Covariate Data for Scripps's Murrelet Egg Size Model

Amelia J. DuVall & Marcela Todd Zaragoza

2021

This is v.2021-12-30

Introduction

This document details steps taken to compile and clean covariate data for a linear mixed model on Scripps's Murrelet (*Synthliboramphus scrippsi*) egg size at Santa Barbara Island within Channel Islands National Park from 2009-2017. Oceanographic indices were pulled from NOAA'S California Current Integrated Ecosystem Assessment Program (CCIEA); data descriptions are provided by CCIEA. Covariates are grouped by spatial scale: local, regional, and large-scale. We also tested for correlation among predictors. Given the differences in spatial and temporal extent of oceanographic variables and their impacts on seabirds, we tested environmental covariates for the Scripps's Murrelet egg size model (see SCMU_model.Rmd) under different scenarios, depending on the availability of the data:

1) Monthly averages for January to June in year t (6 months) to encompass pre-breeding and breeding season ("half").

2) Monthly averages for July in year $t-1$ to June in year t (12 months) to encompass the entire post-breeding, pre-breeding, and breeding season ("full").

3) Monthly averages for January to June in year $t-1$ (6 months) to encompass pre-breeding and breeding season in the previous year ("half_lag").

4) Monthly averages for July in year $t-2$ to June in year $t-1$ (12 months) to encompass the entire post-breeding, pre-breeding, and breeding season in the previous year ("full_lag").

We tested 6 covariates:

1) Larval Anchovy (ANCHL)

2) Biologically Effective Upwelling Transport Index (BEUTI)

3) Sea Surface Temperature (SST)

4) North Pacific Gyre Oscillation Index (NPGO)

5) Pacific Decadal Oscillation Index (PDO)

6) Oceanic Nino Index (ONI)

```
knitr::opts_chunk$set(echo = TRUE)
```

```r
## load libraries
library(here)
library(tidyverse)
library(janitor)
library(ggplot2)
library(lubridate)
library(viridis)
library(Hmisc)
library(stats)
library(faraway)
library(sjPlot)
library(corrplot)

## load egg size data
egg <- read.csv(here("data", "SCMU_egg_data.csv"))%>%
  filter(TrueOrder == TRUE) %>% # select eggs with order known only
  dplyr::select(Year, Observer, Plot, Size, EggOrder)
```

Oceanographic Indices

Local Indices

Sea Surface Temperature

Sea surface temperature was provided by NOAA buoy station 46025. The data description can be found here. We tested this variable under 4 scenarios.

```r
SSTraw <- read_csv(here("data", "covariates", "cciea_OC_SST3_91cf_d165_213f-46025.csv"))

## are there NAs in the data?
sum(is.na(SSTraw$SST))
```

```
## [1] 0
```

```r
## Scenario 1: "half" dataset from Jan-June in time t
SST_half <- SSTraw %>%
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(sst = as.numeric(SST)) %>%
  filter(Month == 1 | Month == 2 | Month == 3 |Month == 4 | Month == 5 | Month == 6) %>%
  group_by(Year) %>%
  summarise(SST = mean(sst, na.rm = TRUE)) %>%
  dplyr::select(Year, SST) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(SST_half = scale(SST)) %>%
  arrange(Year) %>%
  dplyr::select(Year, SST_half)

## Scenario 2: "full" dataset from July in time t-1 to June in time t
SST_full <- SSTraw %>%
  slice(-1) %>%
```

```r
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(sst = as.numeric(SST),
         split_year = ifelse(Month == 7 | Month == 8 | Month == 9 | Month == 10 |
                               Month == 11 | Month == 12,
                             Year + 1, Year)) %>%
  rename(Year = split_year, true_year = Year) %>%
  group_by(Year) %>%
  summarise(SST = mean(sst, na.rm = TRUE)) %>%
  dplyr::select(Year, SST) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(SST_full = scale(SST)) %>%
  arrange(Year) %>%
  dplyr::select(Year, SST_full)

## Scenario 3: "half_lag" dataset from Jan-June in time t-1
SST_half_lag <- SSTraw %>%
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(sst = as.numeric(SST)) %>%
  filter(Month == 1 | Month == 2 | Month == 3 |Month == 4 | Month == 5 |
           Month == 6) %>%
  mutate(Year = Year + 1) %>%
  group_by(Year) %>%
  summarise(SST = mean(sst, na.rm = TRUE)) %>%
  dplyr::select(Year, SST) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(SST_half_lag = scale(SST)) %>%
  arrange(Year) %>%
  dplyr::select(Year, SST_half_lag)

## Scenario 4: "full_lag" dataset from July in time t-2 to June in time t-1
SST_full_lag <- SSTraw %>%
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(sst = as.numeric(SST),
         split_year = ifelse(Month == 7 | Month == 8 | Month == 9 | Month == 10 |
                               Month == 11 | Month == 12,
                             Year + 1, Year)) %>%
  rename(Year = split_year, true_year = Year) %>%
  mutate(Year = Year + 1) %>%
  group_by(Year) %>%
  summarise(SST = mean(sst, na.rm = TRUE)) %>%
  dplyr::select(Year, SST) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(SST_full_lag = scale(SST)) %>%
  arrange(Year) %>%
  dplyr::select(Year, SST_full_lag)
```

```r
## join datasets together
SST1 <- full_join(SST_half, SST_full, by = "Year")
SST2 <- full_join(SST1, SST_half_lag, by = "Year")
SST3 <- full_join(SST2, SST_full_lag, by = "Year")

## join with egg size data
SST_df <- left_join(egg, SST3, by = "Year")

## run models
SST_half_mod <- lm(Size ~ SST_half, data = SST_df)
SST_full_mod <- lm(Size ~ SST_full, data = SST_df)
SST_half_lag_mod <- lm(Size ~ SST_half_lag, data = SST_df)
SST_full_lag_mod <- lm(Size ~ SST_full_lag, data = SST_df)

## model selection table
SST_AIC <- matrix(NA, nrow = 4, ncol = 3) # 4 rows for 6 top models
SST_AIC[1,1] <- AIC(SST_half_mod)
SST_AIC[2,1] <- AIC(SST_full_mod)
SST_AIC[3,1] <- AIC(SST_half_lag_mod)
SST_AIC[4,1] <- AIC(SST_full_lag_mod)
SST_AIC[,2] <- SST_AIC[,1] - min(SST_AIC[,1]) # calculate delta AIC
SST_AIC[,3] <- exp(-0.5*SST_AIC[,2])/
  (sum(exp(-0.5*SST_AIC[,2]))) # calculate model weights
colnames(SST_AIC) <- c("AIC", "deltaAIC", "model_weights")
rownames(SST_AIC) <- c("half", "full", "half_lag", "full_lag")
print(SST_AIC)
```

```
##               AIC deltaAIC model_weights
## half     10724.84 0.000000    0.66899301
## full     10729.84 5.000953    0.05488812
## half_lag 10727.50 2.653379    0.17752012
## full_lag 10728.67 3.829430    0.09859875
```

