Eric Ward

Packages

We'll grab the data using the nwfscAssess package and analysis with sdmTMB

```
#devtools::install_github("nwfsc-assess/nwfscSurvey")
library(nwfscSurvey)
#devtools::install_github("pbs-assess/sdmTMB")
library(sdmTMB)
library(dplyr)
```

Data

```
hauls = nwfscSurvey::PullHaul.fn(SurveyName = "NWFSC.Combo")
hauls$trawl_id = as.character(hauls$trawl_id)
# grab catch data just for eulachon
catch = nwfscSurvey::PullCatch.fn(SciName="Thaleichthys pacificus",
                                  SurveyName="NWFSC.Combo")
names(catch) = tolower(names(catch))
catch$trawl_id = as.character(catch$trawl_id)
combined = dplyr::left_join(hauls, dplyr::select(catch,
  trawl_id, year, pass, cpue_kg_km2))

saveRDS(combined, "wcbts.rds")
```

Modeling

Filter out NAs

```
dat <- readRDS("wcbts.rds")
# drop NAs
dat <- dat[-which(is.na(dat$cpue_kg_km2)),]
```

Convert to UTMs

```
library(sp)
# do spatial projection and scaling
coordinates(dat) <- c("longitude_dd", "latitude_dd")
proj4string(dat) <- CRS("+proj=longlat +datum=WGS84")
newproj <- paste("+proj=utm +zone=10 ellps=WGS84")
dat <- spTransform(dat, CRS(newproj))
dat <- as.data.frame(dat)
dat$lon <- dat$longitude_dd/1000
dat$lat <- dat$latitude_dd/1000
```

Generate day of year

```
library(lubridate)
dat$month <- as.numeric(substr(dat$date_yyyymmdd,5,6))
```

```
dat$day <- as.numeric(substr(dat$date_yyyymmdd,7,8))
dat$yday <- lubridate::yday(lubridate::parse_date_time(paste(dat$year,
  dat$month, dat$day), orders="ymd"))
```

Fit the model using sdmTMB. We'll assume (1) year is a factor, (2) quadratic effect of day of the year, and (3) autoregressive spatial fields

```
spde <- make_mesh(dat, c("lon", "lat"), cutoff = 15)

dat$yday_scaled = scale(dat$yday)
dat$yday_scaled2 = dat$yday_scaled^2
m <- sdmTMB(
  cpue_kg_km2 ~ yday_scaled + yday_scaled2 + as.factor(year),
  data = dat,
  time="year",
  spde = spde,
  family = tweedie(link = "log"),
  ar1_fields=TRUE)
saveRDS(m, "fitted_model.rds")
```

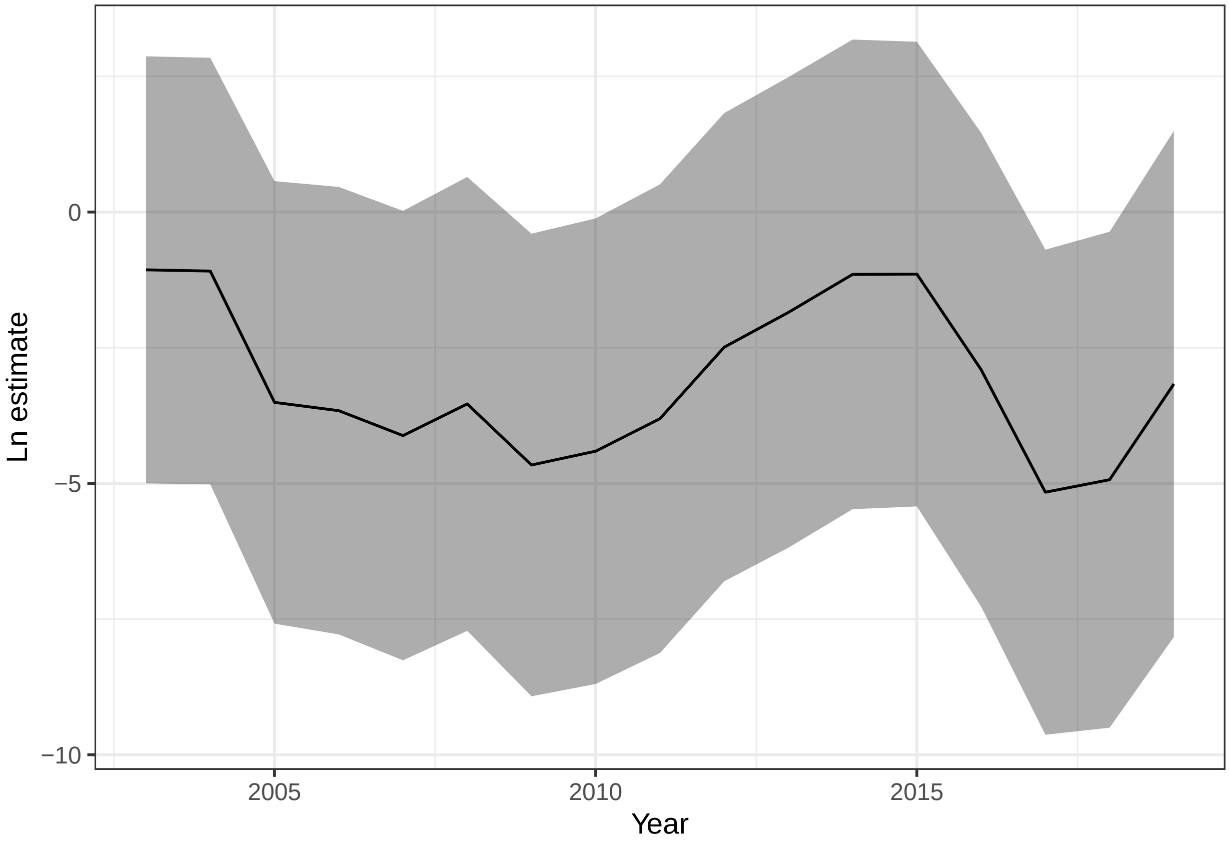
Predictions

```
grid = readRDS("wc_grid.rds")
grid = dplyr::rename(grid, lon = X, lat = Y)
grid = dplyr::mutate(grid,
  depth_scaled = as.numeric(scale(-depth)),
  depth_scaled2 = depth_scaled^2) %>%
  dplyr::select(-log_depth_scaled,
    -log_depth_scaled2)
grid$yday_scaled = 0.5894474 # sept 1
grid$yday_scaled = as.matrix(grid$yday_scaled,ncol=1)
grid$yday_scaled2 = grid$yday_scaled^2
grid$cell = seq(1,nrow(grid))
pred_grid = expand.grid(cell = grid$cell, year = 2003:max(dat$year))
pred_grid = dplyr::left_join(pred_grid, grid)
```

```
m <- readRDS("fitted_model.rds")

prediction = predict(m, newdata = pred_grid, return_tmb_object = TRUE)
index <- get_index(prediction, bias_correct = FALSE)
saveRDS(index,"index.rds")
```

```
index = readRDS("index.rds")
library(ggplot2)
ggplot(index, aes(year, log_est)) +
  geom_ribbon(aes(ymin=log_est-2*se,ymax=log_est+2*se),alpha=0.4) +
  geom_line() + theme_bw() + ylab("Ln estimate") + xlab("Year")
```


Ln estimate
0
−5
−10
2005
2010
2015
Year