

nwfsc_combo_survey

Joshua Zahner

1/30/2023

NWFSC Combo Survey

```
library(nwfscSurvey)

## Loading required package: dplyr
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##   filter, lag
## The following objects are masked from 'package:base':
##   intersect, setdiff, setequal, union
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --
## v ggplot2 3.4.0     v purrr    1.0.1
## v tibble   3.1.8     v stringr  1.5.0
## v tidyr    1.3.0     vforcats  0.5.1
## v readr    2.1.2

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()

source(file=file.path(here::here(), "R", "utils", "plot_comps.R"))

species <- "shortspine_thornyhead"
survey.name <- "NWFSC.Combo"
survey.name.path <- str_replace(tolower(survey.name), "[.]", "_")

data.dir <- here::here("data/") # location of data directory
outputs.dir <- here::here("outputs/")

raw.data.dir <- file.path(data.dir, "raw")
catch.fname <- file.path(raw.data.dir, paste0(survey.name.path, "_survey_catch.csv")) # raw survey catch
bio.fname <- file.path(raw.data.dir, paste0(survey.name.path, "_survey_bio.csv"))      # raw survey bio
out.dir <- file.path(outputs.dir, "surveys", survey.name.path) # send all plot/data outputs here

dir.create(raw.data.dir, recursive=TRUE)
```

```

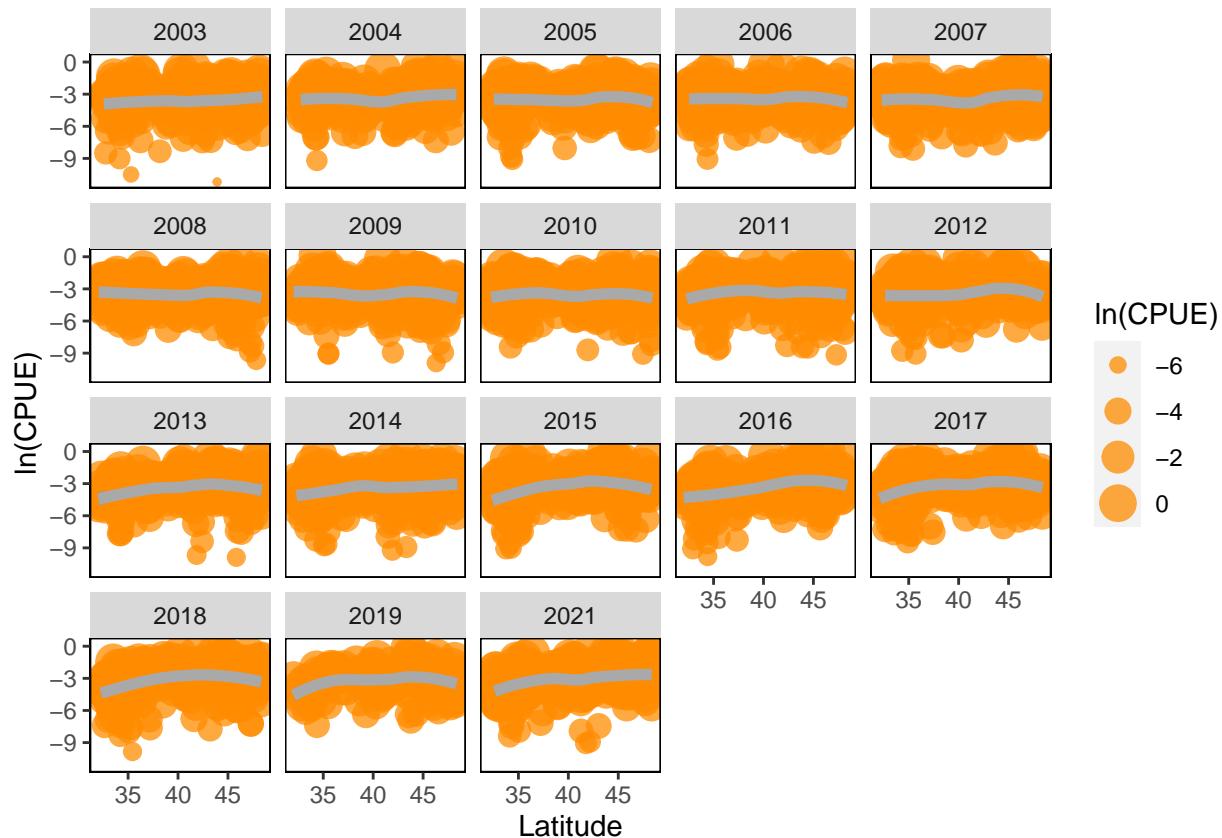
## Warning in dir.create(raw.data.dir, recursive = TRUE):
## '/Users/jzahner/Desktop/shortspine_thornyhead_2023/data//raw' already exists
dir.create(out.dir, recursive=TRUE)

## Warning in dir.create(out.dir, recursive = TRUE):
## '/Users/jzahner/Desktop/shortspine_thornyhead_2023/outputs//surveys/nwfsc_combo'
## already exists

plot_cpue(
  dir = out.dir,
  catch = catch
)

## `geom_smooth()` using formula = 'y ~ x'

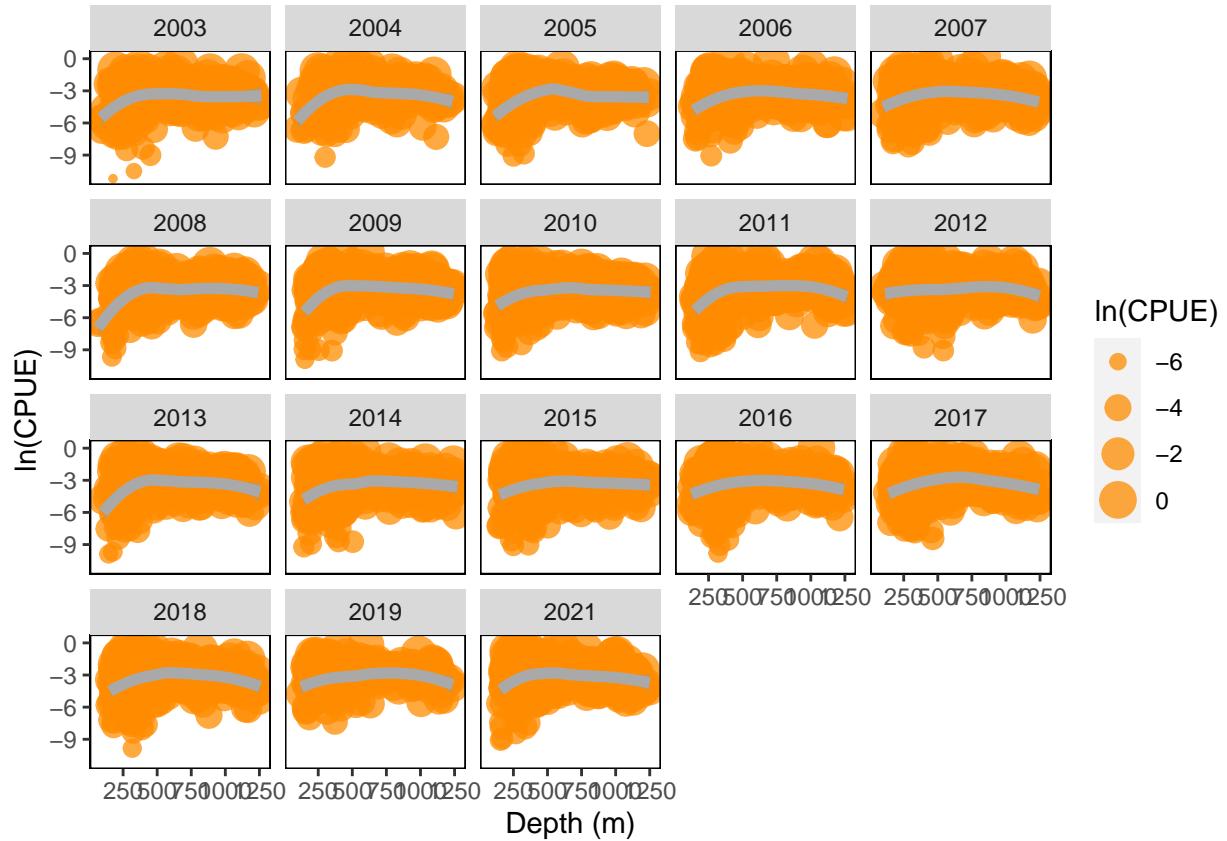
```



```

## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'

