Run sdmTMB

Owen Liu

6/8/2021

Contents

Purpose

Join trawl survey data with ROMS oceanographic data and substrate data, and write an sdmTMB wrapper to run models. Actual modelling done in script `sdmTMB models.Rmd`

Import Data

Trawl Data

Trawl data, including a matching key to link to ROMS data. The hindcast ROMS data has values for all trawl survey locations for all times, but we just want the values matched to the actual trawl survey times/dates.

```
trawl <- read_rds(here('data','nwfsc_trawl_data.rds')) %>%
  # convert date from character to date
  mutate(date=as_date(date))
# roms time refernce
roms_time <- read_rds(here('data','roms_time_date_reference.rds'))
trawl_locs <- read_rds(here::here('data','trawl','trawlID.rds')) %>%
  # add a dummy indicator of a "real" trawl survey location in time/space
  mutate(date=as_datetime(trawl_time,origin="1900-01-01")) %>%
  mutate(date=as_date(date)) %>%
  mutate(realTrawl=1) %>%
  # join the roms_time reference
  left_join(roms_time) %>%
  # select the variable we'll use to match
  dplyr::select(station,date,time,lon_trawl,lat_trawl,depth_trawl,realTrawl)
```

```
## Joining, by = "date"
```

```
trawl <- trawl %>%
  left_join(trawl_locs,by=c("date"="date","longitude_dd"="lon_trawl","latitude_dd"="lat_trawl","depth"=
```

Filter the trawl survey data to fit within the time frame for which we have ROMS hindcast data (1980-2010)

```
trawl <- trawl %>%
  filter(realTrawl==1) %>%
  dplyr::select(-realTrawl) %>%
  # rename time to something more useful
  rename(roms_hindcast_day=time)
glimpse(trawl)
```

```
## Rows: 369,822
## Columns: 25
## $ trawl_id            <dbl> 2.00303e+11, 2.00303e+11, 2.00303e+11, 2.00403e+11, 2.00503e+11, 2.00503
## $ scientific_name     <chr> "sebastes elongatus", "sebastes elongatus", "sebastes elongatus", "sebas
## $ project             <chr> "NWFSC.Combo", "NWFSC.Combo", "NWFSC.Combo", "NWFSC.Combo", "NWFSC.Combo
## $ year                <int> 2003, 2003, 2003, 2004, 2005, 2005, 2005, 2005, 2005, 2005, 2006, 2006,
## $ pass                <int> 1, 2, 1, 2, 2, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2,
## $ vessel              <chr> "Captain Jack", "Excalibur", "Ms. Julie", "Excalibur", "Excalibur", "Ms.
## $ tow                 <int> 111, 92, 119, 67, 205, 13, 38, 65, 116, 162, 169, 48, 154, 123, 169, 1,
## $ date                <date> 2003-07-29, 2003-09-27, 2003-07-30, 2004-09-06, 2005-10-16, 2005-05-31,
## $ longitude_dd        <dbl> -123.1244, -124.6703, -122.5511, -124.4600, -118.6017, -124.2775, -124.4
## $ latitude_dd         <dbl> 38.07694, 41.48583, 37.37694, 43.67444, 33.65722, 45.83417, 46.31750, 43
## $ area_swept_ha       <dbl> 1.837650, 3.047135, 1.651004, 1.821803, 1.178600, 1.530574, 1.453994, 1.
## $ subsample_count     <dbl> 0, 0, 0, 48, 0, 1, 6, 0, 16, 0, 1, 53, 0, 0, 0, 0, 0, 0, 0, 49, 0, 0, 2,
## $ subsample_wt_kg     <dbl> 0.00, 0.00, 0.00, 10.30, 0.00, 0.12, 0.80, 0.00, 1.95, 0.00, 0.02, 14.10
## $ total_catch_numbers <dbl> 0, 0, 0, 48, 0, 1, 6, 0, 16, 0, 1, 242, 0, 0, 0, 0, 0, 0, 0, 81, 0, 0, 2
## $ total_catch_wt_kg   <dbl> 0.00, 0.00, 0.00, 10.30, 0.00, 0.12, 0.80, 0.00, 1.95, 0.00, 0.02, 64.45
## $ cpue_kg_km2         <dbl> 0.000000, 0.000000, 0.000000, 565.373957, 0.000000, 7.840197, 55.020868,
## $ species             <chr> "greenstriped rockfish", "greenstriped rockfish", "greenstriped rockfish
## $ o2                  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
## $ temp                <dbl> 9.0359, 3.6300, 10.0280, 8.1160, 5.9237, 6.9375, 6.2750, 6.5258, 9.1859,
## $ sal                 <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
## $ depth               <dbl> 79.9095, 948.4094, 63.6213, 131.7826, 581.3504, 136.0367, 135.9776, 304.
## $ performance         <chr> "Satisfactory", "Satisfactory", "Satisfactory", "Satisfactory", "Satisfa
## $ survey              <chr> "nwfsc.combo", "nwfsc.combo", "nwfsc.combo", "nwfsc.combo", "nwfsc.combo
## $ station             <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 2
## $ roms_hindcast_day   <int> 8611, 8671, 8612, 9016, 9421, 9283, 9289, 9302, 9321, 9336, 9695, 9740,
```

