

Thesis Goal 1

Eleanor (Ella) Crotty

Contents

Goals	1
Import data	1
Calculate eDNA index	3
Combine eDNA index with environmental data	6
Re-calculate eDNA index for 2021-2022 only	7
Plot eDNA index over time and oxygen	11
Scatterplot eDNA vs hypoxic threshold	44
GAM of eDNA index vs oxygen	46

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

Goal 1: Compare copepod eDNA index, a measure of relative abundance using eDNA detections, to dissolved oxygen data in order to assess whether hypoxia decreases copepod abundance in OCNMS.

Goals

- Plot eDNA index over time and oxygen
- Make scatterplots of eDNA index vs oxygen
- Do a GAM of eDNA index vs oxygen
- Compare eDNA to hypoxic thresholds

Import data

```
envData <- read_csv(here("PMEL-Data", "EnvironmentalDataset1_copy.csv")) %>%
  filter(year != 2023) %>% # Ignoring 2023 due to gaps for now
  mutate(year = as.factor(year)) %>%
  relocate(year, .after = date)
```

```

## New names:
## Rows: 38938 Columns: 11
## -- Column specification
## -----
## (1): source dbl (8): ...1, year, temperature, DO, salinity, potential_density,
## pres, cond dttm (2): date, sampleID
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `...1` -> `...1`

allReads <- read_csv(here("PMEL-Data", "FishPlusCOI_Reads_copy.csv")) %>%
  mutate(year = year(Date_local)) %>%
  relocate(year, .after = Date_local)

## Rows: 1094324 Columns: 71
## -- Column specification -----
## Delimiter: ","
## chr (39): ESV, sequence, Kingdom, Phylum, Class, Order, Family, Genus, Spec...
## dbl (27): X, pctMatch, nReads, Biological_Replicate, Technical_Replicate, D...
## lgl (3): Negative_control, Positive_control, primers_pad
## dttm (2): Date.UTC, Date_local
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

yrReads <- allReads %>%
  filter(year != 2023) %>% # Ignoring 2023 due to gaps for now
  mutate(year = as.factor(year))
copepodNames <- read_csv(here("PMEL-Data", "OCNMS_Copepods_Krill_copy.csv"))

## Rows: 36 Columns: 2
## -- Column specification -----
## Delimiter: ","
## chr (2): Species, Notes
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

eDNAxEnvDataSat <- read_csv(here("PMEL-Data", "eDNAxEnvDataSat_copy.csv"))

## New names:
## Rows: 80442 Columns: 81
## -- Column specification
## -----
## (36): Kingdom, Phylum, Class, Order, Family, Genus, Species, JV243.2, B... dbl
## (35): TotalnReads, Biological_Replicate, Depth_m, Sample_volume_ml, Lat... lgl
## (5): Present, Negative_control, Positive_control, primers_pad, sampleID dttm
## (5): Date.UTC, Date_local, DateMatch, Date_local_hr, date
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `...1` -> `...68`

copepodFiltered <- read_csv(here("CopepodDetectionsFiltered.csv"))

## Rows: 763 Columns: 82
## -- Column specification -----

```

```

## Delimiter: ","
## chr (36): Kingdom, Phylum, Class, Order, Family, Genus, Species, JV243.2, B...
## dbl (35): TotalnReads, Biological_Replicate, Depth_m, Sample_volume_ml, Lat...
## lgl (6): Barcode_mod, Present, Negative_control, Positive_control, primers...
## dttm (5): Date.UTC, Date_local, DateMatch, Date_local_hr, date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
copepodFull <- read_csv(here("CopepodDetectionsFull.csv"))

```

```

## Rows: 3488 Columns: 82
## -- Column specification -----
## Delimiter: ","
## chr (36): Kingdom, Phylum, Class, Order, Family, Genus, Species, JV243.2, B...
## dbl (35): TotalnReads, Biological_Replicate, Depth_m, Sample_volume_ml, Lat...
## lgl (6): Barcode_mod, Present, Negative_control, Positive_control, primers...
## dttm (5): Date.UTC, Date_local, DateMatch, Date_local_hr, date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

```

Calculate eDNA index

```

copepods <- unique(copepodFull$Species)
copepodReads <- allReads %>%
  filter(Species %in% copepods) # Just in case

index1dummy <- allReads %>% # Quick proportions
dplyr::group_by(SampleId) %>%
  mutate(Tot = sum(nReads),
        Row.sums = nReads / Tot) %>% # calculate proportions - 0 reads/0 total = NaN,
        # need to replace with 0 to make max() work
  relocate(c(Tot, Row.sums), .after = SampleId)

# First, combine PCR replicates and average
PCR_reps_combine <- function(df) {
  # print(head(df))

  # Separate out Sample_Name into three informational columns
  df_out <- df %>%
    separate(Sample_Name,
             into=c("E_no", "Cruise1", "PCR_Rep"),
             remove=F,
             fill = "right") %>% # sep is a regular expression which is annoying, but the
             # default recognizes any non-alphanumeric characters so the default works
             # here
    mutate(.)

  # Define which columns to group by
  id_cols1 <- colnames(df_out) # Pull out column names
  id_cols1 <- id_cols1[! id_cols1 %in% c("X", "SampleId", "Sample_Name", "PCR_Rep",
  "pctMatch", "JV_Sample_Name", "Technical_Replicate", "nReads", "Barcode.y")] # Remove
  # SampleId and Sample_Name, as well as other things that differ by PCR replicate

```

```

print(id_cols1)

df_out <- df_out %>%
  group_by_at(id_cols1) %>%
  # dplyr::mutate(SampleId = dplyr::first(SampleId)) %>% # Just ignoring sampleid for
  # now and using E-no as unique
  summarize(nReads = mean(nReads)) %>%
  relocate(nReads, .after = PI)

df_out
}

allReads_PCRcomb <- PCR_reps_combine(allReads) # Spot check E1325 was correct

## [1] "ESV"
## [2] "sequence"
## [3] "Kingdom"
## [4] "Phylum"
## [5] "Class"
## [6] "Order"
## [7] "Family"
## [8] "Genus"
## [9] "Species"
## [10] "JV243.2"
## [11] "Barcode.x"
## [12] "Barcode_mod"
## [13] "PI"
## [14] "E_no"
## [15] "Cruise1"
## [16] "Biological_Replicate"
## [17] "Negative_control"
## [18] "Positive_control"
## [19] "Cruise_ID_short"
## [20] "Cruise_ID_long"
## [21] "Field_Sample_Name"
## [22] "Cast_No."
## [23] "Rosette_position"
## [24] "Field_collection_method"
## [25] "Preservation"
## [26] "Area.within.region"
## [27] "Station"
## [28] "Depth_m"
## [29] "Sample_volume_ml"
## [30] "Personnel_responsible_for_Collecting_Sample"
## [31] "Field_Notes"
## [32] "Date_UTC"
## [33] "Date_local"
## [34] "year"
## [35] "Lat_dec"
## [36] "Lon_dec"
## [37] "Run"
## [38] "Client"
## [39] "ClientId"

```

```

## [40] "BatchId"
## [41] "Replicate"
## [42] "Amplicon"
## [43] "F_name"
## [44] "primers_fwd"
## [45] "R_name"
## [46] "primers_rev"
## [47] "PE"
## [48] "primers_pad"
## [49] "primers_errors"
## [50] "merge_maxdiffs"
## [51] "merge_maxdiffpct"
## [52] "merge_minovlen"
## [53] "ExpL"
## [54] "merge_minlen"
## [55] "merge maxlen"
## [56] "filter_max_ee"
## [57] "denoise_alpha"
## [58] "denoise_minsize"
## [59] "tax_ID"
## [60] "tax_maxaccepts"
## [61] "tax_maxrejects"
## [62] "tax_maxhits"
## [63] "tax_qcov"
## [64] "tax_minwords"
## [65] "tax_wordlen"
## [66] "Marker"

## `summarise()` has grouped output by 'ESV', 'sequence', 'Kingdom', 'Phylum',
## 'Class', 'Order', 'Family', 'Genus', 'Species', 'JV243.2', 'Barcode.x',
## 'Barcode_mod', 'PI', 'E_no', 'Cruise1', 'Biological_Replicate',
## 'Negative_control', 'Positive_control', 'Cruise_ID_short', 'Cruise_ID_long',
## 'Field_Sample_Name', 'Cast_No.', 'Rosette_position', 'Field_collection_method',
## 'Preservation', 'Area.within.region', 'Station', 'Depth_m', 'Sample_volume_ml',
## 'Personnel_responsible_for_Collecting_Sample', 'Field_Notes', 'Date_UTC',
## 'Date_local', 'year', 'Lat_dec', 'Lon_dec', 'Run', 'Client', 'ClientId',
## 'BatchId', 'Replicate', 'Amplicon', 'F_name', 'primers_fwd', 'R_name',
## 'primers_rev', 'PE', 'primers_pad', 'primers_errors', 'merge_maxdiffs',
## 'merge_maxdiffpct', 'merge_minovlen', 'ExpL', 'merge_minlen', 'merge maxlen',
## 'filter_max_ee', 'denoise_alpha', 'denoise_minsize', 'tax_ID',
## 'tax_maxaccepts', 'tax_maxrejects', 'tax_maxhits', 'tax_qcov', 'tax_minwords',
## 'tax_wordlen'. You can override using the `groups` argument.

# Second, combine by species

id_cols <- colnames(allReads_PCRcomb) # Pull out column names
id_cols <- id_cols[! id_cols %in% c("X", "ESV", "sequence", "nReads")] # Remove ESV +
# nreads because those are different within species

allReads_species <- allReads_PCRcomb %>%
  group_by_at(id_cols) %>% # group_by_at can take a vector
  summarize(TotalnReads = sum(nReads)) %>% # Removed , avgpctMatch = mean(pctMatch)
# because I had to remove pctMatch to combine PCR replicates. Sum nReads results in
# taking the sum of all ESVs within a species
  relocate(TotalnReads, .after = PI)

```

```

## `summarise()` has grouped output by 'Kingdom', 'Phylum', 'Class', 'Order',
## 'Family', 'Genus', 'Species', 'JV243.2', 'Barcode.x', 'Barcode_mod', 'PI',
## 'E_no', 'Cruise1', 'Biological_Replicate', 'Negative_control',
## 'Positive_control', 'Cruise_ID_short', 'Cruise_ID_long', 'Field_Sample_Name',
## 'Cast_No.', 'Rosette_position', 'Field_collection_method', 'Preservation',
## 'Area.within.region', 'Station', 'Depth_m', 'Sample_volume_ml',
## 'Personnel_responsible_for_Collecting_Sample', 'Field_Notes', 'Date_UTC',
## 'Date_local', 'year', 'Lat_dec', 'Lon_dec', 'Run', 'Client', 'ClientId',
## 'BatchId', 'Replicate', 'Amplicon', 'F_name', 'primers_fwd', 'R_name',
## 'primers_rev', 'PE', 'primers_pad', 'primers_errors', 'merge_maxdiffs',
## 'merge_maxdiffpct', 'merge_minovlen', 'ExpL', 'merge_minlen', 'merge maxlen',
## 'filter_max_ee', 'denoise_alpha', 'denoise_minsize', 'tax_ID',
## 'tax_maxaccepts', 'tax_maxrejects', 'tax_maxhits', 'tax_qcov', 'tax_minwords',
## 'tax_wordlen'. You can override using the `groups` argument.

write_csv(allReads_species, here("eDNA_Index_Hypoxia", "Data",
→ "CopepodReads_Species.csv")) # Gonna want this later

all_index1 <- allReads_species %>%
  dplyr::group_by(E_no) %>% # Group by E-number, because sampleID had to be removed in
→ the PCR replicate combination step
  mutate(Tot = sum(TotalnReads),
    Row.sums = TotalnReads / Tot) %>% # calculate proportions - 0 reads/0 total =
    → NaN, need to replace with 0 to make max() work
  relocate(c(TotalnReads, Tot, Row.sums), .after = E_no) # Move it somewhere I can see
→ the damn thing

all_eDNA_index <- all_index1 %>%
  dplyr::group_by(Species) %>%
  mutate(Row.sums = case_when(Row.sums == "NaN" ~ 0,
    .default = Row.sums)) %>% # Make 0/0 = 0 and not
    → NaN
  mutate(Colmax = max(Row.sums), Normalized.reads = Row.sums / Colmax) %>% # transforms
  → raw number of reads to eDNA index. Creates same divide by 0 error, so:
  mutate(Normalized.reads = case_when(Normalized.reads == "NaN" ~ 0,
    .default = Normalized.reads)) %>% # Make 0/0 = 0
    → and not NaN
  relocate(c(Colmax, Normalized.reads), .after = Row.sums) # Move it somewhere I can see
→ the damn thing

# Filter to only copepods
copepod_eDNA_index <- all_eDNA_index %>%
  filter(Species %in% copepods) %>%
  rename(eDNA_index = `Normalized.reads`) # to make parsing this easier

write_csv(copepod_eDNA_index, here("eDNA_Index_Hypoxia", "Data",
→ "Copepod_eDNA_index.csv")) # Gonna want this later

```

Combine eDNA index with environmental data

```

# envData = EnvironmentalDataset1
# Based on eDNAxp02.Rmd from summer project
# copepod_eDNA_index is still essentially a version of allReads so I can use the same
# code I did with allReads_Species

EnvRd <- envData %>%
  mutate(DateMatch = round_date(date, unit = "10 minutes")) # Round to the nearest 10
# minutes
DetectRd <- copepod_eDNA_index %>%
  mutate(DateMatch = round_date(Date_UTC, unit = "10 minutes"), Date_local_hr =
# round_date(Date_local, unit = "hour")) # Spot check - looks good.

eDNAindxEnvData <- left_join(DetectRd, EnvRd, by = join_by(DateMatch)) %>%
  relocate(date, .after = Date_UTC) %>%
  relocate(year.x, .after = Date_UTC) %>%
  relocate(year.y, .after = Date_UTC) %>%
  filter(year.x != 2023) %>%
  filter()

investigate <- eDNAindxEnvData %>% select(Species, DateMatch, Date_UTC, Date_local_hr,
# source, temperature, DO, E_no, Rosette_position, Amplicon)

system("say Data Join Complete")

# Export the joined data
write_csv(eDNAindxEnvData, here("eDNA_Index_Hypoxia", "Data",
# "Copepod_eDNAindxEnvData.csv"))

# Make a version without the copepods that are only detected in 2023 because those plots
# are unhelpful
eDNAindxEnvData_clean <- eDNAindxEnvData %>%
  filter(Colmax > 0)

write_csv(eDNAindxEnvData_clean, here("eDNA_Index_Hypoxia", "Data",
# "Copepod_eDNAindxEnvData_clean.csv"))

```

Re-calculate eDNA index for 2021-2022 only

```

index1dummyYr <- yrReads %>% # Quick proportions
dplyr::group_by(SampleId) %>%
  mutate(Tot = sum(nReads),
    Row.sums = nReads / Tot) %>% # calculate proportions - 0 reads/0 total = NaN,
# need to replace with 0 to make max() work
  relocate(c(Tot, Row.sums), .after = SampleId)

yrReads_PCRcomb <- PCR_reps_combine(yrReads) # Spot check E1325 was correct

## [1] "ESV"
## [2] "sequence"
## [3] "Kingdom"
## [4] "Phylum"
## [5] "Class"

```

```

## [6] "Order"
## [7] "Family"
## [8] "Genus"
## [9] "Species"
## [10] "JV243.2"
## [11] "Barcode.x"
## [12] "Barcode_mod"
## [13] "PI"
## [14] "E_no"
## [15] "Cruise1"
## [16] "Biological_Replicate"
## [17] "Negative_control"
## [18] "Positive_control"
## [19] "Cruise_ID_short"
## [20] "Cruise_ID_long"
## [21] "Field_Sample_Name"
## [22] "Cast_No."
## [23] "Rosette_position"
## [24] "Field_collection_method"
## [25] "Preservation"
## [26] "Area.within.region"
## [27] "Station"
## [28] "Depth_m"
## [29] "Sample_volume_ml"
## [30] "Personnel_responsible_for_Collecting_Sample"
## [31] "Field_Notes"
## [32] "Date_UTC"
## [33] "Date_local"
## [34] "year"
## [35] "Lat_dec"
## [36] "Lon_dec"
## [37] "Run"
## [38] "Client"
## [39] "ClientId"
## [40] "BatchId"
## [41] "Replicate"
## [42] "Amplicon"
## [43] "F_name"
## [44] "primers_fwd"
## [45] "R_name"
## [46] "primers_rev"
## [47] "PE"
## [48] "primers_pad"
## [49] "primers_errors"
## [50] "merge_maxdiffs"
## [51] "merge_maxdiffpct"
## [52] "merge_minovlen"
## [53] "ExpL"
## [54] "merge_minlen"
## [55] "merge maxlen"
## [56] "filter_max_ee"
## [57] "denoise_alpha"
## [58] "denoise_minsize"
## [59] "tax_ID"

```

```

## [60] "tax_maxaccepts"
## [61] "tax_maxrejects"
## [62] "tax_maxhits"
## [63] "tax_qcov"
## [64] "tax_minwords"
## [65] "tax_wordlen"
## [66] "Marker"

## `summarise()` has grouped output by 'ESV', 'sequence', 'Kingdom', 'Phylum',
## 'Class', 'Order', 'Family', 'Genus', 'Species', 'JV243.2', 'Barcode.x',
## 'Barcode_mod', 'PI', 'E_no', 'Cruise1', 'Biological_Replicate',
## 'Negative_control', 'Positive_control', 'Cruise_ID_short', 'Cruise_ID_long',
## 'Field_Sample_Name', 'Cast_No.', 'Rosette_position', 'Field_collection_method',
## 'Preservation', 'Area.within.region', 'Station', 'Depth_m', 'Sample_volume_ml',
## 'Personnel_responsible_for_Collecting_Sample', 'Field_Notes', 'Date_UTC',
## 'Date_local', 'year', 'Lat_dec', 'Lon_dec', 'Run', 'Client', 'ClientId',
## 'BatchId', 'Replicate', 'Amplicon', 'F_name', 'primers_fwd', 'R_name',
## 'primers_rev', 'PE', 'primers_pad', 'primers_errors', 'merge_maxdiffs',
## 'merge_maxdiffpct', 'merge_minovlen', 'ExpL', 'merge_minlen', 'merge_maxlen',
## 'filter_max_ee', 'denoise_alpha', 'denoise_minsize', 'tax_ID',
## 'tax_maxaccepts', 'tax_maxrejects', 'tax_maxhits', 'tax_qcov', 'tax_minwords',
## 'tax_wordlen'. You can override using the `groups` argument.

# Second, combine by species

id_cols <- colnames(yrReads_PCRcomb) # Pull out column names
id_cols <- id_cols[! id_cols %in% c("X", "ESV", "sequence", "nReads")] # Remove ESV +
# nreads because those are different within species

allReads_speciesYr <- yrReads_PCRcomb %>%
  group_by_at(id_cols) %>% # group_by_at can take a vector
  summarise(TotalnReads = sum(nReads)) %>% # Removed , avgpctMatch = mean(pctMatch)
# because I had to remove pctMatch to combine PCR replicates. Sum nReads results in
# taking the sum of all ESVs within a species
  relocate(TotalnReads, .after = PI)

## `summarise()` has grouped output by 'Kingdom', 'Phylum', 'Class', 'Order',
## 'Family', 'Genus', 'Species', 'JV243.2', 'Barcode.x', 'Barcode_mod', 'PI',
## 'E_no', 'Cruise1', 'Biological_Replicate', 'Negative_control',
## 'Positive_control', 'Cruise_ID_short', 'Cruise_ID_long', 'Field_Sample_Name',
## 'Cast_No.', 'Rosette_position', 'Field_collection_method', 'Preservation',
## 'Area.within.region', 'Station', 'Depth_m', 'Sample_volume_ml',
## 'Personnel_responsible_for_Collecting_Sample', 'Field_Notes', 'Date_UTC',
## 'Date_local', 'year', 'Lat_dec', 'Lon_dec', 'Run', 'Client', 'ClientId',
## 'BatchId', 'Replicate', 'Amplicon', 'F_name', 'primers_fwd', 'R_name',
## 'primers_rev', 'PE', 'primers_pad', 'primers_errors', 'merge_maxdiffs',
## 'merge_maxdiffpct', 'merge_minovlen', 'ExpL', 'merge_minlen', 'merge_maxlen',
## 'filter_max_ee', 'denoise_alpha', 'denoise_minsize', 'tax_ID',
## 'tax_maxaccepts', 'tax_maxrejects', 'tax_maxhits', 'tax_qcov', 'tax_minwords',
## 'tax_wordlen'. You can override using the `groups` argument.

write_csv(allReads_speciesYr, here("eDNA_Index_Hypoxia", "Data",
# "CopepodReads_Species_no23.csv")) # Gonna want this later

all_index1Yr <- allReads_speciesYr %>%

```

```

dplyr::group_by(E_no) %>% # Group by E-number, because sampleID had to be removed in
→ the PCR replicate combination step
  mutate(Tot = sum(TotalnReads),
        Row.sums = TotalnReads / Tot) %>% # calculate proportions - 0 reads/0 total =
        → NaN, need to replace with 0 to make max() work
  relocate(c(TotalnReads, Tot, Row.sums), .after = E_no) # Move it somewhere I can see
→ the damn thing

yr_eDNA_index <- all_index1Yr %>%
  dplyr::group_by(Species) %>%
  mutate(Row.sums = case_when(Row.sums == "NaN" ~ 0,
                               .default = Row.sums)) %>% # Make 0/0 = 0 and not
→ NaN
  mutate(Colmax = max(Row.sums), Normalized.reads = Row.sums / Colmax) %>% # transforms
→ raw number of reads to eDNA index. Creates same divide by 0 error, so:
  mutate(Normalized.reads = case_when(Normalized.reads == "NaN" ~ 0,
                                       .default = Normalized.reads)) %>% # Make 0/0 = 0
→ and not NaN
  relocate(c(Colmax, Normalized.reads), .after = Row.sums) # Move it somewhere I can see
→ the damn thing

# Filter to only copepods
copepod_eDNA_indexYr <- yr_eDNA_index %>%
  filter(Species %in% copepods) %>%
  rename(eDNA_index = `Normalized.reads`) # to make parsing this easier

write_csv(copepod_eDNA_indexYr, here("eDNA_Index_Hypoxia", "Data",
→ "Copepod_eDNA_index_no23.csv")) # Gonna want this later

# envData = EnvironmentalDataset1
# Based on eDNAxp02.Rmd from summer project
# copepod_eDNA_index is still essentially a version of allReads so I can use the same
→ code I did with allReads_Species

DetectRdYr <- copedp_eDNA_indexYr %>%
  mutate(DateMatch = round_date(Date_UTC, unit = "10 minutes"), Date_local_hr =
→ round_date(Date_local, unit = "hour")) # Spot check - looks good.

eDNAindxEnvDataYr <- left_join(DetectRdYr, EnvRd, by = join_by(DateMatch)) %>%
  relocate(date, .after = Date_UTC) %>%
  relocate(year.x, .after = Date_UTC) %>%
  relocate(year.y, .after = Date_UTC) %>%
  filter(year.x != 2023) %>%
  filter()

investigateYr <- eDNAindxEnvDataYr %>% select(Species, DateMatch, Date_UTC,
→ Date_local_hr, source, temperature, DO, E_no, Rosette_position, Amplicon)

system("say Data Join Without 2023 Complete")

# Export the joined data
write_csv(eDNAindxEnvDataYr, here("eDNA_Index_Hypoxia", "Data",
→ "Copepod_eDNAindxEnvData_no23.csv"))

```

```

# Make a version without the copepods that are only detected in 2023 because those plots
# are unhelpful
eDNAindxEnvData_cleanYr <- eDNAindxEnvDataYr %>%
  filter(Colmax > 0)

write_csv(eDNAindxEnvData_cleanYr, here("eDNA_Index_Hypoxia", "Data",
  "Copepod_eDNAindxEnvData_clean_no23.csv"))

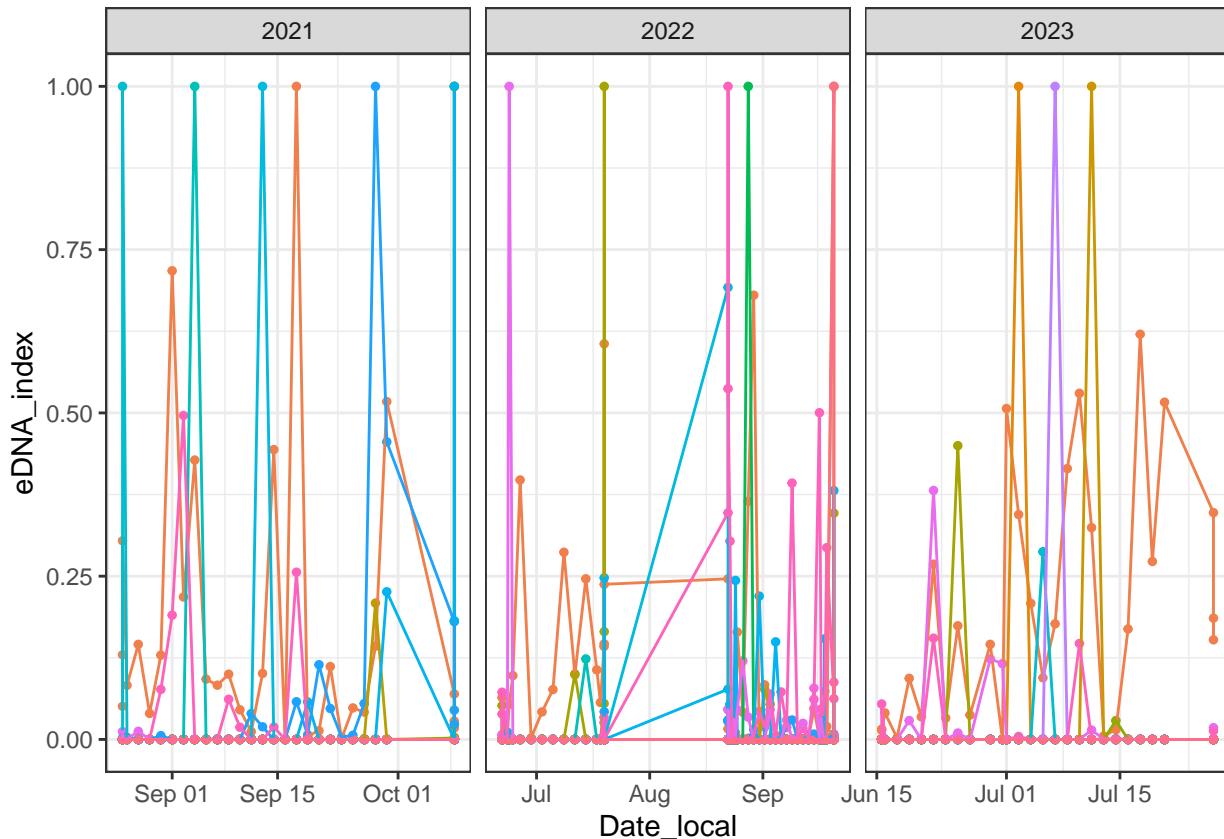
```