```



```
plot_bio_patterns(
```

```
  dir = out.dir,
```

```
  bio = bio,
```

```
  col_name = "Length_cm")
```

```
## Warning: Removed 70 rows containing non-finite values (`stat_summary()`).
```

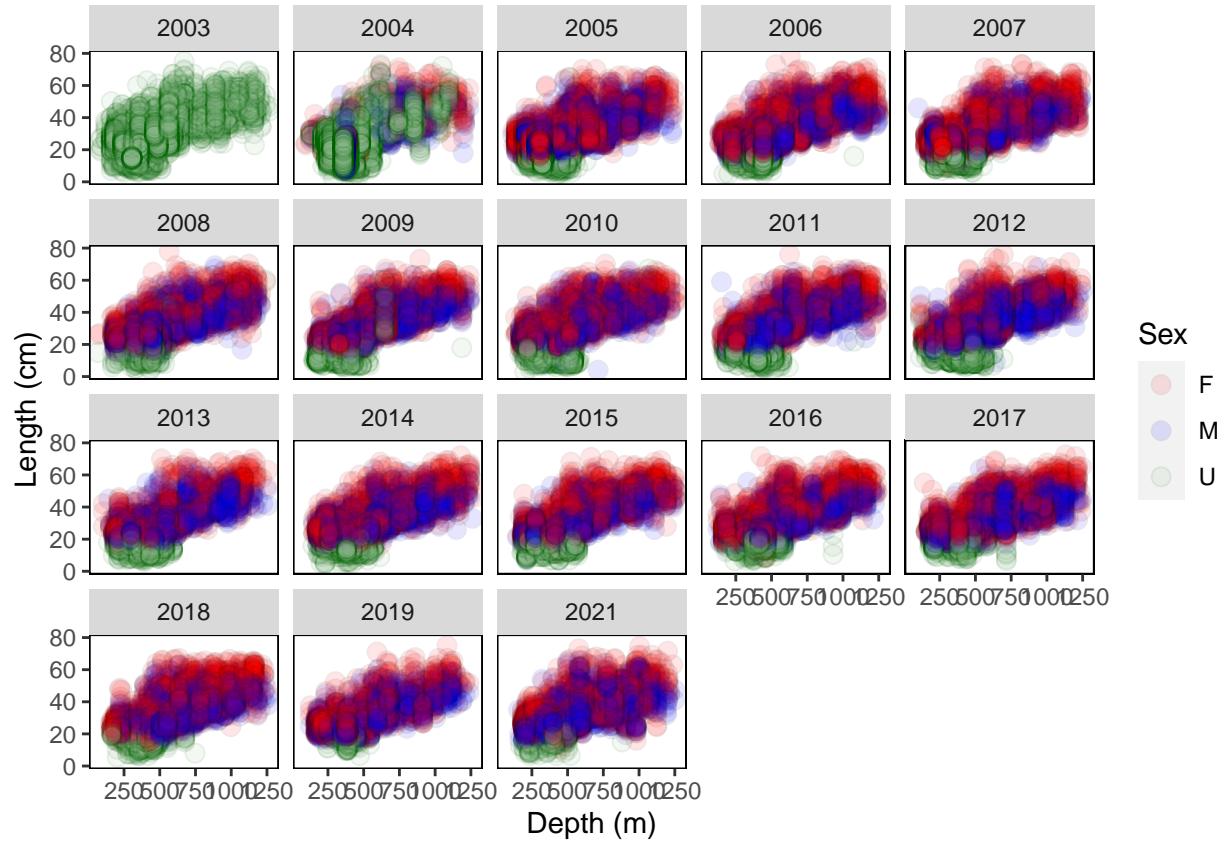
```
## Warning: Removed 70 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 70 rows containing non-finite values (`stat_summary()`).
```