ROMS data

Hindcast ROMS data matched to trawl survey locations and times

```
roms <- read_rds(here::here('data','joined_30d_lagged_t_o.rds')) %>%
  # join the trawl_locs and filter by actual trawl locations and times
  left_join(trawl_locs) %>%
  filter(realTrawl==1) %>%
  select(-realTrawl) %>%
  # rename time to something more useful
  rename(roms_hindcast_day=time)
```

```
## Joining, by = c("station", "time", "lon_trawl", "lat_trawl", "depth_trawl")
```

```
glimpse(roms)
```

```
## Rows: 4,832
## Columns: 13
## $ temp_roms           <dbl> 4.184411, 5.565145, 6.866655, 3.951781, 7.336621, 7.267605, 3.350144, 6.
## $ oxygen_roms         <dbl> 44.78112, 78.12999, 137.91451, 41.38925, 168.28668, 166.14552, 28.76366,
## $ station             <int> 274, 4671, 4772, 4773, 4845, 120, 4672, 4774, 4775, 4783, 4825, 275, 484
## $ roms_hindcast_day   <int> 8576, 8576, 8576, 8576, 8576, 8577, 8577, 8577, 8577, 8577, 8577, 8578,
## $ trawl_time          <dbl> 3265444800, 3265444800, 3265444800, 3265444800, 3265444800, 3265531200,
## $ lon_trawl           <dbl> -124.7761, -124.7389, -124.5156, -124.7325, -124.5428, -124.8156, -125.1
## $ lat_trawl           <dbl> 46.09611, 46.02472, 46.15667, 46.50389, 46.75500, 47.60194, 46.81083, 47
## $ depth_trawl         <dbl> 564.9317, 310.0056, 140.7280, 606.3237, 107.4615, 106.0853, 797.7154, 17
## $ temp_trawl          <dbl> 4.8597, 5.6246, 6.7312, 4.6011, 6.8820, 6.9012, 3.8907, 6.7184, 6.7229,
## $ oxygen_trawl        <dbl> NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN, Na
## $ mean_temp_roms_30   <dbl> 4.227865, 5.564605, 6.970570, 4.016127, 7.363900, 7.492323, 3.361807, 6.
## $ mean_oxygen_roms_30 <dbl> 45.80141, 81.00598, 141.60814, 42.14824, 170.73278, 178.97508, 28.82358,
## $ date                <date> 2003-06-24, 2003-06-24, 2003-06-24, 2003-06-24, 2003-06-24, 2003-06-25,
```

