

Combining Species Presence/Absence with Environmental Data

Eleanor (Ella) Crotty

Contents

Setup	1
Sampling x environmental data graph	2
Data Join	3
Joined data graphs	3
DO x Temp graphs	6
Binomial Regression	8
Check for outliers	8
Filter out outliers - currently not using	9
Check for linearity	10
Calculate regression	13

Setup

```
# Warnings and startup messages suppressed
library(tidyverse)
library(patchwork)
library(scales)
library(ggrepel)
library(readxl)
library(here)

# Import data
SpeciesDetections <- read_csv(here("OCNMS_Project", "Outputs",
    "SOI_IDs_Species10Detections.csv")) %>%
    filter(year(Date_UTC) != 2023)
EnvData1 <- read_csv(here("OCNMS_Project", "Data", "EnvironmentalDataset1.csv")) %>% # 
    # Using EnvironmentalDataset1 because the satellite + NEMO data in
    # EnvironmentalDataset2 didn't turn out to be good at gap filling
    filter(year != 2023) %>% # Ignoring 2023 due to gaps for now
    mutate(year = as.factor(year))
SamplingDates2 <- read_csv(here("OCNMS_Project", "Data", "SamplingDates2.csv")) %>% #
    # Exported from EnvironmentalDataxSampleDates.Rmd, simple dataframe of all datetimes of
    # samples
    filter(year(Date) != 2023) %>% # Ignoring 2023 due to gaps for now
    mutate(Source = case_when(Source == "Bottle_DNA" ~ "Bottle_DNA_Sampled", Source ==
        "PPS_DNA" ~ "Automated_DNA_Sampler"))
```

```

# Date check

# Let's take a quick look at whether the dates match up
sampdates <- data.frame(Date = SamplingDates2$Date, Source = "Metadata")
detectdates <- data.frame(Date = SpeciesDetections$Date_UTC, Source = "Detections")
datescomp <- rbind(sampdates,detectdates)

ggplot() +
  geom_vline(data = datescomp, mapping = aes(xintercept = Date, color = Source, linetype
    ↵   = Source)) +
  scale_linetype_manual(values = c(2,1)) +
  theme_bw() +
  facet_wrap(facets = vars(year(Date)), scales = "free_x", ncol = 1) #+
#scale_x_datetime(date_breaks = "4 days", date_labels = "%y-%b-%d") # This is fucked
  ↵  but tbh I don't need it

```

Sampling x environmental data graph

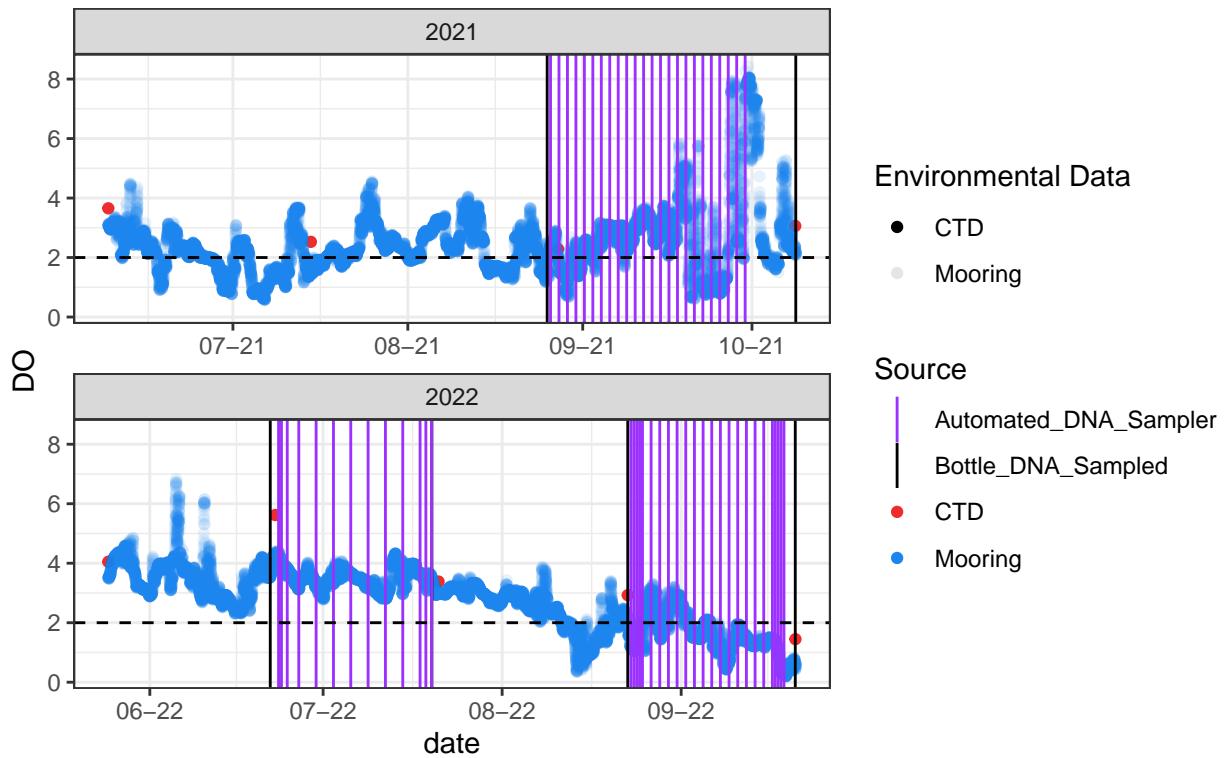
```

ggplot(EnvData1, aes(x = date, y = DO, color = source, alpha = source)) +
  geom_point() +
  scale_alpha_manual(values = c(1,0.1)) + # doesn't currently do anything
  scale_x_datetime(date_breaks = "1 month", date_labels = "%m-%y") +
  theme_bw() +
  facet_wrap(facets = vars(year), scales = "free_x", ncol = 1) +
  geom_vline(data = SamplingDates2, aes(xintercept = Date, color = Source)) +
  scale_color_manual(values = c("purple1", "black", "firebrick2", "dodgerblue2")) +
  geom_hline(aes(yintercept = 2), color = "black", linetype = "dashed") +
  labs(title = "Dissolved Oxygen Data + Sampling Dates", caption = "Dotted line = hypoxic
    ↵ threshold of 2 mg/L dissolved oxygen, vertical lines = DNA sampling times", alpha =
    ↵ "Environmental Data", color = "Source")

## Warning: Removed 236 rows containing missing values or values outside the scale range
## (`geom_point()`).