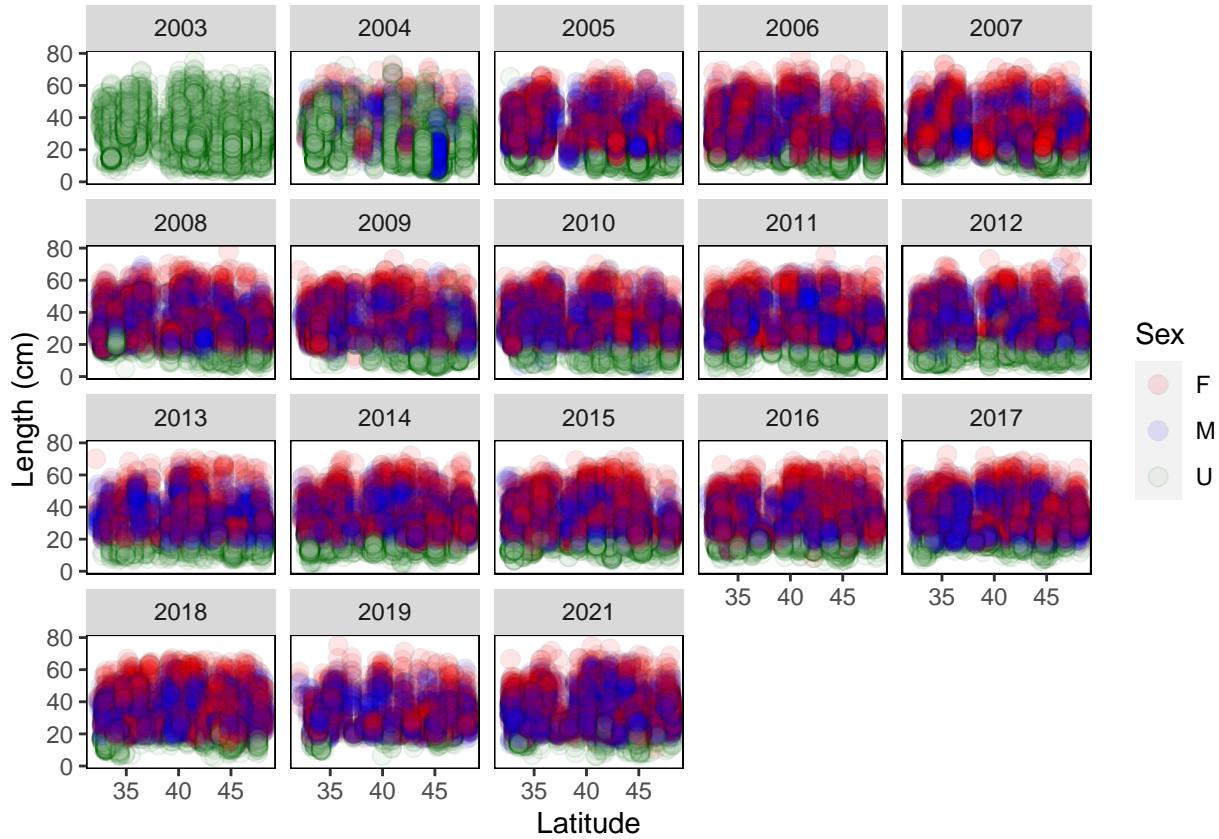
```
## Warning: Removed 70 rows containing missing values (`geom_point()`).
```

```
## Removed 70 rows containing missing values (`geom_point()`).
```

```
## Removed 70 rows containing missing values (`geom_point()`).
```



```
## Warning: Removed 70 rows containing missing values (`geom_point()`).
## Removed 70 rows containing missing values (`geom_point()`).
```



```

# Made a small change to the default wh_plot_proportion() function in nwfscSurvey so
# that the plots are shown in line as well as saved to files.
wh_plot_proportion <- function (data_catch, data_bio, dir = file.path(getwd(), "plots"),
                                bar_width = c("n", "equal"))
{
  stopifnot(any(c(!missing(data_catch), !missing(data_bio))))
  files_all <- file.path(dir, t(outer(X = c(if (!missing(data_catch)) {
    "presence-absence"
  }, if (!missing(data_bio)) {
    "sex"
  }), Y = paste0("_by_", c("depth", "latitude"), ".png"), FUN = paste0)))
  data <- c(if (!missing(data_catch)) {
    dplyr::mutate(data_catch, the_factor = factor(cpue_kg_km2 <=
      0, levels = c(FALSE, TRUE), labels = c("Present",
      "Absent")))) %>% purrr::rerun(.n = 2)
  }, if (!missing(data_bio)) {
    dplyr::mutate(data_bio, the_factor = codify_sex(Sex)) %>%
      purrr::rerun(.n = 2)
  })
  gg_all <- purrr::pmap(.l = list(pdata = data, x = rep(ggplot2::quos(Depth_m,
    Latitude_dd), length(data)/2), width = rep(c(50, 1),
    length(data)/2)), .f = function(pdata, x, width, bar_width) {
    gg <- plot_proportion(data = pdata, column_factor = the_factor,
      column_bin = !!x, width = width, bar_width = bar_width,
      boundary = 0)
  }), bar_width = match.arg(bar_width))
}