The half SST model has more support.

Regional Indices

Larval Anchovy (ANCHL)

Derived from spring California Cooperative Oceanic Fisheries Investigations (CalCOFI) surveys. Larval fish data summed across all stations of the CalCOFI survey in spring (units are in number under 10 sq. m of surface area; ln(abundance+1); CalCOFI lines 76.7 - 93.3; stations 28.0 - 120.0). Sampling data is only available at a yearly sampling interval. Therefore, we tested this variable under only two scenarios: 1) yearly value in time t (full), and 2) yearly value in time $t-1$ (full_lag).

```r
## load data
ANCHLraw  <- read.csv(here("data", "covariates", "cciea_EI_FBS_2020_a29e_1fd0_409d.csv"),
                      na.strings = "NaN")

## are there NAs in the data?
sum(is.na(ANCHLraw$relative_abundance))
```

```
## [1] 0
```

```r
## Scenario 1: create "full" dataset from 2009-2017 for values in year t
ANCHL_full <- ANCHLraw %>%
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time)) %>%
  mutate(ANCHL = as.numeric(relative_abundance)) %>%
  arrange(Year) %>%
  dplyr::select(Year, ANCHL) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(ANCHL_full = scale(ANCHL)) %>%
  dplyr::select(Year, ANCHL_full)

## Scenario 2: create "full_lag" dataset from 2008-2016 for values in year t-1
ANCHL_full_lag <- ANCHLraw %>%
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time)) %>%
  mutate(ANCHL = as.numeric(relative_abundance)) %>%
  arrange(Year) %>%
  dplyr::select(Year, ANCHL) %>%
  filter(Year >= 2008 & Year <= 2016) %>%
  mutate(TrueYear = Year,
         Year = TrueYear + 1,
         ANCHL_full_lag = scale(ANCHL)) %>%
  dplyr::select(Year, ANCHL_full_lag)

## join full and full_lag datasets
ANCHL1 <- full_join(ANCHL_full, ANCHL_full_lag, by = "Year")

## join with egg size data
ANCHLdf1 <- left_join(egg, ANCHL1, by = "Year")

## run models
ANCHL_full_mod <- lm(Size ~ ANCHL_full , data = ANCHLdf1)
ANCHL_full_lag_mod <- lm(Size ~ ANCHL_full_lag, data = ANCHLdf1)

## model selection table
ANHCL_AIC <- matrix(NA, nrow = 2, ncol = 3)
ANHCL_AIC[1,1] <- AIC(ANCHL_full_mod)
ANHCL_AIC[2,1] <- AIC(ANCHL_full_lag_mod)
ANHCL_AIC[,2] <- ANHCL_AIC[,1] - min(ANHCL_AIC[,1]) # calculate delta AIC
ANHCL_AIC[,3] <- exp(-0.5*ANHCL_AIC[,2])/
  (sum(exp(-0.5*ANHCL_AIC[,2]))) # calculate model weights
colnames(ANHCL_AIC) <- c("AIC", "deltaAIC", "model_weights")
rownames(ANHCL_AIC) <- c("full", "full_lag")
print(ANHCL_AIC)
```

```
##               AIC deltaAIC model_weights
## full     10719.99 0.000000    0.91359752
## full_lag 10724.71 4.716747    0.08640248
```

The ANCHL full model has more support.