```

Dissolved Oxygen Data + Sampling Dates



Toxic threshold of 2 mg/L dissolved oxygen, vertical lines = DNA sampling times

Data Join

```
EnvRd <- EnvData1 %>%
  mutate(DateMatch = round_date(date, unit = "10 minutes")) # Round to the nearest 10
  ↵ minutes?
DetectRd <- SpeciesDetections %>%
  mutate(DateMatch = round_date(Date_UTC, unit = "10 minutes"), Date_local_hr =
  ↵ round_date(Date_local, unit = "hour")) # Spot check - looks good.
eDNAxEnvData <- left_join(DetectRd, EnvRd, by = join_by(DateMatch))

investigate <- eDNAxEnvData %>% select(Species, DateMatch, Date_UTC, Date_local_hr,
  ↵ source, temperature, DO, SampleId, Rosette_position, Amplicon)
```

Joined data graphs

```
# Currently hidden because these aren't super helpful
ggplot(eDNAxEnvData, aes(x = DateMatch, y = DO, shape = Present, size = Present, color =
  ↵ Present)) +
  scale_shape_manual(values = c(1, 19)) +
  scale_size_manual(values = c(1,1)) +
  geom_point() +
  theme_bw() +
  facet_wrap(facets = vars(Species))
```

```

## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).

ggplot(data = EnvData1, aes(x = date, y = DO)) +
  geom_line(color = "gray90") +
  geom_point(data = eDNAxEnvData, aes(x = DateMatch, y = DO, shape = Present, size =
    ↪ Present, color = Present)) +
  scale_shape_manual(values = c(1, 19)) +
  scale_size_manual(values = c(1,1)) +
  theme_bw() +
  facet_wrap(facets = vars(Species, year(date)), scales = "free_x", ncol = 2)

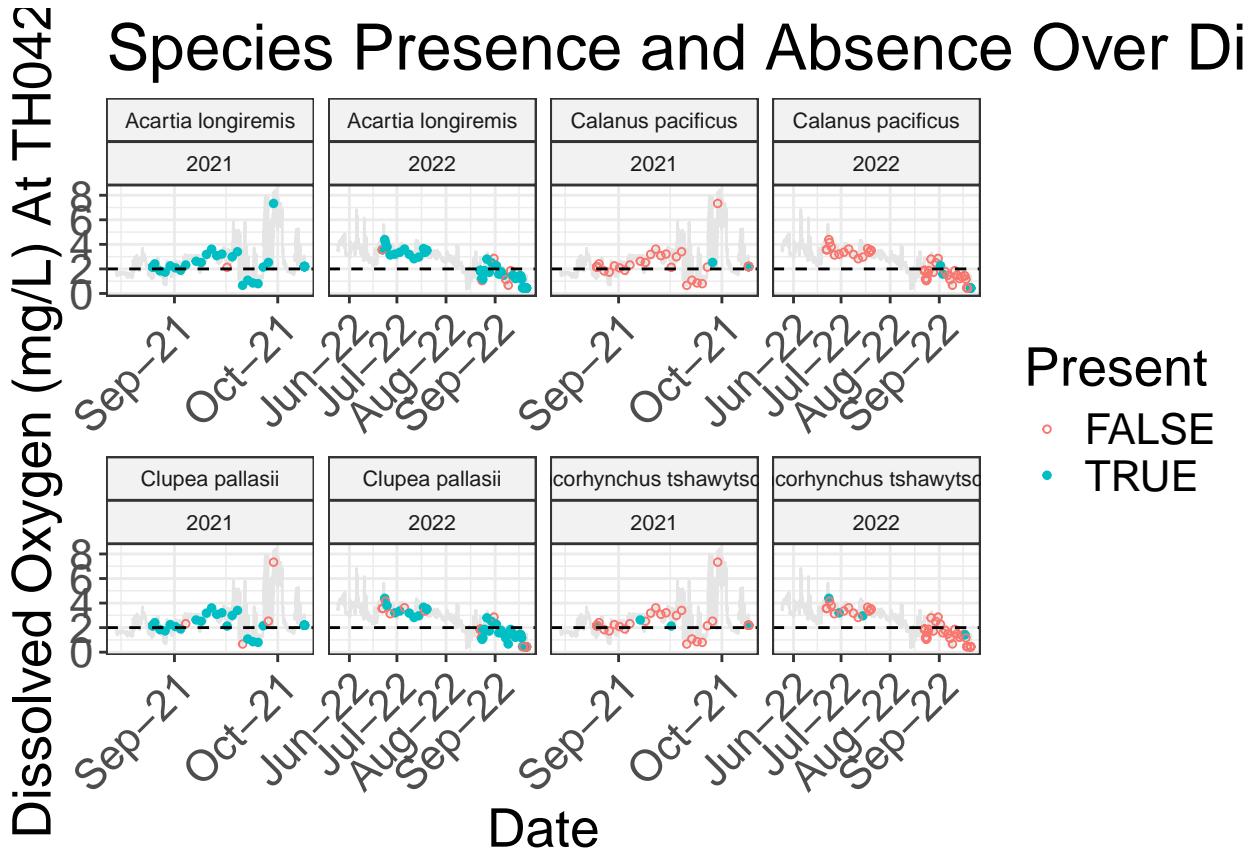
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).

EnvDataRange <- EnvData1 %>%
  filter(date > as.POSIXct("2021-08-15 00:00:00", tz = "UTC"))

ggplot(data = EnvDataRange, aes(x = date, y = DO)) +
  geom_line(color = "gray90") +
  geom_point(data = eDNAxEnvData, aes(x = DateMatch, y = DO, shape = Present, size =
    ↪ Present, color = Present)) +
  scale_shape_manual(values = c(1, 19)) +
  scale_size_manual(values = c(1,1)) +
  theme_bw() +
  facet_wrap(facets = vars(Species, year(date)), scales = "free_x", ncol = 4) +
  scale_x_datetime(breaks = "month", date_labels = "%b-%y") +
  theme(text = element_text(size = 20), axis.text.x = element_text(angle = 45, hjust =
    ↪ 1), strip.text = element_text(size = 8), strip.background = element_rect(fill =
    ↪ "gray95")) +
  geom_hline(yintercept = 2, linetype = 2) +
  labs(title = "Species Presence and Absence Over Dissolved Oxygen", x = "Date", y =
    ↪ "Dissolved Oxygen (mg/L) At TH042")

## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).