Plot eDNA index over time and oxygen

```

# Explore eDNA index by species
ggplot(copepod_eDNA_index, aes(x = Date_local, y = eDNA_index, color = Species)) +
  geom_point(show.legend = F, size = 1) +
  facet_wrap(facets = vars(year(Date_local)), scales = "free_x") +
  geom_line(show.legend = F) +
  theme_bw()

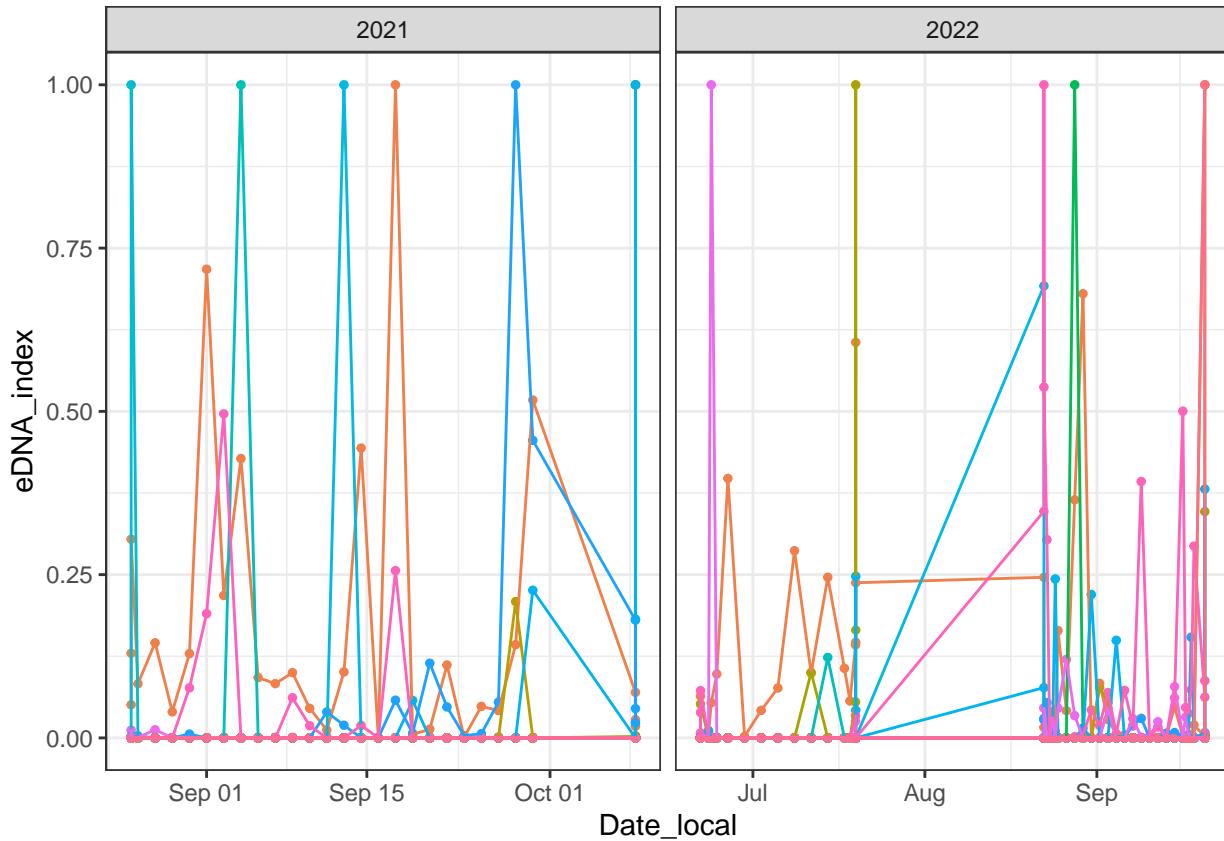
```



```

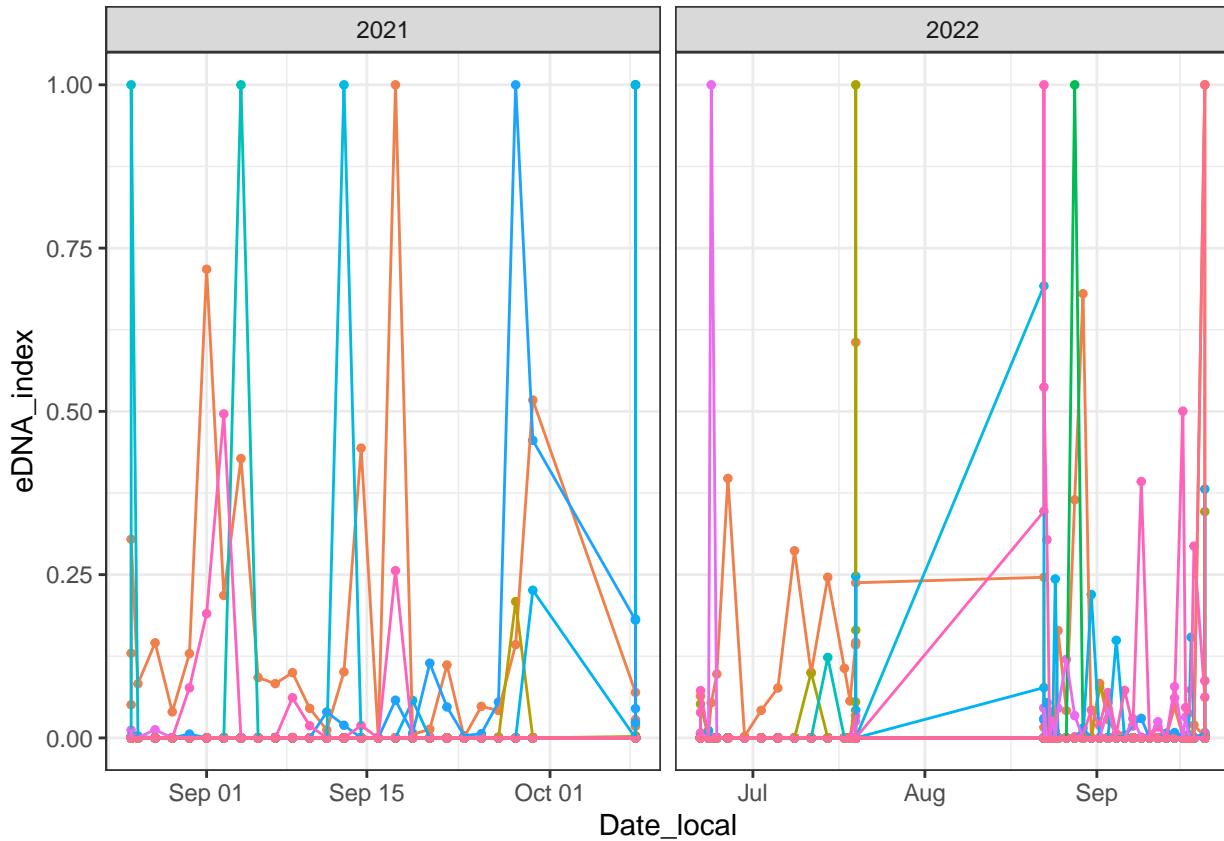
ggplot(eDNAindxEnvData, aes(x = Date_local, y = eDNA_index, color = Species)) +
  geom_point(show.legend = F, size = 1) +
  facet_wrap(facets = vars(year(Date_local)), scales = "free_x") +
  geom_line(show.legend = F) +
  theme_bw()

```



Seems legit!

```
ggplot(eDNAindxEnvDataYr, aes(x = Date_local, y = eDNA_index, color = Species)) +
  geom_point(show.legend = F, size = 1) +
  facet_wrap(facets = vars(year(Date_local)), scales = "free_x") +
  geom_line(show.legend = F) +
  theme_bw()
```



```

source(here("eDNA_Index_Hypoxia", "eDNA_EnvGraphFunction.R"))

## ggbreak v0.1.2
##
## If you use ggbreak in published research, please cite the following
## paper:
## S Xu, M Chen, T Feng, L Zhan, L Zhou, G Yu. Use ggbreak to effectively
## utilize plotting space to deal with large datasets and outliers.
## Frontiers in Genetics. 2021, 12:774846. doi: 10.3389/fgene.2021.774846
## Warning in geom_rect(data = sampleHighlight, inherit.aes = FALSE, mapping =
## aes(xmin = x1b, : Ignoring unknown parameters: `stroke`
## Warning in geom_rect(data = sampleHighlight, inherit.aes = FALSE, mapping =
## aes(xmin = x2b, : Ignoring unknown parameters: `stroke`
## Warning in geom_rect(data = sampleHighlight, inherit.aes = FALSE, mapping =
## aes(xmin = x3b, : Ignoring unknown parameters: `stroke`

eDNAGraph(eDNAindxEnvData_clean,
           envCond = "DO",
           envCondName = "Oxygen",
           filepath = here("eDNA_Index_Hypoxia", "Plots", "eDNAXDO"),
           ylab = "Dissolved Oxygen (mg/L)",
           widthpx = 3000, # make it longer
           threshold = F,
           thresholdLvl = 2
)

```

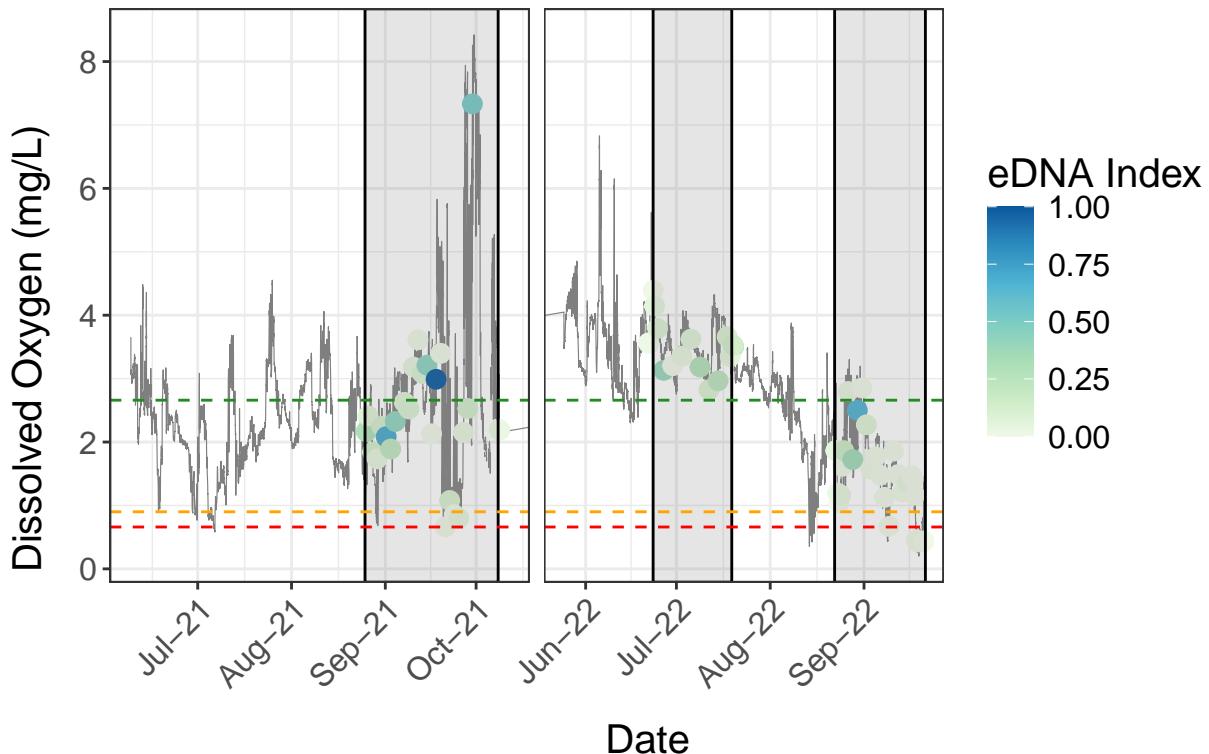
```

## [1] "HEADS UP: Date/time must be called exactly date and be in POSIXct, and envCond must be entered as a character vector"
## [1] "If you don't want a threshold line, set threshold = F instead of setting a thresholdLvl"
## [1] "Also for some reason you have to press 1 to confirm this function. Don't worry about it."
## [1] "Acartia longiremis"
## [1] "Acartia longiremis eDNA Index vs Oxygen"

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```

Acartia longiremis eDNA Index vs Oxygen

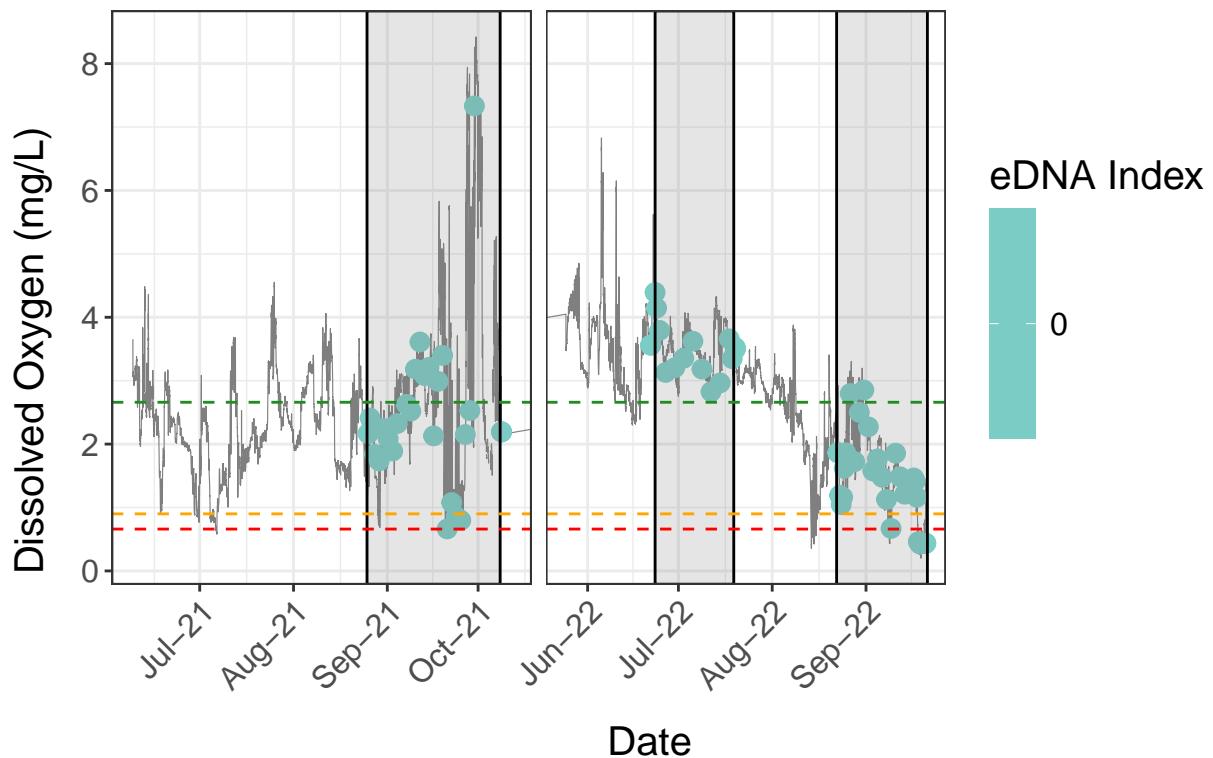