```

```

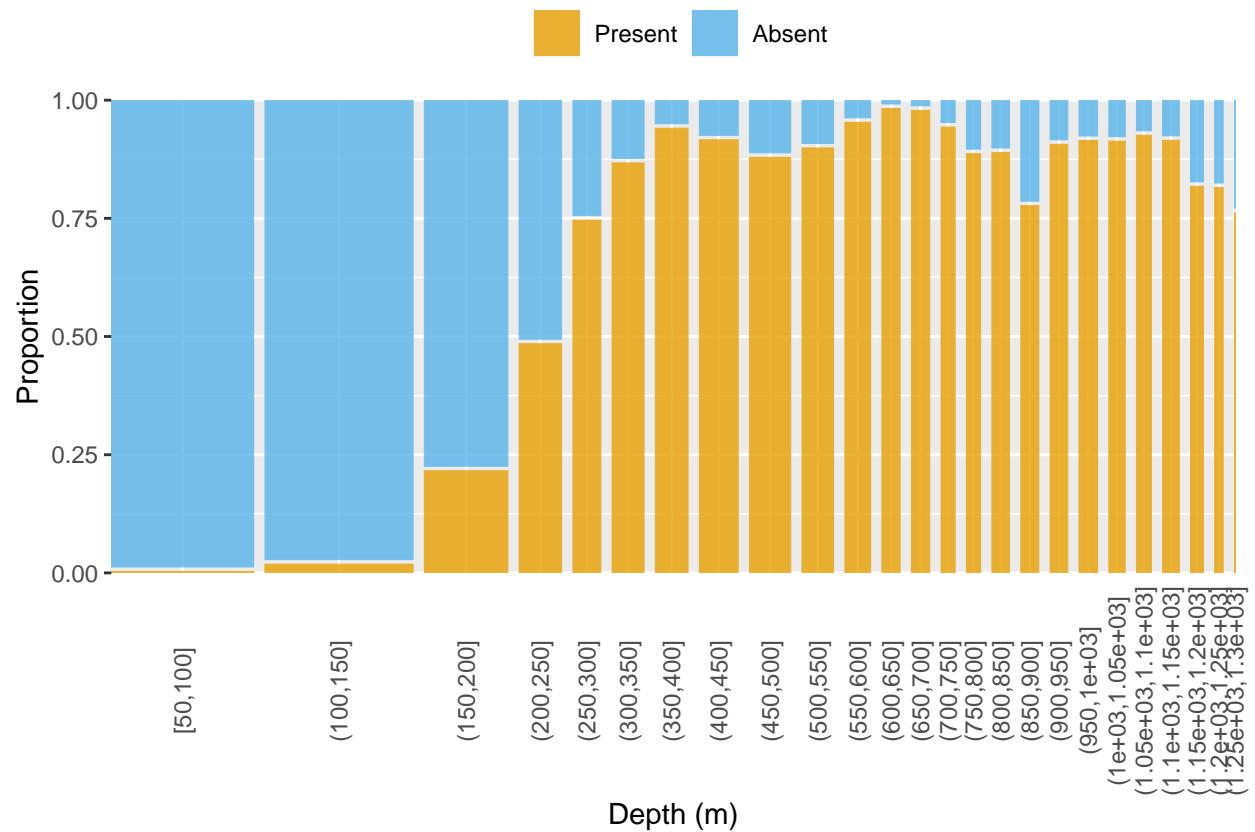
    files_out <- purrr::map2_chr(.x = files_all, .y = gg_all,
      .f = ggplot2::ggsave, height = 7, width = 7)
    return(list(files_out, gg_all))
}

wh_plot_proportion(
  dir = file.path(out.dir, "plots"),
  data_catch = catch,
  data_bio = bio
)

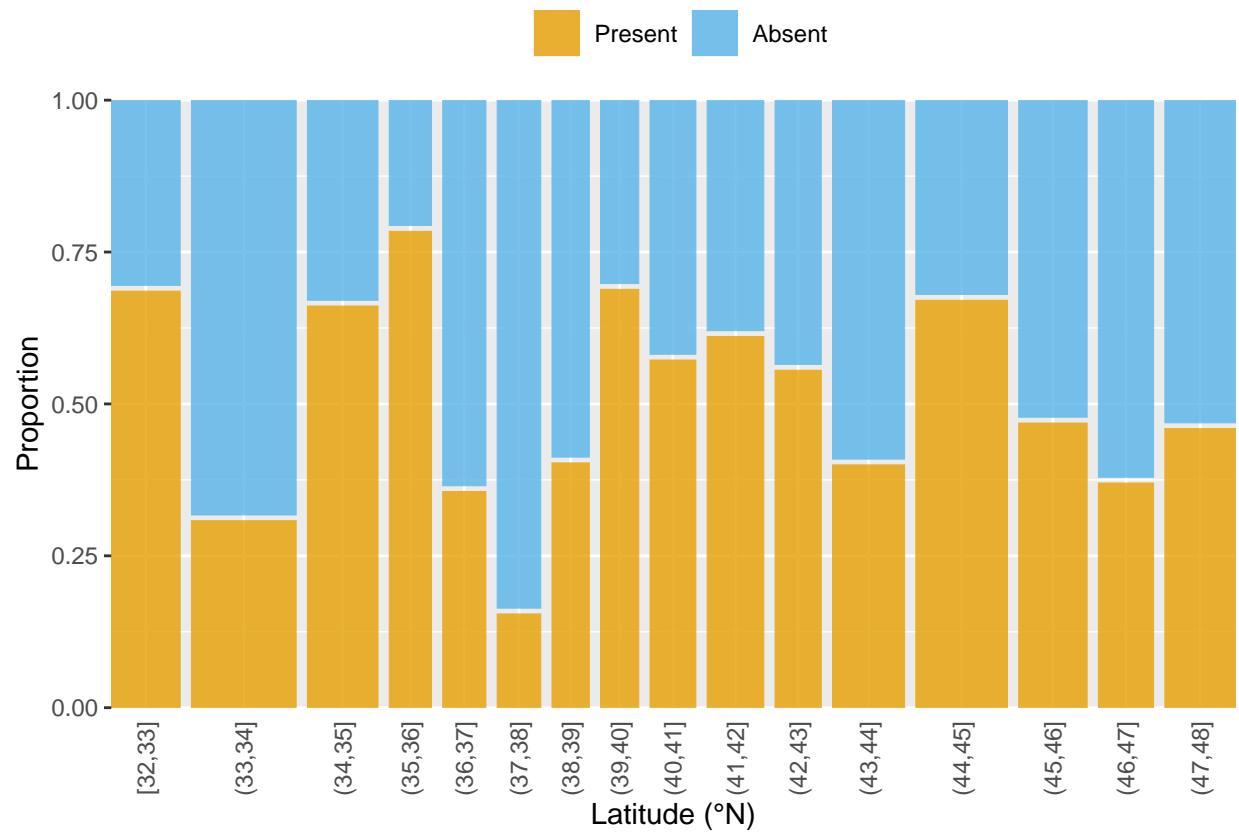
## Warning: `rerun()` was deprecated in purrr 1.0.0.
## i Please use `map()` instead.
##   # Previously
##   rerun(2, .)
##
##   # Now
##   map(1:2, ~.)

## [[1]]
## [1] "/Users/jzahner/Desktop/shortspine_thornyhead_2023/outputs//surveys/nwfsc_combo/plots/presence-absence_by_latin_name"
## [2] "/Users/jzahner/Desktop/shortspine_thornyhead_2023/outputs//surveys/nwfsc_combo/plots/presence-absence_by_latin_name"
## [3] "/Users/jzahner/Desktop/shortspine_thornyhead_2023/outputs//surveys/nwfsc_combo/plots/sex_by_department"
## [4] "/Users/jzahner/Desktop/shortspine_thornyhead_2023/outputs//surveys/nwfsc_combo/plots/sex_by_latitude"
##
## [[2]]
## [[2]][[1]]

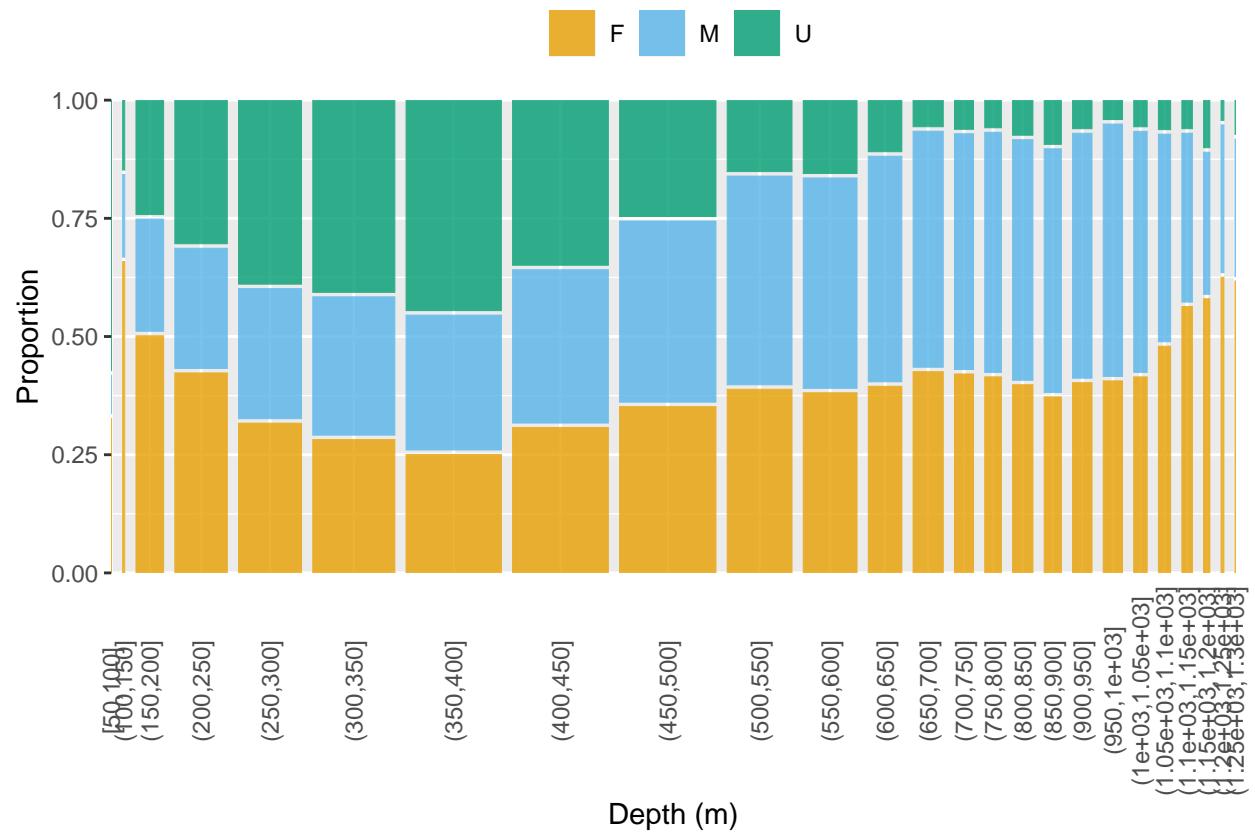
```