Substrate Data

Here is the substrate data that Blake Feist matched to individual trawl tows.

```
substrate <- read_rds(here('data','substrate','substrate_by_trawlID.rds'))
glimpse(substrate)
```

```
## Rows: 20,746
## Columns: 11
## $ TRAWL_ID                                          <dbl> 1.97706e+11, 1.97706e+11, 1.97706e+11, 1.
## $ `Length_of_towline_outside_substrate_domain_(m)`   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
## $ `Length_of_towline_traversing_hard_substrate_(m)`  <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
## $ `Length_of_towline_traversing_mixed_substrate_(m)` <dbl> 0.00, 0.00, 169.37, 0.00, 0.00, 0.00, 0.0
## $ `Length_of_towline_traversing_soft_substrate_(m)` <dbl> 2628.42, 2484.34, 2899.37, 2570.00, 2639.2
## $ `Total_length_of_towline_(m)`                      <dbl> 2628.42, 2484.34, 3068.74, 2570.00, 2639.2
## $ Proportion_outside_substrate_domain               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
## $ Proportion_hard                                   <dbl> 0.0000000, 0.0000000, 0.0000000, 0.0000000
## $ Proportion_mixed                                  <dbl> 0.00000000, 0.00000000, 0.05519243, 0.0000
## $ Proportion_soft                                   <dbl> 1.00000000, 1.00000000, 0.94480757, 1.0000
## $ prop_hard_mixed                                   <dbl> 0.00000000, 0.00000000, 0.05519243, 0.0000
```