```



```

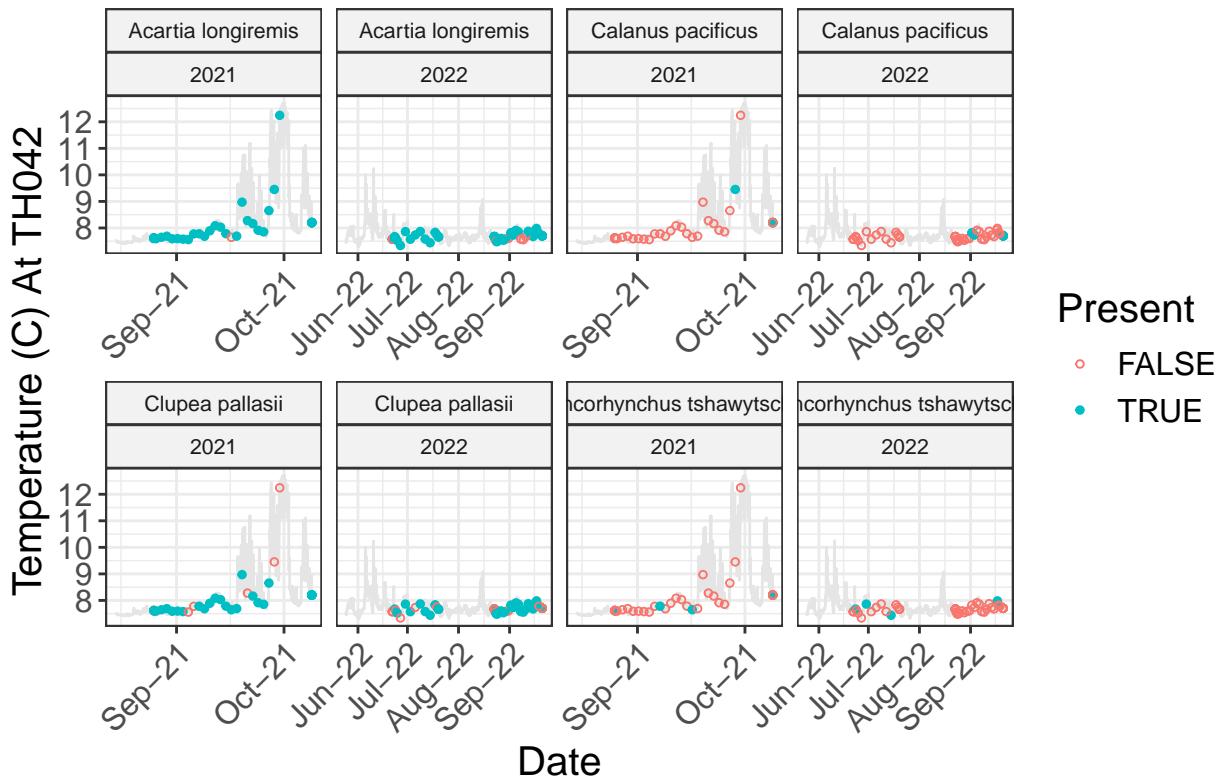
ggsave(filename = here("OCNMS_Project", "Plots",
  "SpeciesPresence_Oxygen_Preliminary.png"), width = 2500, height = 2000, units = "px")

## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).

ggplot(data = EnvDataRange, aes(x = date, y = temperature)) +
  geom_line(color = "gray90") +
  geom_point(data = eDNAxEnvData, aes(x = DateMatch, y = temperature, shape = Present,
    size = Present, color = Present)) +
  scale_shape_manual(values = c(1, 19)) +
  scale_size_manual(values = c(1,1)) +
  theme_bw() +
  facet_wrap(facets = vars(Species, year(date)), scales = "free_x", ncol = 4) +
  scale_x_datetime(breaks = "month", date_labels = "%b-%y") +
  theme(text = element_text(size = 15), axis.text.x = element_text(angle = 45, hjust =
    1), strip.text = element_text(size = 8), strip.background = element_rect(fill =
    "gray95")) +
  labs(title = "Species Presence and Absence Over Temperature", x = "Date", y =
    "Temperature (C) At TH042")

