20 mm index calculation R script

Interagency Ecological Program, California Department of Fish and Wildlife

15 July 2022

The following code is used to calculate the 20 mm index by the California Department of Fish and Wildlife as of the date of this document.

```r
# Index calculations

# Loading libraries --------------------------------------------------------
library(dplyr)
library(readr)
library(ggplot2)
library(lubridate)
library(tidyr)
library(ggsci)
library(ggdark)

# Geometric mean formula
geometric.mean <- function(x, na.rm=TRUE) exp(mean(log(x), na.rm=na.rm))

myTheme <- dark_theme_bw(base_size = 24) +
  theme(panel.grid.minor = element_blank(),
        plot.background = element_rect(fill = "#2c2828", color = NA),
        panel.background = element_blank(),
        # panel.grid.major = element_blank(),
        panel.grid.major = element_line(color = "#646464", size = 0.2),
        legend.background = element_blank(),
        legend.key = element_blank())

theme_set(myTheme)

# Reading in the 20 mm -----------------------------------------------------

TTMM <- read_csv(file.path("data-raw", "20mm", "TMM.csv"),
                 col_types = cols(
                   Source = col_character(),
                   Station = col_double(),
                   Latitude = col_double(),
                   Longitude = col_double(),
                   Date = col_date(format = ""),
                   Datetime = col_datetime(format = ""),
                   Survey = col_double(),
                   TowNum = col_double(),
                   Depth = col_double(),
                   SampleID = col_character(),
                   Method = col_character(),
                   Tide = col_character(),
                   Sal_surf = col_double(),
                   Temp_surf = col_double(),
                   Secchi = col_double(),
                   Tow_volume = col_double(),
                   Cable_length = col_double(),
                   Taxa = col_character(),
                   Length = col_double(),
                   Count = col_double(),
                   Length_NA_flag = col_character(),
                   Notes_survey = col_character(),
                   Notes_station = col_character(),
                   Notes_gear = col_character()
                 )) %>%
  filter(Survey %in% 1:9)

# Calculating index -------------------------------------------------------

countTows <- TTMM %>%
  distinct(Date, Survey, Station, TowNum) %>%
  group_by(Date, Survey, Station) %>%
  tally(name = "countTows")

indexStations <- read_csv(file.path("docs", "StationCords_20mmTN.csv"),
                          col_types = cols(
                            Station = col_double(),
                            Lat = col_character(),
                            Long = col_character(),
                            Status = col_character()
                          )) %>%
  mutate(Index = ifelse(grepl("20mmStation|Non-Index", Status),
                        T, F)) %>%
  filter(Index %in% T)

meanForkLength <- TTMM %>%
  filter(Taxa == "Hypomesus transpacificus",
         Station %in% indexStations$Station,
         Length < 50 & month(Date) < 6 | Length < 60 & month(Date) >= 6) %>%
  group_by(year = year(Date),
           Survey) %>%
  summarize(meanForkLength = sum(Length * Count, na.rm = T)/sum(Count, na.rm = T)) %>%
  mutate(threshold20 = ifelse(meanForkLength > 20, T, NA))

# ***IMPORTANT***: The subsampling events here are expanded out to the non-measured fish
# This means that there are fractional counts in there. For now, Lauren Damon,
# Vanessa Mora, and Trinh Nguyen have decided that there is no rounding required.
# This may need to be revisited in the future.

geometricMeans <- TTMM %>%
  filter(Station %in% indexStations$Station) %>%
  left_join(countTows,
            by = c("Date", "Survey", "Station")) %>%
```

```r
  ungroup() %>%
  mutate(month = month(Date),
         # This is where the adult filter comes in
         # 50 for < June and 60 >= June
         Count = case_when(Taxa == "Hypomesus transpacificus" & Length < 50 & month < 6 ~ Count,
                           Taxa == "Hypomesus transpacificus" & Length < 60 & month >= 6 ~ Count)) %>%
  group_by(Date, Survey, Station, countTows) %>%
  summarise(# This standardizes CPUE by # of tows to be summed later. This is the same as
    # calculating mean after summation
    CPUE = sum(((Count/Tow_volume) * 10000)/countTows, na.rm = T)) %>%
  group_by(year = year(Date),
           Survey) %>%
  summarize(geoMean = geometric.mean(CPUE + 1) - 1)

historicalIndex <- read_csv(file.path("docs", "20mmHistoricalIndex.csv"),
                            col_types = cols(
                              YEAR = col_double(),
                              INDEX = col_double()
                            )) %>%
  rename(year = YEAR,
         indexHistorical = INDEX)

index <- geometricMeans %>%
  left_join(meanForkLength,
            by = c("year", "Survey")) %>%
  group_by(year, threshold20) %>%
  mutate(firstOccurence = ifelse(row_number() == 1 & threshold20 == T, T, NA),
         meanForkLength = ifelse(is.na(meanForkLength), 0, meanForkLength)) %>%
  group_by(year) %>%
  mutate(oneSurveyBack = lead(firstOccurence, 1),
         twoSurveyBack = lead(firstOccurence, 2),
         oneSurveyForward = firstOccurence,
         twoSurveyForward = lag(firstOccurence, 1),
         indexSurveys = coalesce(oneSurveyBack, twoSurveyBack,
                                 oneSurveyForward, twoSurveyForward),
         # what was the previous meanFL?
         previousMeanFL = ifelse(firstOccurence, lag(meanForkLength, 1), NA),
         noCatchesSeason = sum(geoMean)) %>%
  filter(indexSurveys|noCatchesSeason == 0) %>%
  # Adding in # of indexed surveys warning
  group_by(year) %>%
  add_count(name = "indexSurveyCount") %>%
  fill(previousMeanFL, .direction = "updown") %>%
  summarize(index = round(sum(geoMean), 1),
            indexSurveyCount = mean(indexSurveyCount),
            previousMeanFL = mean(previousMeanFL),
            .groups = "drop") %>%
  left_join(historicalIndex,
            by = "year") %>%
  # If there are not 4 index surveys, then return NA
  mutate(index = ifelse(indexSurveyCount != 4, NA, index))

# Plotting it
```

```r
index %>%
  select(-c(indexSurveyCount, previousMeanFL)) %>%
  pivot_longer(-year, names_to = "era", values_to = "index") %>%
  ggplot(aes(x = year, y = index, color = factor(era))) +
  geom_line(size = 1) +
  geom_point(size = 4) +
  scale_color_npg() +
  labs(x = "Year",
       y = "Index",
       color = "Variant") +
  theme(legend.position = "top")

# Relevant warnings
warning(c(index %>%
            filter(indexSurveyCount == 3) %>%
            pull(year) %>%
            paste0("Years in which the number of index surveys were < 4: ", ., "\n"),
          index %>%
            filter(previousMeanFL == 0) %>%
            pull(year) %>%
            paste(., collapse = ", ") %>%
            paste("Years in which the previous survey mean fork length was NA or 0:", .)),
        call. = F)
```