Look at the form of these data

```r
glimpse(trawl)
```

```
## Rows: 369,822
## Columns: 25
## $ trawl_id            <dbl> 2.00303e+11, 2.00303e+11, 2.00303e+11, 2.00403e+11, 2.00503e+11, 2.00503e
## $ scientific_name     <chr> "sebastes elongatus", "sebastes elongatus", "sebastes elongatus", "sebast
## $ project             <chr> "NWFSC.Combo", "NWFSC.Combo", "NWFSC.Combo", "NWFSC.Combo", "NWFSC.Combo
## $ year                <int> 2003, 2003, 2003, 2004, 2005, 2005, 2005, 2005, 2005, 2005, 2006, 2006, 2
## $ pass                <int> 1, 2, 1, 2, 2, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2
## $ vessel              <chr> "Captain Jack", "Excalibur", "Ms. Julie", "Excalibur", "Excalibur", "Ms.
## $ tow                 <int> 111, 92, 119, 67, 205, 13, 38, 65, 116, 162, 169, 48, 154, 123, 169, 1, 2
## $ date                <date> 2003-07-29, 2003-09-27, 2003-07-30, 2004-09-06, 2005-10-16, 2005-05-31,
## $ longitude_dd        <dbl> -123.1244, -124.6703, -122.5511, -124.4600, -118.6017, -124.2775, -124.46
## $ latitude_dd         <dbl> 38.07694, 41.48583, 37.37694, 43.67444, 33.65722, 45.83417, 46.31750, 43
## $ area_swept_ha       <dbl> 1.837650, 3.047135, 1.651004, 1.821803, 1.178600, 1.530574, 1.453994, 1.5
## $ subsample_count     <dbl> 0, 0, 0, 48, 0, 1, 6, 0, 16, 0, 1, 53, 0, 0, 0, 0, 0, 0, 0, 49, 0, 0, 2,
## $ subsample_wt_kg     <dbl> 0.00, 0.00, 0.00, 10.30, 0.00, 0.12, 0.80, 0.00, 1.95, 0.00, 0.02, 14.10
## $ total_catch_numbers <dbl> 0, 0, 0, 48, 0, 1, 6, 0, 16, 0, 1, 242, 0, 0, 0, 0, 0, 0, 0, 81, 0, 0, 2
## $ total_catch_wt_kg   <dbl> 0.00, 0.00, 0.00, 10.30, 0.00, 0.12, 0.80, 0.00, 1.95, 0.00, 0.02, 64.45
## $ cpue_kg_km2         <dbl> 0.000000, 0.000000, 0.000000, 565.373957, 0.000000, 7.840197, 55.020868,
## $ species             <chr> "greenstriped rockfish", "greenstriped rockfish", "greenstriped rockfish
## $ o2                  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
## $ temp                <dbl> 9.0359, 3.6300, 10.0280, 8.1160, 5.9237, 6.9375, 6.2750, 6.5258, 9.1859,
## $ sal                 <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
## $ depth               <dbl> 79.9095, 948.4094, 63.6213, 131.7826, 581.3504, 136.0367, 135.9776, 304.9
## $ performance         <chr> "Satisfactory", "Satisfactory", "Satisfactory", "Satisfactory", "Satisfac
## $ survey              <chr> "nwfsc.combo", "nwfsc.combo", "nwfsc.combo", "nwfsc.combo", "nwfsc.combo
## $ station             <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21
## $ roms_hindcast_day   <int> 8611, 8671, 8612, 9016, 9421, 9283, 9289, 9302, 9321, 9336, 9695, 9740,
```

```r
glimpse(roms)
```

```
## Rows: 4,832
## Columns: 13
## $ temp_roms           <dbl> 4.184411, 5.565145, 6.866655, 3.951781, 7.336621, 7.267605, 3.350144, 6.2
## $ oxygen_roms         <dbl> 44.78112, 78.12999, 137.91451, 41.38925, 168.28668, 166.14552, 28.76366,
## $ station             <int> 274, 4671, 4772, 4773, 4845, 120, 4672, 4774, 4775, 4783, 4825, 275, 4840
## $ roms_hindcast_day   <int> 8576, 8576, 8576, 8576, 8576, 8577, 8577, 8577, 8577, 8577, 8577, 8578, 8
## $ trawl_time          <dbl> 3265444800, 3265444800, 3265444800, 3265444800, 3265444800, 3265531200, 3
## $ lon_trawl           <dbl> -124.7761, -124.7389, -124.5156, -124.7325, -124.5428, -124.8156, -125.16
## $ lat_trawl           <dbl> 46.09611, 46.02472, 46.15667, 46.50389, 46.75500, 47.60194, 46.81083, 47
## $ depth_trawl         <dbl> 564.9317, 310.0056, 140.7280, 606.3237, 107.4615, 106.0853, 797.7154, 170
## $ temp_trawl          <dbl> 4.8597, 5.6246, 6.7312, 4.6011, 6.8820, 6.9012, 3.8907, 6.7184, 6.7229,
## $ oxygen_trawl        <dbl> NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN, Na
## $ mean_temp_roms_30   <dbl> 4.227865, 5.564605, 6.970570, 4.016127, 7.363900, 7.492323, 3.361807, 6.4
## $ mean_oxygen_roms_30 <dbl> 45.80141, 81.00598, 141.60814, 42.14824, 170.73278, 178.97508, 28.82358,
## $ date                <date> 2003-06-24, 2003-06-24, 2003-06-24, 2003-06-24, 2003-06-24, 2003-06-25,
```

```r
glimpse(substrate)
```

```
## Rows: 20,746
## Columns: 11
## $ TRAWL_ID                                              <dbl> 1.97706e+11, 1.97706e+11, 1.97706e+11, 1.9
## $ `Length_of_towline_outside_substrate_domain_(m)`      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
## $ `Length_of_towline_traversing_hard_substrate_(m)`     <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
## $ `Length_of_towline_traversing_mixed_substrate_(m)`    <dbl> 0.00, 0.00, 169.37, 0.00, 0.00, 0.00, 0.0
## $ `Length_of_towline_traversing_soft_substrate_(m)`     <dbl> 2628.42, 2484.34, 2899.37, 2570.00, 2639.
## $ `Total_length_of_towline_(m)`                         <dbl> 2628.42, 2484.34, 3068.74, 2570.00, 2639.
## $ Proportion_outside_substrate_domain                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
## $ Proportion_hard                                       <dbl> 0.0000000, 0.0000000, 0.0000000, 0.000000
## $ Proportion_mixed                                      <dbl> 0.00000000, 0.00000000, 0.05519243, 0.0000
## $ Proportion_soft                                       <dbl> 1.00000000, 1.00000000, 0.94480757, 1.0000
## $ prop_hard_mixed                                       <dbl> 0.00000000, 0.00000000, 0.05519243, 0.0000
```

For the ROMS data (for now), we are using modelled temperature and oxygen, lagged 30 days from each trawl survey location and time.

```r
roms_thin <- roms %>%
  dplyr::select(station,lon_trawl,lat_trawl,depth_trawl,mean_temp_roms_30,mean_oxygen_roms_30)
```