```

Species Presence and Absence Over Temperature



```
ggsave(filename = here("OCNMS_Project", "Plots", "SpeciesPresence_Temp_Preliminary.png"),
       width = 2500, height = 2000, units = "px")
```

DO x Temp graphs

```
full <- ggplot(EnvDataRange, aes(x = temperature, y = DO, shape = source, color =
  as.factor(year))) +
  scale_shape_manual(values = c(4,19)) +
  scale_color_manual(values = c("blue", "black")) +
  geom_point(shape = 1, alpha = 0.5) +
  theme_bw() +
  labs(title = "DO vs Temp during sampling years")
ggsave(here("OCNMS_eDNA", "Plots", "DO_vs_Temp.png"), plot = full)
```

```
## Saving 6.5 x 4.5 in image
full_yr <- ggplot(EnvDataRange, aes(x = temperature, y = DO, shape = source, color =
  as.factor(year))) +
  scale_shape_manual(values = c(4,19)) +
  scale_color_manual(values = c("blue", "black")) +
  geom_point(shape = 1, alpha = 0.5) +
  theme_bw() +
  facet_wrap(facets = vars(year)) +
  labs(title = "DO vs Temp during sampling years")
ggsave(here("OCNMS_eDNA", "Plots", "DO_vs_TempYear.png"), plot = full_yr)
```