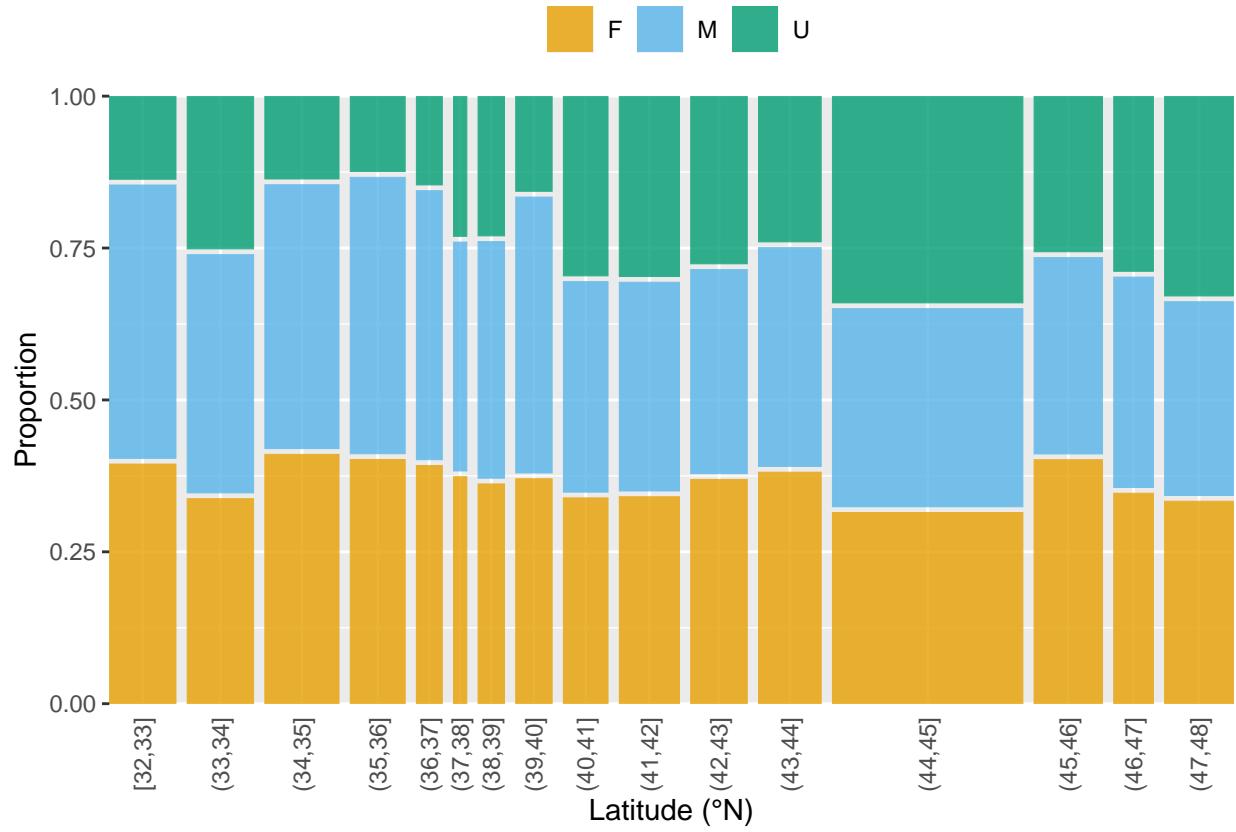
```
##  
## [[2]] [[2]]
```



```
##  
## [[2]] [[3]]
```



```
##  
## [[2]][[4]]
```



```
# Set spatial strata (UPDATE THESE)
strata = CreateStrataDF.fn(
  names      = c("shallow_south", "deep_south", "shallow_cen", "deep_cen", "shallow_north", "deep_north"),
  depths.shallow = c(55,      500,      55,      500,      55,      500),
  depths.deep   = c(500,    1280,    500,    1280,    500,    1280),
  lats.south    = c(32,      32,      40.5,    40.5,    43,      43),
  lats.north    = c(40.5,    40.5,    43,      43,      49,      49)
)
```

