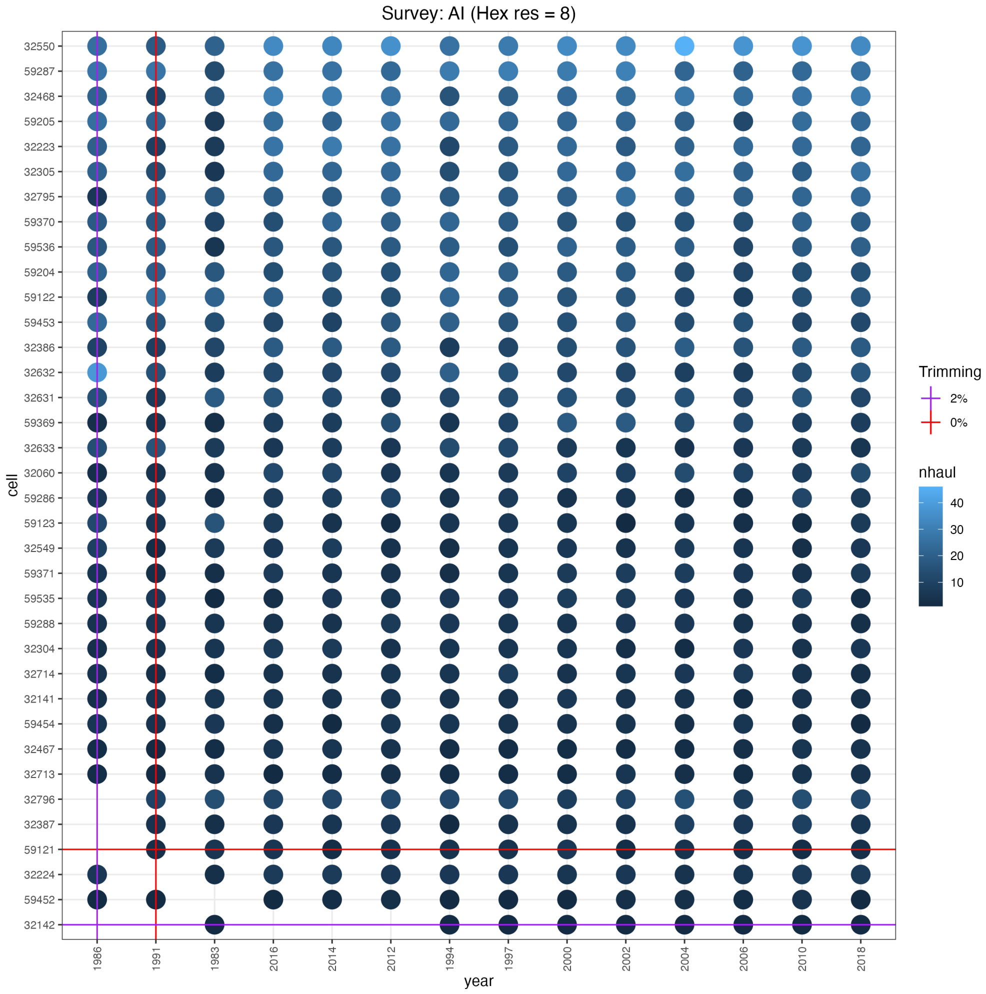
10. Spatio-temporal standardization

a. Standardization method 1

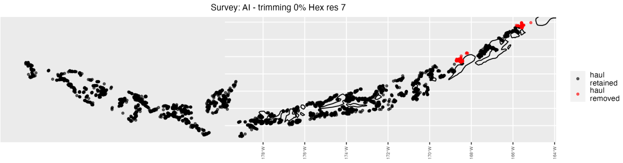
This standardization method was adapted from https://github.com/zoekitchel/trawl_spatial_turnover/blob/master/data_prep_code/species/explore_NorthSea_trimming.Rmd
It was run for hex resolution 7 and 8.