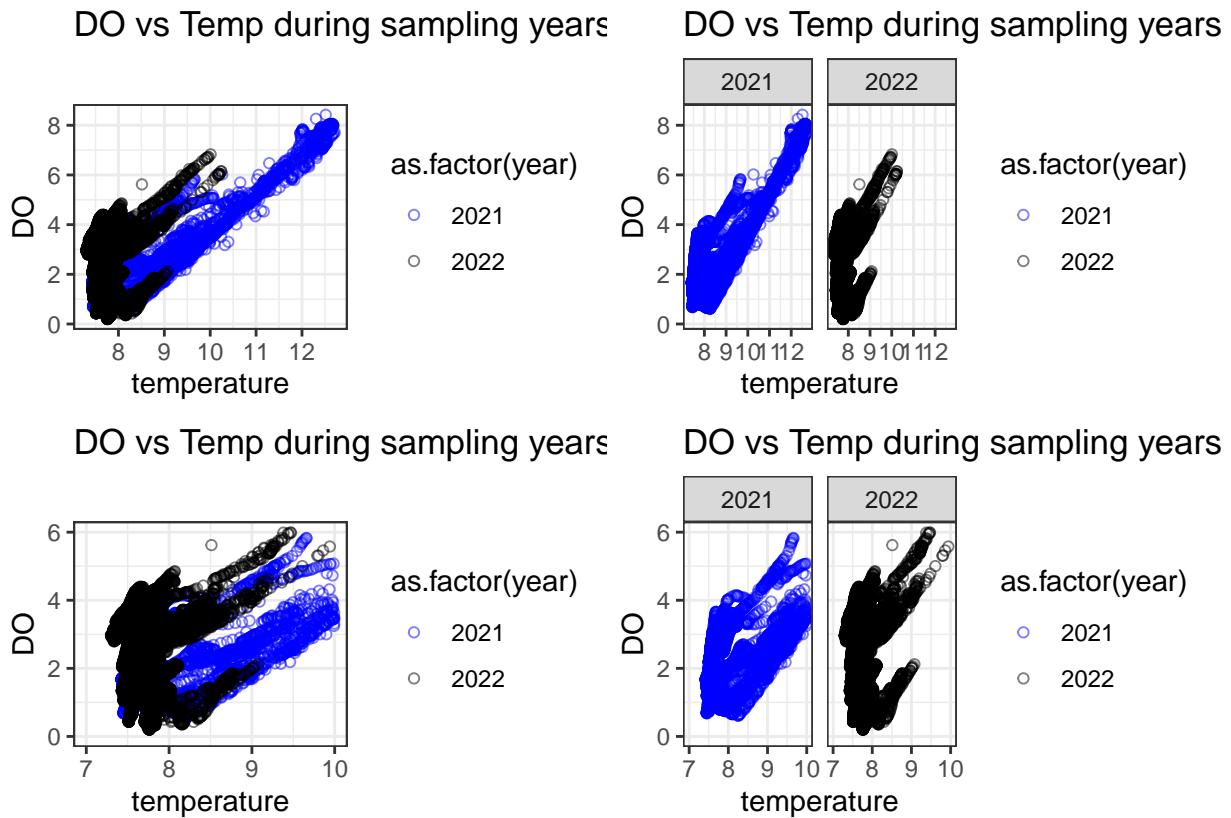
```

## Saving 6.5 x 4.5 in image
zoom <- ggplot(EnvDataRange, aes(x = temperature, y = DO, shape = source, color =
  as.factor(year))) +
  scale_shape_manual(values = c(4,19)) +
  scale_color_manual(values = c("blue", "black")) +
  geom_point(shape = 1, alpha = 0.5) +
  scale_x_continuous(limits = c(7, 10)) +
  scale_y_continuous(limits = c(0,6)) +
  theme_bw() +
  labs(title = "DO vs Temp during sampling years (Zoomed in)")
ggsave(here("OCNMS_eDNA", "Plots", "DO_vs_Temp_Zoomed.png"), plot = zoom)

## Saving 6.5 x 4.5 in image
zoom_yr <- ggplot(EnvDataRange, aes(x = temperature, y = DO, shape = source, color =
  as.factor(year))) +
  scale_shape_manual(values = c(4,19)) +
  scale_color_manual(values = c("blue", "black")) +
  geom_point(shape = 1, alpha = 0.5) +
  scale_x_continuous(limits = c(7, 10)) +
  scale_y_continuous(limits = c(0,6)) +
  theme_bw() +
  facet_wrap(facets = vars(year)) +
  labs(title = "DO vs Temp during sampling years (Zoomed in)")
ggsave(here("OCNMS_eDNA", "Plots", "DO_vs_Temp_ZoomedYear.png"), plot = zoom_yr)

## Saving 6.5 x 4.5 in image
(full + full_yr) / (zoom + zoom_yr)

```



Binomial Regression

Binomial Regression Assumptions

1. Independent observations - *Check!*
2. No multicollinearity: Continuous independent variables should not be correlated with each other - *NOT MET - shouldn't do a DO x Temp x Presence model*
3. Binomial dependent variable, with two **mutually exclusive** values - *Check!*
4. Continuous independent variable - *Check!*
5. Linearity between the binomial and continuous variables - this is checked with a logit transformation - *See below, this varies by species*
6. Sufficient obeservations - the source I found recommends > 15 - *Check! 10 should be alright but I will note it in the results*
7. No outliers - *One outlier, which appears to be part of a real heatwave - I have not filtered it out yet*

Check for outliers

```
ggplot(EnvDataRange, aes(x = DO, y = temperature)) +
  geom_point(color = "gray90", alpha = 0.5) +
  geom_point(eDNAxEnvData, mapping = aes(x = DO, y = temperature, color = Present, shape =
  ↵ = Present), inherit.aes = F) +
  scale_shape_manual(values = c(1, 19)) +
  theme_bw() +
  theme(text = element_text(size = 15), strip.text = element_text(size = 8),
  ↵ strip.background = element_rect(fill = "gray95")) +
  labs(title = "Dissolved Oxygen vs. Temp + Species Presence", y = "Temperature") +
  facet_wrap(facets = vars(Species))
```