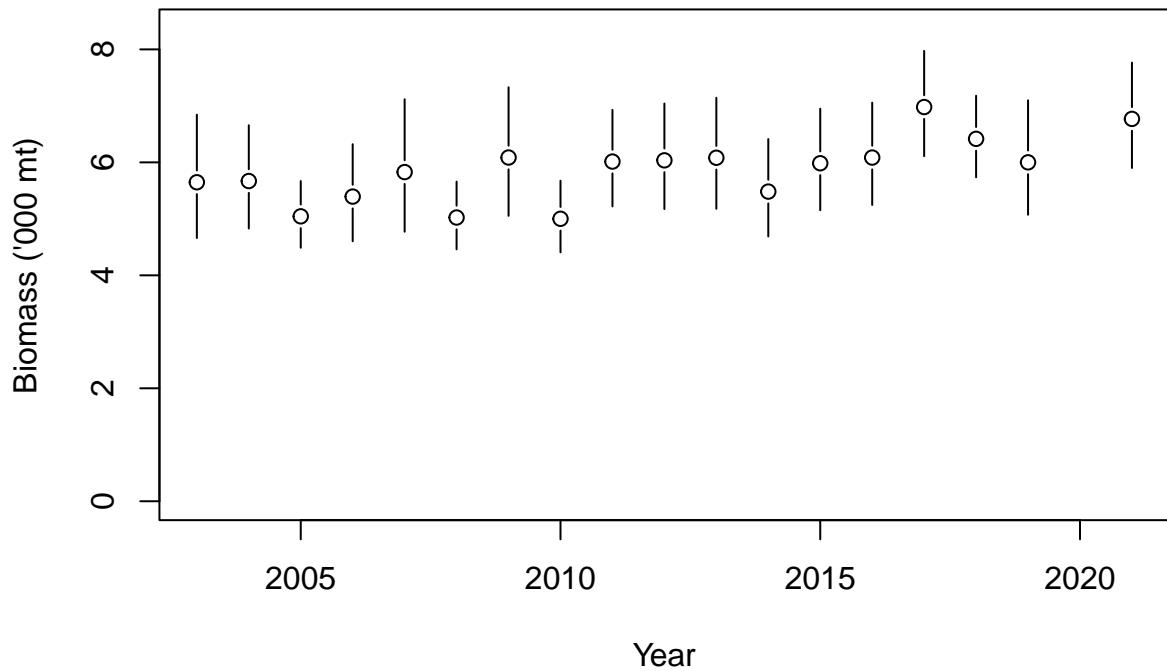
Design Based Indices

```
## Calculate design-based index

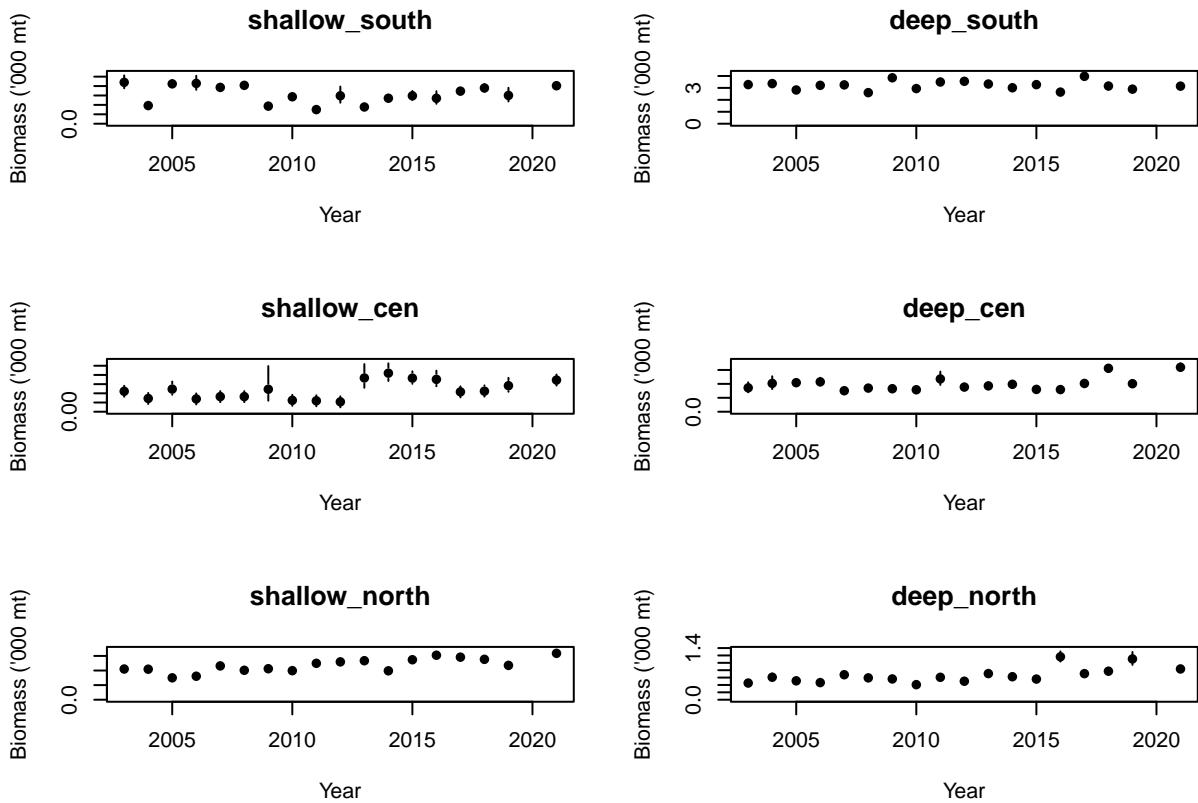
biomass.idx = Biomass.fn(dir = out.dir,
                         dat = catch,
                         strat.df = strata)

PlotBio.fn(
  dir = NULL, # set to `out.dir` to save as file
  dat = biomass.idx,
  scalar = 1000, # convert to kg like csv file
) # produces '_designed_based_index.png'

## Output will not be saved in `dir` because dir = NULL.
```



```
PlotBioStrata.fn(  
  dir = NULL, # set to `out.dir` to save as file  
  dat = biomass.idx,  
  scalar = 1000, # convert to kg like csv file  
) # produces '_designed_based_by_strata_index.png'  
  
## Output will not be saved in `dir` because dir = NULL.
```