Biologically Effective Upwelling Transport Index (BEUTI)

Summary: BEUTI is a new upwelling index that leverages state-of-the-art ocean models as well as satellite and in situ data to improve upon historically available upwelling indices for the U.S. west coast. BEUTI provides estimates of vertical nitrate flux near the coast (i.e., the amount of nitrate upwelled/downwelled), which may be more relevant than upwelling strength when considering some biological responses. See Jacox, M. G., C. A. Edwards, E. L. Hazen, and S. J. Bograd (2018) Coastal upwelling revisited: Ekman, Bakun, and improved upwelling indices for the U.S. west coast. Journal of Geophysical Research, doi:10.1029/2018JC014187. We tested this variable under 4 scenarios.

```
BEUTIraw <- read_csv(here("data", "covariates", "cciea_OC_BEUTI_784c_ef9f_af6f.csv"))

## are there NAs in the data?
sum(is.na(BEUTIraw$beuti))
```

```
## [1] 0
```

```
## Scenario 1: "half" dataset from Jan-June in time t
BEUTI_half <- BEUTIraw %>%
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(beuti = as.numeric(beuti)) %>%
  filter(Month == 1 | Month == 2 | Month == 3 |Month == 4 | Month == 5 |
           Month == 6) %>%
  group_by(Year) %>%
  summarise(BEUTI = mean(beuti, na.rm = TRUE)) %>%
  dplyr::select(Year, BEUTI) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(BEUTI_half = scale(BEUTI)) %>%
  arrange(Year) %>%
  dplyr::select(Year, BEUTI_half)

## Scenario 2: "full" dataset from July in time t-1 to June in time t
BEUTI_full <- BEUTIraw %>%
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(beuti = as.numeric(beuti),
         split_year = ifelse(Month == 7 | Month == 8 | Month == 9 | Month == 10 |
                               Month == 11 | Month == 12,
                             Year + 1, Year)) %>%
  rename(Year = split_year, true_year = Year) %>%
  group_by(Year) %>%
  summarise(BEUTI = mean(beuti, na.rm = TRUE)) %>%
  dplyr::select(Year, BEUTI) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(BEUTI_full = scale(BEUTI)) %>%
  arrange(Year) %>%
  dplyr::select(Year, BEUTI_full)

## Scenario 3: "half_lag" dataset from Jan-June in time t-1
```

```r
BEUTI_half_lag <- BEUTIraw %>%
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(beuti = as.numeric(beuti)) %>%
  filter(Month == 1 | Month == 2 | Month == 3 |Month == 4 | Month == 5 |
           Month == 6) %>%
  mutate(Year = Year + 1) %>%
  group_by(Year) %>%
  summarise(BEUTI = mean(beuti, na.rm = TRUE)) %>%
  dplyr::select(Year, BEUTI) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(BEUTI_half_lag = scale(BEUTI)) %>%
  arrange(Year) %>%
  dplyr::select(Year, BEUTI_half_lag)

## Scenario 4: "full_lag" dataset from July in time t-2 to June in time t-1
BEUTI_full_lag <- BEUTIraw %>%
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(beuti = as.numeric(beuti),
         split_year = ifelse(Month == 7 | Month == 8 | Month == 9 | Month == 10 |
                               Month == 11 | Month == 12,
                             Year + 1, Year)) %>%
  rename(Year = split_year, true_year = Year) %>%
  mutate(Year = Year + 1) %>%
  group_by(Year) %>%
  summarise(BEUTI = mean(beuti, na.rm = TRUE)) %>%
  dplyr::select(Year, BEUTI) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(BEUTI_full_lag = scale(BEUTI)) %>%
  arrange(Year) %>%
  dplyr::select(Year, BEUTI_full_lag)

## join datasets together
BEUTI1 <- full_join(BEUTI_half, BEUTI_full, by = "Year")
BEUTI2 <- full_join(BEUTI1, BEUTI_half_lag, by = "Year")
BEUTI3 <- full_join(BEUTI2, BEUTI_full_lag, by = "Year")

## join with egg size data
BEUTI_df <- left_join(egg, BEUTI3, by = "Year")

## run models
BEUTI_half_mod <- lm(Size ~ BEUTI_half, data = BEUTI_df)
BEUTI_full_mod <- lm(Size ~ BEUTI_full, data = BEUTI_df)
BEUTI_half_lag_mod <- lm(Size ~ BEUTI_half_lag, data = BEUTI_df)
BEUTI_full_lag_mod <- lm(Size ~ BEUTI_full_lag, data = BEUTI_df)

## model selection table
BEUTI_AIC <- matrix(NA, nrow = 4, ncol = 3) # 4 rows for 6 top models
```

```r
BEUTI_AIC[1,1] <- AIC(BEUTI_half_mod)
BEUTI_AIC[2,1] <- AIC(BEUTI_full_mod)
BEUTI_AIC[3,1] <- AIC(BEUTI_half_lag_mod)
BEUTI_AIC[4,1] <- AIC(BEUTI_full_lag_mod)
BEUTI_AIC[,2] <- BEUTI_AIC[,1] - min(BEUTI_AIC[,1]) # calculate delta AIC
BEUTI_AIC[,3] <- exp(-0.5*BEUTI_AIC[,2])/
  (sum(exp(-0.5*BEUTI_AIC[,2]))) # calculate model weights
colnames(BEUTI_AIC) <- c("AIC", "deltaAIC", "model_weights")
rownames(BEUTI_AIC) <- c("half", "full", "half_lag", "full_lag")
print(BEUTI_AIC)
```

```
##               AIC  deltaAIC model_weights
## half     10733.29 11.613474   0.002660237
## full     10731.98 10.297204   0.005137416
## half_lag 10725.89  4.213694   0.107587579
## full_lag 10721.68  0.000000   0.884614768
```

The BEUTI full_lag model has more support.