```

## [1] "Ameira sp. FHL 1"
## [1] "Ameira sp. FHL 1 eDNA Index vs Oxygen"

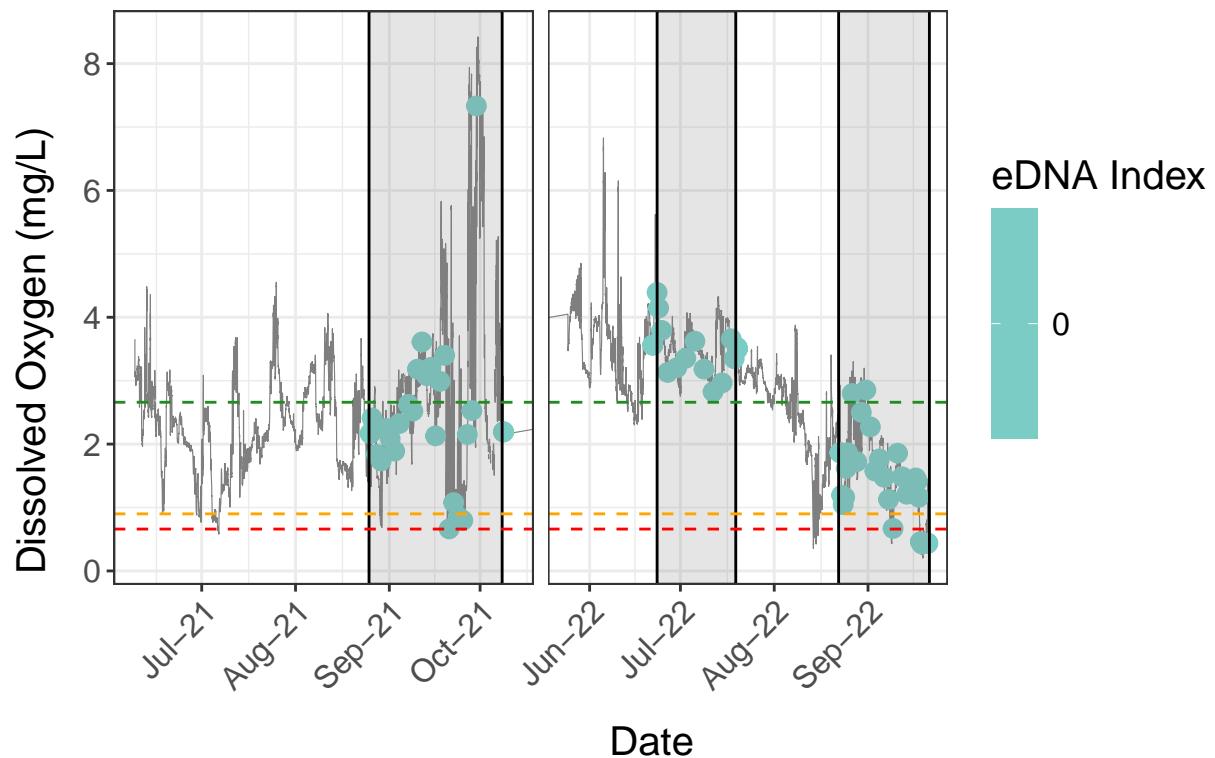
```

Ameira sp. FHL 1 eDNA Index vs Oxygen



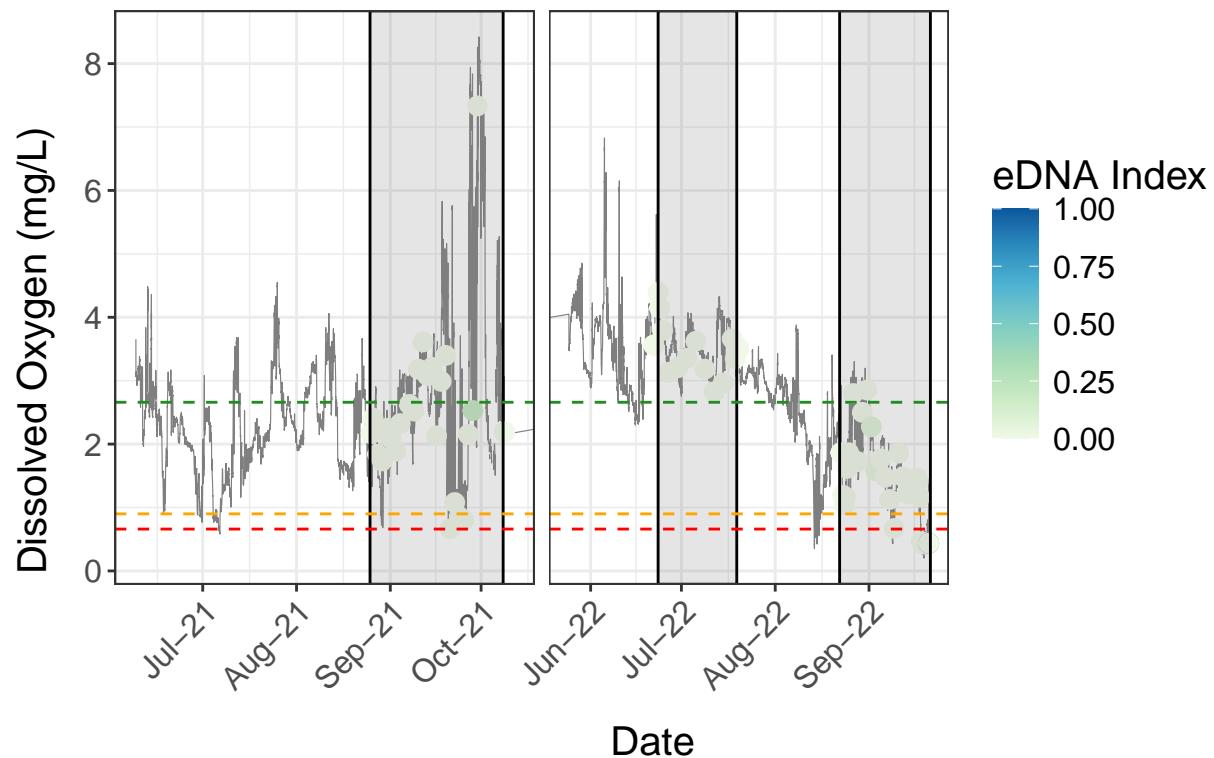
```
## [1] "Calanus marshallae"  
## [1] "Calanus marshallae eDNA Index vs Oxygen"
```

Calanus marshallae eDNA Index vs Oxygen



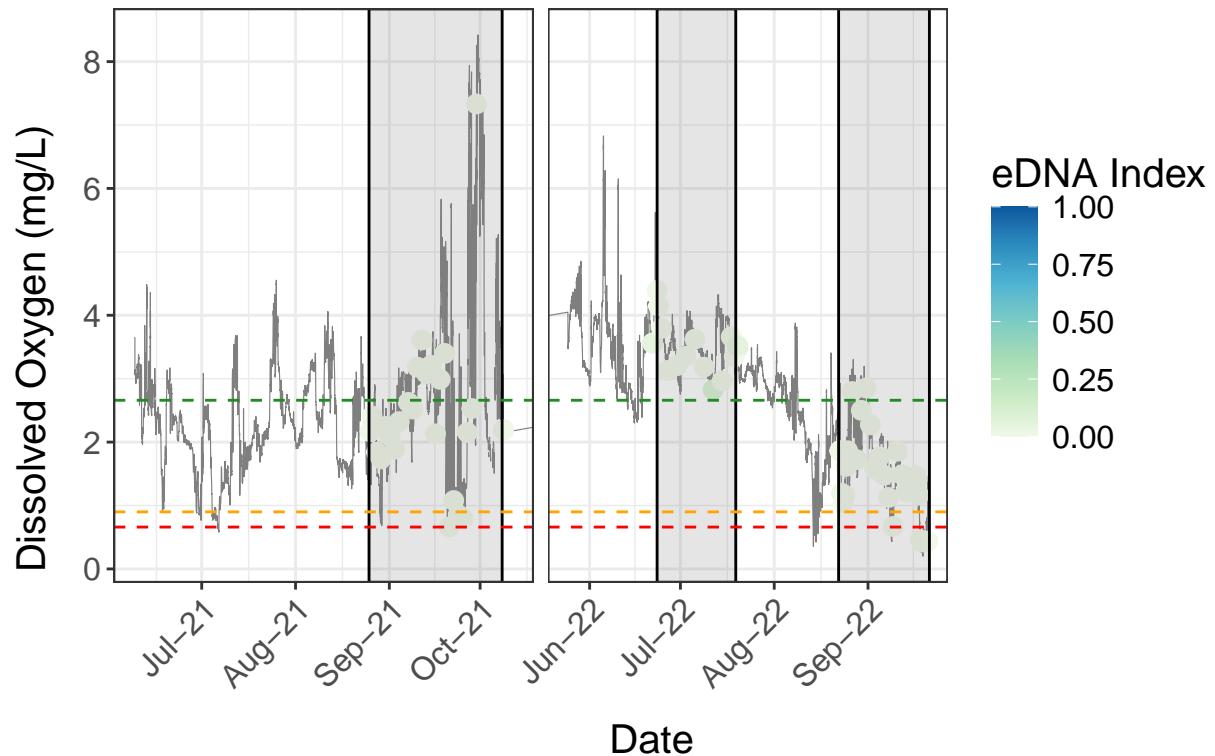
```
## [1] "Calanus pacificus"  
## [1] "Calanus pacificus eDNA Index vs Oxygen"
```

Calanus pacificus eDNA Index vs Oxygen



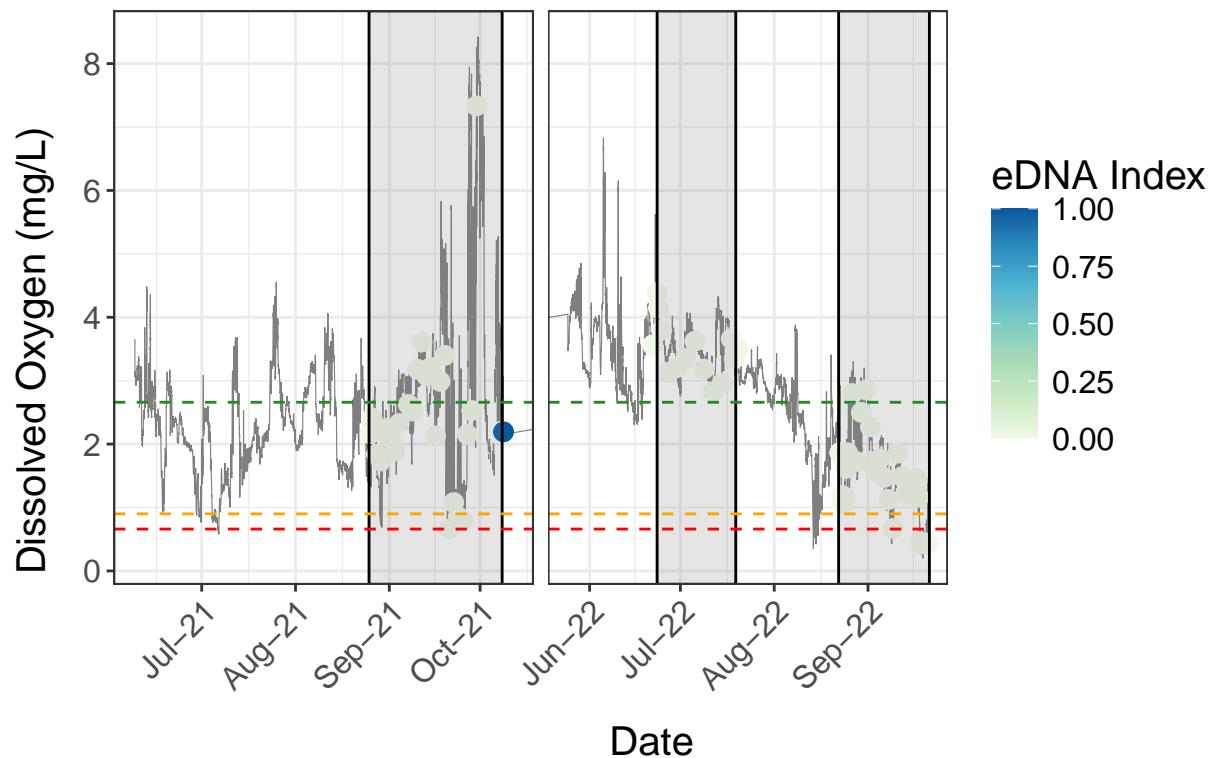
```
## [1] "Centropages abdominalis"  
## [1] "Centropages abdominalis eDNA Index vs Oxygen"
```

Centropages abdominalis eDNA Index vs Oxygen



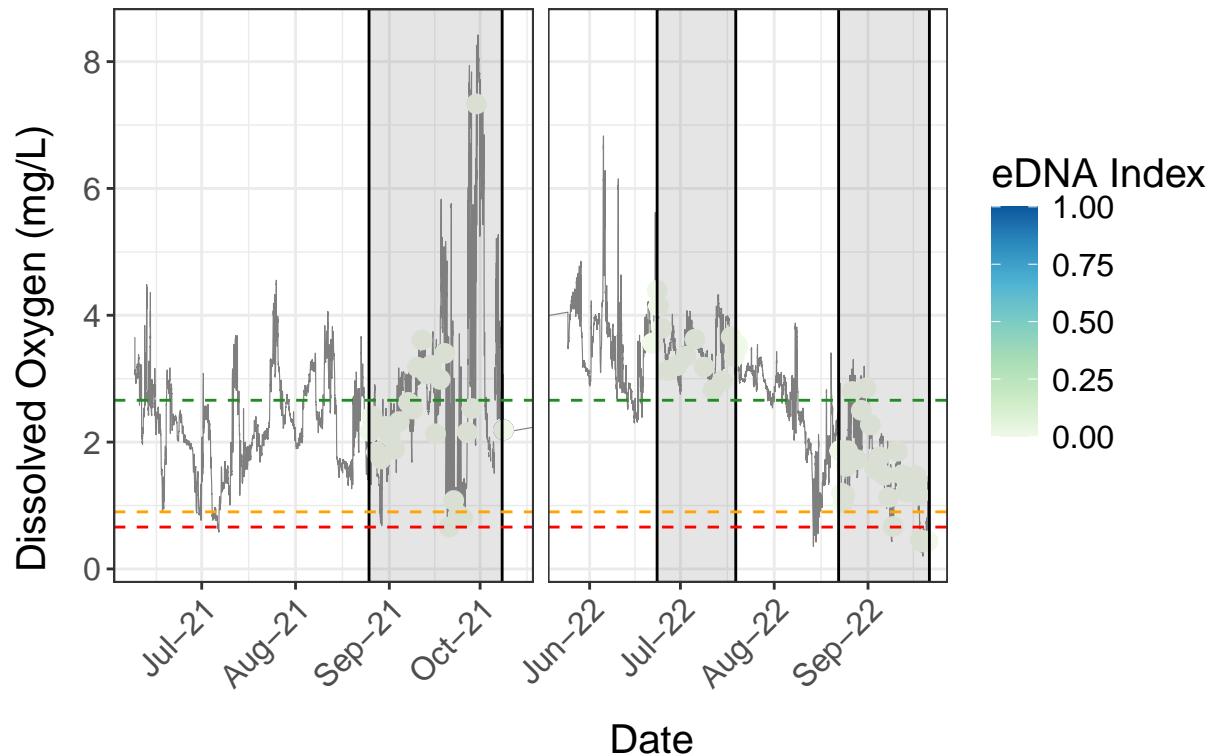
```
## [1] "Clausocalanus parapergens"  
## [1] "Clausocalanus parapergens eDNA Index vs Oxygen"
```

Clausocalanus parapergens eDNA Index vs Oxygen



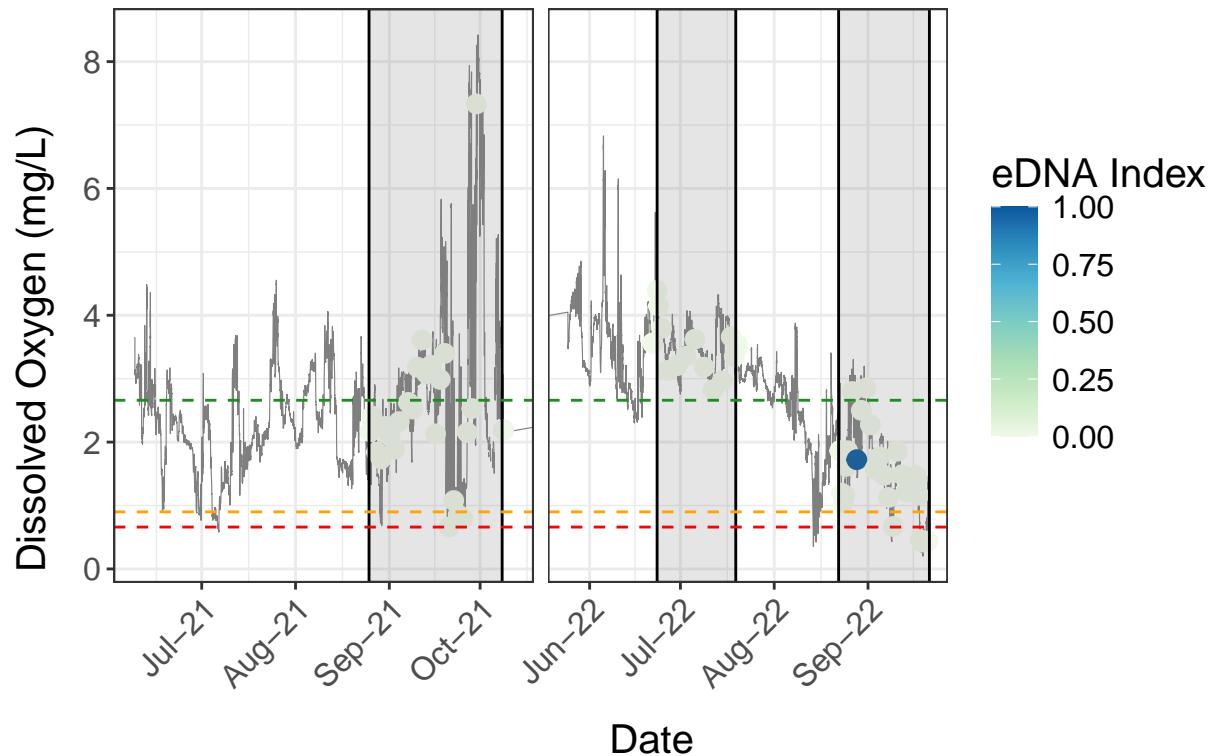
```
## [1] "Clausocalanus pergens"  
## [1] "Clausocalanus pergens eDNA Index vs Oxygen"
```

Clausocalanus pergens eDNA Index vs Oxygen



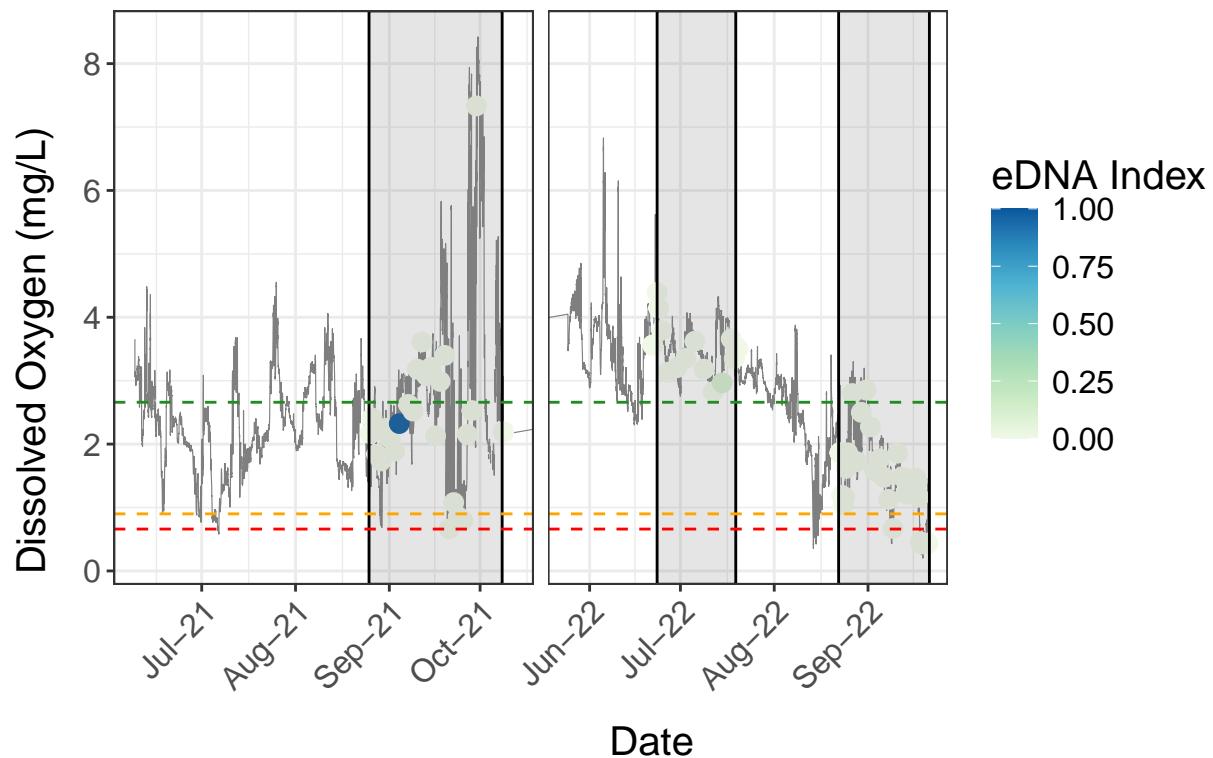
```
## [1] "Diacyclops incolotaenia"  
## [1] "Diacyclops incolotaenia eDNA Index vs Oxygen"
```

Diacyclops incolaenia eDNA Index vs Oxygen



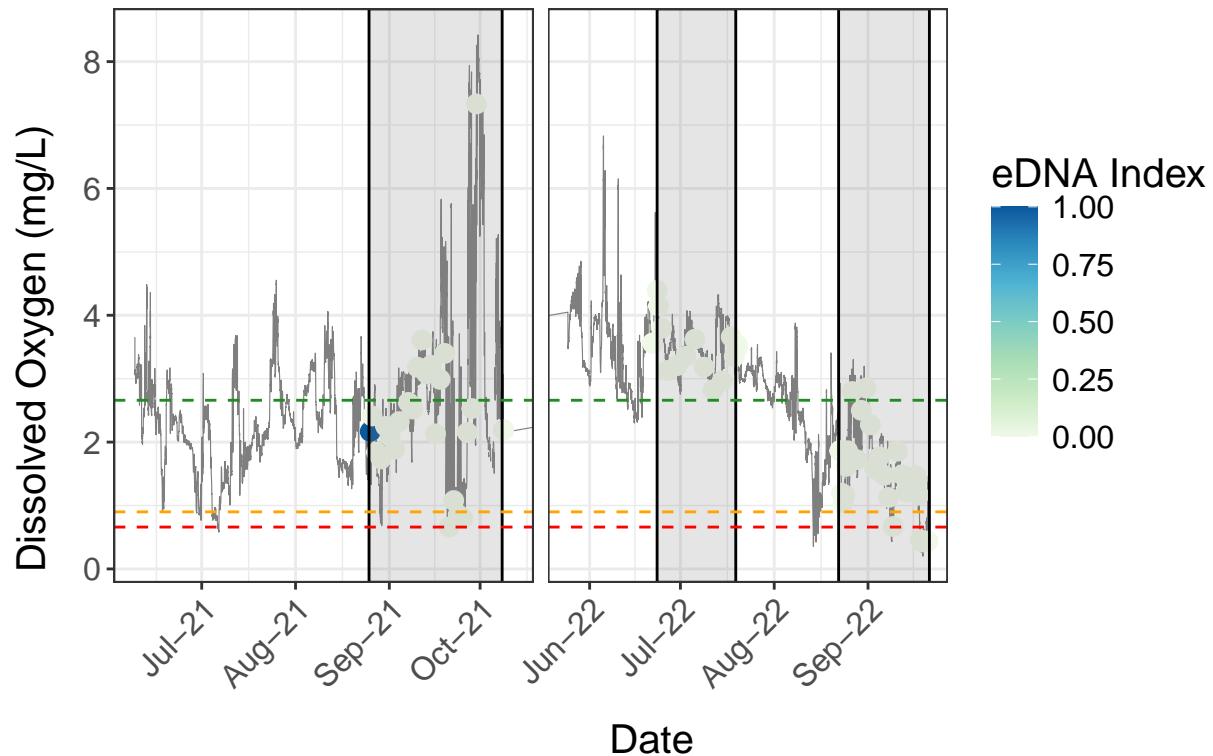
```
## [1] "Lucicutia flavigornis"  
## [1] "Lucicutia flavigornis eDNA Index vs Oxygen"
```

Lucicutia flavigornis eDNA Index vs Oxygen



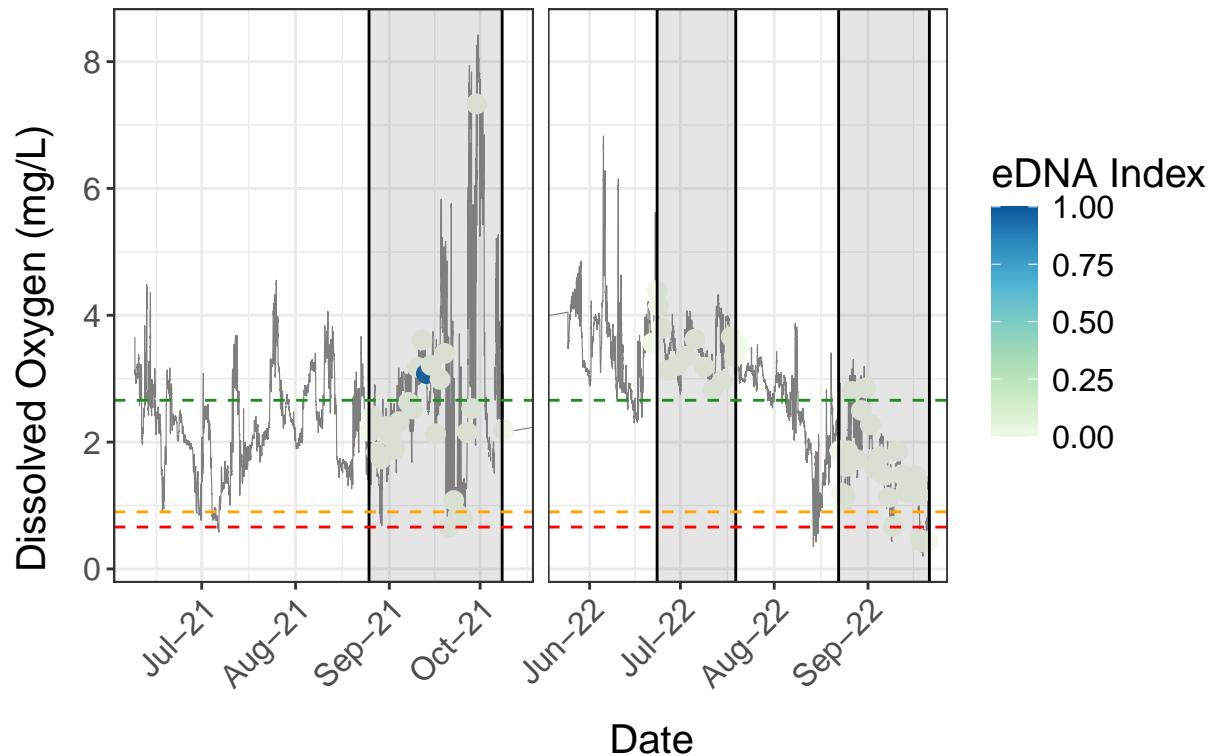
```
## [1] "Metridia lucens"  
## [1] "Metridia lucens eDNA Index vs Oxygen"
```

Metridia lucens eDNA Index vs Oxygen



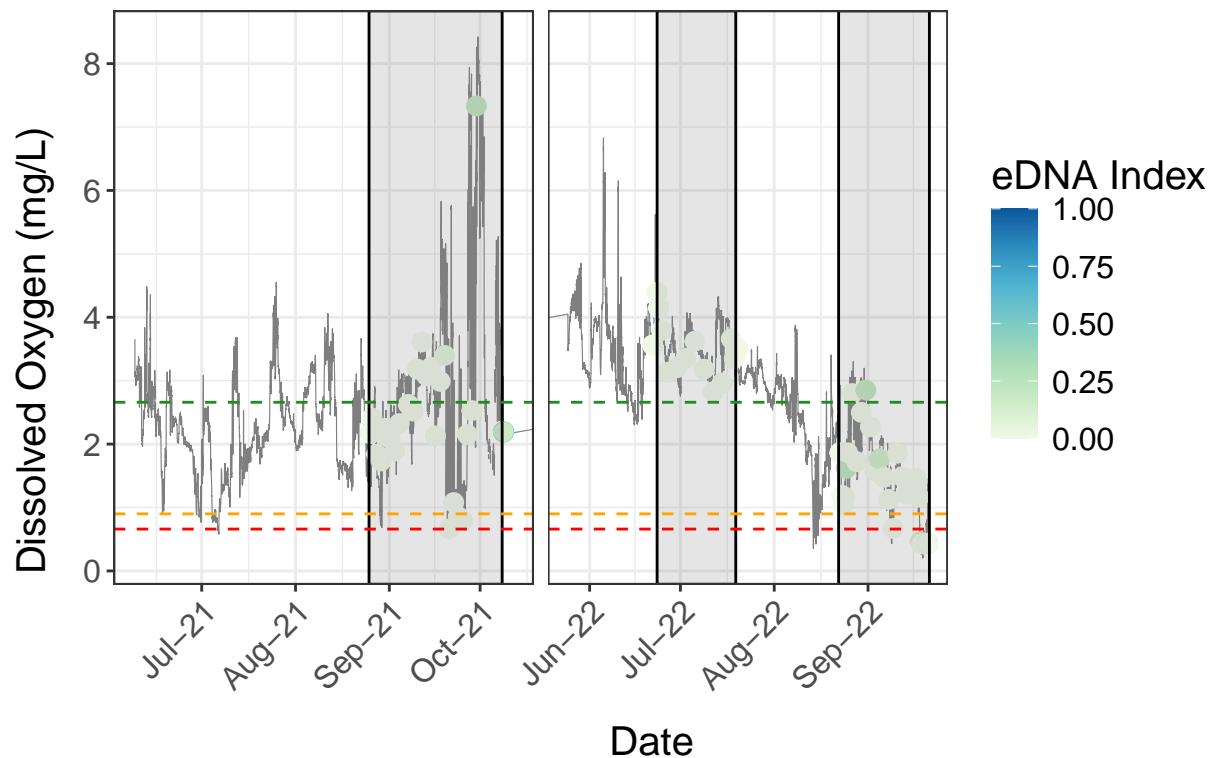
```
## [1] "Metridia pacifica"  
## [1] "Metridia pacifica eDNA Index vs Oxygen"
```

Metridia pacifica eDNA Index vs Oxygen



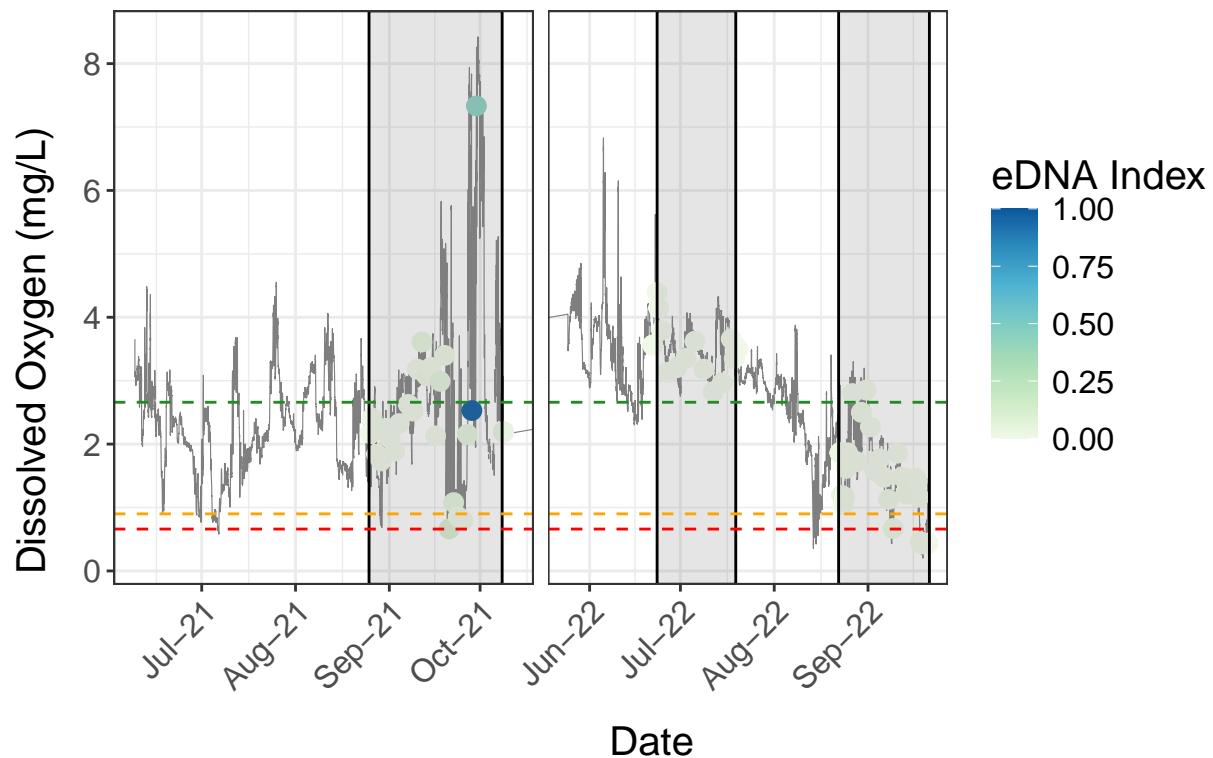
```
## [1] "Oithona similis"  
## [1] "Oithona similis eDNA Index vs Oxygen"
```

Oithona similis eDNA Index vs Oxygen



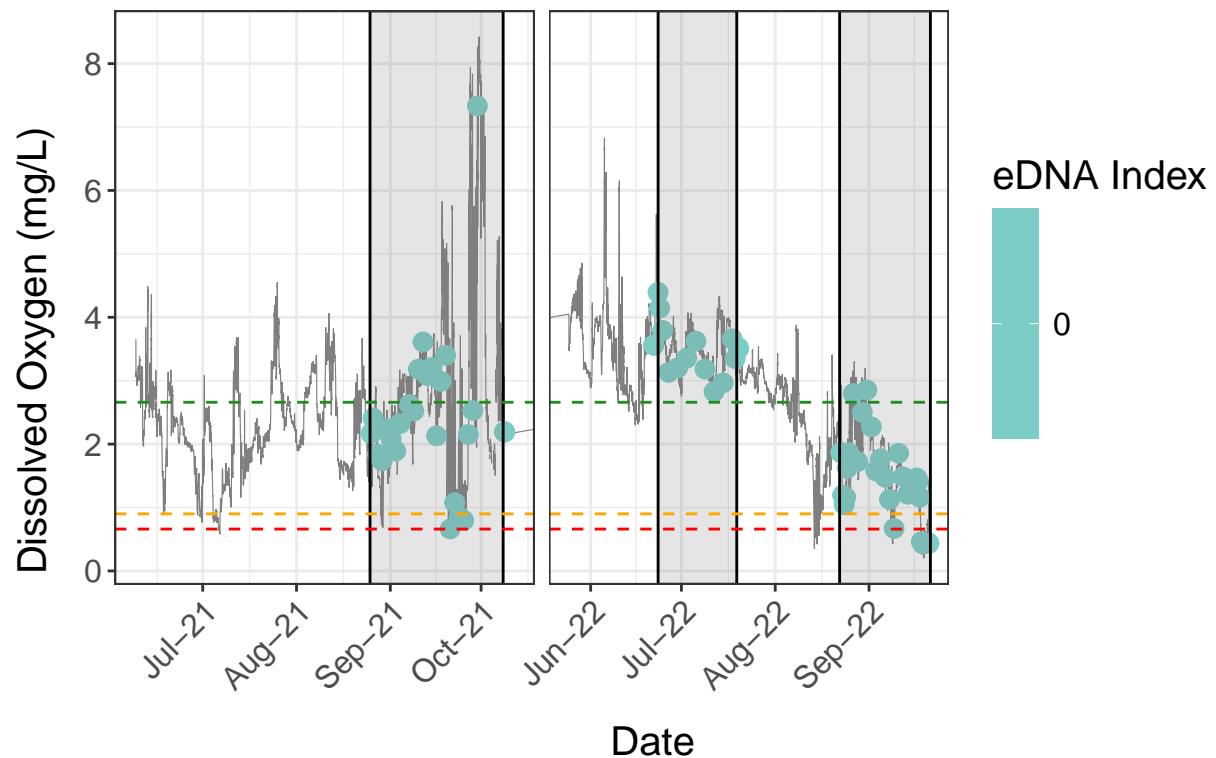
```
## [1] "Paracalanus sp. C AC-2013"
## [1] "Paracalanus sp. C AC-2013 eDNA Index vs Oxygen"
```

Paracalanus sp. C AC-2013 eDNA Index vs Oxygen



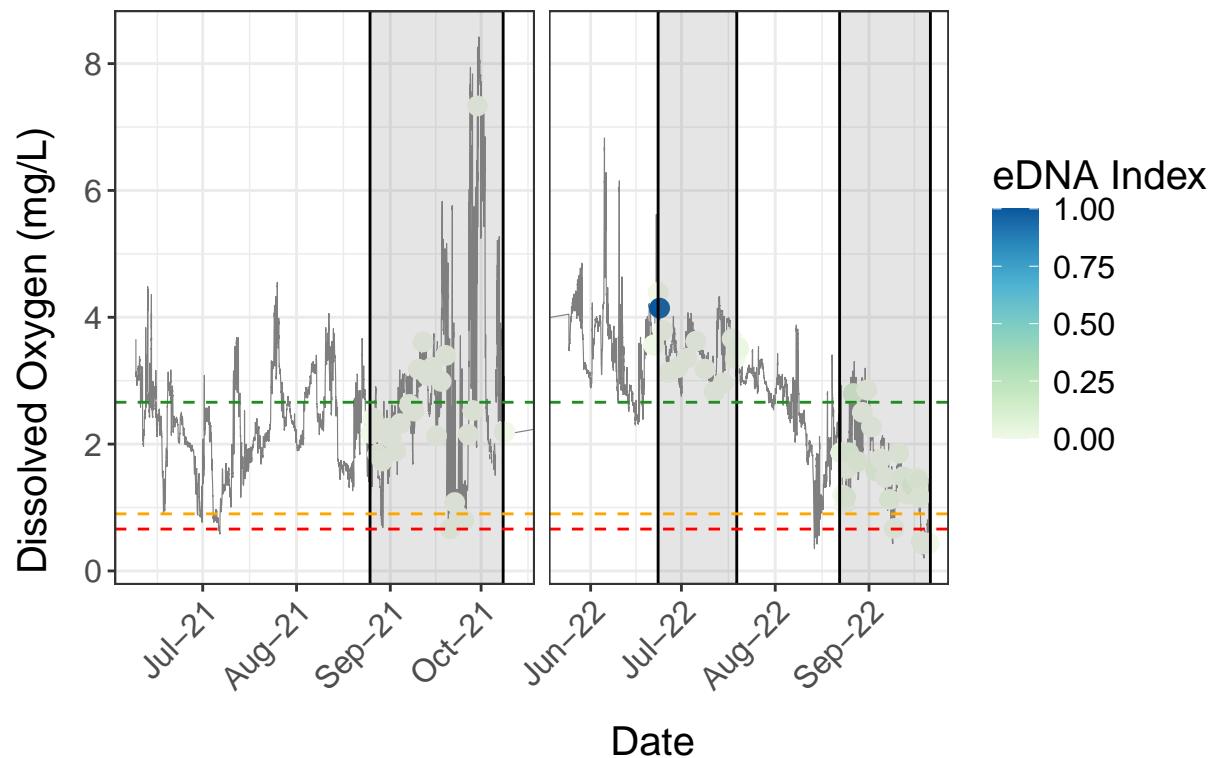
```
## [1] "Pseudobradya minor"  
## [1] "Pseudobradya minor eDNA Index vs Oxygen"
```

Pseudobradya minor eDNA Index vs Oxygen



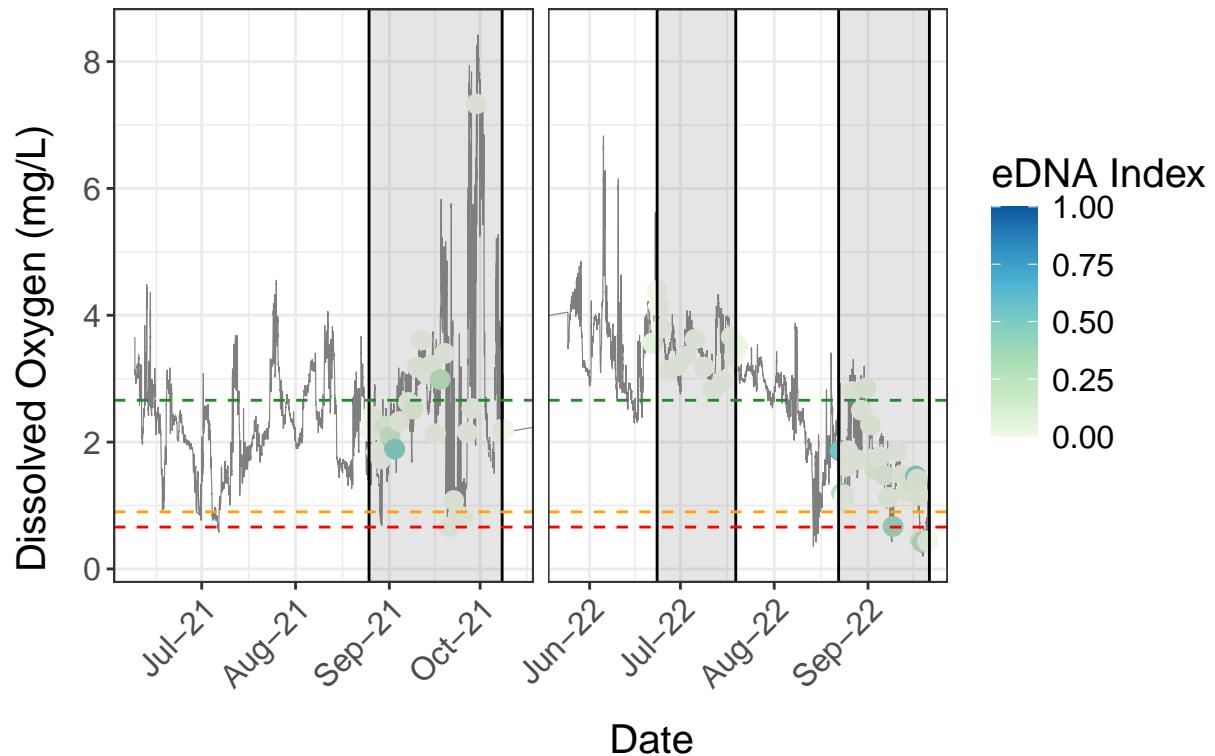
```
## [1] "Pseudocalanus newmani"  
## [1] "Pseudocalanus newmani eDNA Index vs Oxygen"
```

Pseudocalanus newmani eDNA Index vs Oxygen



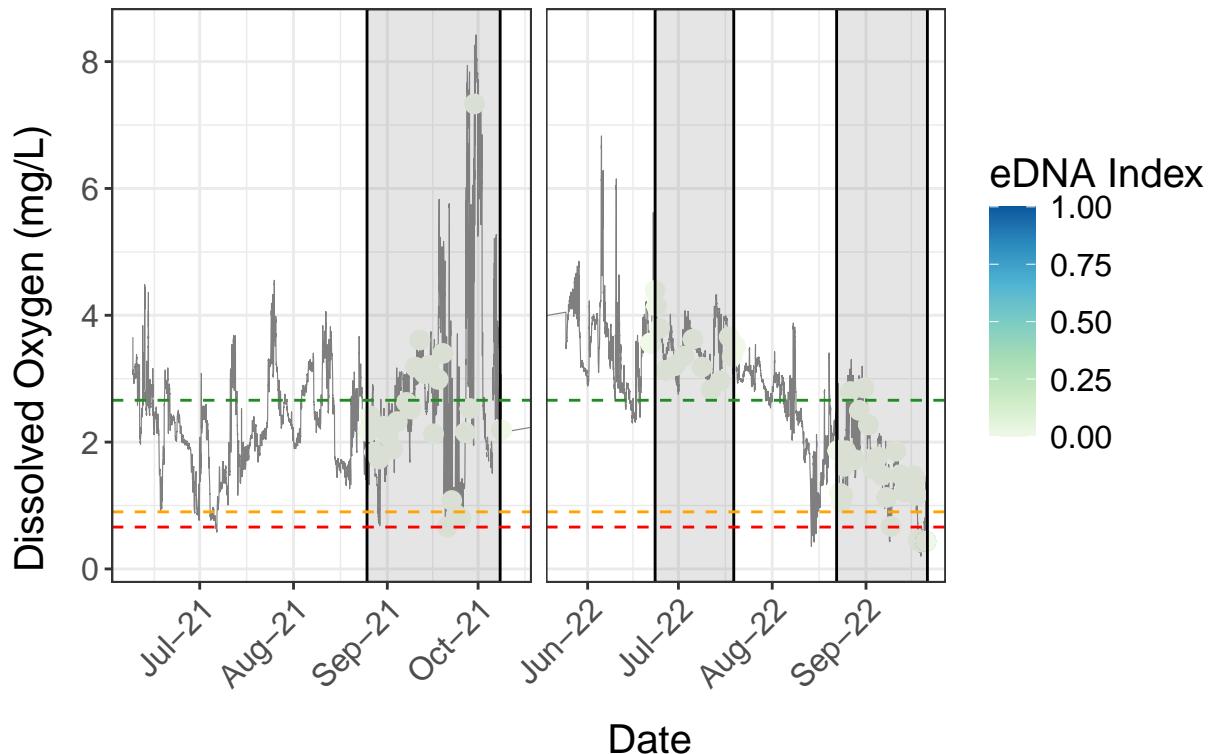
```
## [1] "Thermocyclops inversus"  
## [1] "Thermocyclops inversus eDNA Index vs Oxygen"
```

Thermocyclops inversus eDNA Index vs Oxygen