Length Compositions

```

n.len.raw <- nrow(bio[!is.na(bio$Length_cm),]) # check is data available

if(n.len.raw > 0){
  n <- GetN.fn(dir = out.dir,
    dat = bio,
    type = "length",
    species = "thorny")

  len_bins <- seq(6, 72, 2) # from last assessment (p. 125)

  Length_Freq <- SurveyLFs.fn(dir = out.dir,
    datL = bio,
    datTows = catch,
    strat.df = strata,
    lgthBins = len_bins)

  PlotFreqData.fn(dir = NULL, # set to `out.dir` to save as file
    dat = Length_Freq)

  plot_comps(dir = out.dir, data = Length_Freq) # uses version in R/utils/
} else{
  print(n.len.raw)
}

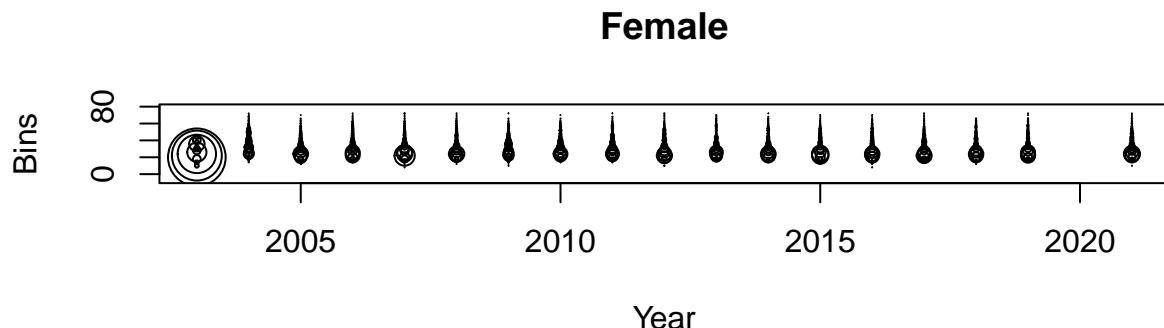
```

```

}

##
## The effN sample size is calculated using the thorny multiplier of 6.91. This number is multiplied by
##
## There are 16 tows where fish were observed but no lengths/ages taken.
## These tows contain 1147 lengths/ages that comprise 0.5 percent of total sampled fish.
## There are 89855 records kept out of 89925 records after removing missing records.
##
## NOTE: Files have been saved the the printfolder directory.
## The first file has the 999 column showing fish smaller or younger than the initial bin.
## Check to make sure there is not a large number of fish smaller or younger than the initial bin.
## The second file has combined the 999 with the first bin and is ready for use in SS.
##
## Output will not be saved in `dir` because dir = NULL.
##
## Warning: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.

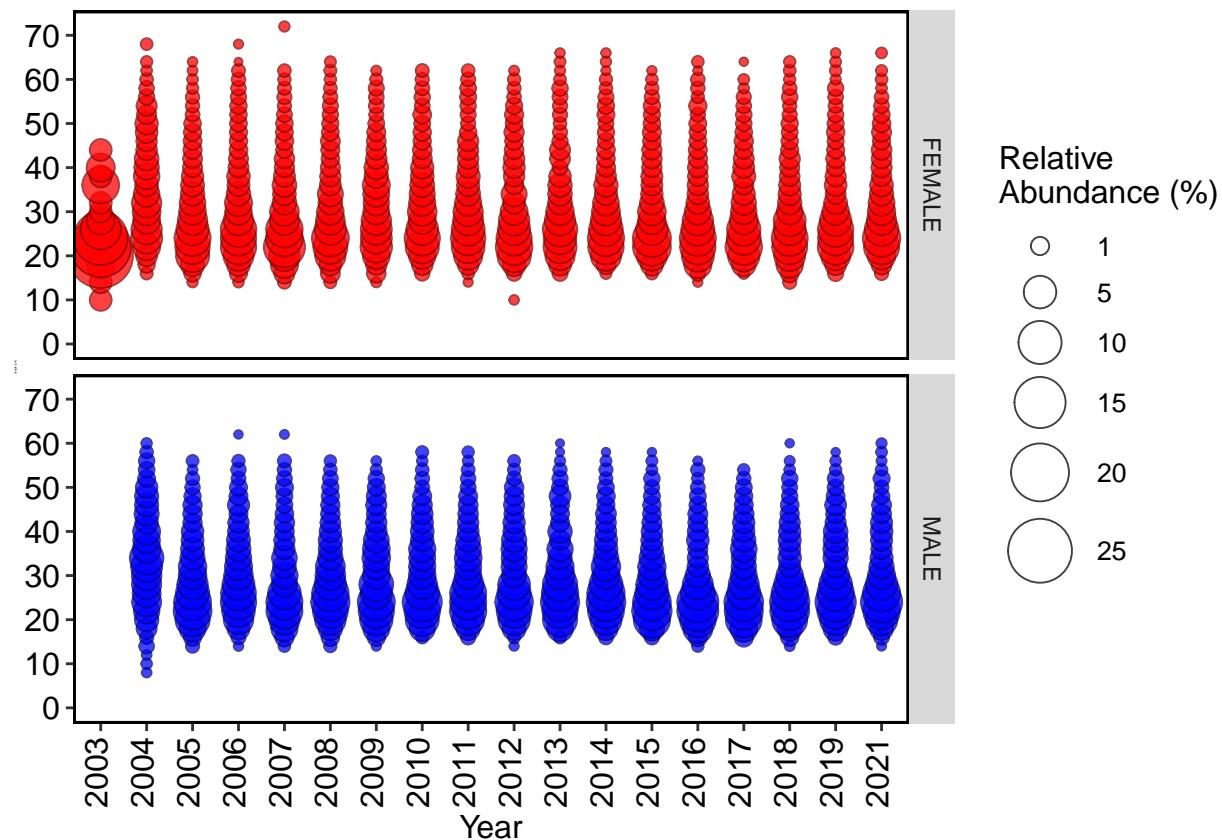
```