Large-scale Indices

North Pacific Gyre Oscillation Index (NPGO)

Summary: The NPGO is calculated from an Empirical Orthogonal Function analysis of sea-surface height in the Northeast Pacific. The NPGO is the second dominant mode. We tested this variable under 4 scenarios.

```r
NPGOraw <- read_csv(here("data", "covariates", "cciea_OC_NPGO_712b_5843_9069.csv"))[-1,]

## are there NAs in the data?
sum(is.na(NPGOraw$NPGO))
```

```
## [1] 0
```

```r
## Scenario 1: "half" dataset from Jan-June in time t
NPGO_half <- NPGOraw %>%
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(npgo = as.numeric(NPGO)) %>%
  filter(Month == 1 | Month == 2 | Month == 3 |Month == 4 | Month == 5 |
           Month == 6) %>%
  group_by(Year) %>%
  summarise(NPGO = mean(npgo, na.rm = TRUE)) %>%
  dplyr::select(Year, NPGO) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(NPGO_half = scale(NPGO)) %>%
  arrange(Year) %>%
  dplyr::select(Year, NPGO_half)

## Scenario 2: "full" dataset from July in time t-1 to June in time t
NPGO_full <- NPGOraw %>%
```

```
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(npgo = as.numeric(NPGO),
         split_year = ifelse(Month == 7 | Month == 8 | Month == 9 | Month == 10 |
                               Month == 11 | Month == 12,
                             Year + 1, Year)) %>%
  rename(Year = split_year, true_year = Year) %>%
  group_by(Year) %>%
  summarise(NPGO = mean(npgo, na.rm = TRUE)) %>%
  dplyr::select(Year, NPGO) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(NPGO_full = scale(NPGO)) %>%
  arrange(Year) %>%
  dplyr::select(Year, NPGO_full)

## Scenario 3: "half_lag" dataset from Jan-June in time t-1
NPGO_half_lag <- NPGOraw %>%
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(npgo = as.numeric(NPGO)) %>%
  filter(Month == 1 | Month == 2 | Month == 3 |Month == 4 | Month == 5 |
           Month == 6) %>%
  mutate(Year = Year + 1) %>%
  group_by(Year) %>%
  summarise(NPGO = mean(npgo, na.rm = TRUE)) %>%
  dplyr::select(Year, NPGO) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(NPGO_half_lag = scale(NPGO)) %>%
  arrange(Year) %>%
  dplyr::select(Year, NPGO_half_lag)

## Scenario 4: "full_lag" dataset from July in time t-2 to June in time t-1
NPGO_full_lag <- NPGOraw %>%
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(npgo = as.numeric(NPGO),
         split_year = ifelse(Month == 7 | Month == 8 | Month == 9 | Month == 10 |
                               Month == 11 | Month == 12,
                             Year + 1, Year)) %>%
  rename(Year = split_year, true_year = Year) %>%
  mutate(Year = Year + 1) %>%
  group_by(Year) %>%
  summarise(NPGO = mean(npgo, na.rm = TRUE)) %>%
  dplyr::select(Year, NPGO) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(NPGO_full_lag = scale(NPGO)) %>%
  arrange(Year) %>%
```

```
  dplyr::select(Year, NPGO_full_lag)

## join datasets together
NPGO1 <- full_join(NPGO_half, NPGO_full, by = "Year")
NPGO2 <- full_join(NPGO1, NPGO_half_lag, by = "Year")
NPGO3 <- full_join(NPGO2, NPGO_full_lag, by = "Year")

## join with egg size data
NPGO_df <- left_join(egg, NPGO3, by = "Year")

## run models
NPGO_half_mod <- lm(Size ~ NPGO_half, data = NPGO_df)
NPGO_full_mod <- lm(Size ~ NPGO_full, data = NPGO_df)
NPGO_half_lag_mod <- lm(Size ~ NPGO_half_lag, data = NPGO_df)
NPGO_full_lag_mod <- lm(Size ~ NPGO_full_lag, data = NPGO_df)

## model selection table
NPGO_AIC <- matrix(NA, nrow = 4, ncol = 3) # 4 rows for 6 top models
NPGO_AIC[1,1] <- AIC(NPGO_half_mod)
NPGO_AIC[2,1] <- AIC(NPGO_full_mod)
NPGO_AIC[3,1] <- AIC(NPGO_half_lag_mod)
NPGO_AIC[4,1] <- AIC(NPGO_full_lag_mod)
NPGO_AIC[,2] <- NPGO_AIC[,1] - min(NPGO_AIC[,1]) # calculate delta AIC
NPGO_AIC[,3] <- exp(-0.5*NPGO_AIC[,2])/
  (sum(exp(-0.5*NPGO_AIC[,2]))) # calculate model weights
colnames(NPGO_AIC) <- c("AIC", "deltaAIC", "model_weights")
rownames(NPGO_AIC) <- c("half", "full", "half_lag", "full_lag")
print(NPGO_AIC)
```

```
##               AIC  deltaAIC model_weights
## half     10710.26  0.000000  9.696585e-01
## full     10718.03  7.770044  1.992394e-02
## half_lag 10731.24 20.983985  2.691562e-05
## full_lag 10719.33  9.072072  1.039066e-02
```