```
## [1] "Triconia minuta"  
## [1] "Triconia minuta eDNA Index vs Oxygen"
```

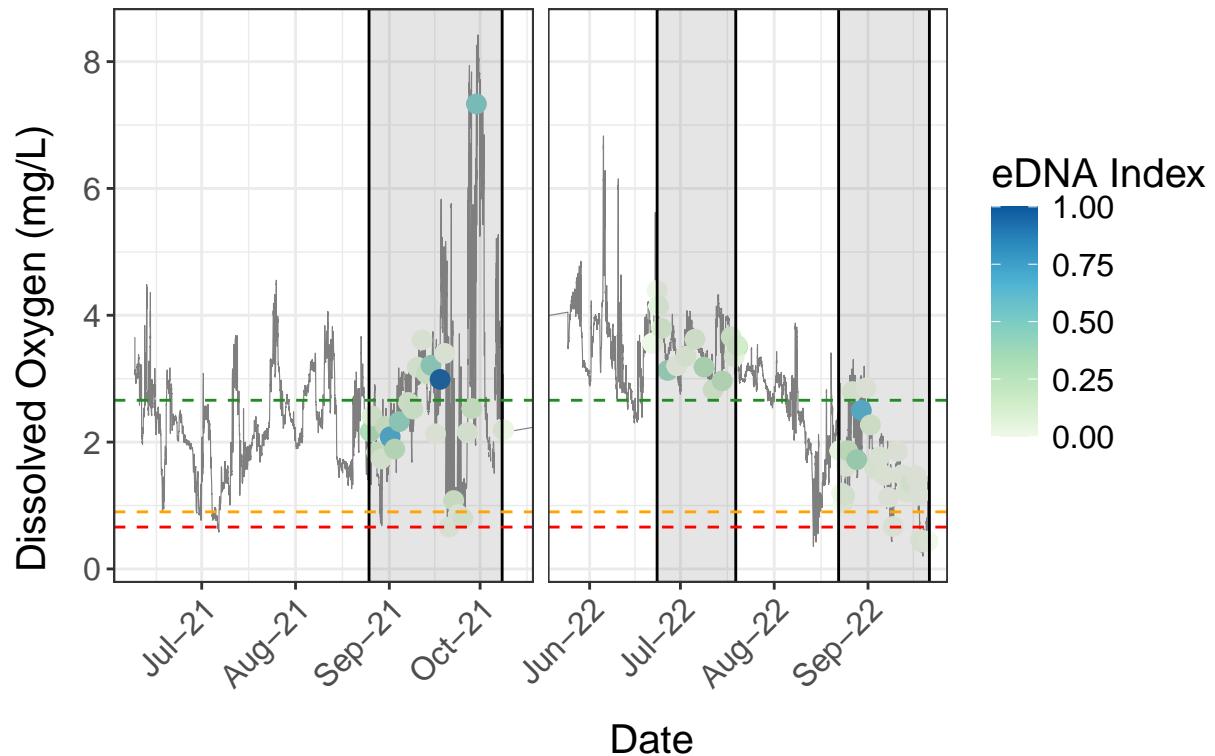
Triconia minuta eDNA Index vs Oxygen



```
eDNAGraph(eDNAindxEnvData_cleanYr, # ignoring 2023
            envCond = "DO",
            envCondName = "Oxygen",
            filepath = here("eDNA_Index_Hypoxia", "Plots", "eDNAXDO_no23"),
            ylab = "Dissolved Oxygen (mg/L)",
            widthpx = 3000, # make it longer
            threshold = F,
            thresholdLvl = 2
            )

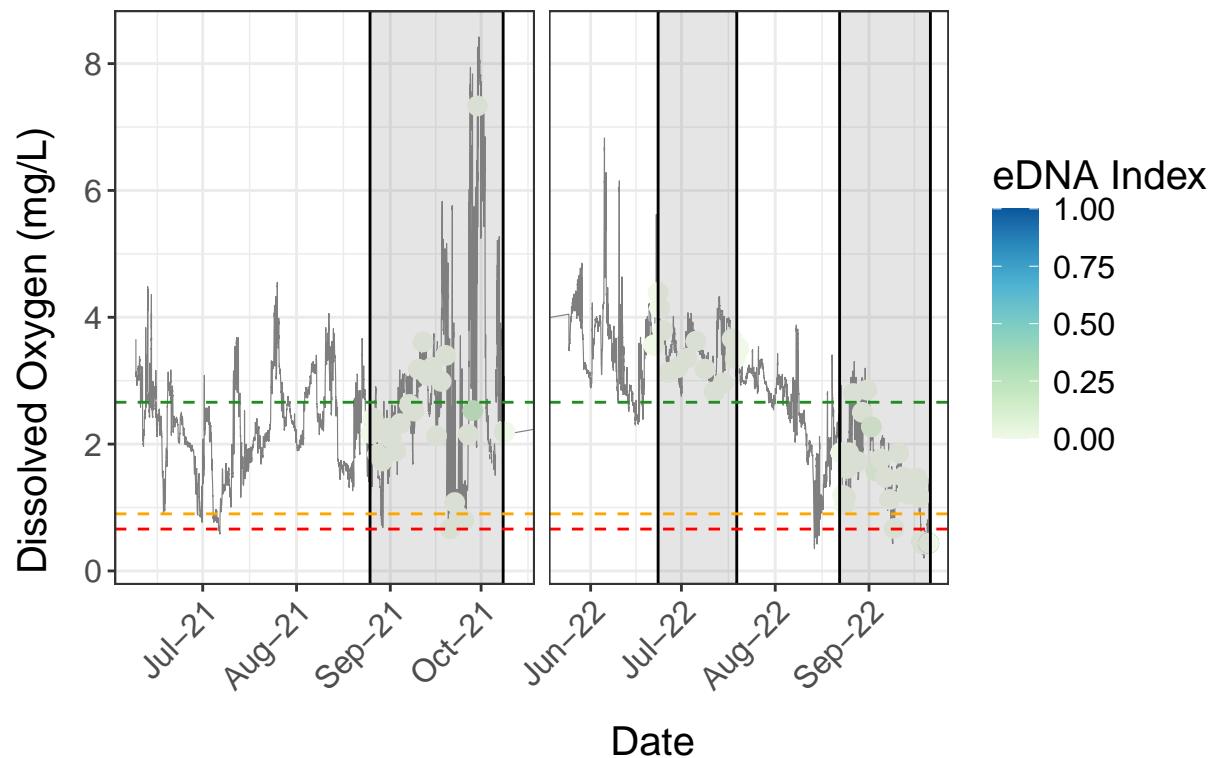
## [1] "HEADS UP: Date/time must be called exactly date and be in POSIXct, and envCond must be entered as a string"
## [1] "If you don't want a threshold line, set threshold = F instead of setting a thresholdLvl"
## [1] "Also for some reason you have to press 1 to confirm this function. Don't worry about it."
## [1] "Acartia longiremis"
## [1] "Acartia longiremis eDNA Index vs Oxygen"
```

Acartia longiremis eDNA Index vs Oxygen



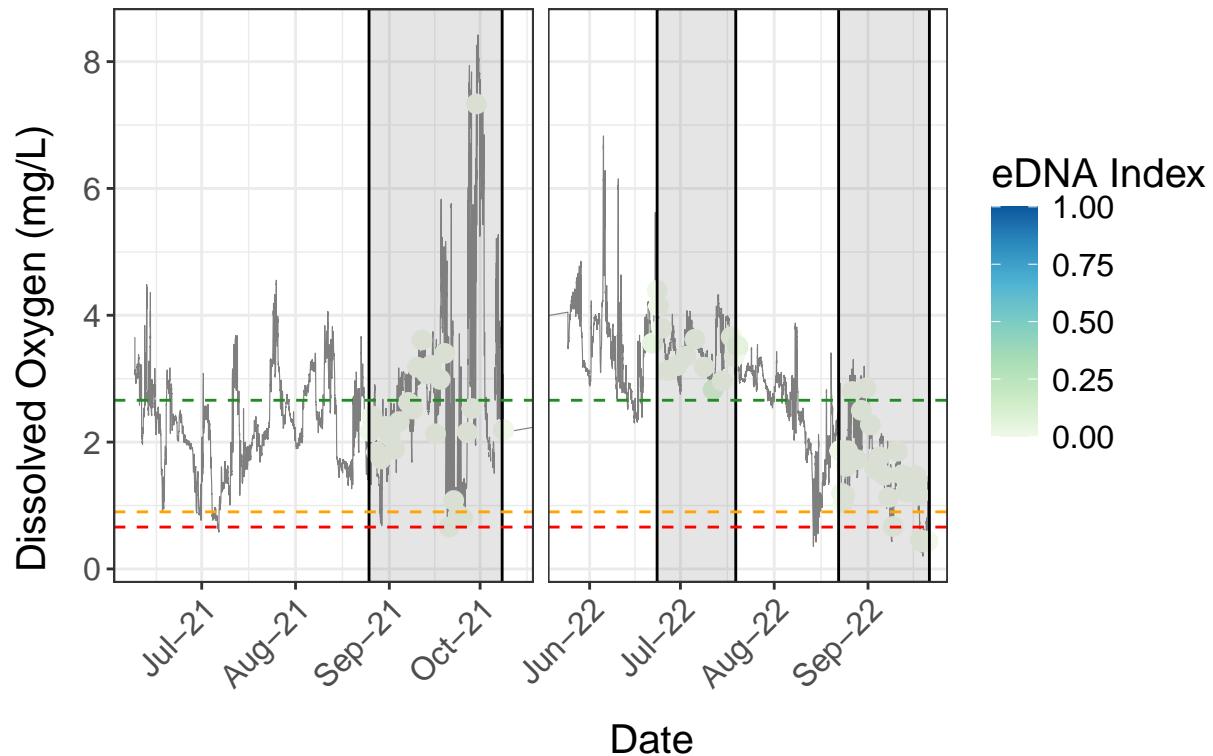
```
## [1] "Calanus pacificus"  
## [1] "Calanus pacificus eDNA Index vs Oxygen"
```

Calanus pacificus eDNA Index vs Oxygen



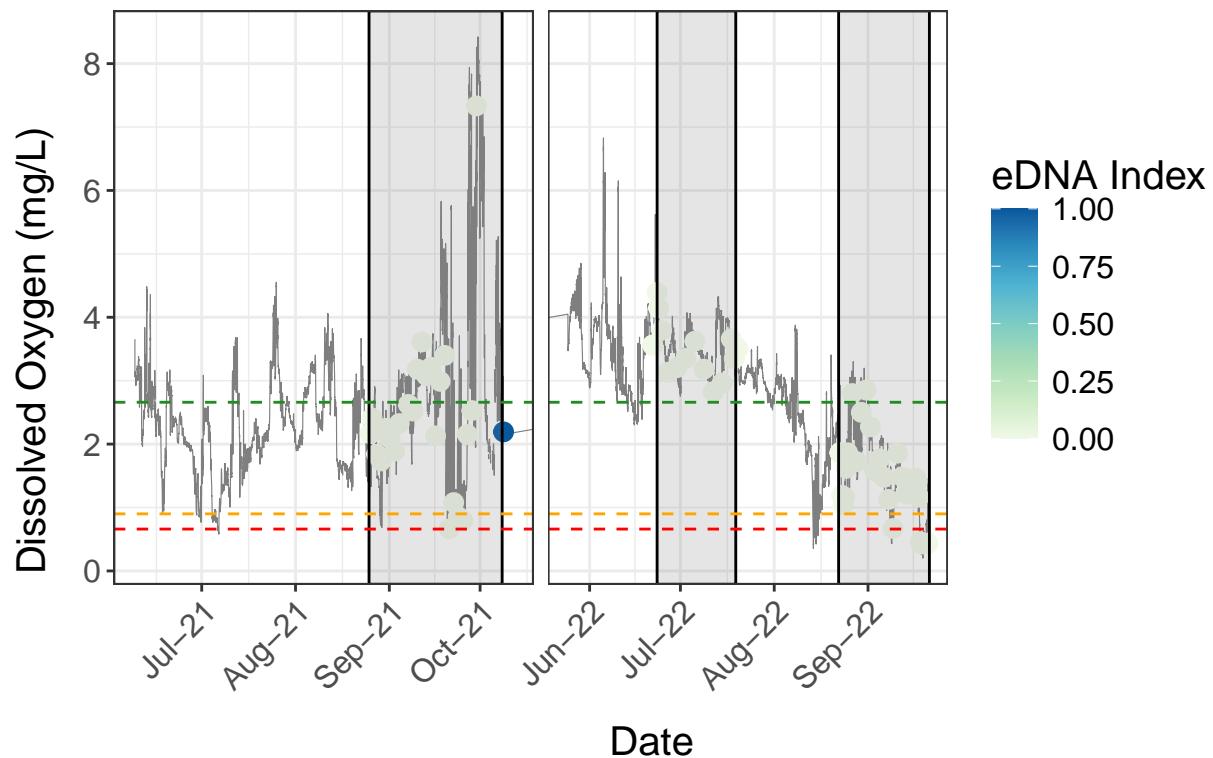
```
## [1] "Centropages abdominalis"  
## [1] "Centropages abdominalis eDNA Index vs Oxygen"
```

Centropages abdominalis eDNA Index vs Oxygen



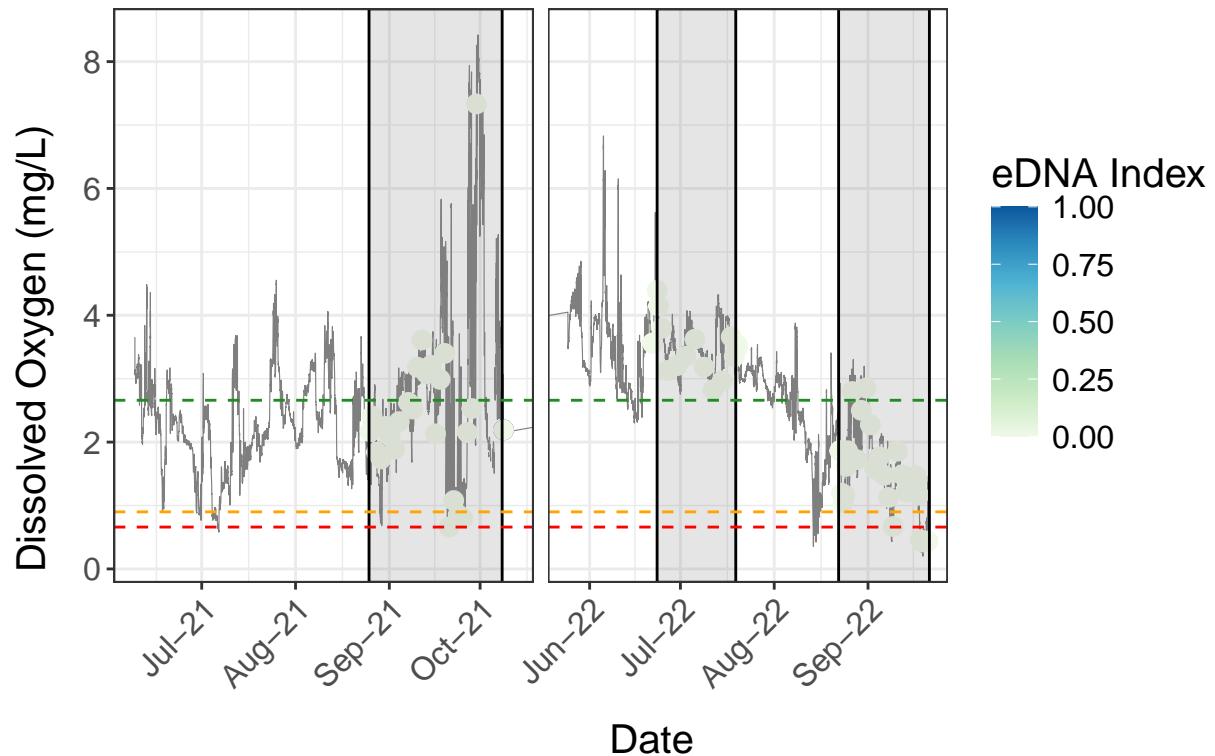
```
## [1] "Clausocalanus parapergens"  
## [1] "Clausocalanus parapergens eDNA Index vs Oxygen"
```

Clausocalanus parapergens eDNA Index vs Oxygen



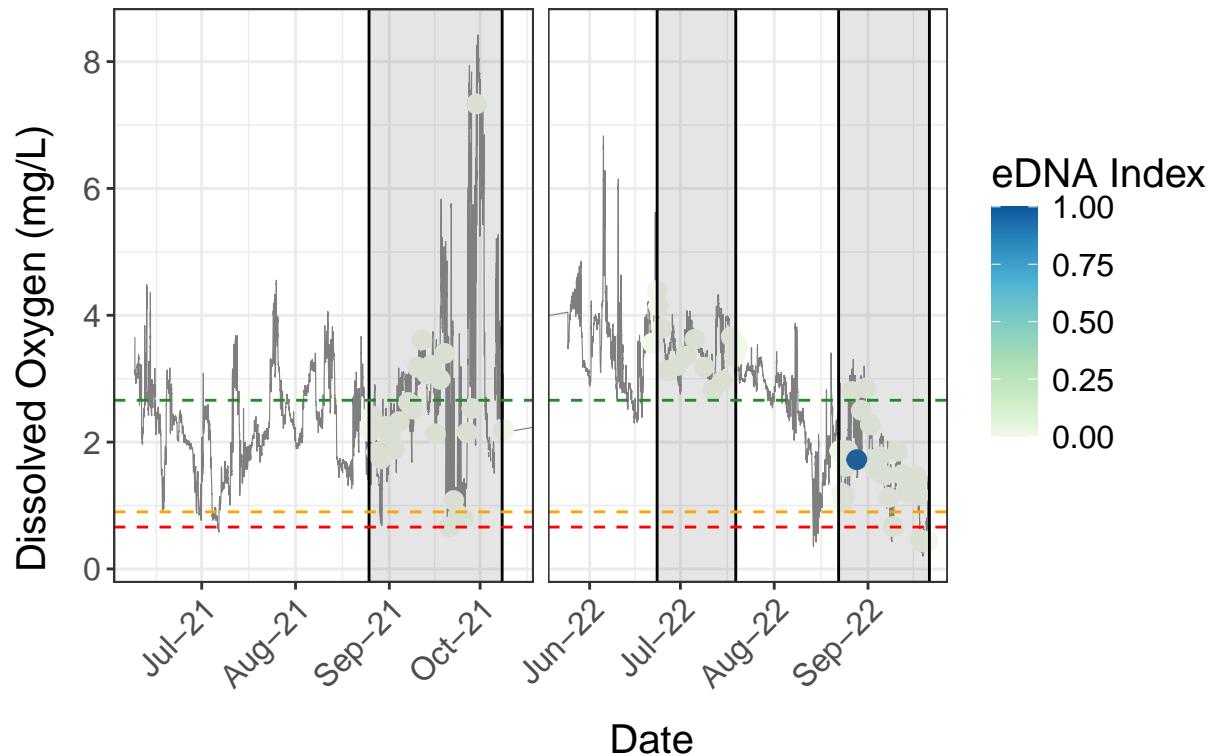
```
## [1] "Clausocalanus pergens"  
## [1] "Clausocalanus pergens eDNA Index vs Oxygen"
```

Clausocalanus pergens eDNA Index vs Oxygen



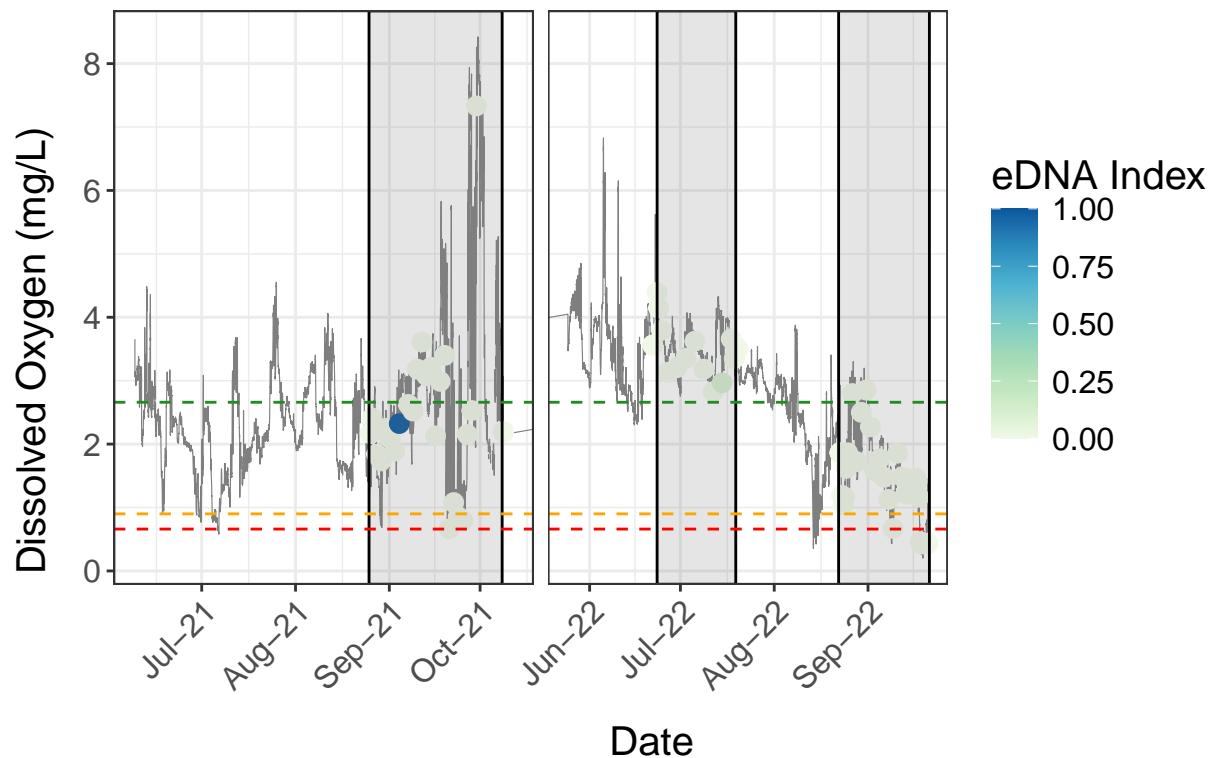
```
## [1] "Diacyclops incolotaenia"  
## [1] "Diacyclops incolotaenia eDNA Index vs Oxygen"
```

Diacyclops incolaenia eDNA Index vs Oxygen



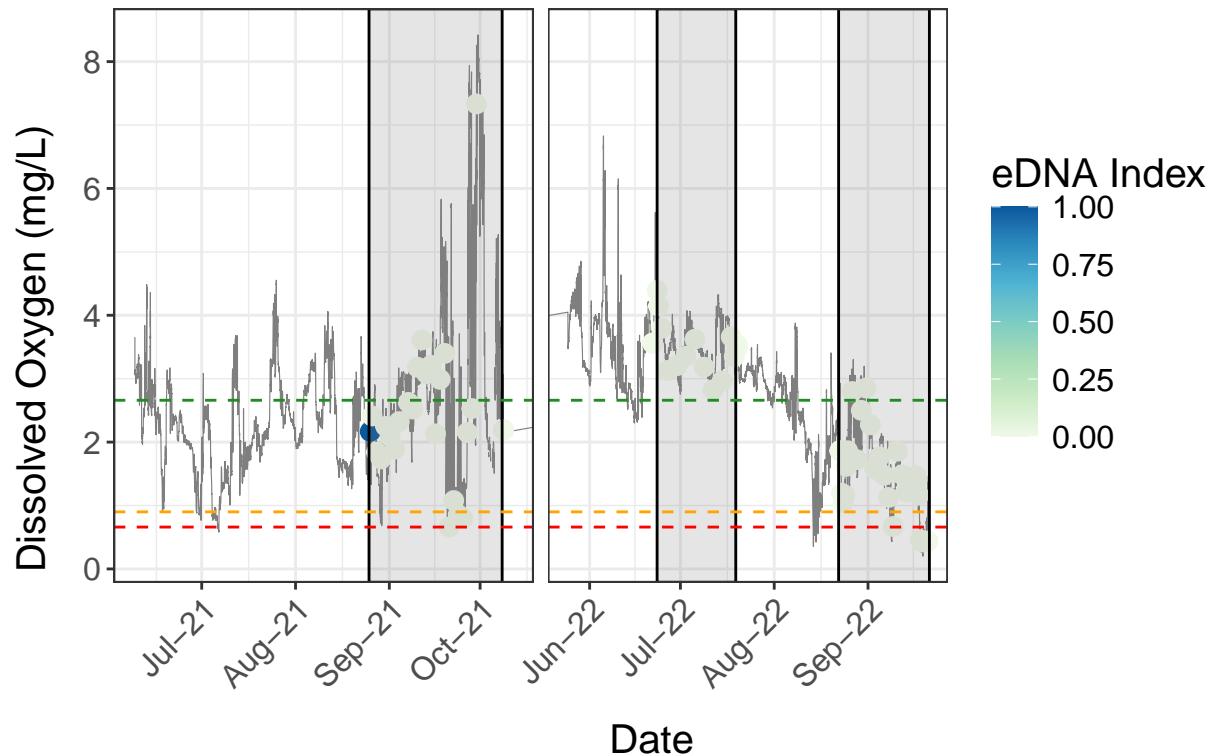
```
## [1] "Lucicutia flavigornis"  
## [1] "Lucicutia flavigornis eDNA Index vs Oxygen"
```

Lucicutia flavigornis eDNA Index vs Oxygen



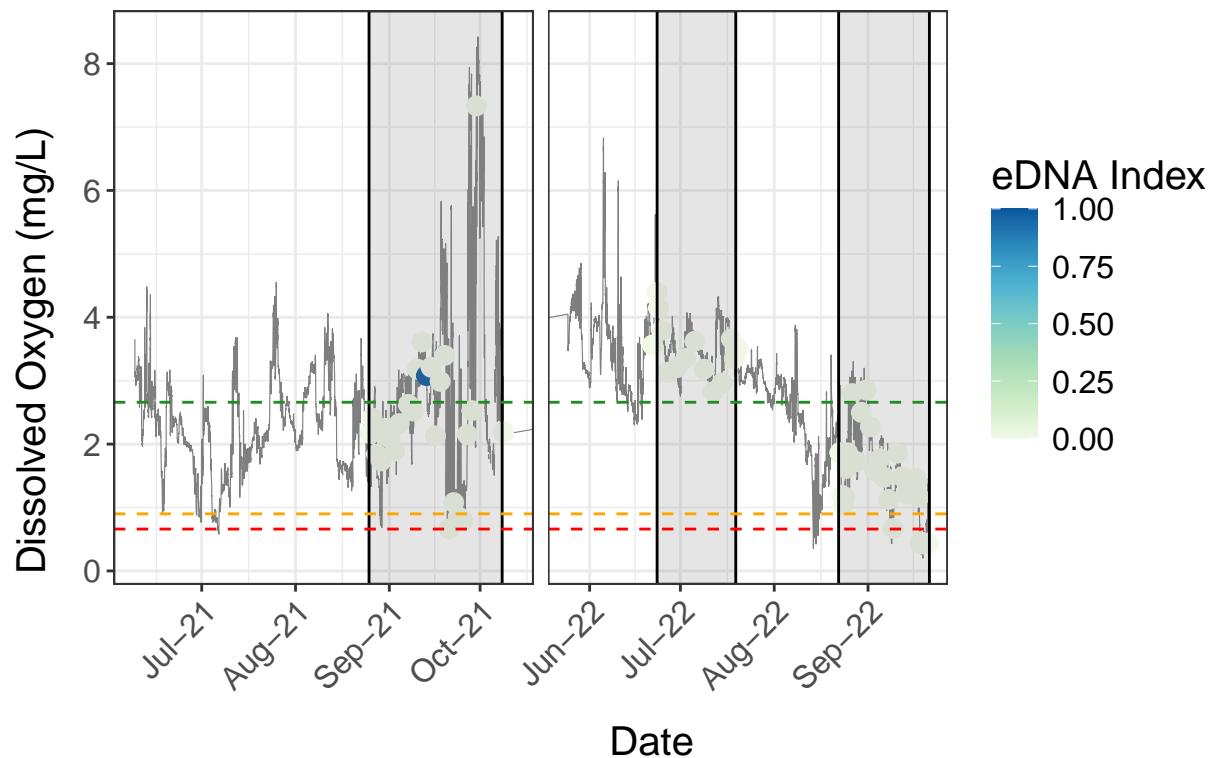
```
## [1] "Metridia lucens"  
## [1] "Metridia lucens eDNA Index vs Oxygen"
```

Metridia lucens eDNA Index vs Oxygen



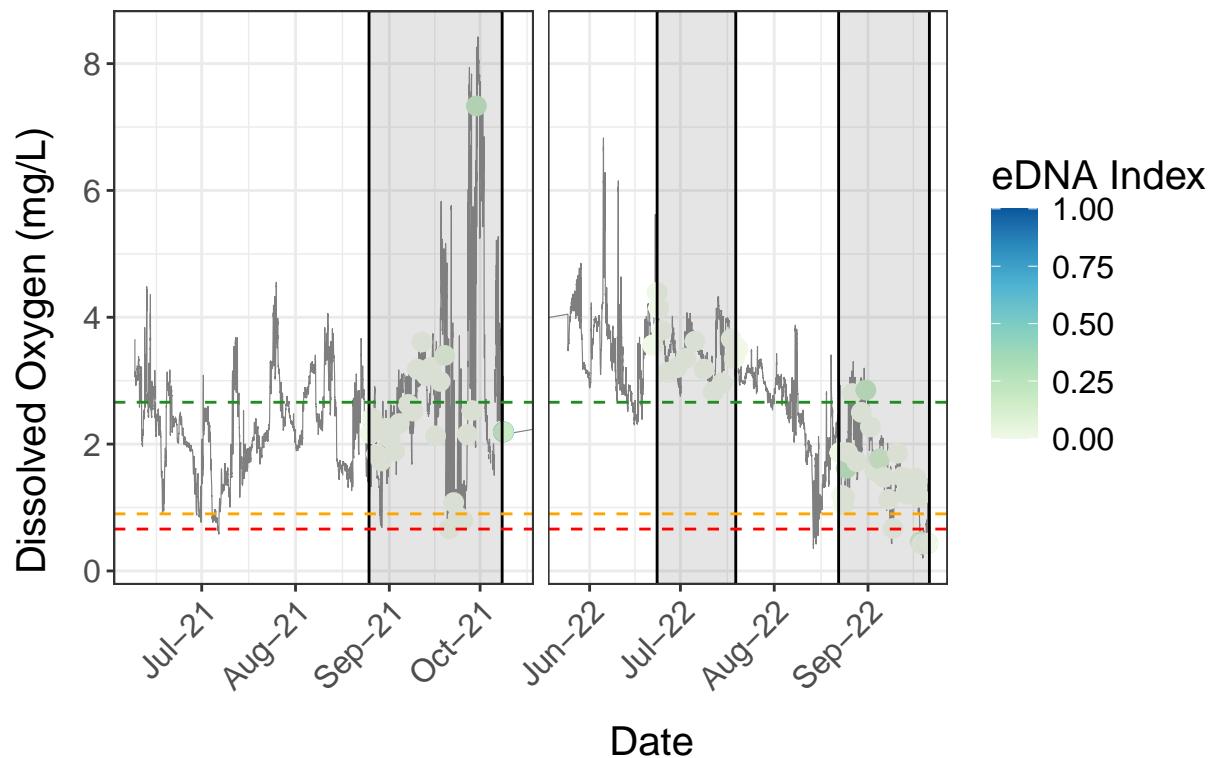
```
## [1] "Metridia pacifica"  
## [1] "Metridia pacifica eDNA Index vs Oxygen"
```

Metridia pacifica eDNA Index vs Oxygen



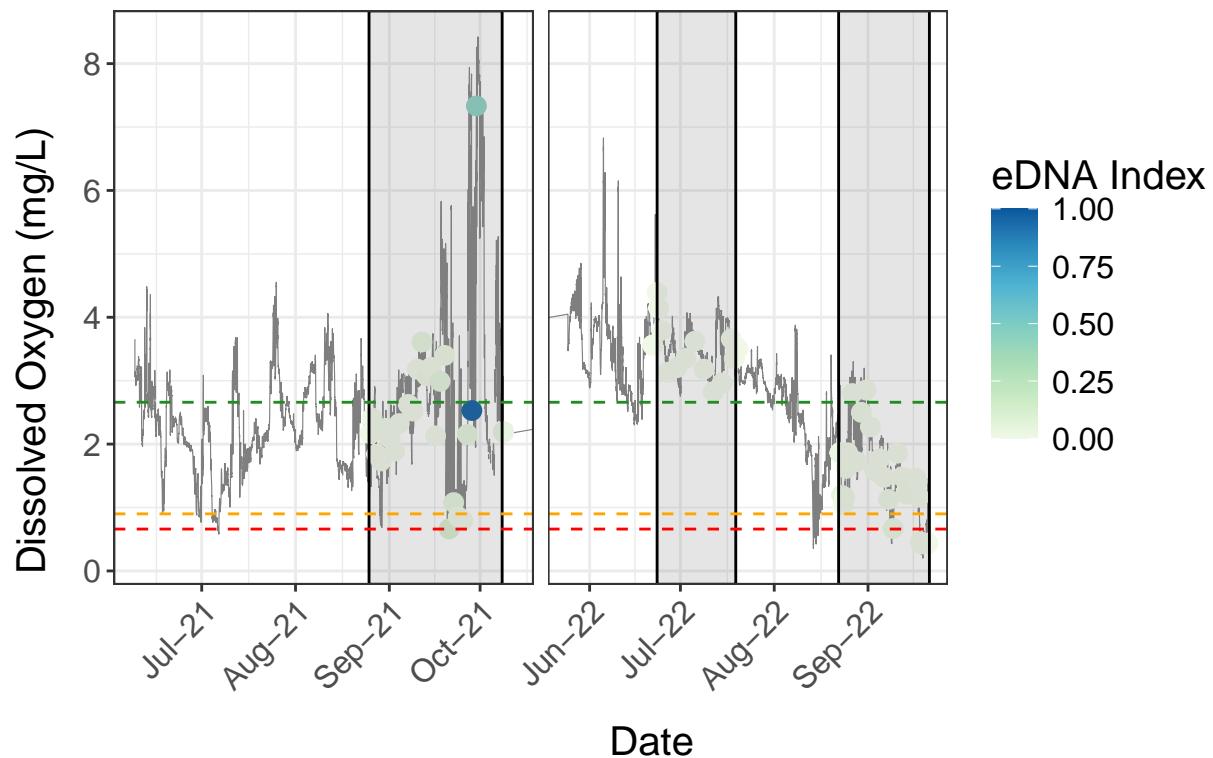
```
## [1] "Oithona similis"  
## [1] "Oithona similis eDNA Index vs Oxygen"
```

Oithona similis eDNA Index vs Oxygen



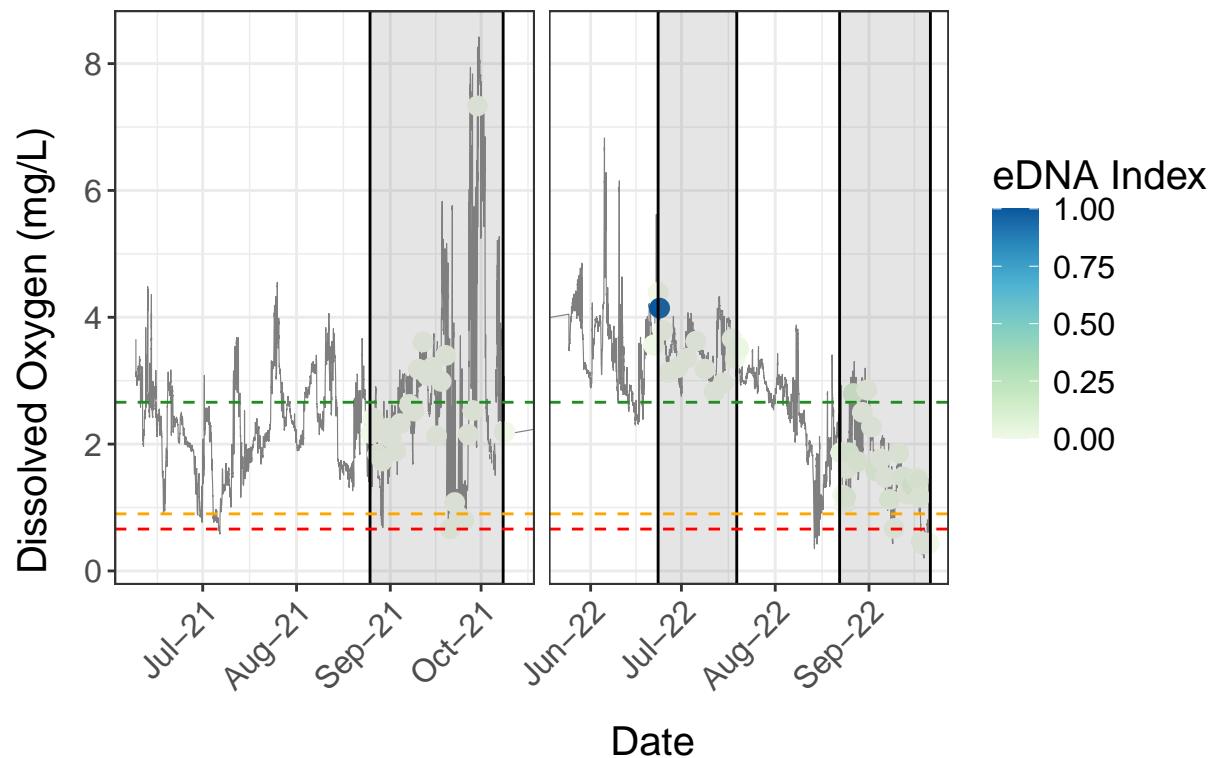
```
## [1] "Paracalanus sp. C AC-2013"
## [1] "Paracalanus sp. C AC-2013 eDNA Index vs Oxygen"
```

Paracalanus sp. C AC-2013 eDNA Index vs Oxygen



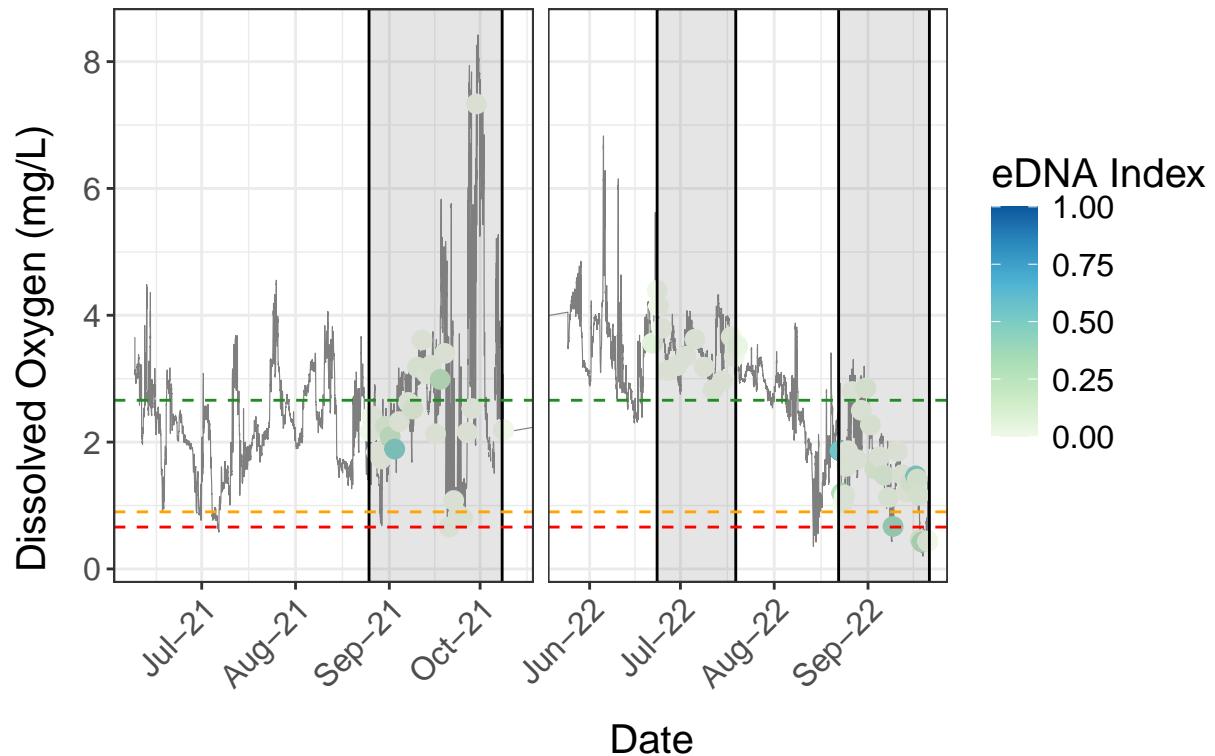
```
## [1] "Pseudocalanus newmani"  
## [1] "Pseudocalanus newmani eDNA Index vs Oxygen"
```

Pseudocalanus newmani eDNA Index vs Oxygen

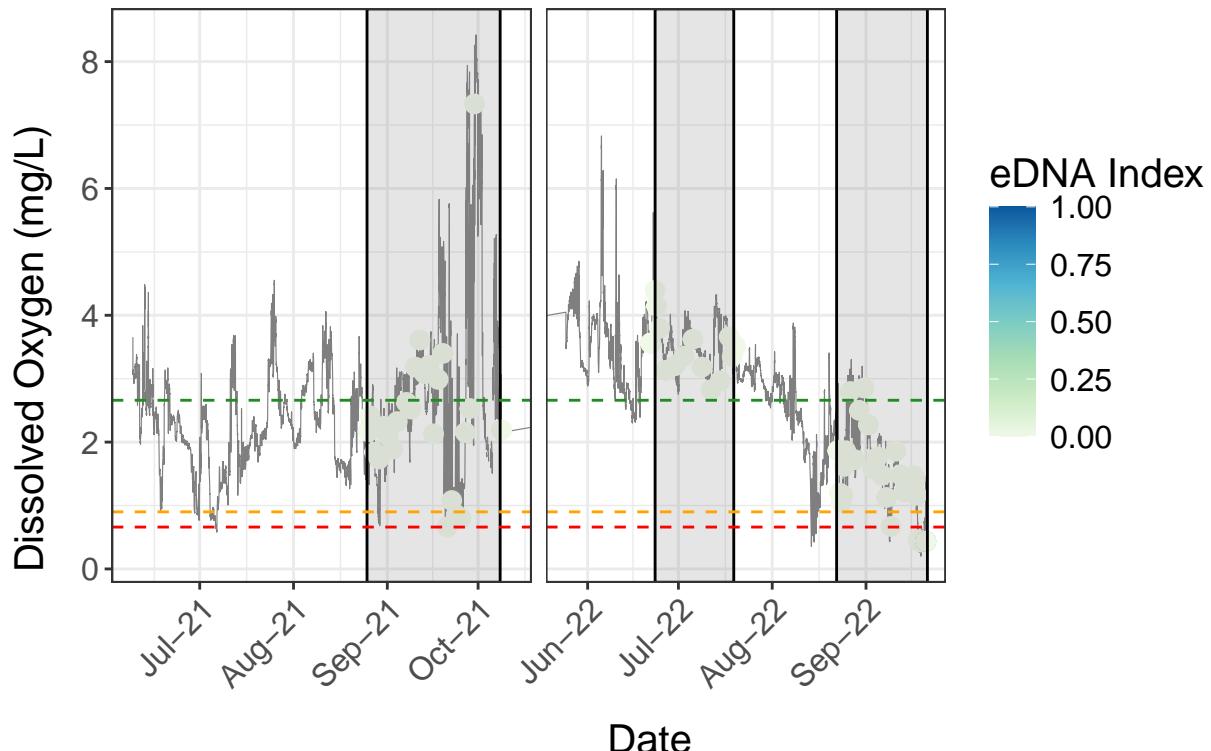