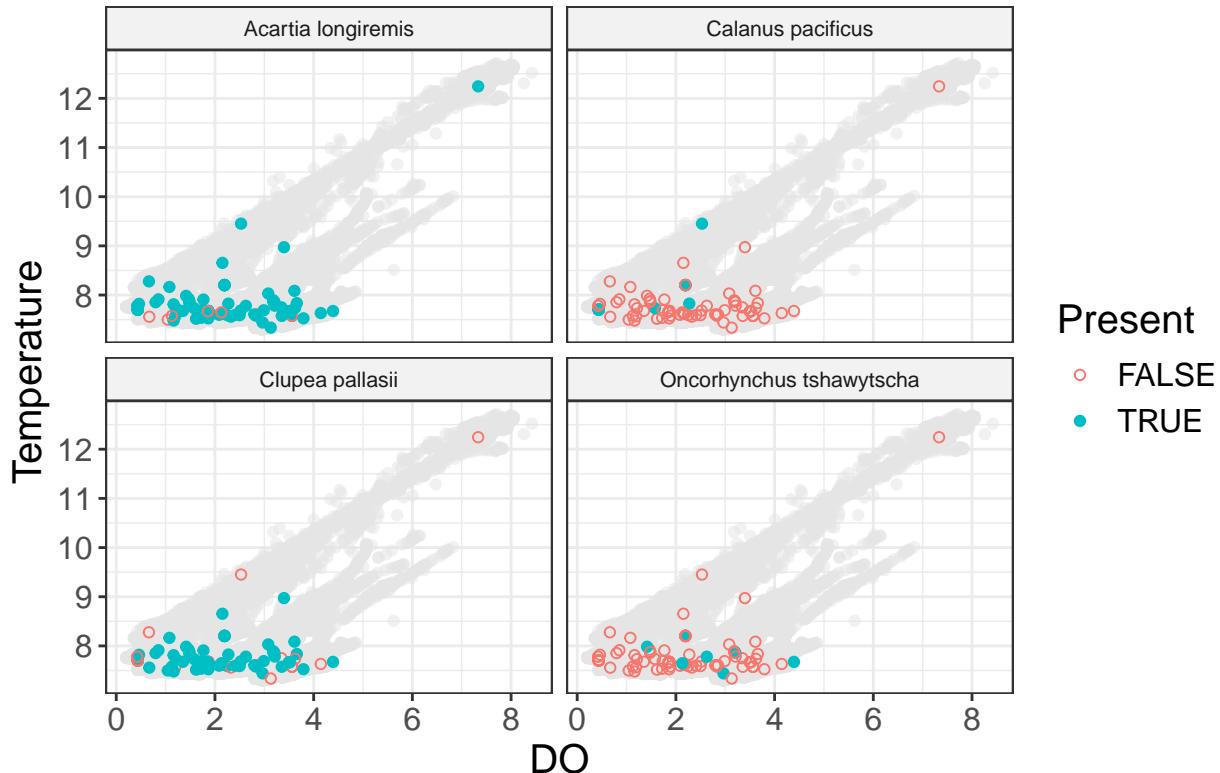
```

## Warning: Removed 924 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).

```

Dissolved Oxygen vs. Temp + Species Presence



```

ggsave(filename = here("OCNMS_Project", "Plots", "SpeciesPresence_TempxD0.png"), width =
  2500, height = 2000, units = "px")

```

```

## Warning: Removed 924 rows containing missing values or values outside the scale range
## (`geom_point()`).
## Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).

```

Filter out outliers - currently not using

```

eDNAxEnvData_forlm <- eDNAxEnvData %>%
  filter(temperature < 11)

```

```

ggplot(EnvDataRange, aes(x = DO, y = temperature)) +
  geom_point(color = "gray90", alpha = 0.5) +
  geom_point(eDNAxEnvData_forlm, mapping = aes(x = DO, y = temperature, color = Present,
  shape = Present), inherit.aes = F) +
  scale_shape_manual(values = c(1, 19)) +
  theme_bw() +
  theme(text = element_text(size = 15), strip.text = element_text(size = 8),
  strip.background = element_rect(fill = "gray95")) +
  labs(title = "Dissolved Oxygen vs. Temp + Species Presence", y = "Temperature") +

```

```

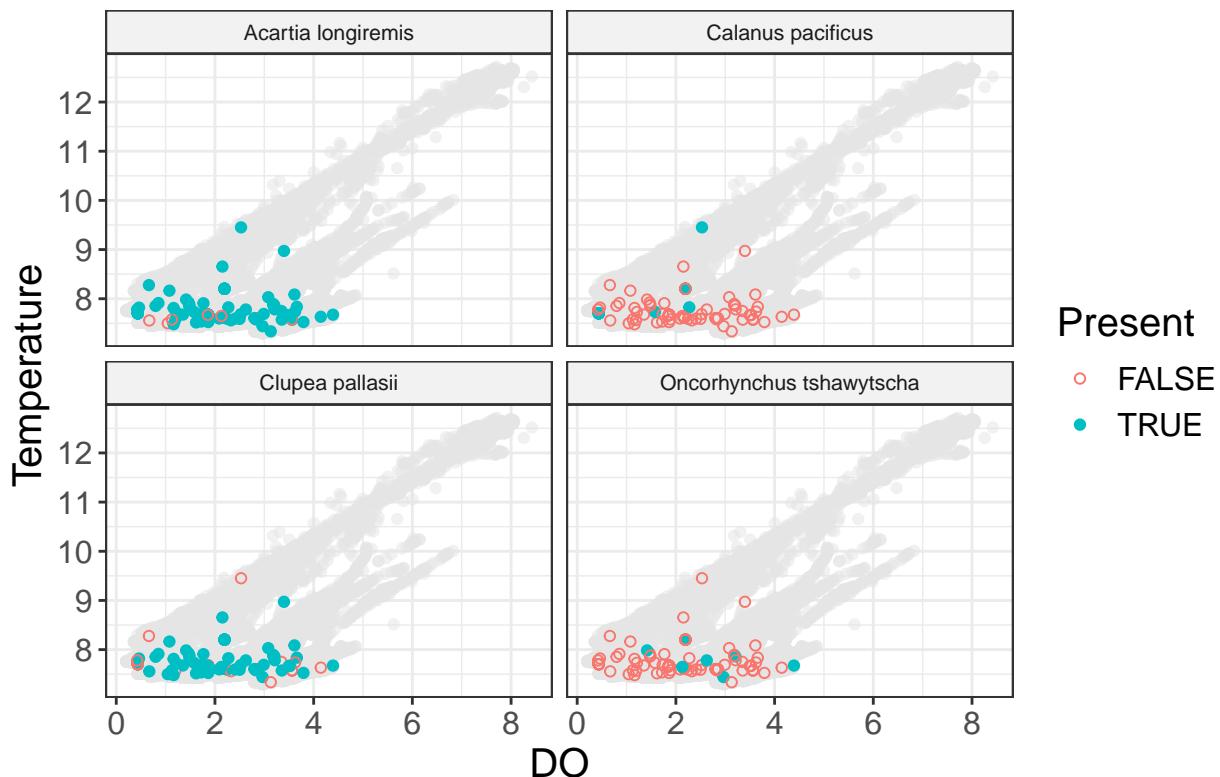
facet_wrap(facets = vars(Species))

## Warning: Removed 924 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).