The half NPGO model has the most support.

Pacific Decadal Oscillation Index (PDO)

Summary: The PDO is calculated from an Empirical Orthogonal analysis of sea surface temperature anomalies in the North Pacific. The PDO is the first dominant mode. We tested this variable under 4 scenarios.

```
PDOraw <- read_csv(here("data", "covariates", "cciea_OC_PDO_712b_5843_9069.csv"))[-1,]

## are there NAs in the data?
sum(is.na(PDOraw$PDO))
```

```
## [1] 0
```

```
## Scenario 1: "half" dataset from Jan-June in time t
PDO_half <- PDOraw %>%
  slice(-1) %>%
```

```r
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(pdo = as.numeric(PDO)) %>%
  filter(Month == 1 | Month == 2 | Month == 3 |Month == 4 | Month == 5 |
           Month == 6) %>%
  group_by(Year) %>%
  summarise(PDO = mean(pdo, na.rm = TRUE)) %>%
  dplyr::select(Year, PDO) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(PDO_half = scale(PDO)) %>%
  arrange(Year) %>%
  dplyr::select(Year, PDO_half)

## Scenario 2: "full" dataset from July in time t-1 to June in time t
PDO_full <- PDOraw %>%
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(pdo = as.numeric(PDO),
         split_year = ifelse(Month == 7 | Month == 8 | Month == 9 | Month == 10 |
                               Month == 11 | Month == 12,
                             Year + 1, Year)) %>%
  rename(Year = split_year, true_year = Year) %>%
  group_by(Year) %>%
  summarise(PDO = mean(pdo, na.rm = TRUE)) %>%
  dplyr::select(Year, PDO) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(PDO_full = scale(PDO)) %>%
  arrange(Year) %>%
  dplyr::select(Year, PDO_full)

## Scenario 3: "half_lag" dataset from Jan-June in time t-1
PDO_half_lag <- PDOraw %>%
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(pdo = as.numeric(PDO)) %>%
  filter(Month == 1 | Month == 2 | Month == 3 |Month == 4 | Month == 5 |
           Month == 6) %>%
  mutate(Year = Year + 1) %>%
  group_by(Year) %>%
  summarise(PDO = mean(pdo, na.rm = TRUE)) %>%
  dplyr::select(Year, PDO) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(PDO_half_lag = scale(PDO)) %>%
  arrange(Year) %>%
  dplyr::select(Year, PDO_half_lag)

## Scenario 4: "full_lag" dataset from July in time t-2 to June in time t-1
PDO_full_lag <- PDOraw %>%
```

```
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(pdo = as.numeric(PDO),
         split_year = ifelse(Month == 7 | Month == 8 | Month == 9 | Month == 10 |
                               Month == 11 | Month == 12,
                             Year + 1, Year)) %>%
  rename(Year = split_year, true_year = Year) %>%
  mutate(Year = Year + 1) %>%
  group_by(Year) %>%
  summarise(PDO = mean(pdo, na.rm = TRUE)) %>%
  dplyr::select(Year, PDO) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(PDO_full_lag = scale(PDO)) %>%
  arrange(Year) %>%
  dplyr::select(Year, PDO_full_lag)

## join datasets together
PDO1 <- full_join(PDO_half, PDO_full, by = "Year")
PDO2 <- full_join(PDO1, PDO_half_lag, by = "Year")
PDO3 <- full_join(PDO2, PDO_full_lag, by = "Year")

## join with egg size data
PDO_df <- left_join(egg, PDO3, by = "Year")

## run models
PDO_half_mod <- lm(Size ~ PDO_half, data = PDO_df)
PDO_full_mod <- lm(Size ~ PDO_full, data = PDO_df)
PDO_half_lag_mod <- lm(Size ~ PDO_half_lag, data = PDO_df)
PDO_full_lag_mod <- lm(Size ~ PDO_full_lag, data = PDO_df)

## model selection table
PDO_AIC <- matrix(NA, nrow = 4, ncol = 3) # 4 rows for 6 top models
PDO_AIC[1,1] <- AIC(PDO_half_mod)
PDO_AIC[2,1] <- AIC(PDO_full_mod)
PDO_AIC[3,1] <- AIC(PDO_half_lag_mod)
PDO_AIC[4,1] <- AIC(PDO_full_lag_mod)
PDO_AIC[,2] <- PDO_AIC[,1] - min(PDO_AIC[,1]) # calculate delta AIC
PDO_AIC[,3] <- exp(-0.5*PDO_AIC[,2])/
  (sum(exp(-0.5*PDO_AIC[,2]))) # calculate model weights
colnames(PDO_AIC) <- c("AIC", "deltaAIC", "model_weights")
rownames(PDO_AIC) <- c("half", "full", "half_lag", "full_lag")
print(PDO_AIC)
```

```
##               AIC  deltaAIC model_weights
## half     10729.46 19.158830  6.352108e-05
## full     10728.48 18.178855  1.036852e-04
## half_lag 10710.30  0.000000  9.187661e-01
## full_lag 10715.15  4.855518  8.106673e-02
```