```
## [1] "Thermocyclops inversus"
## [1] "Thermocyclops inversus eDNA Index vs Oxygen"
```

Thermocyclops inversus eDNA Index vs Oxygen



Triconia minuta eDNA Index vs Oxygen



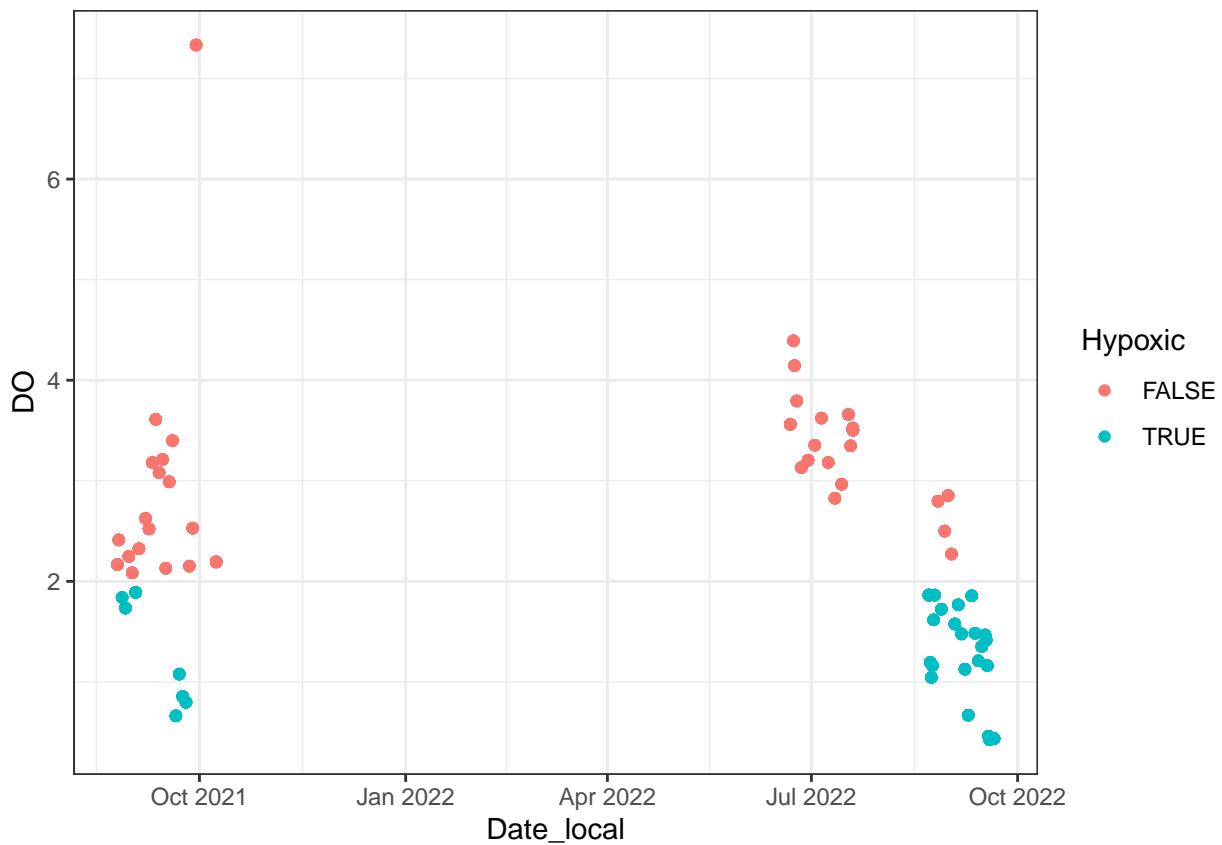
```
system("say Graphs complete")
```

Scatterplot eDNA vs hypoxic threshold

```
eDNAindxEnvData_cleanYr <- eDNAindxEnvData_cleanYr %>%
  mutate(Hypoxic = case_when(DO < 2 ~ T, .default = F))

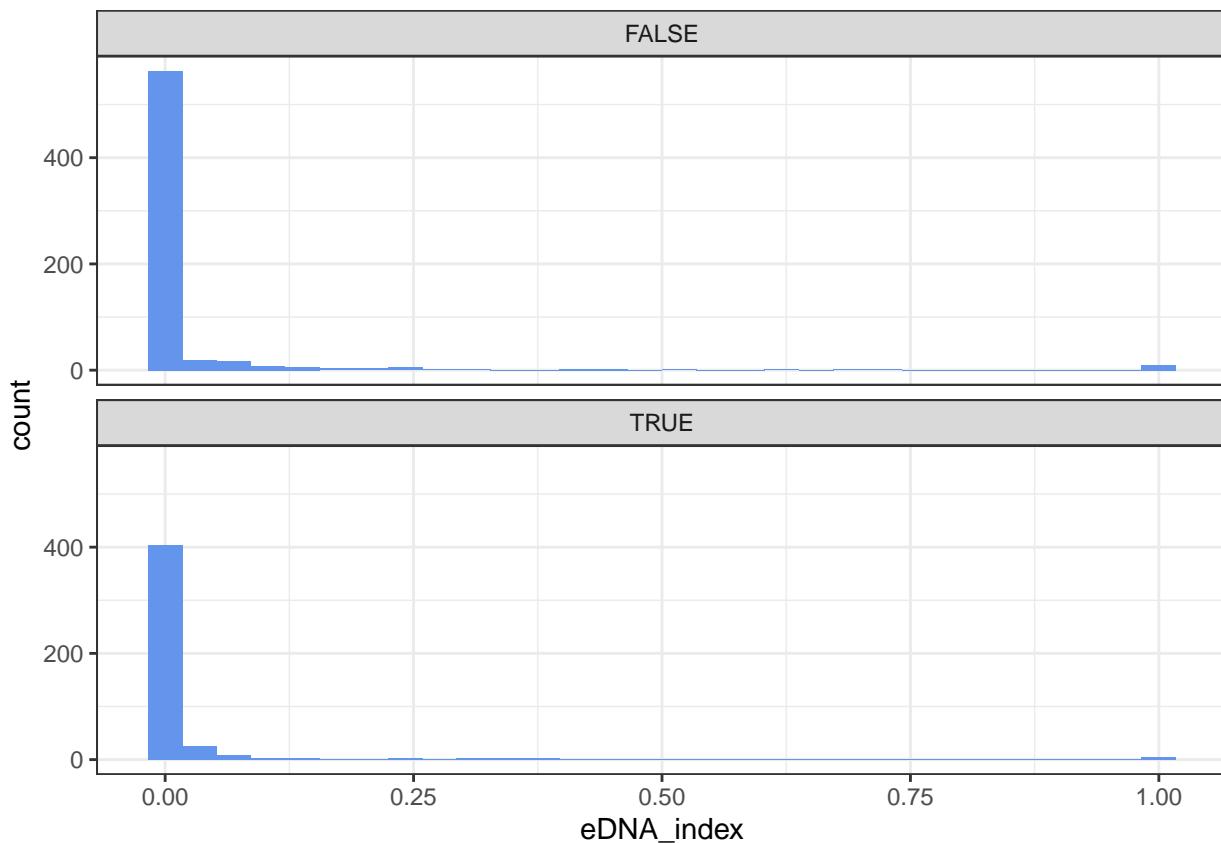
ggplot(eDNAindxEnvData_cleanYr, aes(x = Date_local, y = DO, color = Hypoxic)) +
  theme_bw() +
  geom_point()

## Warning: Removed 14 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



```
ggplot(eDNAindxEnvData_cleanYr, aes(x = eDNA_index)) +
  geom_histogram(fill = "cornflowerblue") +
  theme_bw() +
  facet_wrap(facets = vars(Hypoxic), ncol = 1)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



GAM of eDNA index vs oxygen

```

dfsplit <- split(eDNAindxEnvData_cleanYr, eDNAindxEnvData_cleanYr$Species) # Split by
#> species

for (i in 1:length(dfsplit)) { # For each species:
  species <- dfsplit[[i]]$Species[1] # Species name
  print(species)
  title <- paste(species, sep = " ", "eDNA Index vs Dissolved Oxygen") # Plot title,
#> changed for eDNA
  print(title)
  print(ggplot(dfsplit[[i]], aes(x = DO, y = eDNA_index)) + # plot this species
    geom_point(color = "orange2", alpha = 0.7) +
    geom_smooth(method = 'gam', se = F, color = "cornflowerblue") + # visualize GAM
    ggtitle(title) +
    theme_bw())
  ggsave(filename = paste(species, sep = "_", "eDNA_DO_GAM.png"), path =
#> here("eDNA_Index_Hypoxia", "Plots", "GAM_Initial"), width = 2500, height = 2000,
#> units = "px")
}

## [1] "Acartia longiremis"
## [1] "Acartia longiremis eDNA Index vs Dissolved Oxygen"
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range

```

```

## (`stat_smooth()`).

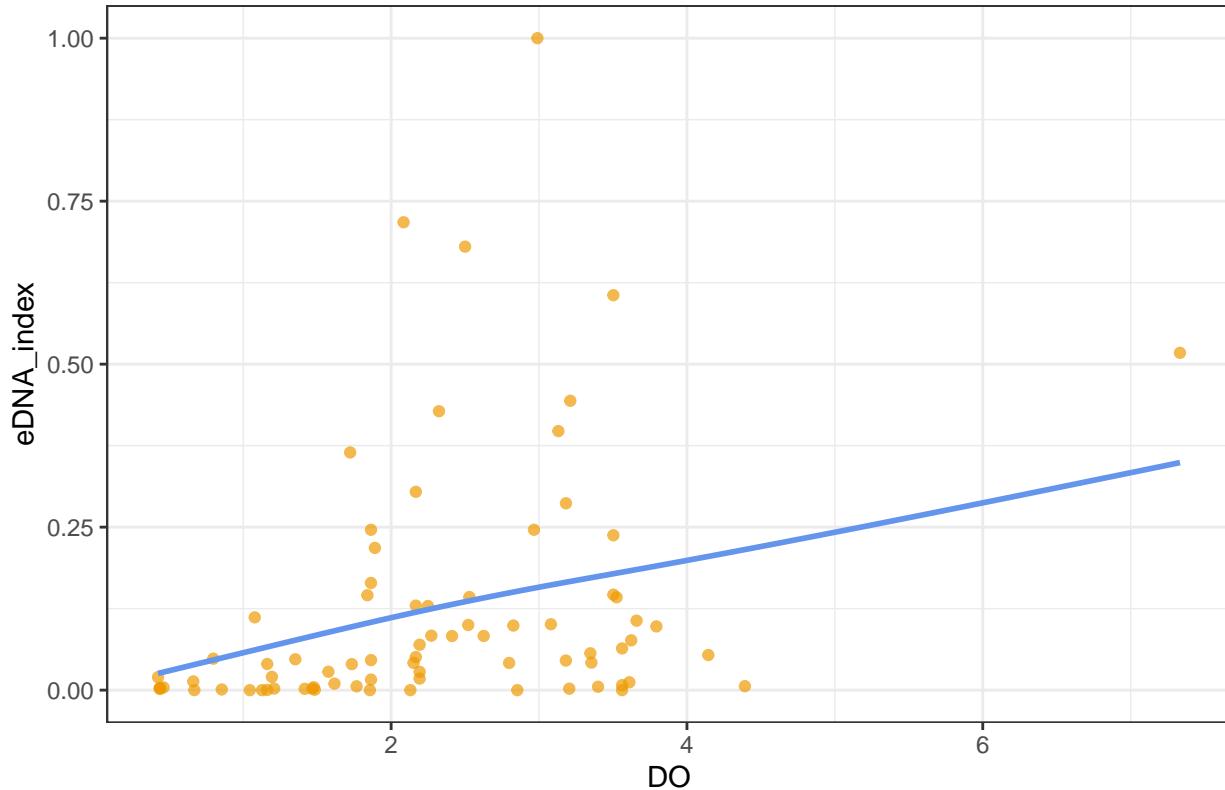
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).

## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'