Join Datasets

Join Trawl and ROMS

Join the two datasets together, such that we have the appropriately-matched ROMS outputs

```r
trawl_roms <- trawl %>% left_join(roms,by = c("date", "station", "roms_hindcast_day")) %>%
  #clean up some columns
  dplyr::select(date,trawl_id,station,lon_trawl,lat_trawl,depth_trawl,mean_temp_roms_30,mean_oxygen_roms
  # drop any rows with NAs
  drop_na()
```

```r
  # test <- trawl %>% left_join(roms,by = c("date", "station", "roms_hindcast_day")) %>%
  #   #clean up some columns
  #   dplyr::select(date,trawl_id,station,lon_trawl,lat_trawl,depth_trawl,temp_roms,oxygen_roms,mean_te
  #   # drop any rows with NAs
  #   drop_na()
```

Join Trawl and Substrate

Join the substrate data by trawl ID number.

```
substrate_thin <- substrate %>%
  dplyr::select(TRAWL_ID,prop_hard_mixed)
trawl_roms <- trawl_roms %>%
  left_join(substrate_thin,by=c('trawl_id'="TRAWL_ID")) %>%
  drop_na()
```

Prepare Data for sdmTMB

Convert the trawl spatial data to UTM.

```
# convert to UTM
trawl_roms_utm <- trawl_roms %>%
  # convert to sf object
  st_as_sf(coords=c('lon_trawl','lat_trawl'),crs=4326) %>%
  # transform to UTM zone 10
  st_transform(crs = "+proj=utm +zone=10 +datum=WGS84 +units=km") %>%
  # add new coords as vars
  mutate(latitude = sf::st_coordinates(.)[,2],
         longitude = sf::st_coordinates(.)[,1]) %>%
  # convert back to normal df
  st_set_geometry(NULL)
```

We can save this version of the data so we do not have to run the join every time.

```
write_rds(trawl_roms_utm,here::here('data','trawl_roms_joined.rds'))
```

Functions to Run a Model

We'll write two functions, one to prepare a specific species for an sdmTMB model, and another to actually run an sdmTMB model with custom options

Prepare Species' Data

This function selects a species' data from the trawl survey data, converts the spatial data to UTM, does a couple of filters for missing data, and then joins the ROMS hindcast data to it by time and location.

```
prepare_species <- function(dat,spp){
  dat_sub <- dat %>%
    filter(species==spp) %>%

    # rescale depth, oxygen, and temp to be N(0,1)
    mutate(across(c(depth_trawl,mean_temp_roms_30,mean_oxygen_roms_30),~(scale(.) %>% as.vector()),.name

    # add a year indicator
    mutate(year=lubridate::year(date))
}
```

Try an example for sablefish

```
sablefish_dat <- prepare_species(trawl_roms_utm,spp="sablefish")
glimpse(sablefish_dat)
```