The PDO half_lag has more support.

Oceanic Nino Index (ONI)

Summary: The ONI is the 3 month running mean of sea surface temperature anomalies in the Nino 3.4 region. We tested this variable under 4 scenarios.

```
ONIraw <- read_csv(here("data", "covariates", "cciea_OC_ONI_712b_5843_9069.csv"))[-1,]

## are there NAs in the data?
sum(is.na(ONIraw$ONI))
```

```
## [1] 0
```

```
## Scenario 1: "half" dataset from Jan-June in time t
ONI_half <- ONIraw %>%
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(oni = as.numeric(ONI)) %>%
  filter(Month == 1 | Month == 2 | Month == 3 |Month == 4 | Month == 5 |
           Month == 6) %>%
  group_by(Year) %>%
  summarise(ONI = mean(oni, na.rm = TRUE)) %>%
  dplyr::select(Year, ONI) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(ONI_half = scale(ONI)) %>%
  arrange(Year) %>%
  dplyr::select(Year, ONI_half)

## Scenario 2: "full" dataset from July in time t-1 to June in time t
ONI_full <- ONIraw %>%
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(oni = as.numeric(ONI),
         split_year = ifelse(Month == 7 | Month == 8 | Month == 9 | Month == 10 |
                               Month == 11 | Month == 12,
                             Year + 1, Year)) %>%
  rename(Year = split_year, true_year = Year) %>%
  group_by(Year) %>%
  summarise(ONI = mean(oni, na.rm = TRUE)) %>%
  dplyr::select(Year, ONI) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(ONI_full = scale(ONI)) %>%
  arrange(Year) %>%
  dplyr::select(Year, ONI_full)

## Scenario 3: "half_lag" dataset from Jan-June in time t-1
ONI_half_lag <- ONIraw %>%
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
```

```r
  mutate(oni = as.numeric(ONI)) %>%
  filter(Month == 1 | Month == 2 | Month == 3 |Month == 4 | Month == 5 |
           Month == 6) %>%
  mutate(Year = Year + 1) %>%
  group_by(Year) %>%
  summarise(ONI = mean(oni, na.rm = TRUE)) %>%
  dplyr::select(Year, ONI) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(ONI_half_lag = scale(ONI)) %>%
  arrange(Year) %>%
  dplyr::select(Year, ONI_half_lag)

## Scenario 4: "full_lag" dataset from July in time t-2 to June in time t-1
ONI_full_lag <- ONIraw %>%
  slice(-1) %>%
  mutate(time = ymd_hms(time)) %>%
  mutate(Year = year(time),
         Month = month(time)) %>%
  mutate(oni = as.numeric(ONI),
         split_year = ifelse(Month == 7 | Month == 8 | Month == 9 | Month == 10 |
                               Month == 11 | Month == 12,
                             Year + 1, Year)) %>%
  rename(Year = split_year, true_year = Year) %>%
  mutate(Year = Year + 1) %>%
  group_by(Year) %>%
  summarise(ONI = mean(oni, na.rm = TRUE)) %>%
  dplyr::select(Year, ONI) %>%
  filter(Year >= 2009 & Year <= 2017) %>%
  mutate(ONI_full_lag = scale(ONI)) %>%
  arrange(Year) %>%
  dplyr::select(Year, ONI_full_lag)

## join datasets together
ONI1 <- full_join(ONI_half, ONI_full, by = "Year")
ONI2 <- full_join(ONI1, ONI_half_lag, by = "Year")
ONI3 <- full_join(ONI2, ONI_full_lag, by = "Year")

## join with egg size data
ONI_df <- left_join(egg, ONI3, by = "Year")

## run models
ONI_half_mod <- lm(Size ~ ONI_half, data = ONI_df)
ONI_full_mod <- lm(Size ~ ONI_full, data = ONI_df)
ONI_half_lag_mod <- lm(Size ~ ONI_half_lag, data = ONI_df)
ONI_full_lag_mod <- lm(Size ~ ONI_full_lag, data = ONI_df)

## model selection table
ONI_AIC <- matrix(NA, nrow = 4, ncol = 3) # 4 rows for 6 top models
ONI_AIC[1,1] <- AIC(ONI_half_mod)
ONI_AIC[2,1] <- AIC(ONI_full_mod)
ONI_AIC[3,1] <- AIC(ONI_half_lag_mod)
ONI_AIC[4,1] <- AIC(ONI_full_lag_mod)
ONI_AIC[,2] <- ONI_AIC[,1] - min(ONI_AIC[,1]) # calculate delta AIC
```

```
ONI_AIC[,3] <- exp(-0.5*ONI_AIC[,2])/
  (sum(exp(-0.5*ONI_AIC[,2]))) # calculate model weights
colnames(ONI_AIC) <- c("AIC", "deltaAIC", "model_weights")
rownames(ONI_AIC) <- c("half", "full", "half_lag", "full_lag")
print(ONI_AIC)
```