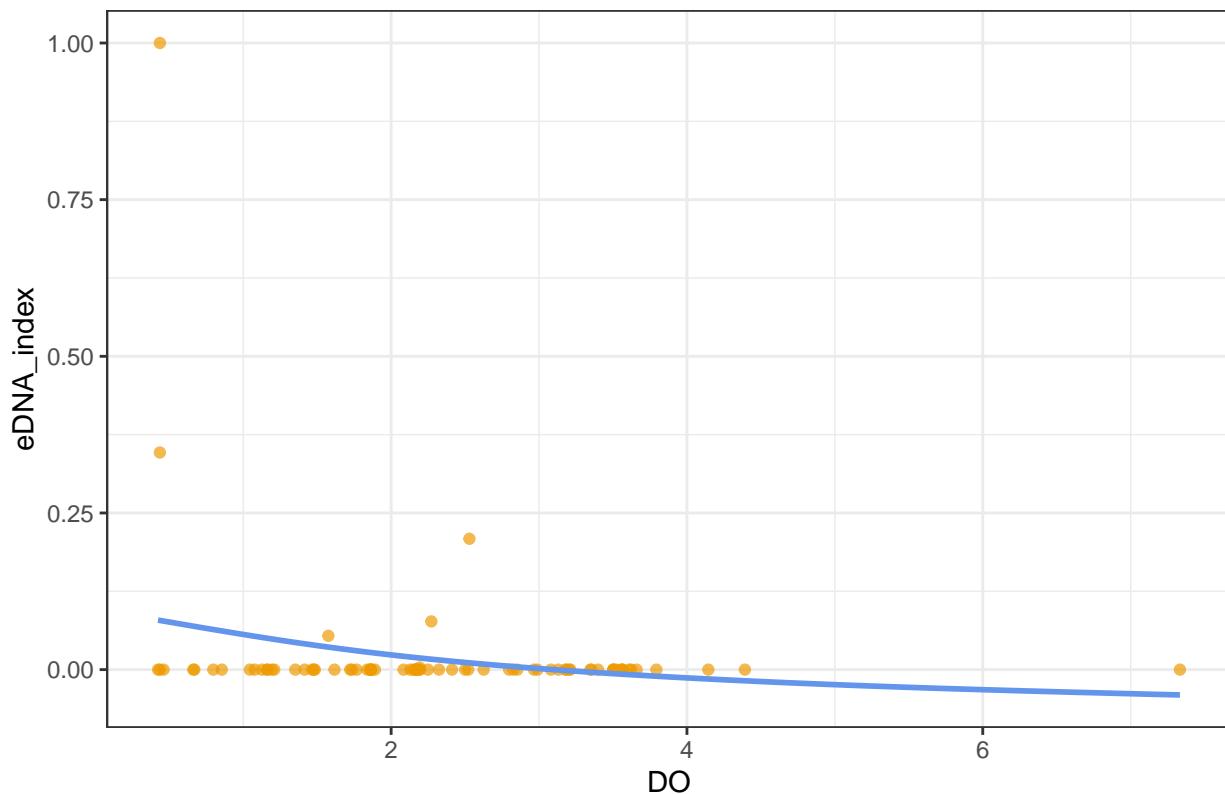
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).

```

Acartia longiremis eDNA Index vs Dissolved Oxygen

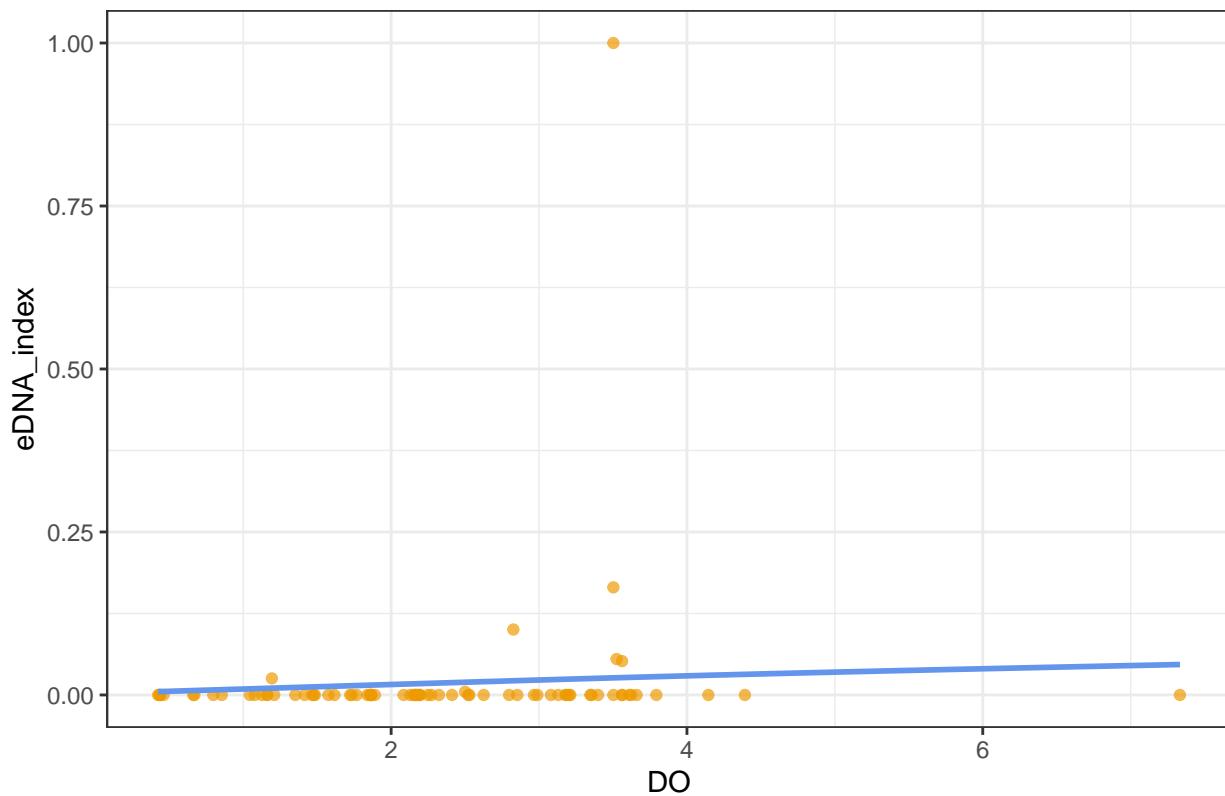


Calanus pacificus eDNA Index vs Dissolved Oxygen



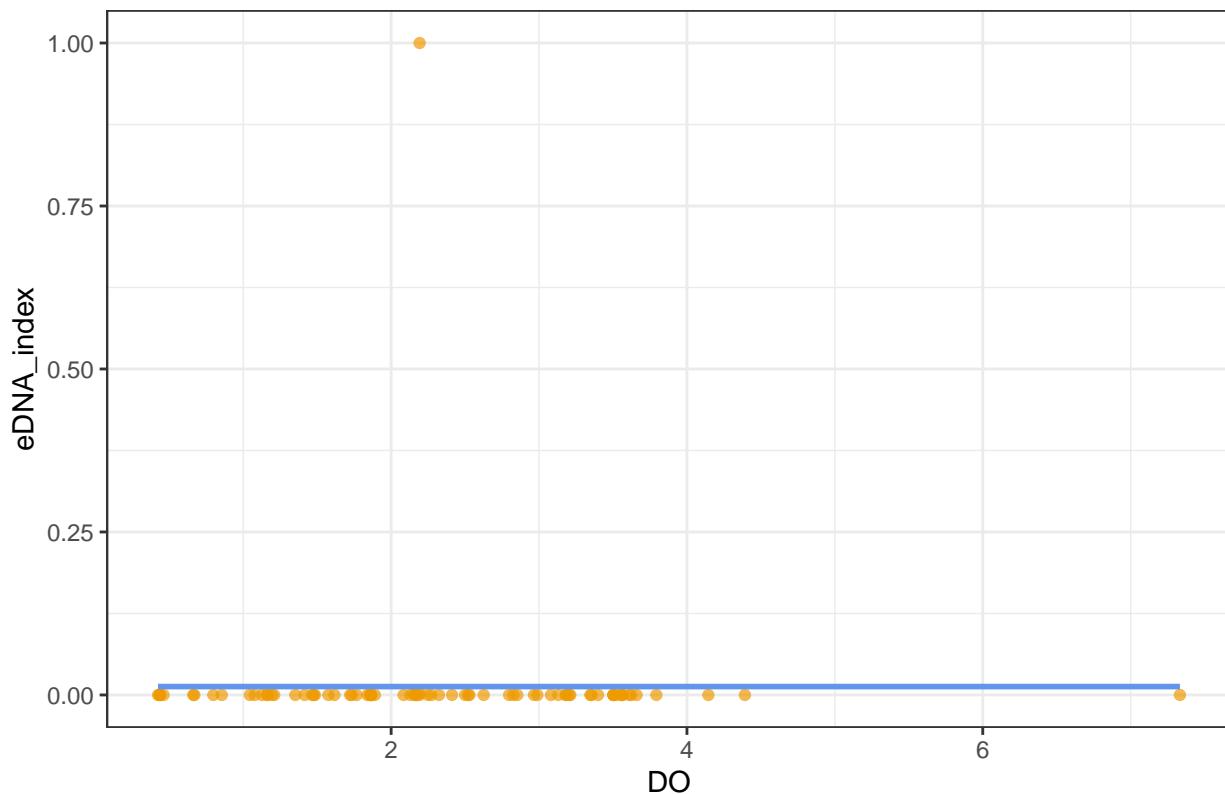
```
## [1] "Centropages abdominalis"
## [1] "Centropages abdominalis eDNA Index vs Dissolved Oxygen"
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

Centropages abdominalis eDNA Index vs Dissolved Oxygen



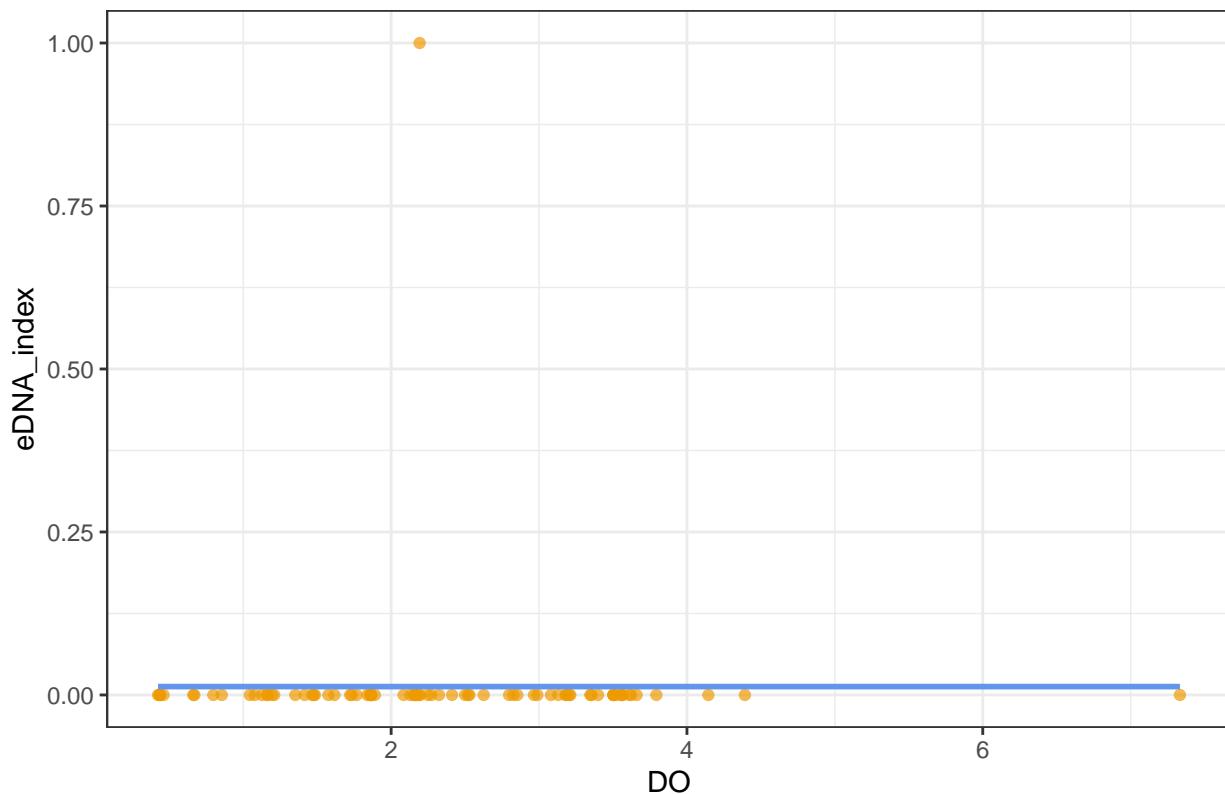
```
## [1] "Clausocalanus parapergens"
## [1] "Clausocalanus parapergens eDNA Index vs Dissolved Oxygen"
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

Clausocalanus parapergens eDNA Index vs Dissolved Oxygen



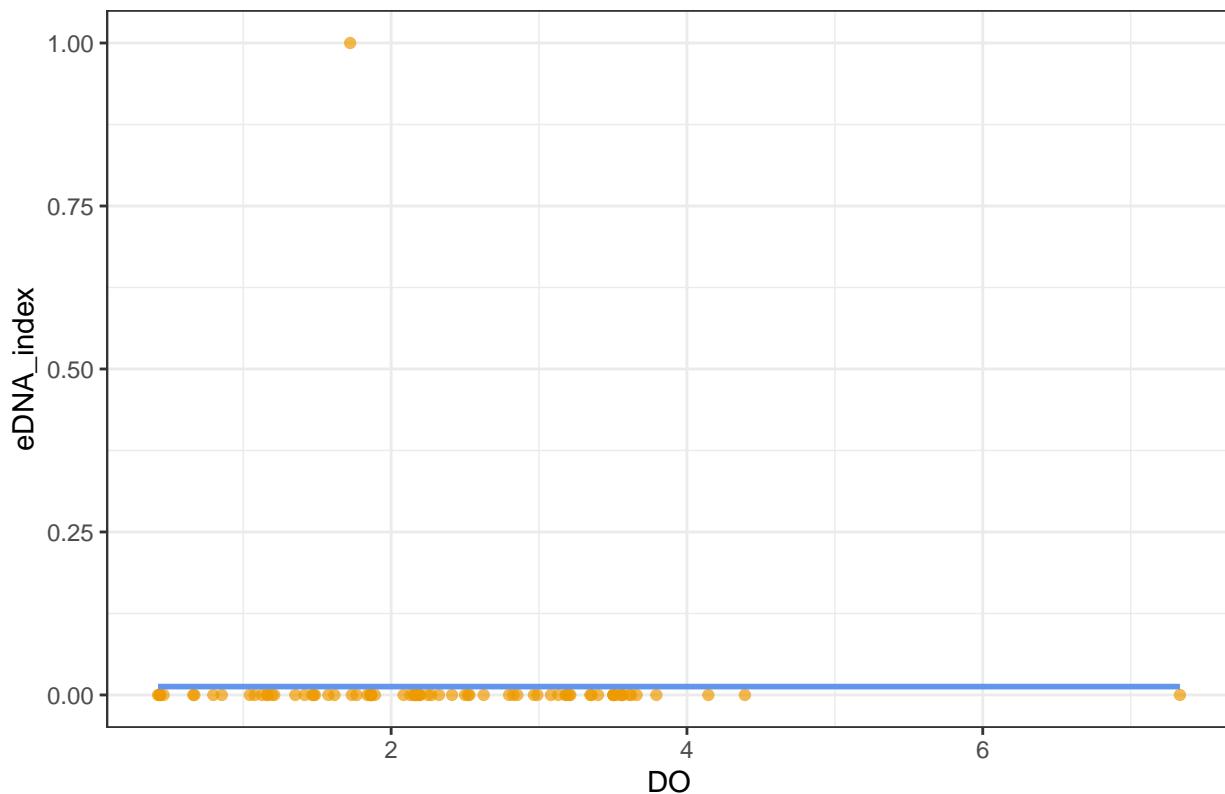
```
## [1] "Clausocalanus pergens"
## [1] "Clausocalanus pergens eDNA Index vs Dissolved Oxygen"
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

Clausocalanus pergens eDNA Index vs Dissolved Oxygen



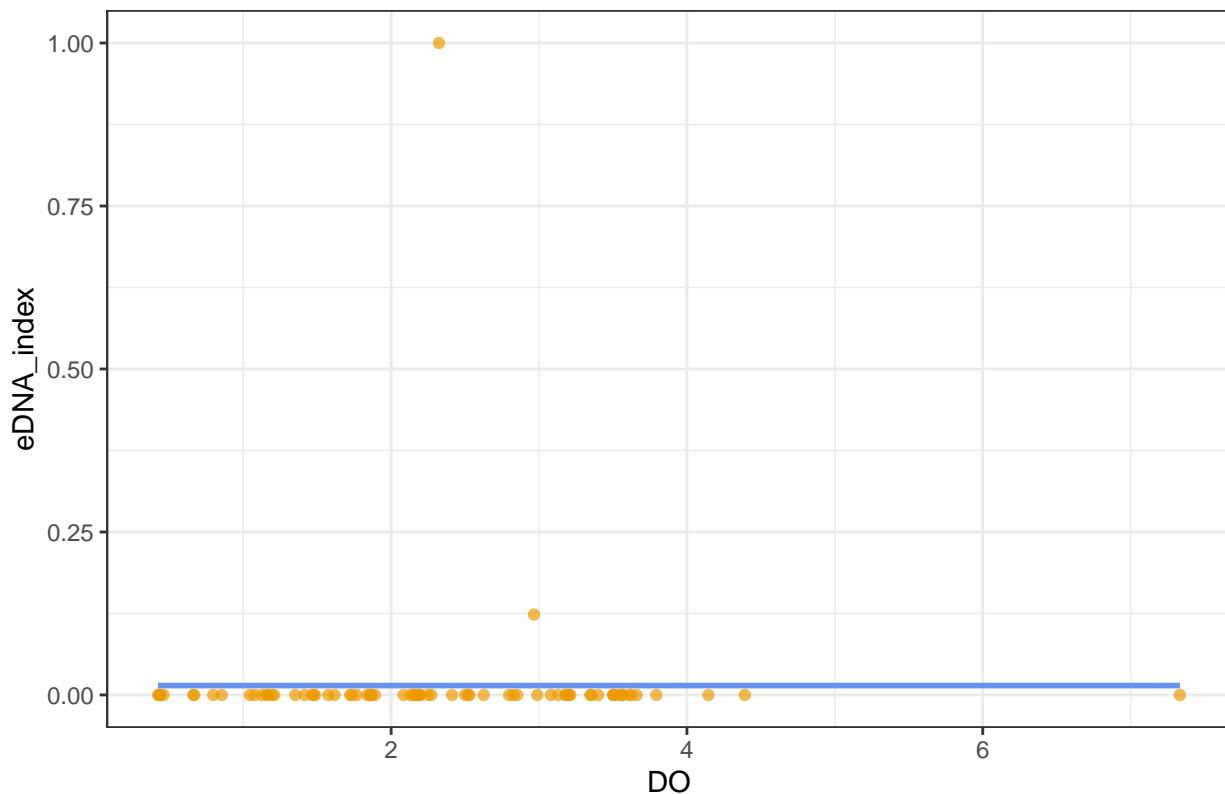
```
## [1] "Diacyclops incolotaenia"
## [1] "Diacyclops incolotaenia eDNA Index vs Dissolved Oxygen"
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

Diacyclops incolotaenia eDNA Index vs Dissolved Oxygen



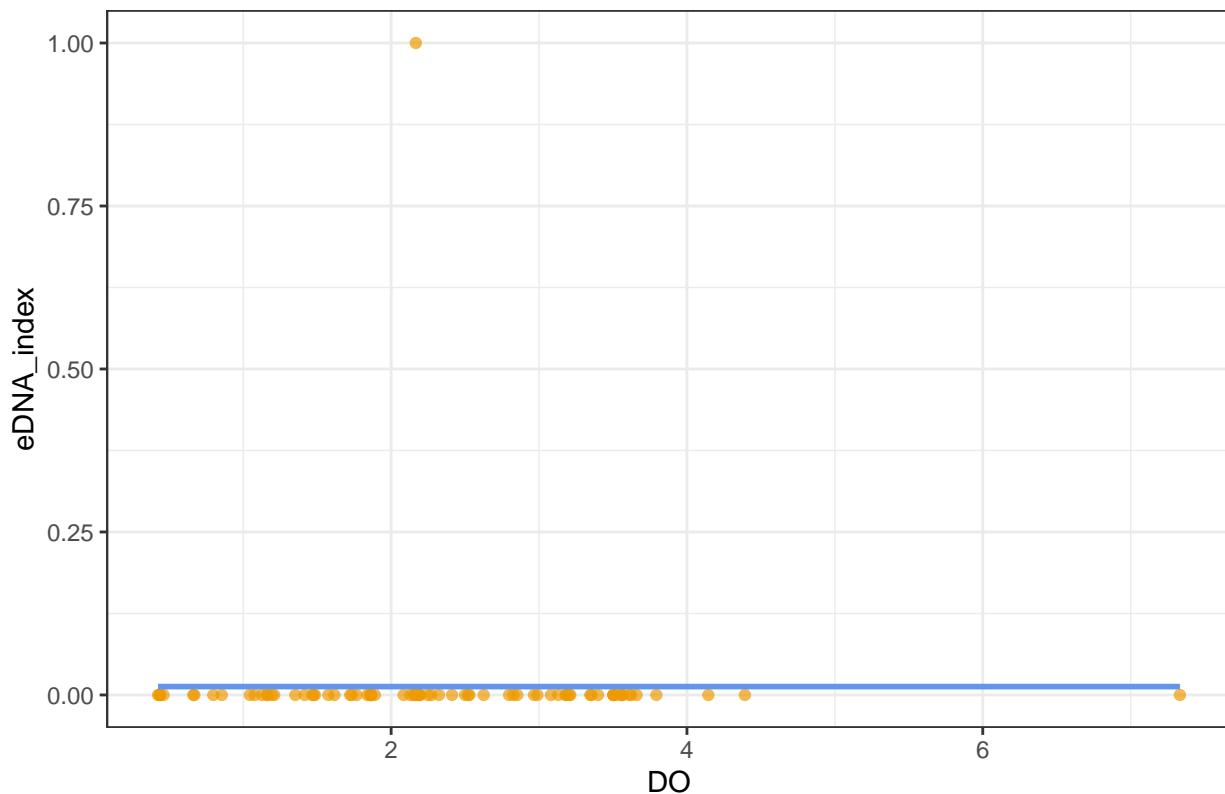
```
## [1] "Lucicutia flavigornis"
## [1] "Lucicutia flavigornis eDNA Index vs Dissolved Oxygen"
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

Lucicutia flavigornis eDNA Index vs Dissolved Oxygen



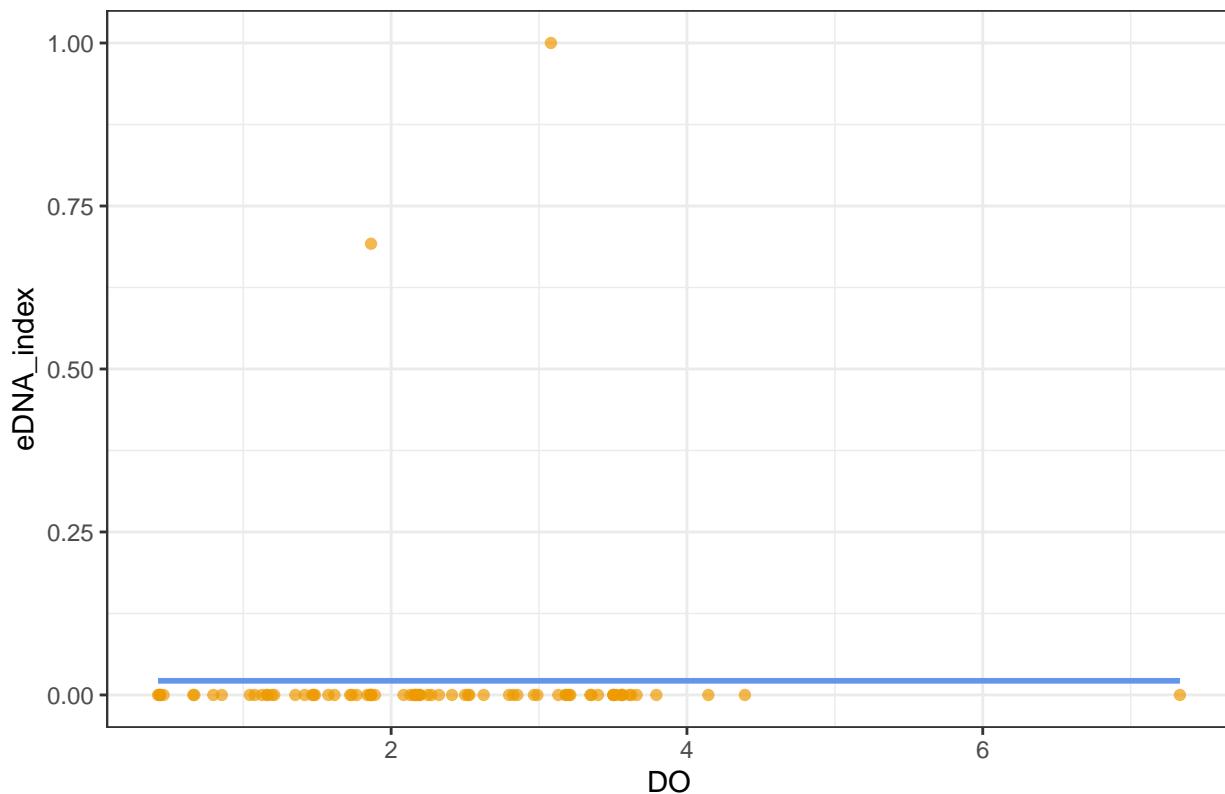
```
## [1] "Metridia lucens"
## [1] "Metridia lucens eDNA Index vs Dissolved Oxygen"
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

Metridia lucens eDNA Index vs Dissolved Oxygen



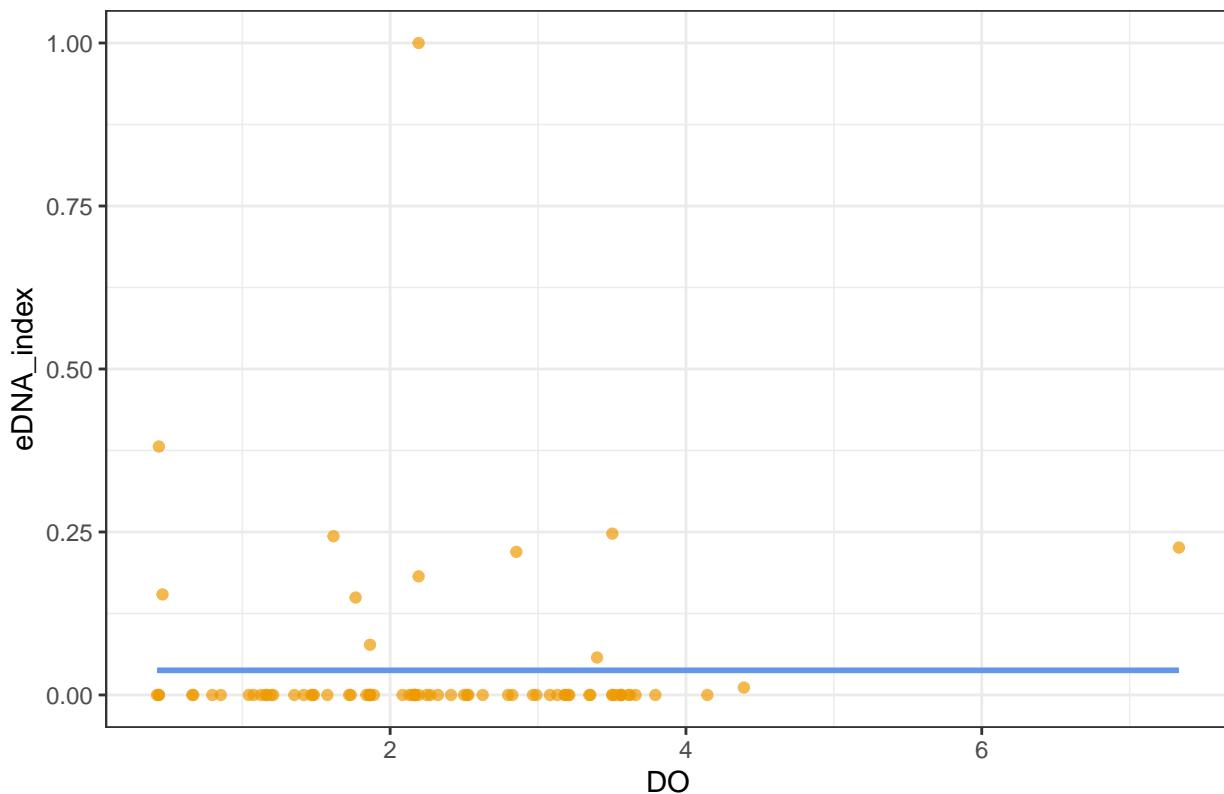
```
## [1] "Metridia pacifica"
## [1] "Metridia pacifica eDNA Index vs Dissolved Oxygen"
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

Metridia pacifica eDNA Index vs Dissolved Oxygen



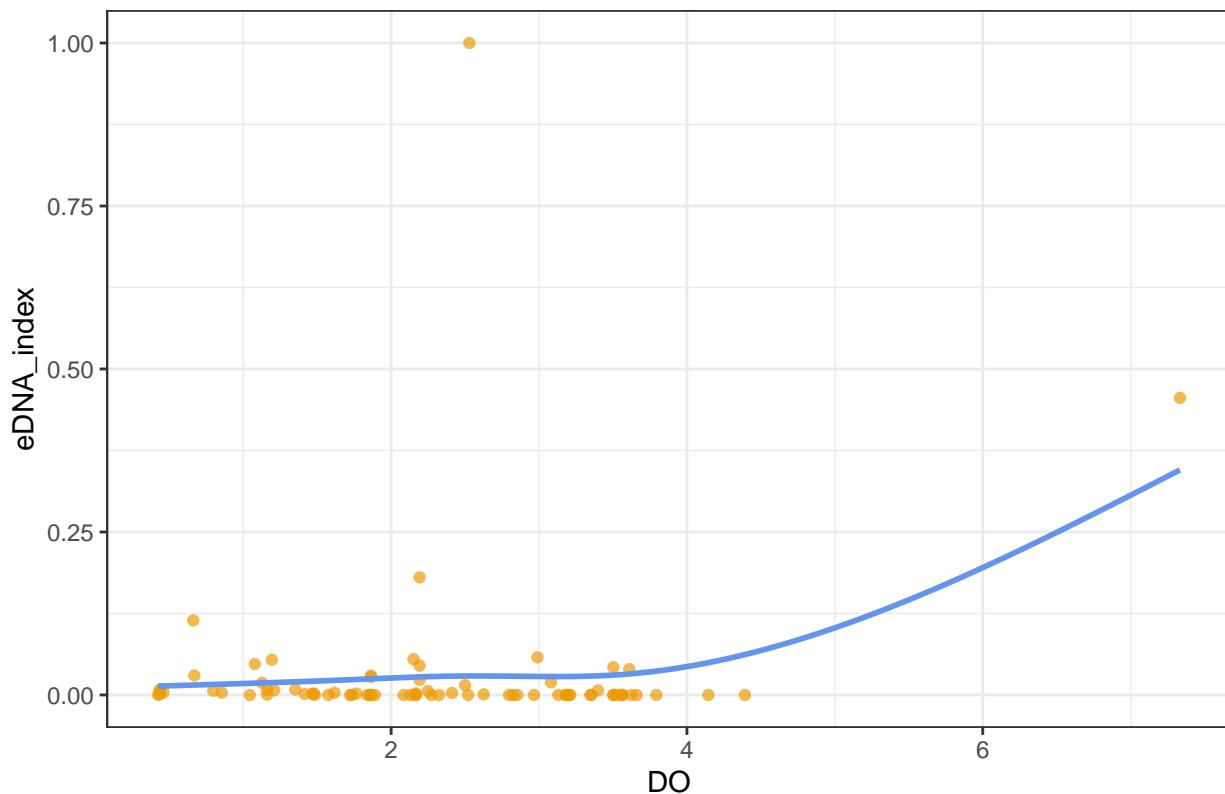
```
## [1] "Oithona similis"
## [1] "Oithona similis eDNA Index vs Dissolved Oxygen"
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

Oithona similis eDNA Index vs Dissolved Oxygen



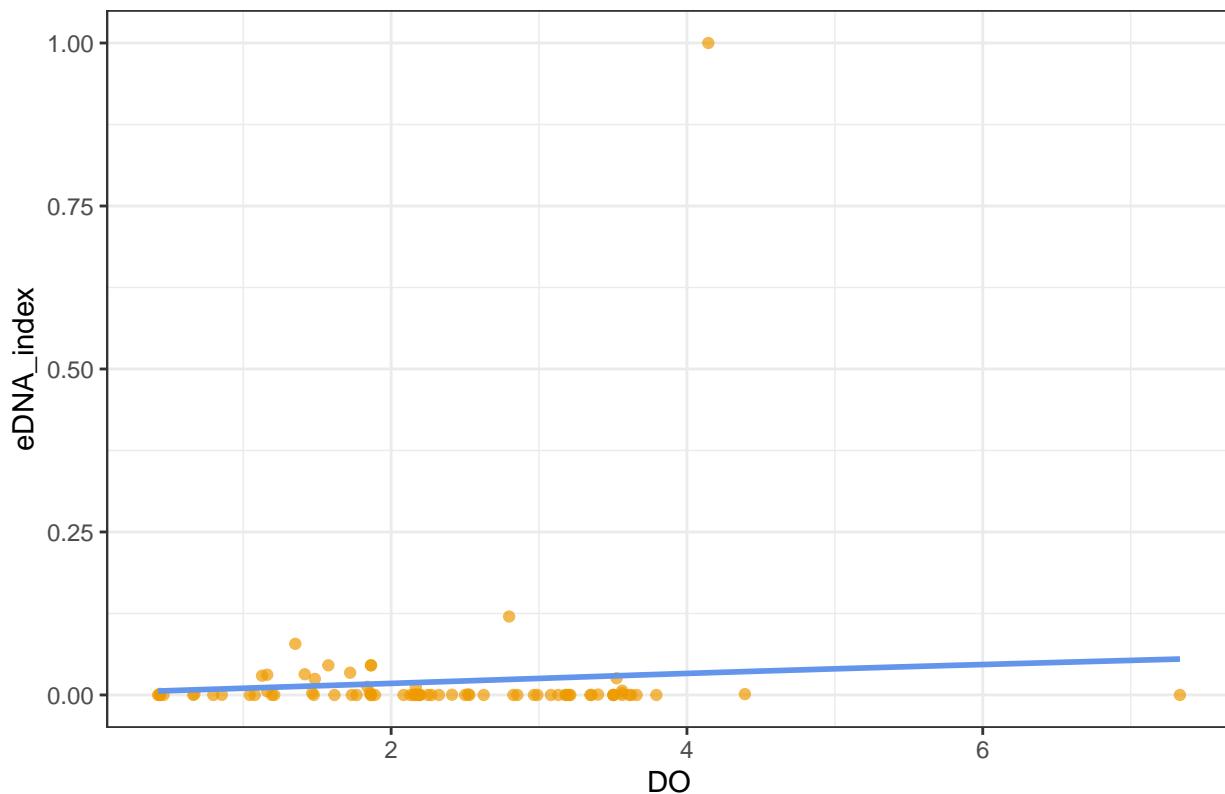
```
## [1] "Paracalanus sp. C AC-2013"
## [1] "Paracalanus sp. C AC-2013 eDNA Index vs Dissolved Oxygen"
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