```

Dissolved Oxygen vs. Temp + Species Presence



```

ggsave(filename = here("OCNMS_Project", "Plots", "SpeciesPresence_TempxD0.png"), width =
  ↵ 2500, height = 2000, units = "px")

```

```

## Warning: Removed 924 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).

```

Check for linearity

“Assumption #5 involves the necessity of a linear relationship between the continuous independent variables and the logit transformation of the dependent variable. This linearity assumption implies that **for continuous independent variables** like income level, hours of exercise per week, and blood sugar levels, **there should be a linear relationship with the logit of the dependent variable**, such as the probability of developing diabetes. Various methods can be employed to assess this linearity, with one common approach being the **Box-Tidwell procedure**. This technique involves creating interaction terms between each continuous independent variable and its natural logarithm and adding these to the logistic regression model. This technique can be implemented using software like **SPSS Statistics**, which offers the **Binary Logistic procedure to test for this assumption**. The results of this test are then interpreted to decide the next steps in the analysis, depending on whether the linearity assumption holds or is violated. If the assumption is

met, the analysis can proceed as planned. However, if the assumption is not met, adjustments to the model or alternative methods may be necessary to address the non-linearity appropriately." - [Binomial Logistic Regression](#)

Basically, this equation needs to have a somewhat linear relationship with the independent variable:

$$g(p) = \log\left(\frac{p}{1-p}\right)$$

Where p = probability of 1 (if 1, p. if 0, 1-p)?

```
joinSpeciesList <- split(eDNAxEnvData, eDNAxEnvData$Species) # Split by species

library(car) # Has a function for the Box-Tidwell procedure

## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##   recode
##
## The following object is masked from 'package:purrr':
##   some

?boxTidwell

joinSpeciesList <- lapply(joinSpeciesList, drop_na, DO)

# joinSpeciesList, glm, formula = Present ~ DO

# Single test
boxTidwell(Present ~ DO, other.x = ~ year, data = joinSpeciesList[[1]]) # other.x = any
# factors not to be transformed. i had to make year into a factor to make it accept
# this lmao, so I'm not sure this is valid!

## Warning in boxTidwell.default(y, X1, X2, max.iter = max.iter, tol = tol, :
## maximum iterations exceeded

## MLE of lambda Score Statistic (t) Pr(>|t|)
##      229.08          0.3369    0.7369
##
## iterations = 26

# Loop to test all for linearity with temperature
for (i in 1:length(joinSpeciesList)) {
  print(paste(names(joinSpeciesList)[i], sep = " ", "Presence vs DO"))
  print(boxTidwell(Present ~ DO, other.x = ~ year, data = joinSpeciesList[[i]]))
}

## [1] "Acartia longiremis Presence vs DO"
## Warning in boxTidwell.default(y, X1, X2, max.iter = max.iter, tol = tol, :
## maximum iterations exceeded

## MLE of lambda Score Statistic (t) Pr(>|t|)
```

```

##          229.08      0.3369   0.7369
##
## iterations =  26
## [1] "Calanus pacificus Presence vs DO"
## Warning in boxTidwell.default(y, X1, X2, max.iter = max.iter, tol = tol, :
## maximum iterations exceeded

##  MLE of lambda Score Statistic (t) Pr(>|t|)
##          -7.4201      2.8025  0.006112 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## iterations =  26
## [1] "Clupea pallasii Presence vs DO"
##  MLE of lambda Score Statistic (t) Pr(>|t|)
##          -29.763      -2.8066  0.006041 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## iterations =  11
## [1] "Oncorhynchus tshawytscha Presence vs DO"
##  MLE of lambda Score Statistic (t) Pr(>|t|)
##          0.043619     -0.7706   0.4428
##
## iterations =  15

# Loop to test all for linearity with DO
for (i in 1:length(joinSpeciesList)) {
  print(paste(names(joinSpeciesList)[i], sep = " ", "Presence vs Temp"))
  print(boxTidwell(Present ~ temperature, other.x = ~ year, data = joinSpeciesList[[i]]))
}

## [1] "Acartia longiremis Presence vs Temp"
## Warning in boxTidwell.default(y, X1, X2, max.iter = max.iter, tol = tol, :
## maximum iterations exceeded

##  MLE of lambda Score Statistic (t) Pr(>|t|)
##          -13.576      -1.2963   0.1979
##
## iterations =  26
## [1] "Calanus pacificus Presence vs Temp"
##  MLE of lambda Score Statistic (t) Pr(>|t|)
##          -11.882      -2.008   0.04739 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## iterations =  8
## [1] "Clupea pallasii Presence vs Temp"
## Warning in boxTidwell.default(y, X1, X2, max.iter = max.iter, tol = tol, :
## maximum iterations exceeded

##  MLE of lambda Score Statistic (t) Pr(>|t|)
##          5.2474      -1.1528   0.2518
##