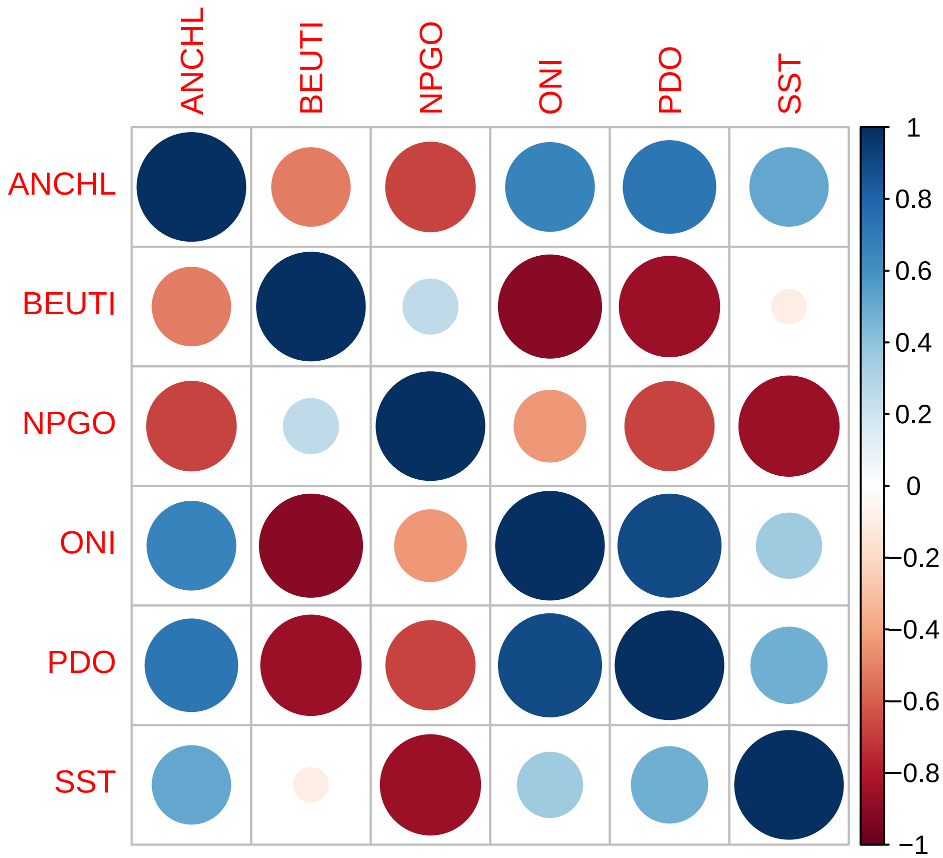
```
##                AIC   deltaAIC model_weights
## half     10733.90 17.2117112  0.0001053966
## full     10733.88 17.1942420  0.0001063212
## half_lag 10716.69  0.0000000  0.5758404504
## full_lag 10717.30  0.6124404  0.4239478318
```

There is equal support for the ONI half_lag and full_lag model.

Correlation

Check for correlation between the oceanographic indices (absolute value of Pearson's correlation coefficient > 0.65).

```
## bind covars
covars <- cbind(ANCHL_full[,2], BEUTI_full_lag[,2], NPGO_half[,2], ONI_half_lag[,2],
                PDO_half_lag[,2], SST_half[,2])
colnames(covars) <- c("ANCHL", "BEUTI", "NPGO", "ONI", "PDO", "SST")

## check for correlation between predictors (cutoff >0.65)
cp <- as.data.frame((round(cor(covars, use="complete.obs"), 2)))

## this function flattens your data in a particular way, used below
flattenCorrMatrix <- function(cormat, pmat) {
  ut <- upper.tri(cormat)
  data.frame(
    row = rownames(cormat)[row(cormat)[ut]],
    column = rownames(cormat)[col(cormat)[ut]],
    cor  =(cormat)[ut],
    p = pmat[ut]
  )
}

cor <- rcorr(as.matrix(covars))

## create a new formatted df
cor_vals <- flattenCorrMatrix(cor$r, cor$P) %>% arrange(cor)

## plot Pearson's correlation coefficient
corrplot(cor$r)
```




```
## print the covariates that should not be included in the same model
print(cor_vals[abs(cor_vals$cor) > 0.65, ] %>% arrange(row))
```

```
##     row column        cor            p
## 1 ANCHL   NPGO -0.6775876 0.0449154953
## 2 ANCHL    ONI  0.6647046 0.0507921812
## 3 ANCHL    PDO  0.7236809 0.0275209084
## 4 BEUTI    ONI -0.9004575 0.0009284770
## 5 BEUTI    PDO -0.8527556 0.0034818260
## 6  NPGO    SST -0.8504496 0.0036677485
## 7  NPGO    PDO -0.6717674 0.0475132254
## 8   ONI    PDO  0.8995266 0.0009583229
```

Positive correlations are displayed in blue and negative correlations in red color. Color intensity and the size of the circle are proportional to the correlation coefficients.