Paracalanus sp. C AC-2013 eDNA Index vs Dissolved Oxygen



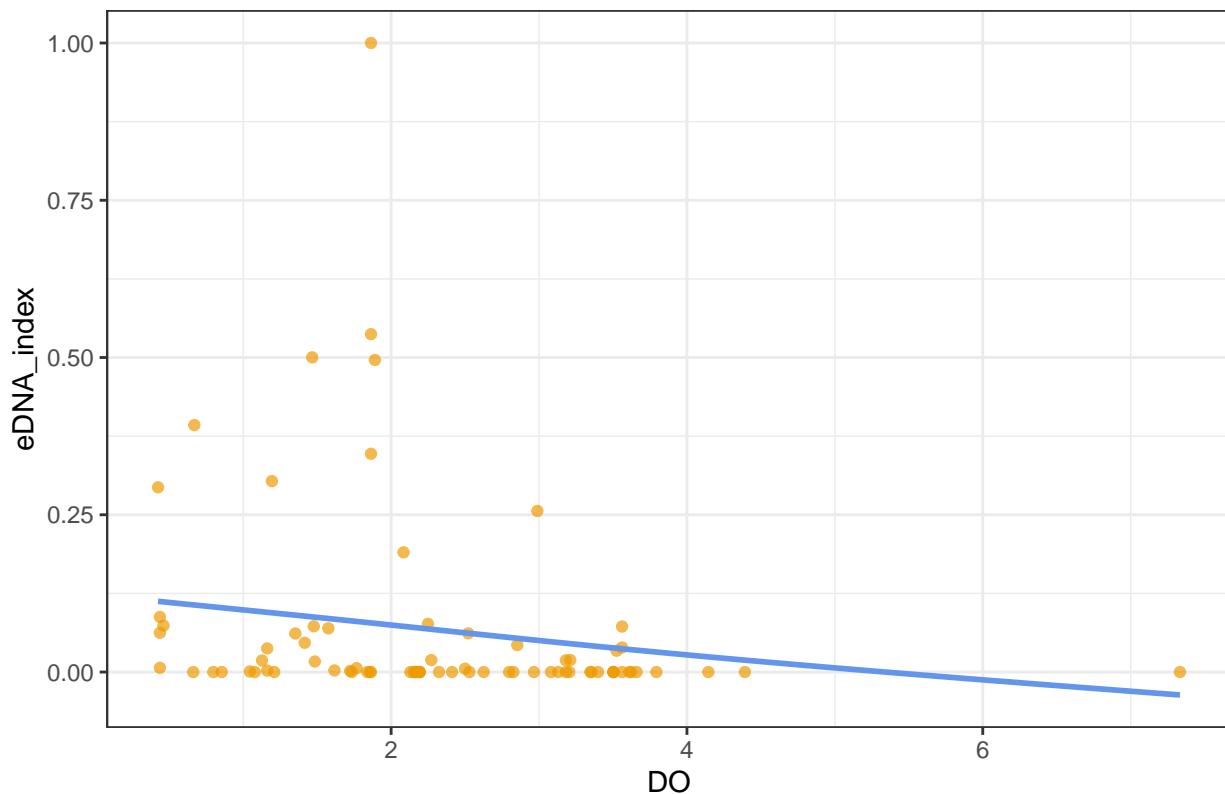
```
## [1] "Pseudocalanus newmani"
## [1] "Pseudocalanus newmani eDNA Index vs Dissolved Oxygen"
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

Pseudocalanus newmani eDNA Index vs Dissolved Oxygen



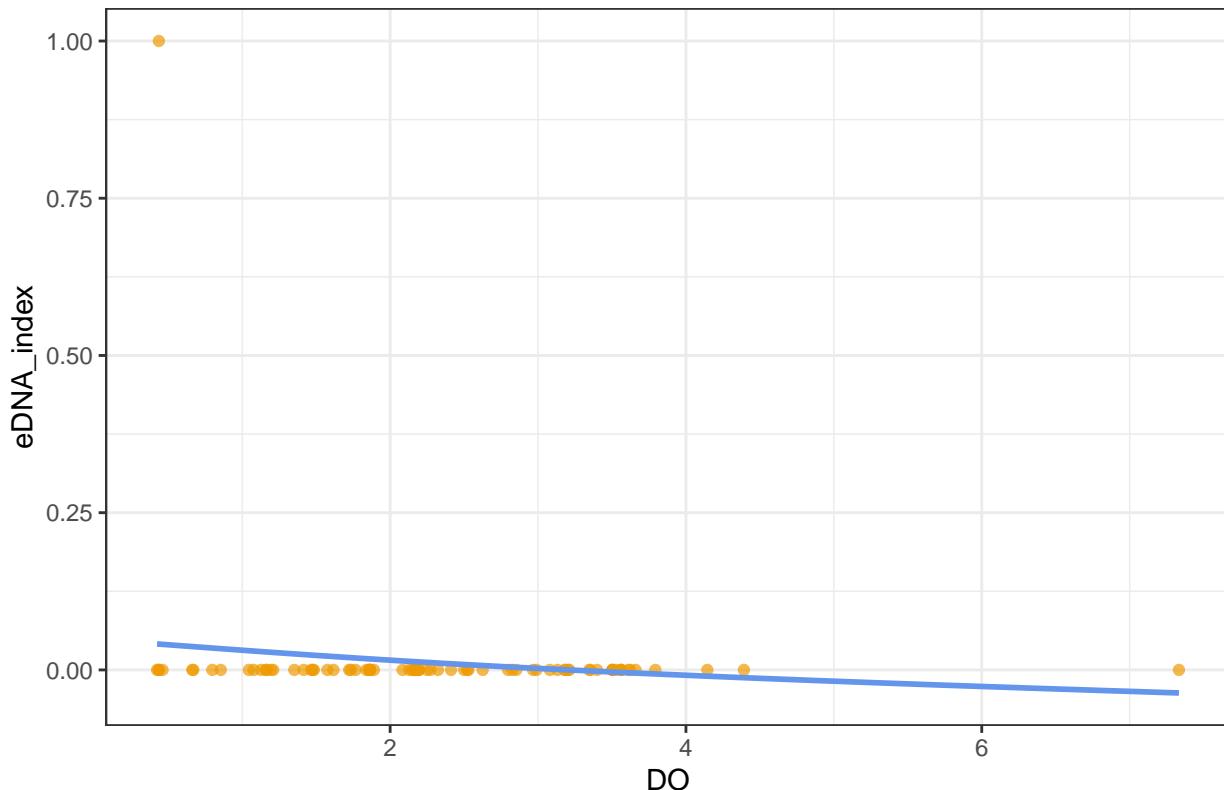
```
## [1] "Thermocyclops inversus"
## [1] "Thermocyclops inversus eDNA Index vs Dissolved Oxygen"
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

Thermocyclops inversus eDNA Index vs Dissolved Oxygen



```
## [1] "Triconia minuta"
## [1] "Triconia minuta eDNA Index vs Dissolved Oxygen"
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

Triconia minuta eDNA Index vs Dissolved Oxygen



```
# I think I'm still having some issues w.r.t. the 2023 elimination issue

for (i in 1:length(dfsplit)) { # For each species:
  species <- dfsplit[[i]]$Species[1] # Species name for title and export png name
  title <- paste(species, sep = " ", "eDNA Index vs Dissolved Oxygen") # Plot title,
  ↪ changed for eDNA
  print(title)

  ind_DO_GAM <- gam(eDNA_index ~ s(DO), bs = "cr", data = dfsplit[[i]])
  print(summary(ind_DO_GAM))

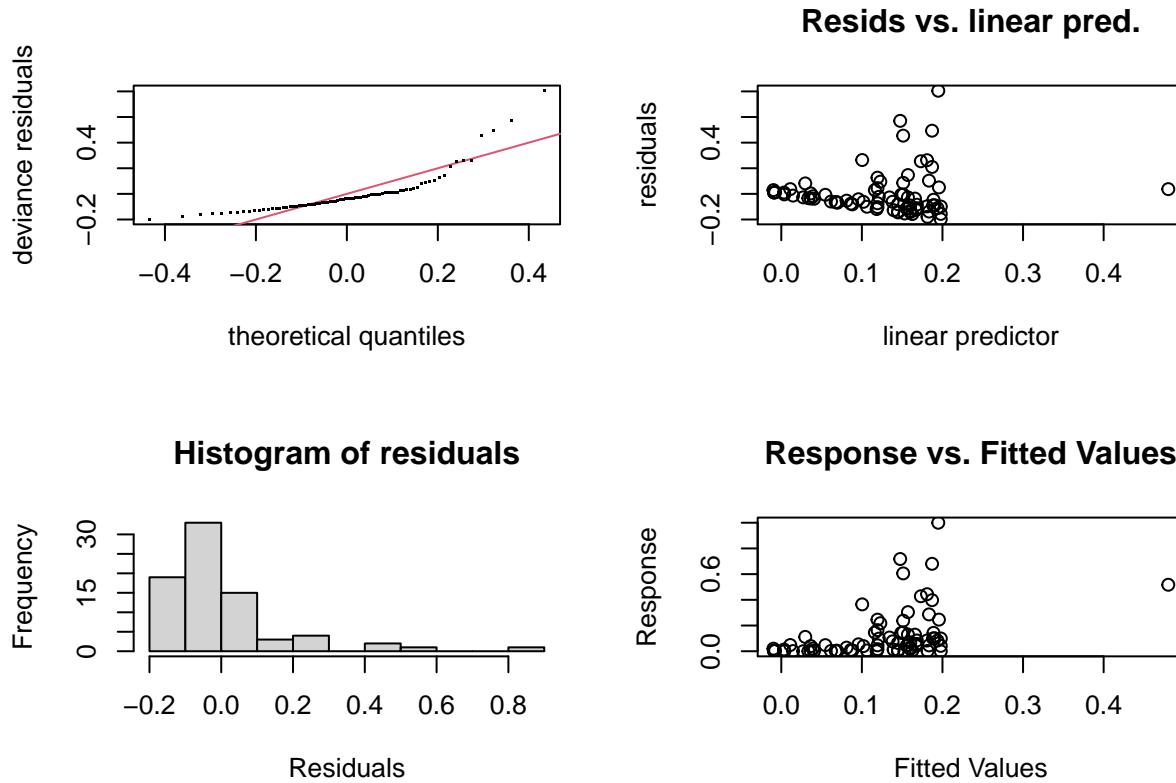
  print("GAM quality")
  print(paste("AIC: ", sep = "", AIC(ind_DO_GAM))) # quality metric
  print(summary(ind_DO_GAM)$sp.criterion)
  print(gam.check(ind_DO_GAM, k.rep = 1000)) # Bunch of graphs and also text output,
  ↪ which is used to check how the GAM is doing
}

## [1] "Acartia longiremis eDNA Index vs Dissolved Oxygen"
## 
## Family: gaussian
## Link function: identity
## 
## Formula:
## eDNA_index ~ s(DO)
## 
## Parametric coefficients:
```

```

##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.12233   0.01975   6.195 3.1e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Approximate significance of smooth terms:
##          edf Ref.df    F p-value
## s(DO) 3.918 4.742 3.529 0.00822 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## R-sq.(adj) =  0.16 Deviance explained = 20.3%
## GCV = 0.032463 Scale est. = 0.030416 n = 78
## [1] "GAM quality"
## [1] "AIC: -44.3267161286312"
##      GCV.Cp
## 0.03246284

```



```

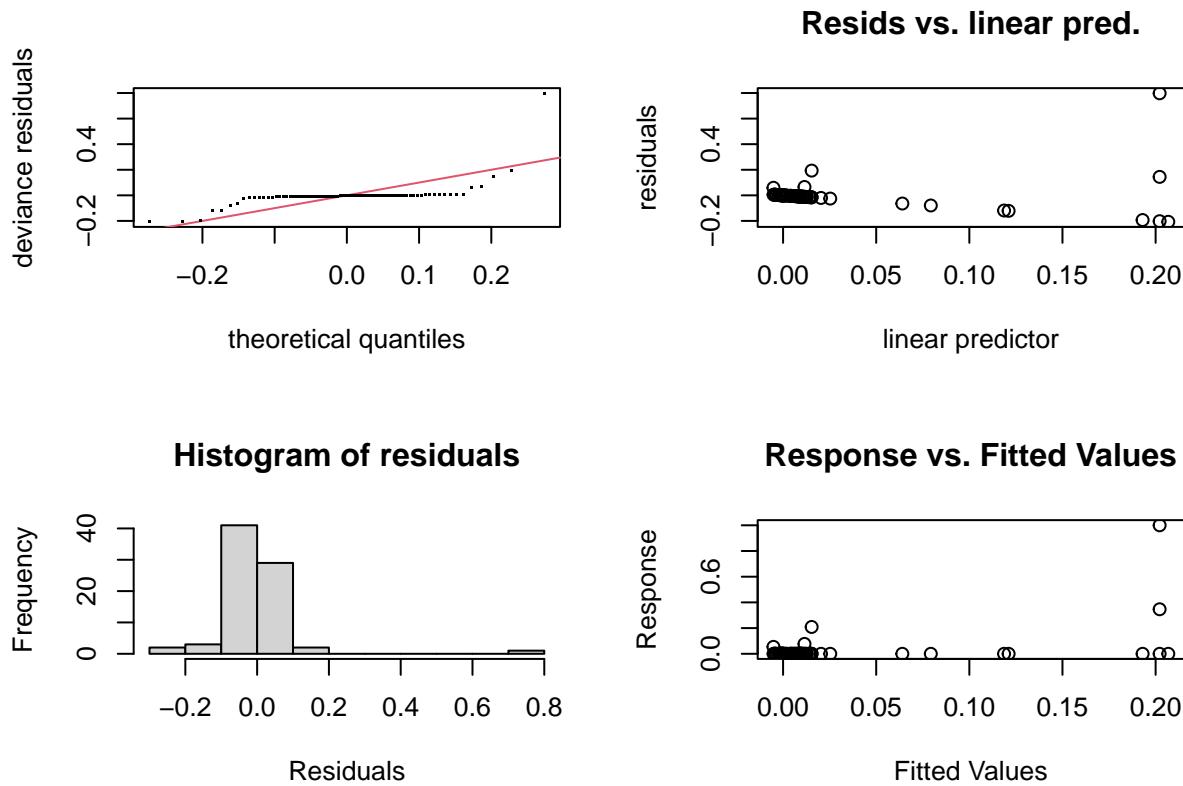
## 
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 4 iterations.
## The RMS GCV score gradient at convergence was 9.432252e-07 .
## The Hessian was positive definite.
## Model rank = 10 / 10
## 
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
## 
##          k'    edf k-index p-value
## s(DO) 9.00 3.92     1.13    0.87

```

```

## $mfrow
## [1] 2 2
##
## [1] "Calanus pacificus eDNA Index vs Dissolved Oxygen"
##
## Family: gaussian
## Link function: identity
##
## Formula:
## eDNA_index ~ s(DO)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.02166   0.01244   1.742   0.0858 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df    F p-value
## s(DO) 5.147 6.151 3.165 0.00818 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.182 Deviance explained = 23.7%
## GCV = 0.013095 Scale est. = 0.012063 n = 78
## [1] "GAM quality"
## [1] "AIC: -115.327310101293"
##      GCV.Cp
## 0.01309512

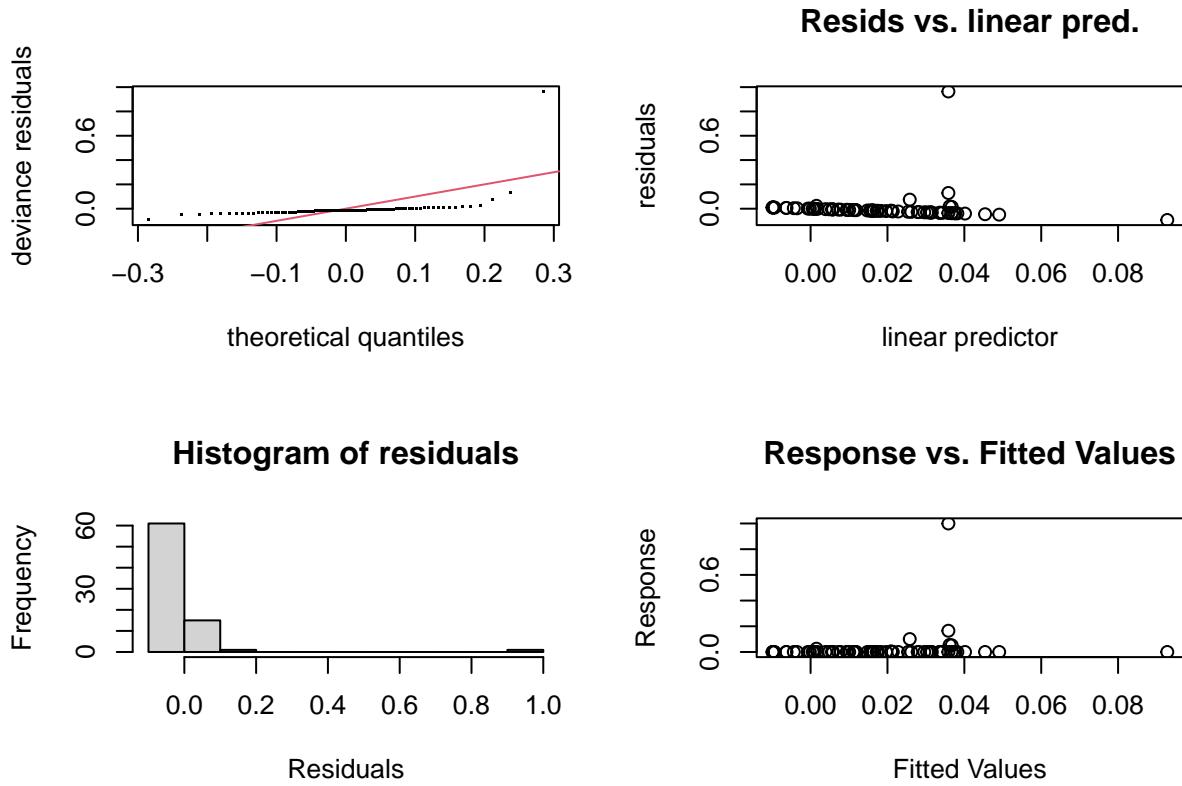
```



```

##
## Method: GCV Optimizer: magic
## Smoothing parameter selection converged after 5 iterations.
## The RMS GCV score gradient at convergence was 1.620426e-07 .
## The Hessian was positive definite.
## Model rank = 10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##          k'  edf k-index p-value
## s(DO) 9.00 5.15    0.98    0.19
## $mfrow
## [1] 2 2
##
## [1] "Centropages abdominalis eDNA Index vs Dissolved Oxygen"
##
## Family: gaussian
## Link function: identity
##
## Formula:
## eDNA_index ~ s(DO)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.01798   0.01296   1.388   0.169
##
## Approximate significance of smooth terms:
##          edf Ref.df      F p-value
## s(DO)    1     1 1.769   0.187
##
## R-sq.(adj) = 0.00989 Deviance explained = 2.27%
## GCV = 0.013441 Scale est. = 0.013096 n = 78
## [1] "GAM quality"
## [1] "AIC: -112.834263153192"
##      GCV.Cp
## 0.01344104

```

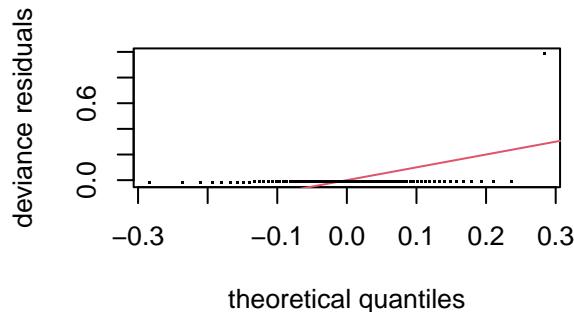


```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 8 iterations.
## The RMS GCV score gradient at convergence was 1.582062e-07 .
## The Hessian was positive definite.
## Model rank = 10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##      k' edf k-index p-value
## s(DO) 9    1    0.89   0.046 *
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## $mfrow
## [1] 2 2
##
## [1] "Clausocalanus parapergens eDNA Index vs Dissolved Oxygen"
##
## Family: gaussian
## Link function: identity
##
## Formula:
## eDNA_index ~ s(DO)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.01286   0.01290   0.997   0.322
##
```

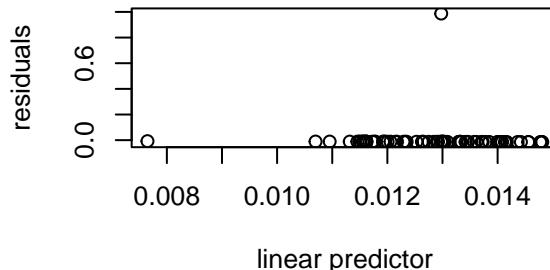
```

## Approximate significance of smooth terms:
##          edf Ref.df      F p-value
## s(DO)    1     1 0.009  0.926
## 
## R-sq.(adj) = -0.013  Deviance explained = 0.0114%
## GCV = 0.013328  Scale est. = 0.012987 n = 78
## [1] "GAM quality"
## [1] "AIC: -113.490730747714"
##      GCV.Cp
## 0.0133284

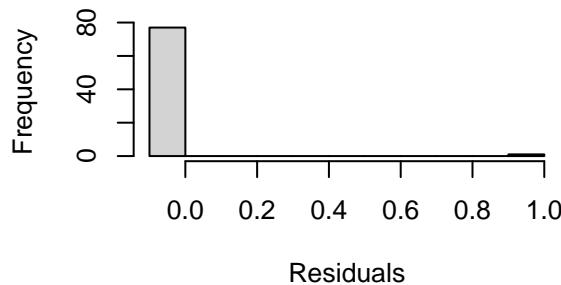
```



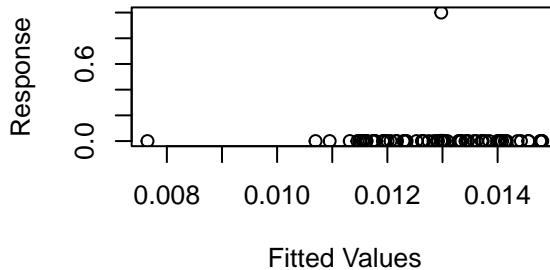
Resids vs. linear pred.



Histogram of residuals



Response vs. Fitted Values



```

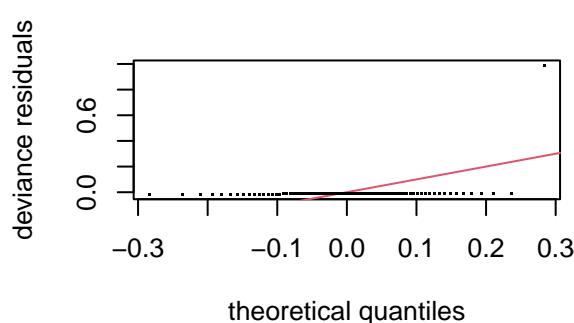
## 
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 9 iterations.
## The RMS GCV score gradient at convergence was 7.296207e-08 .
## The Hessian was positive definite.
## Model rank = 10 / 10
## 
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
## 
##          k' edf k-index p-value
## s(DO)    9     1    1.03    0.53
## $mfrow
## [1] 2 2
## 
## [1] "Clausocalanus pergens eDNA Index vs Dissolved Oxygen"
## 
## Family: gaussian
## Link function: identity

```

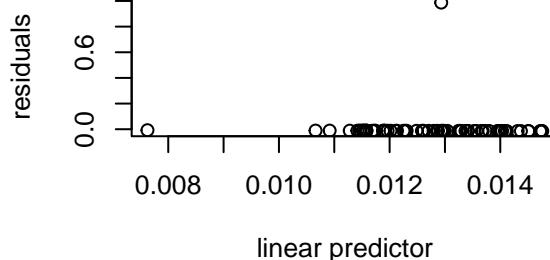
```

##
## Formula:
## eDNA_index ~ s(DO)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01282   0.01290   0.994   0.324
##
## Approximate significance of smooth terms:
##          edf Ref.df    F p-value
## s(DO)     1     1 0.009  0.926
##
## R-sq.(adj) = -0.013  Deviance explained = 0.0113%
## GCV = 0.01333  Scale est. = 0.012988 n = 78
## [1] "GAM quality"
## [1] "AIC: -113.484174498507"
##      GCV.Cp
## 0.01332952

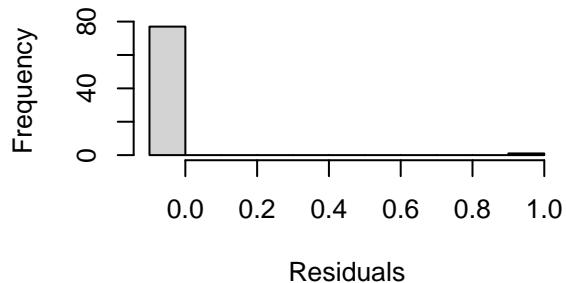
```



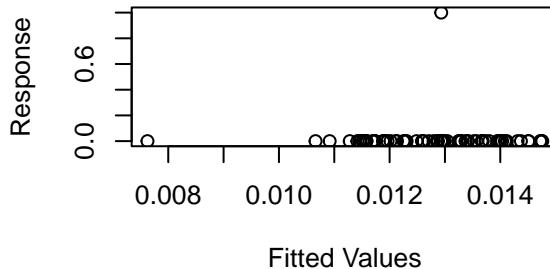
Resids vs. linear pred.



Histogram of residuals



Response vs. Fitted Values



```

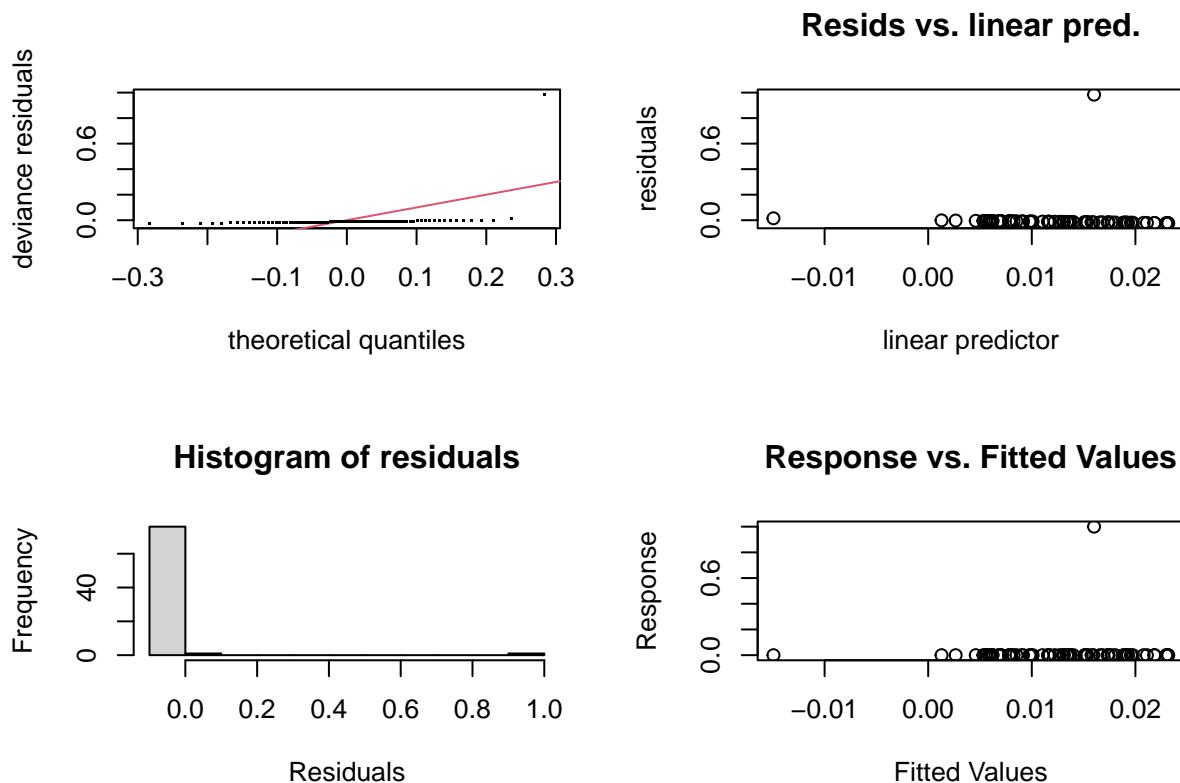
##
## Method: GCV Optimizer: magic
## Smoothing parameter selection converged after 9 iterations.
## The RMS GCV score gradient at convergence was 7.323085e-08 .
## The Hessian was positive definite.
## Model rank = 10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##      k' edf k-index p-value

```