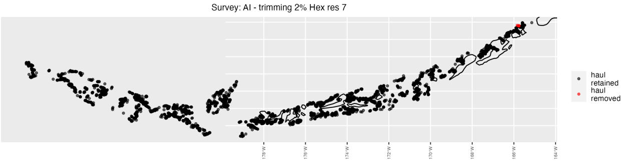
Plot of number of cells x years with overlaid flagging options



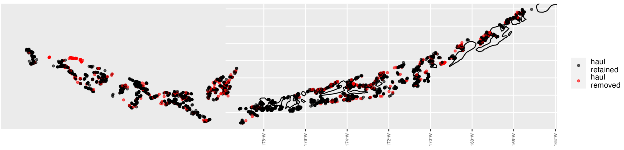


Map of hauls retained and removed per flagging method and threshold

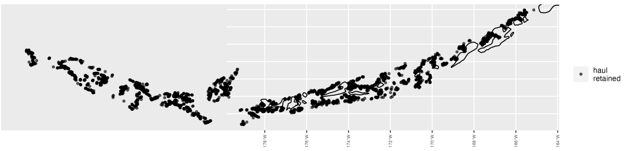




Survey: AI - trimming 0% Hex res 8

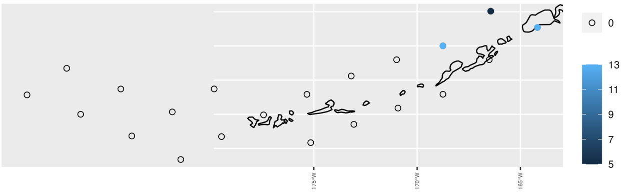


Survey: AI - trimming 2% Hex res 8

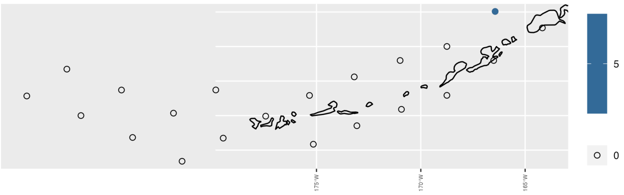


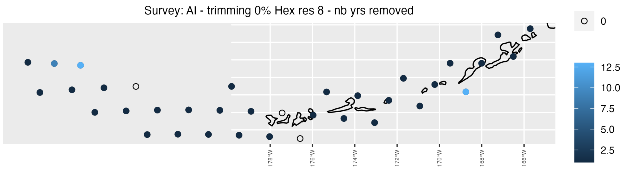
Map of numbers of years removed per grid cell and flagging method/threshold

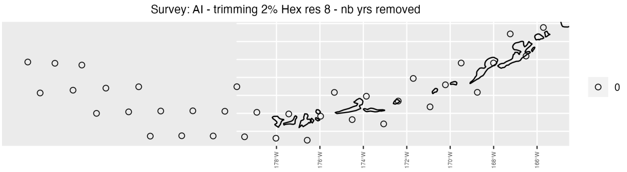
Survey: AI - trimming 0% Hex res 7 - nb yrs removed



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



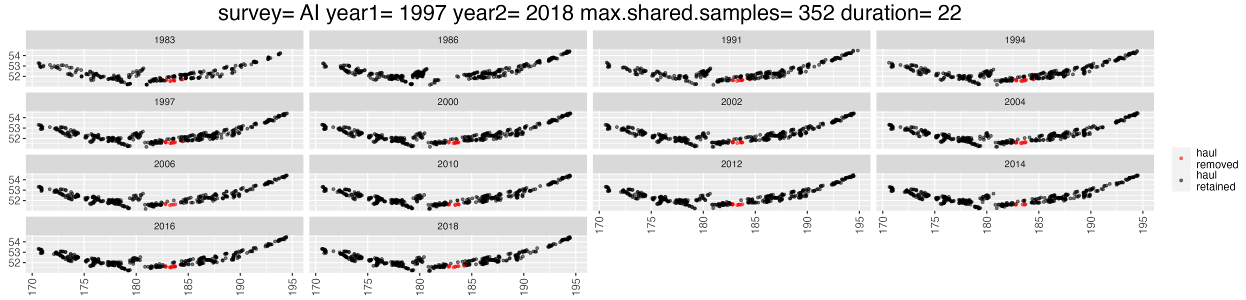




b. Standardization method 2

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	115.0	6.0	483.0	0	1204.0
percentage of hauls removed	2.1	0.1	8.8	0	2.3