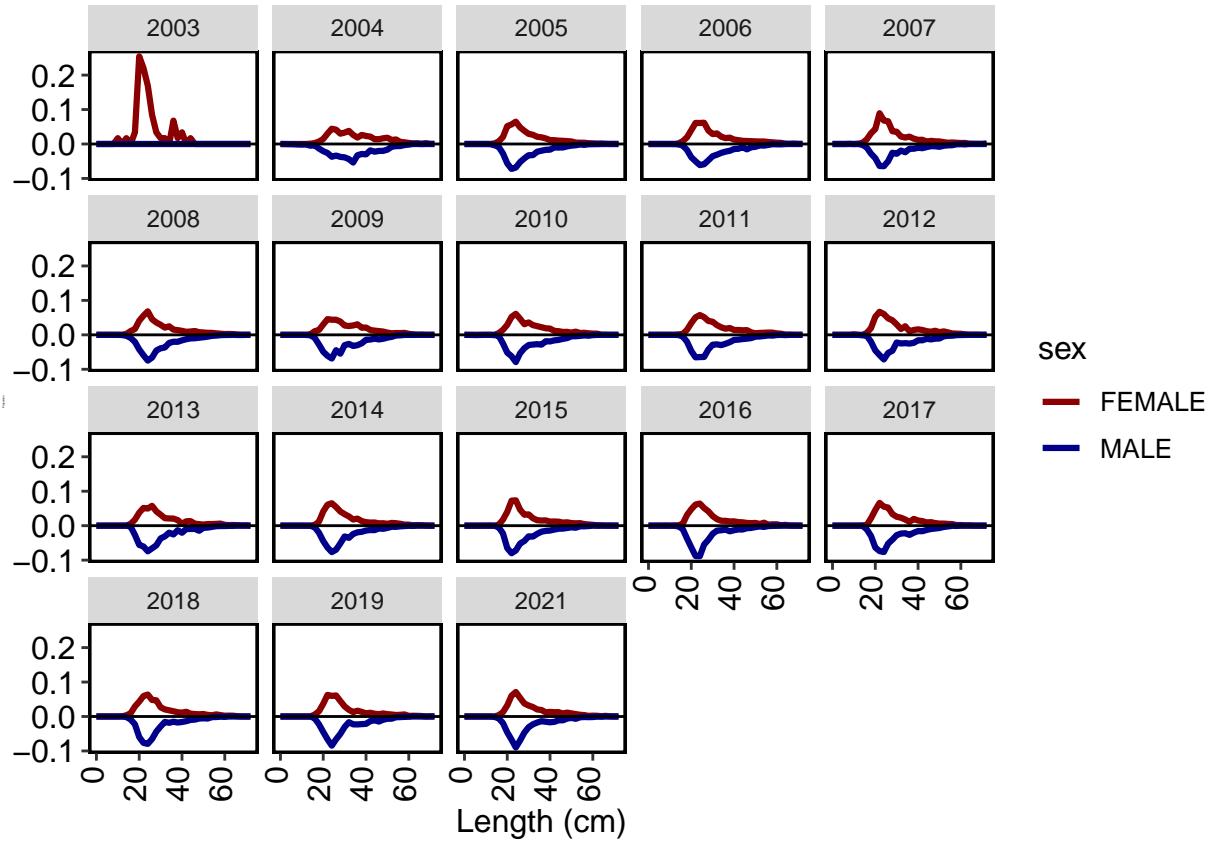


```

## Warning: Removed 430 rows containing missing values (`geom_point()`).
## Warning: Removed 430 rows containing missing values (`geom_point()`).
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.

```





Age Compositions

There is no age data associated with this species and survey, and thus age composition analysis is not possible.

```
n.age.raw <- nrow(bio[!is.na(bio$Age),]) # check is data available

if(n.age.raw > 0){

  n <- GetN.fn(dir = out.dir,
                dat = bio,
                type = "age",
                species = "thorny")

  age_bins <- 1:22 # arbitrarily set right now as not used in previous assessment

  Ages <- SurveyAFs.fn(dir = out.dir,
                        datA = bio,
                        datTows = catch,
                        strat.df = strata,
                        ageBins = age_bins,
                        nSamps = n)

  PlotFreqData.fn(
    dir = out.dir,
    dat = Ages
```

```

)
plot_comps(
  dir = out.dir,
  data = Ages
) # uses the version in R/utils/

} else{
  print(n.age.raw)
}

## [1] 0

```

Conditional Age-at-Length

There are no ages associated with this species and survey, so CAAL analyses are not possible.

```

if(n.age.raw > 0){
  caal <- SurveyAgeAtLen.fn(dir = out.dir,
                            datAL = bio,
                            datTows = catch,
                            strat.df = strata,
                            lgthBins = len_bins,
                            ageBins = age_bins)
}

```