We selected a Pearson's correlation coefficient of +/- 0.65 as our cutoff for correlation. The following covariates should not be in the same model:

1) ANCHL and NPGO (-0.68)

2) ANCHL and ONI (0.66)

3) ANCHL and PDO (0.72)

4) BEUTI and ONI (-0.90)

5) BEUTI and PDO (-0.85)

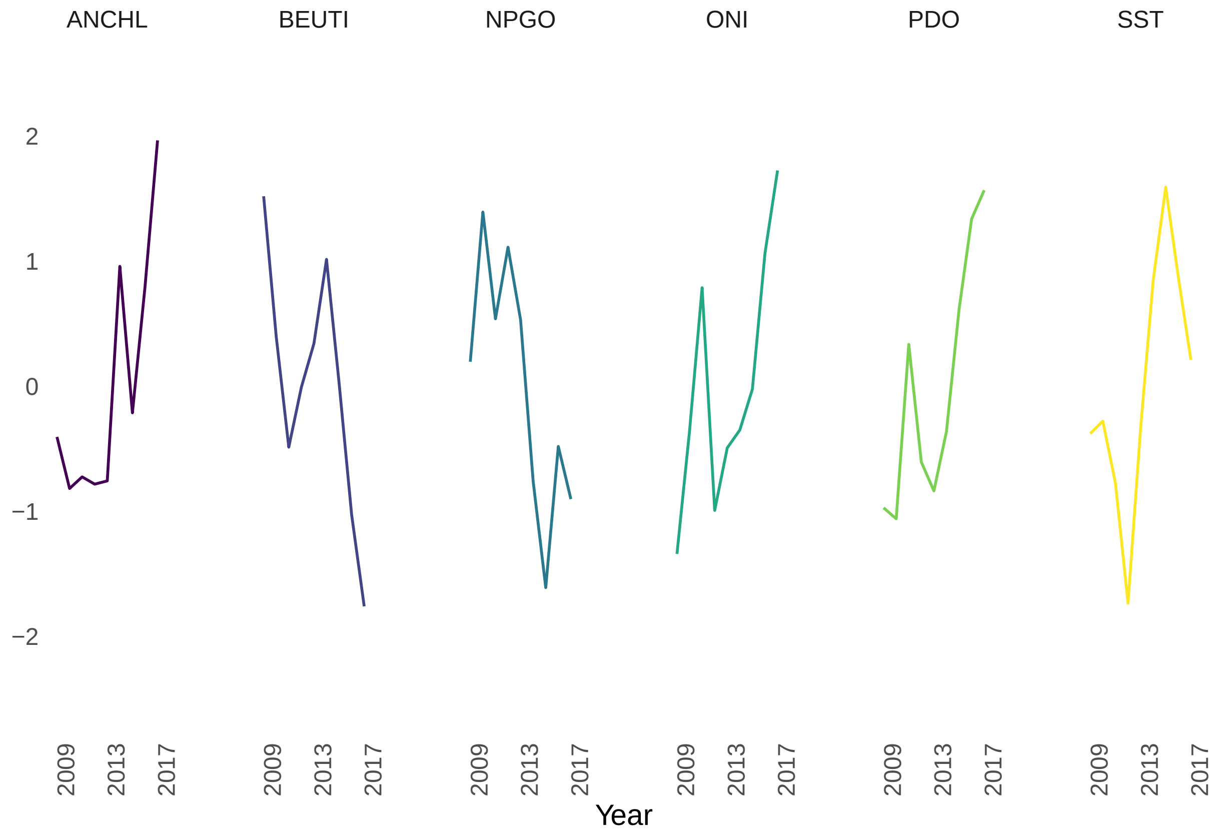
6) NPGO and PDO (-0.67)

7) NPGO and SST (-0.85)

8) ONI and PDO (0.90)

Covariate Plot

```
covarsplot <- covars %>%
  mutate(year = 2009:2017) %>%
  pivot_longer(cols = 1:6, names_to = "covariate", values_to = "value")

ggplot(data = covarsplot, aes(x = year, y = value, color = covariate)) +
  geom_line() +
  facet_wrap(~covariate, nrow = 1) +
  scale_color_viridis(discrete = TRUE) +
  ylim(c(-2.5, 2.5)) +
  theme_minimal() + xlab("Year") + ylab("") +
  scale_x_continuous(breaks = c(2009, 2013, 2017), limits = c(2009, 2017)) +
  theme(legend.position = "none",
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.text.x = element_text(angle = 90),
        panel.spacing = unit(2.5, "lines"))
```



Export Data

```
finalcovars <- covars %>%
  mutate(Year = 2009:2017) %>%
  select(Year, everything())
write.csv(x = finalcovars, file = here("data", "covariates", "covars.csv"),
          row.names = FALSE)
```