```

```

## iterations = 26
## [1] "Oncorhynchus tshawytscha Presence vs Temp"
## MLE of lambda Score Statistic (t) Pr(>|t|)
##      1.6114          -0.0447   0.9644
##
## iterations = 15

```

Calculate regression

```

oxmodels <- lapply(joinSpeciesList, glm, formula = Present ~ DO, family = "binomial") #
  ↪ Make the model for each species
lapply(oxmodels, summary) # Print the summary of each model

```

```

## $`Acartia longiremis`
##
## Call:
## FUN(formula = ..1, family = "binomial", data = X[[i]])
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.0986    0.8390   2.501   0.0124 *
## DO          0.2513    0.3884   0.647   0.5176
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 51.015 on 101 degrees of freedom
## Residual deviance: 50.572 on 100 degrees of freedom
## AIC: 54.572
##
## Number of Fisher Scoring iterations: 5
##
##
## $`Calanus pacificus`
##
## Call:
## FUN(formula = ..1, family = "binomial", data = X[[i]])
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.07917   0.68168   0.116  0.90754
## DO         -1.40949   0.46551  -3.028  0.00246 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 65.434 on 101 degrees of freedom
## Residual deviance: 52.869 on 100 degrees of freedom
## AIC: 56.869
##
## Number of Fisher Scoring iterations: 6
##

```

```

## 
## $`Clupea pallasii`
## 
## Call:
## FUN(formula = ..1, family = "binomial", data = X[[i]])
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 1.02402   0.48112  2.128   0.0333 *
## DO         -0.04661   0.19676 -0.237   0.8127
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
## Null deviance: 121.78  on 101  degrees of freedom
## Residual deviance: 121.73  on 100  degrees of freedom
## AIC: 125.73
## 
## Number of Fisher Scoring iterations: 4
## 
## 
## $`Oncorhynchus tshawytscha`
## 
## Call:
## FUN(formula = ..1, family = "binomial", data = X[[i]])
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.3263    0.8270 -4.022 5.77e-05 ***
## DO          0.3668    0.2896  1.267    0.205
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
## Null deviance: 56.084  on 101  degrees of freedom
## Residual deviance: 54.591  on 100  degrees of freedom
## AIC: 58.591
## 
## Number of Fisher Scoring iterations: 5

tmodels <- lapply(joinSpeciesList, glm, formula = Present ~ temperature, family =
  "binomial") # Make the model for each species (temperature)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
lapply(tmodels, summary) # Print the summary of each model

## $`Acartia longiremis`
## 
## Call:
## FUN(formula = ..1, family = "binomial", data = X[[i]])
## 
## Coefficients:

```

```

##           Estimate Std. Error z value Pr(>|z|)
## (Intercept) -58.90      29.74  -1.980  0.0477 *
## temperature   8.03       3.91   2.053  0.0400 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 51.015  on 101  degrees of freedom
## Residual deviance: 44.025  on 100  degrees of freedom
## AIC: 48.025
##
## Number of Fisher Scoring iterations: 8
##
##
## $`Calanus pacificus`
##
## Call:
## FUN(formula = ..1, family = "binomial", data = X[[i]])
##
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.7025    3.5401  -1.328  0.184
## temperature  0.3157    0.4456   0.709  0.479
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 65.434  on 101  degrees of freedom
## Residual deviance: 65.015  on 100  degrees of freedom
## AIC: 69.015
##
## Number of Fisher Scoring iterations: 5
##
##
## $`Clupea pallasii`
##
## Call:
## FUN(formula = ..1, family = "binomial", data = X[[i]])
##
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.2800    3.0301   1.082  0.279
## temperature -0.3007    0.3853  -0.780  0.435
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 121.78  on 101  degrees of freedom
## Residual deviance: 121.17  on 100  degrees of freedom
## AIC: 125.17
##
## Number of Fisher Scoring iterations: 4
##
##
## $`Oncorhynchus tshawytscha`
```

```
##  
## Call:  
## FUN(formula = ..1, family = "binomial", data = X[[i]])  
##  
## Coefficients:  
##             Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -0.5336    7.4162  -0.072   0.943  
## temperature -0.2475    0.9522  -0.260   0.795  
##  
## (Dispersion parameter for binomial family taken to be 1)  
##  
##     Null deviance: 56.084  on 101  degrees of freedom  
## Residual deviance: 56.000  on 100  degrees of freedom  
## AIC: 60  
##  
## Number of Fisher Scoring iterations: 5
```