sdmTMB Model Function

Write a function that runs sdmTMB. It wil call the previous function to make the appropriate species data. For now, the environmental variable names are not generic (always `mean_temp_roms_30_norm` and `mean_oxygen_roms_30_norm`)

```
# nknots=400;use_depth=F;time_vary=F;spatial_field=T;hab_spline=F;env_spline=F;spline_k=3
# rm(time_varying,spatial_field,hab_spline,env_spline,spline_k,modeldat,spde,formula,substrate,enviro,d
run_sdmTMB <- function(dat,spp,nknots=400,use_depth=F,time_vary=F,spatial_field=T,hab_spline=F,env_splin
  # filter data for species
  modeldat <- prepare_species(dat,spp=spp)

  # make spde
  spde <- make_mesh(modeldat,xy_cols = c('longitude','latitude'),
                   cutoff = 20)

  # model formula
  formula <- paste0("cpue_kg_km2 ~ ")

  # substrate relationship
  substrate <- paste("prop_hard_mixed + I(prop_hard_mixed^2)")
  #wiggly habitat relationship?
  substrate <- ifelse(hab_spline, paste0("s(prop_hard_mixed,k=",spline_k,")"),
                      substrate)

  # make the environmental effects
  enviro <- paste("mean_temp_roms_30_norm +
                  I(mean_temp_roms_30_norm^2) +
                  mean_oxygen_roms_30_norm +
                  I(mean_oxygen_roms_30_norm^2)")
  # wiggly environmental relationships?
  enviro <- ifelse(env_spline, paste0("s(mean_temp_roms_30_norm,k=",spline_k,") + ",
                                      "s(mean_oxygen_roms_30_norm,k=",spline_k,")"),
                   enviro)
  # if depth effect, add to model formla
  if(use_depth) {
    formula = paste0(formula, " + depth + I(depth^2)")
  }

  time_formula = "~ -1"
  if(time_vary) {
    time_formula = paste0(time_formula, " + ", substrate, " + ", enviro)
    time_varying = as.formula(time_formula)
    time = "year"
  } else {
    formula = paste0(formula, " + ", substrate, " + ", enviro)
    time_varying = NULL
    time = "year"
  }
  # fit model. EW commented out quadratic roots, since those are still experimental and won't work for
  # set.seed(41389) # for reproducibility
  # test_set = sample(1:nrow(modeldat), size = round(0.1*nrow(modeldat)), replace=FALSE)
  # modeldat$fold = 1
  # modeldat$fold[test_set] = 2
  # anisotropy off for now
  print('running model.')
  m <- try( sdmTMB(
    formula = as.formula(formula),
    time_varying = time_varying,
    spde = spde,
    time = time,
    family = tweedie(link = "log"),
    data = modeldat,
    anisotropy = FALSE,
    spatial_only = T,
    #extra_time argument necessary for prediction?
    extra_time=1980:2100,
    control=sdmTMBcontrol(map_rf=ifelse(spatial_field,F,T))
  ),
  silent=F)


  # predicted values for the 2nd fold (test)
  # m_cv$data$cv_predicted[which(m_cv$data$cv_fold==2)]
  # log likelihood values for the 2nd fold (test)
  # m_cv$data$cv_loglik[which(m_cv$data$cv_fold==2)]

    # sum(m_cv$data$cv_loglik[which(m_cv$data$cv_fold==2)])

  # if(class(m)!="try-error") {
  #   write_rds(m, file=here::here('model output',
  #                                paste0(spp,'.rds')))
  # }
  if(class(m)=="try-error"){
    print(paste("Error."))
  }else{
    print(paste("Model for",spp,"complete."))
  }

  # return(m)
  return(m)

}
```

```
test <- run_sdmTMB(dat=trawl_roms_utm,spp="sablefish",hab_spline = F,env_spline = F)
```

##CV formula

```
run_sdmTMB_cv <- function(dat,spp,nknots=400,use_depth=F,time_vary=F,spatial_field=T,hab_spline=F,env_s
  # filter data for species
  modeldat <- prepare_species(dat,spp=spp)
```

```r
# make spde
spde <- make_mesh(modeldat,xy_cols = c('longitude','latitude'),
                 cutoff = 20)

# model formula
formula <- paste0("cpue_kg_km2 ~ ")

# substrate relationship
substrate <- paste("prop_hard_mixed + I(prop_hard_mixed^2)")
#wiggly habitat relationship?
substrate <- ifelse(hab_spline, paste0("s(prop_hard_mixed,k=",spline_k,")"),
                    substrate)

# make the environmental effects
enviro <- paste("mean_temp_roms_30_norm + I(mean_temp_roms_30_norm^2) + mean_oxygen_roms_30_norm + I(
# wiggly environmental relationships?
enviro <- ifelse(env_spline, paste0("s(mean_temp_roms_30_norm,k=",spline_k,") + ",
                                    "s(mean_oxygen_roms_30_norm,k=",spline_k,")"),
                 enviro)
# if depth effect, add to model formla
if(use_depth) {
  formula = paste0(formula, " + depth + I(depth^2)")
}

time_formula = "~ -1"

if(time_vary) {
  time_formula = paste0(time_formula, " + ", substrate, " + ", enviro)
  time_varying = as.formula(time_formula)
  time = "year"
} else {
  formula = paste0(formula, " + ", substrate, " + ", enviro)
  time_varying = NULL
  time = "year"
}

# fit model. EW commented out quadratic roots, since those are still experimental and won't work for
set.seed(41389) # for reproducibility
test_set = sample(1:nrow(modeldat), size = round(0.1*nrow(modeldat)), replace=FALSE)
modeldat$fold = 1
modeldat$fold[test_set] = 2

print('running 2-fold CV.')

m_cv <- try( sdmTMB_cv(
  formula = as.formula(formula),
  k_folds=2,
  parallel = FALSE,
  fold_ids = modeldat$fold,
  time_varying = time_varying,
  spde = spde,
  time = time,
  family = tweedie(link = "log"),
    data = modeldat,
    anisotropy = FALSE,
    spatial_only = T,
    #extra_time argument necessary for prediction?
    # extra_time=1980:2100,
    control=sdmTMBcontrol(map_rf=ifelse(spatial_field,F,T))
  ),
  silent=T)
  if(class(m_cv)=='try-error'){
    print(paste('Error.'))
  } else{
    # tem <- m_cv %>% pluck('data')
    # print(paste('data is class',class(tem)))
    total_pred_ll = m_cv %>%
      pluck('data') %>%
      dplyr::filter(cv_fold==2) %>%
      pluck('cv_loglik') %>%
      sum()
    if(return_what=='model') return(m_cv)
    else return(total_pred_ll)
  }
}
```

```
test_cv <- run_sdmTMB_cv(dat=trawl_roms_utm,spp='sablefish',return_what = 'model')
```

Cross Validation options

If we wanted to use cross validation, we could do that in a couple ways. For example, with a single train/test split, we could assign 10% of the observations to the test set and not fit the model for all the folds (fitting to all folds is the default in sdmTMB_cv).

```
set.seed(41389) # for reproducibility
test_set = sample(1:nrow(modeldat), size = round(0.1*nrow(modeldat)), replace=FALSE)
modeldat$fold = 1
modeldat$fold[test_set] = 2
head(modeldat$fold,25)
```

Alternatively we could do something like assign all points for a given year (e.g. 2018) to the test set.

```
# modeldat$fold = ifelse(modeldat$year=="2018",2,1)
```

A third option is to use blockCV to assign the folds. If you have raster data, there's a few functions in that package (spatialAutoRange, rangeExplorer) to estimate the range – but because those are probably difficult to estimate with the kind of data we have, I've generally used ranges in the 50-75km, which is about what's estimated for many WCBTS species.

```
the_data <- sf::st_as_sf(modeldat, coords = c("longitude", "latitude"),crs="+proj=utm +zone=10 +datum=W
sb <- spatialBlock(
    speciesData = the_data,
    species = "xxxxx",
    theRange = 5000, # range should be in meters
    # k = 10,
    selection = "systematic",
    showBlocks = FALSE
  )
modeldat$fold = ifelse(sb$fold==1,2,1)
```

And then we can use sdmTMB_cv to do the estimation for each fold. Because we did the test-train split, we'll fit the model 2x, but just be interested in the 1st fit.

```
m_cv <- try( sdmTMB_cv(
    formula = as.formula(formula),
    # time_varying = time_varying,
    spde = spde,
    k_folds = 2,
    fold_ids = "fold",
    time = 'year',
    family = tweedie(link = "log"),
    data = modeldat,
    anisotropy = FALSE,
    spatial_only = T,
    #extra_time argument necessary for prediction?
    extra_time=1980:2100,
    map_rf=T
    # map_rf=ifelse(spatial_field,F,T)
  ),
  silent=TRUE)

# predicted values for the 2nd fold (test)
m_cv$data$cv_predicted[which(m_cv$data$cv_fold==2)]
# log likelihood values for the 2nd fold (test)
m_cv$data$cv_loglik[which(m_cv$data$cv_fold==2)]

total_pred_ll = sum(m_cv$data$cv_loglik[which(m_cv$data$cv_fold==2)])
```

This is the end of this script. Moving the actual modelling (i.e., the calling of this function) to a new script.