```

## s(DO) 9 1 1.03 0.52
## $mfrow
## [1] 2 2
##
## [1] "Diacyclops incolotaenia eDNA Index vs Dissolved Oxygen"
##
## Family: gaussian
## Link function: identity
##
## Formula:
## eDNA_index ~ s(DO)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.01282   0.01288  0.995   0.323
##
## Approximate significance of smooth terms:
##          edf Ref.df    F p-value
## s(DO)     1      1 0.246  0.621
##
## R-sq.(adj) = -0.00988 Deviance explained = 0.323%
## GCV = 0.013288 Scale est. = 0.012947 n = 78
## [1] "GAM quality"
## [1] "AIC: -113.727748143016"
## GCV.Cp
## 0.01328796

```



```

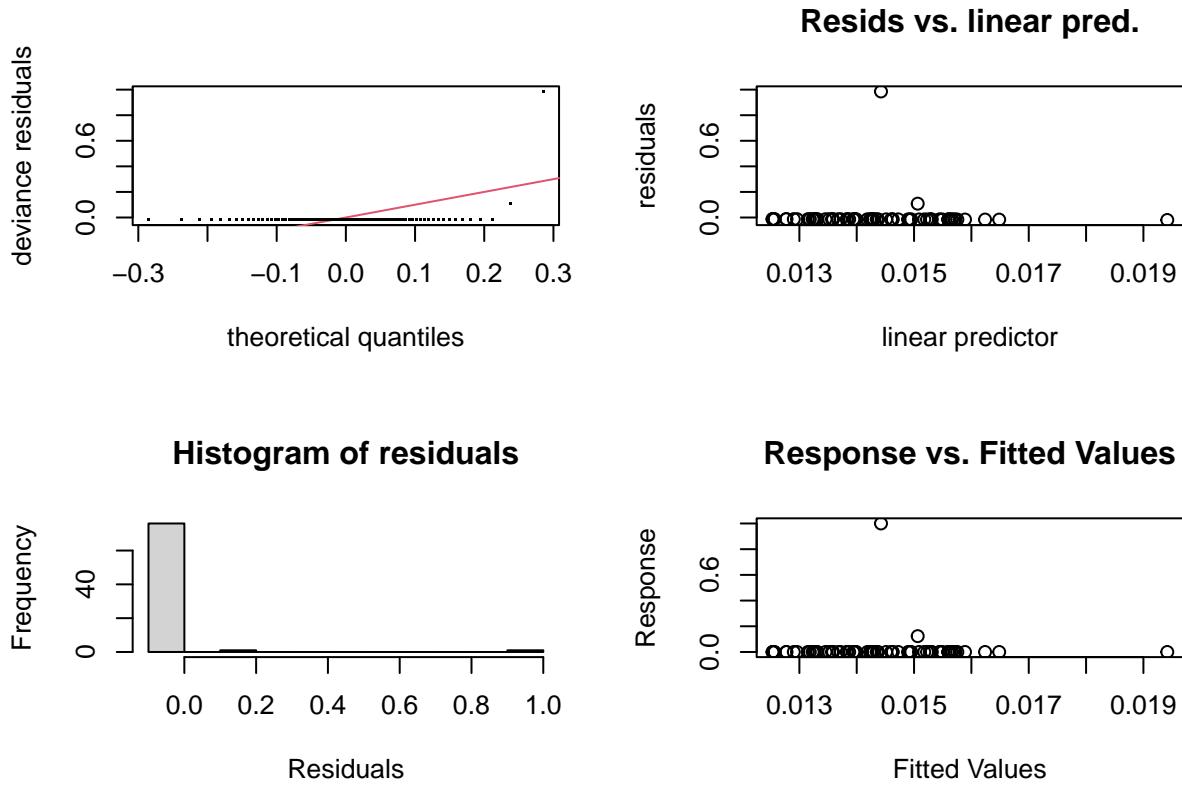
##
## Method: GCV Optimizer: magic
## Smoothing parameter selection converged after 9 iterations.

```

```

## The RMS GCV score gradient at convergence was 8.729725e-08 .
## The Hessian was positive definite.
## Model rank = 10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##      k' edf k-index p-value
## s(DO) 9     1     1.03    0.69
## $mfrow
## [1] 2 2
##
## [1] "Lucicutia flavicornis eDNA Index vs Dissolved Oxygen"
##
## Family: gaussian
## Link function: identity
##
## Formula:
## eDNA_index ~ s(DO)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.01440   0.01298   1.109   0.271
##
## Approximate significance of smooth terms:
##      edf Ref.df    F p-value
## s(DO) 1     1 0.008  0.929
##
## R-sq.(adj) = -0.0131 Deviance explained = 0.0104%
## GCV = 0.01349 Scale est. = 0.013144 n = 78
## [1] "GAM quality"
## [1] "AIC: -112.553063140206"
##      GCV.Cp
## 0.01348959

```



```

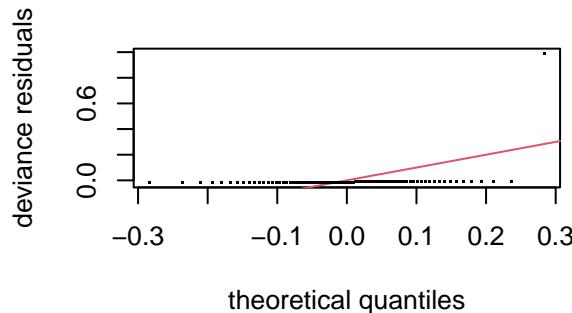
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 9 iterations.
## The RMS GCV score gradient at convergence was 1.050211e-07 .
## The Hessian was positive definite.
## Model rank = 10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##      k' edf k-index p-value
## s(DO) 9     1    1.03    0.61
## $mfrow
## [1] 2 2
##
## [1] "Metridia lucens eDNA Index vs Dissolved Oxygen"
##
## Family: gaussian
## Link function: identity
##
## Formula:
## eDNA_index ~ s(DO)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.01282   0.01290   0.994   0.324
##
## Approximate significance of smooth terms:
##          edf Ref.df    F p-value

```

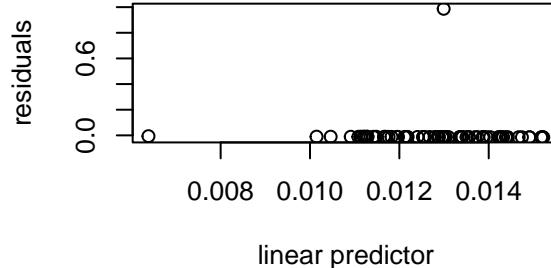
```

## s(DO)    1      1  0.013   0.909
##
## R-sq.(adj) = -0.013  Deviance explained = 0.0174%
## GCV = 0.013329  Scale est. = 0.012987 n = 78
## [1] "GAM quality"
## [1] "AIC: -113.488891135054"
##     GCV.Cp
## 0.01332871

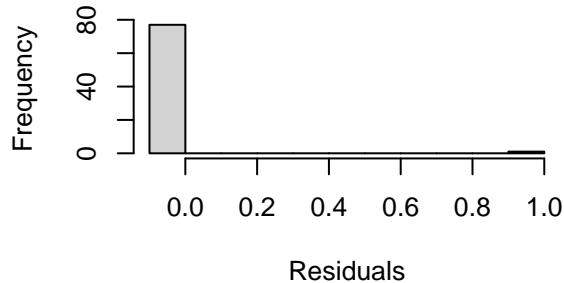
```



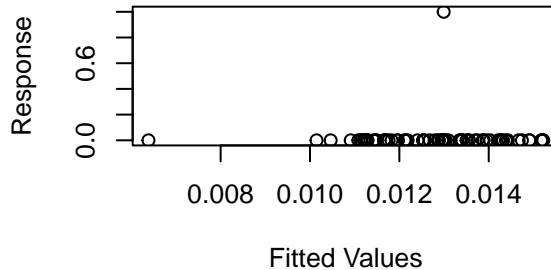
Resids vs. linear pred.



Histogram of residuals



Response vs. Fitted Values



```

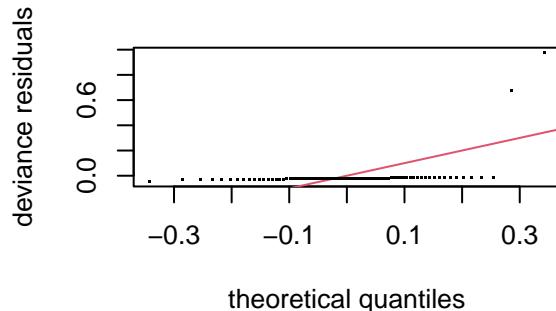
##
## Method: GCV Optimizer: magic
## Smoothing parameter selection converged after 9 iterations.
## The RMS GCV score gradient at convergence was 7.384722e-08 .
## The Hessian was positive definite.
## Model rank = 10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##          k' edf k-index p-value
## s(DO)    9    1    1.03    0.53
## $mfrow
## [1] 2 2
##
## [1] "Metridia pacifica eDNA Index vs Dissolved Oxygen"
##
## Family: gaussian
## Link function: identity
##
## Formula:

```

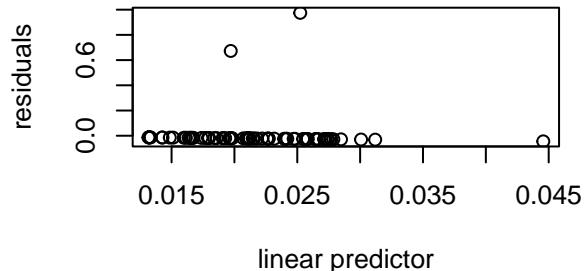
```

## eDNA_index ~ s(DO)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.02169   0.01559   1.392   0.168
##
## Approximate significance of smooth terms:
##          edf Ref.df    F p-value
## s(DO)     1     1 0.114   0.736
##
## R-sq.(adj) = -0.0116 Deviance explained = 0.15%
## GCV = 0.019448 Scale est. = 0.018949 n = 78
## [1] "GAM quality"
## [1] "AIC: -84.0181242154277"
##      GCV.Cp
## 0.01944813

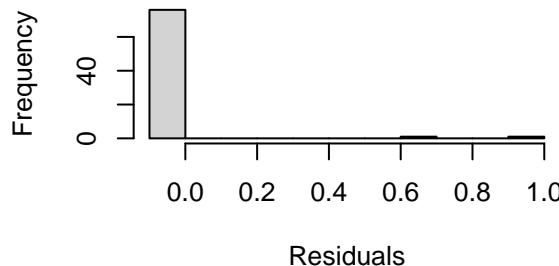
```



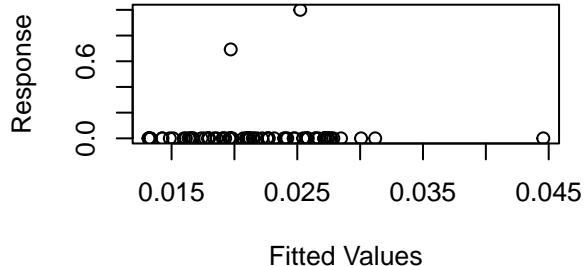
Resids vs. linear pred.



Histogram of residuals



Response vs. Fitted Values



```

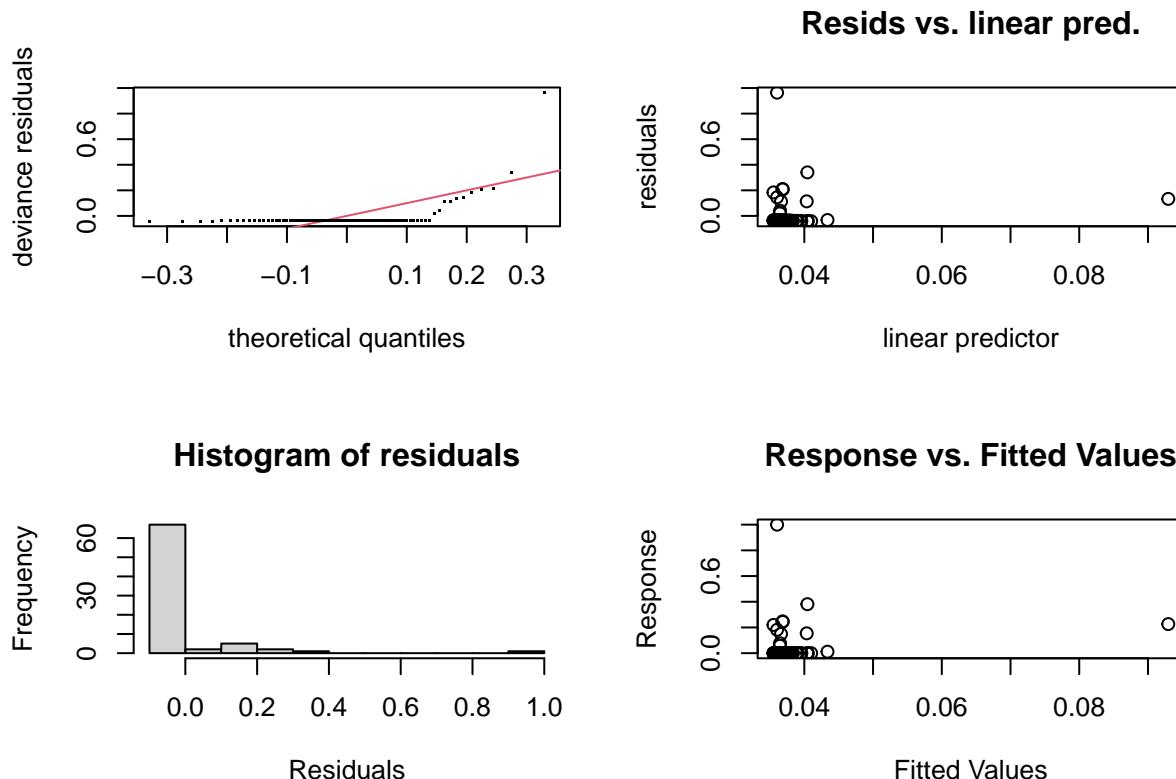
##
## Method: GCV Optimizer: magic
## Smoothing parameter selection converged after 9 iterations.
## The RMS GCV score gradient at convergence was 1.129112e-07 .
## The Hessian was positive definite.
## Model rank = 10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##          k' edf k-index p-value
## s(DO)    9    1    1.04    0.63
## $mfrow

```

```

## [1] 2 2
##
## [1] "Oithona similis eDNA Index vs Dissolved Oxygen"
##
## Family: gaussian
## Link function: identity
##
## Formula:
## eDNA_index ~ s(DO)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03781   0.01499   2.523   0.0137 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df    F p-value
## s(DO) 1.407 1.711 0.267  0.789
##
## R-sq.(adj) = -0.00515  Deviance explained = 1.32%
## GCV = 0.018078  Scale est. = 0.01752 n = 78
## [1] "GAM quality"
## [1] "AIC: -89.7414115555766"
##      GCV.Cp
## 0.01807768

```



```

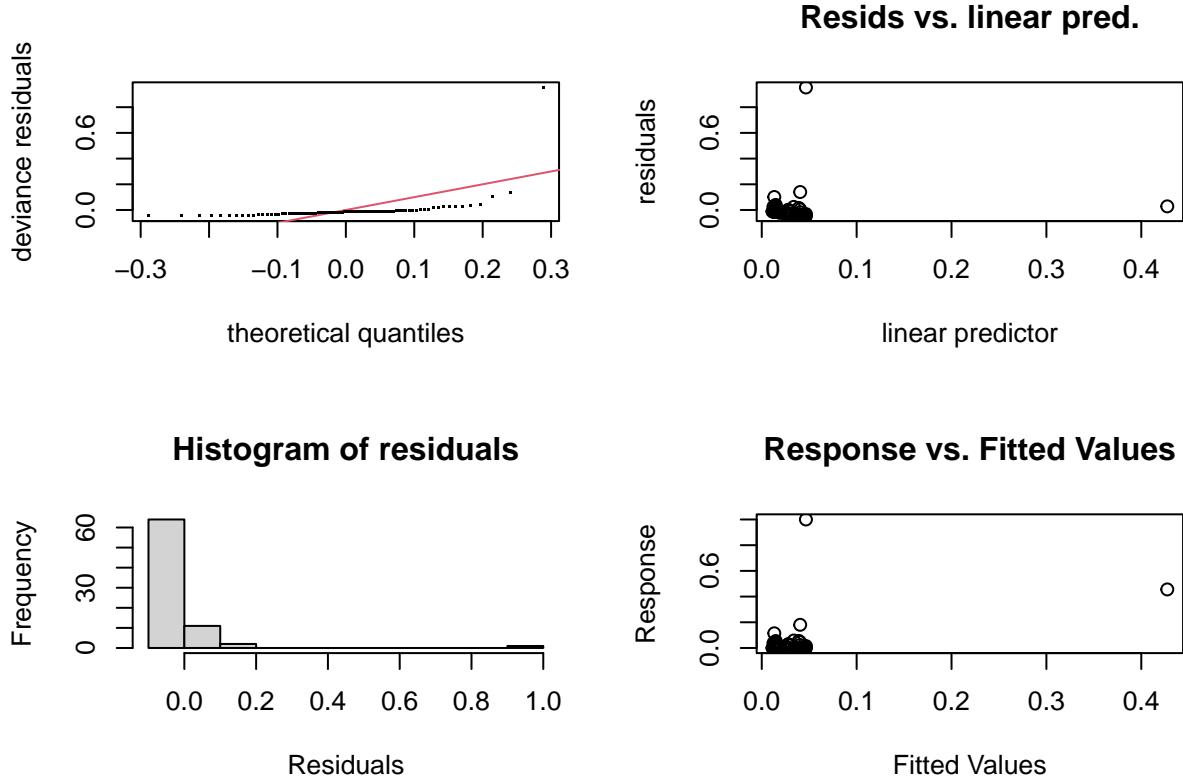
##
## Method: GCV Optimizer: magic
## Smoothing parameter selection converged after 5 iterations.

```

```

## The RMS GCV score gradient at convergence was 3.187024e-07 .
## The Hessian was positive definite.
## Model rank = 10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##          k'   edf k-index p-value
## s(DO) 9.00 1.41    0.93    0.17
## $mfrow
## [1] 2 2
##
## [1] "Paracalanus sp. C AC-2013 eDNA Index vs Dissolved Oxygen"
##
## Family: gaussian
## Link function: identity
##
## Formula:
## eDNA_index ~ s(DO)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.02998    0.01313   2.283   0.0253 *
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df F p-value
## s(DO) 3.691 4.473 3.131 0.0156 *
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.14 Deviance explained = 18.2%
## GCV = 0.014306 Scale est. = 0.013446 n = 78
## [1] "GAM quality"
## [1] "AIC: -108.210096766989"
##      GCV.Cp
## 0.01430627

```



```

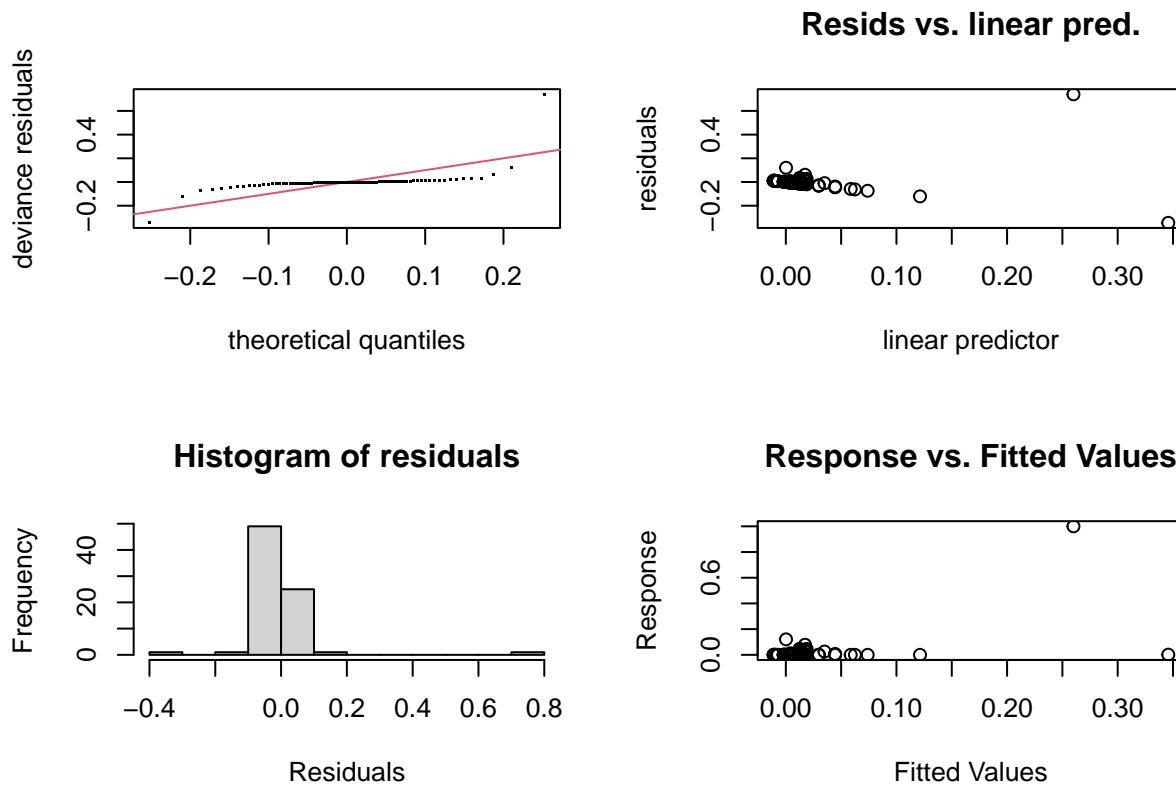
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 4 iterations.
## The RMS GCV score gradient at convergence was 5.122255e-11 .
## The Hessian was positive definite.
## Model rank = 10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##      k'    edf  k-index p-value
## s(DO) 9.00 3.69     1.08    0.98
## $mfrow
## [1] 2 2
##
## [1] "Pseudocalanus newmani eDNA Index vs Dissolved Oxygen"
##
## Family: gaussian
## Link function: identity
##
## Formula:
## eDNA_index ~ s(DO)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.02007   0.01147    1.75   0.0845 .
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##

```

```

## Approximate significance of smooth terms:
##          edf Ref.df      F p-value
## s(DO) 6.059 7.112 3.231 0.00492 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## R-sq.(adj) =  0.211  Deviance explained = 27.3%
## GCV = 0.011282  Scale est. = 0.010261 n = 78
## [1] "GAM quality"
## [1] "AIC: -127.120364551944"
## GCV.Cp
## 0.011282

```



```

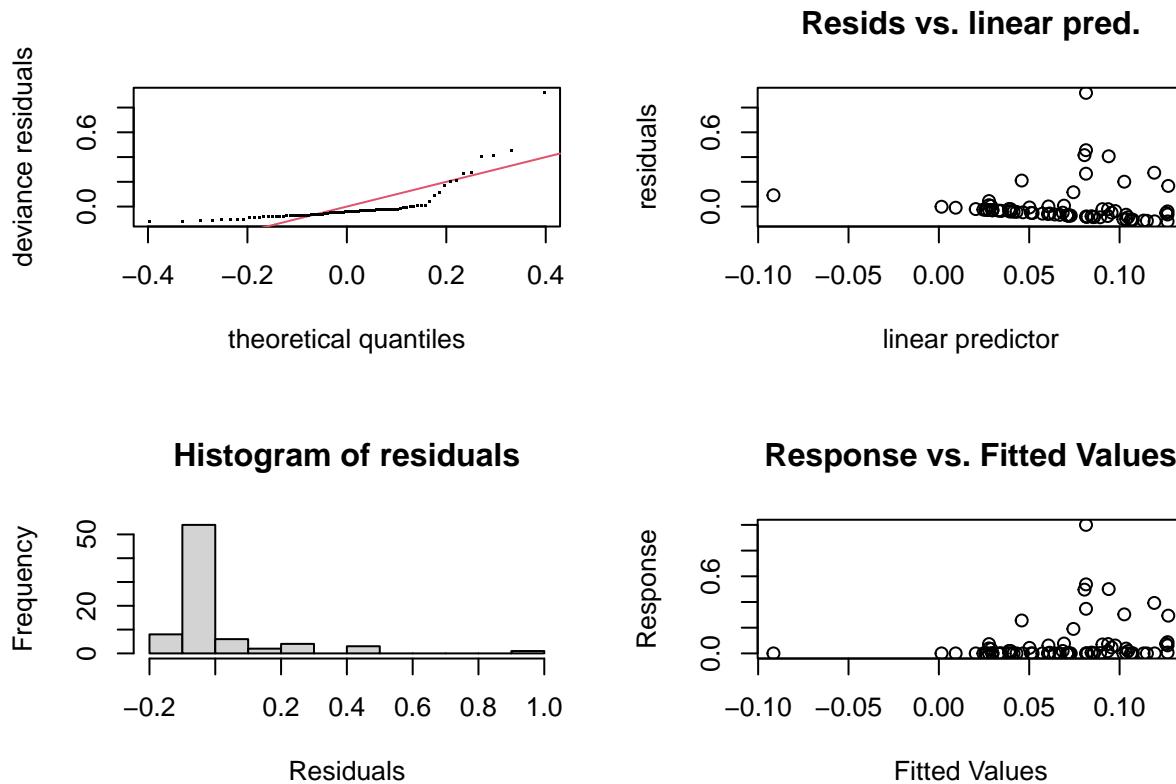
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 6 iterations.
## The RMS GCV score gradient at convergence was 1.074935e-06 .
## The Hessian was positive definite.
## Model rank = 10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##          k'    edf  k-index p-value
## s(DO) 9.00 6.06     1.46      1
## $mfrow
## [1] 2 2
##
## [1] "Thermocyclops inversus eDNA Index vs Dissolved Oxygen"
## 

```

```

## Family: gaussian
## Link function: identity
##
## Formula:
## eDNA_index ~ s(DO)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.06759   0.01806   3.742 0.000352 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df      F p-value
## s(DO)    1     1 4.118  0.0459 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0389  Deviance explained = 5.14%
## GCV = 0.02612  Scale est. = 0.02545 n = 78
## [1] "GAM quality"
## [1] "AIC: -61.0125342830331"
##       GCV.Cp
## 0.02611981

```



```

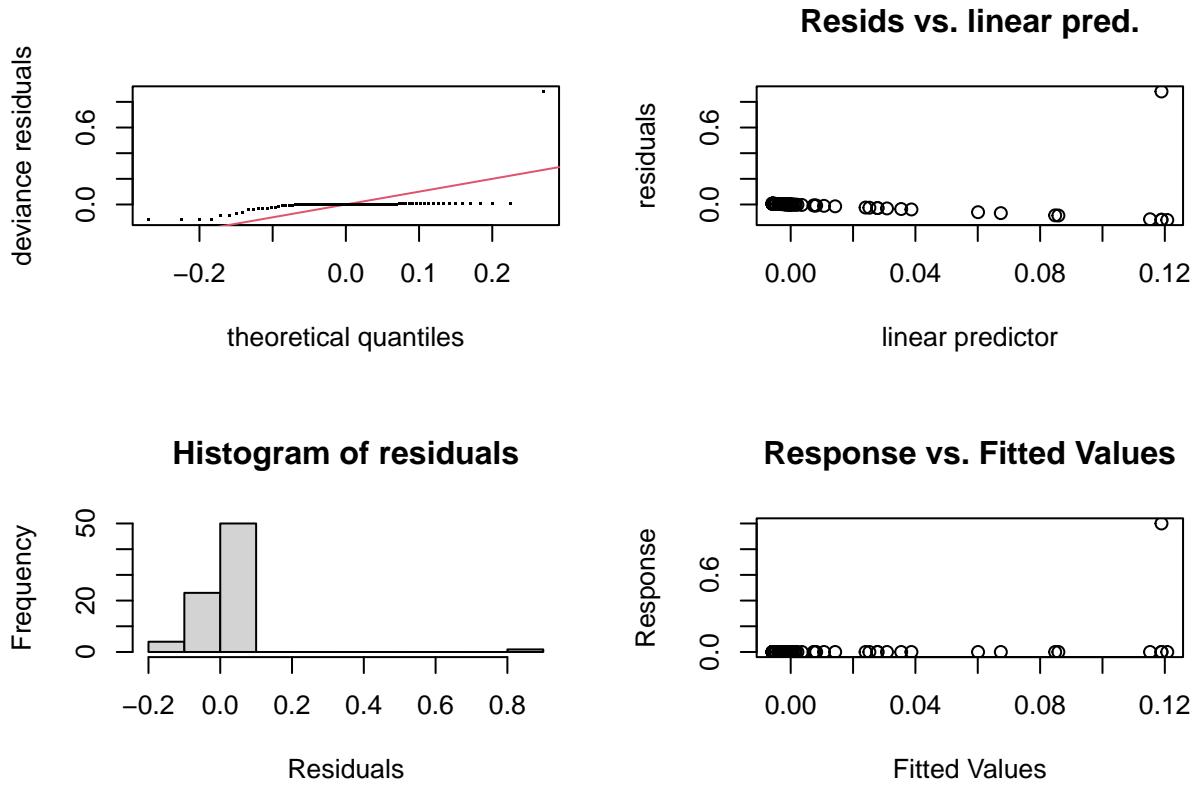
##
## Method: GCV Optimizer: magic
## Smoothing parameter selection converged after 10 iterations.
## The RMS GCV score gradient at convergence was 6.221389e-08 .
## The Hessian was positive definite.

```

```

## Model rank = 10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##      k' edf k-index p-value
## s(DO) 9     1     0.53   0.001 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## $mfrow
## [1] 2 2
##
## [1] "Triconia minuta eDNA Index vs Dissolved Oxygen"
##
## Family: gaussian
## Link function: identity
##
## Formula:
## eDNA_index ~ s(DO)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.01282   0.01227   1.045   0.299
##
## Approximate significance of smooth terms:
##          edf Ref.df    F p-value
## s(DO) 3.516 4.264 1.961 0.0941 .
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## R-sq.(adj) = 0.0841 Deviance explained = 12.6%
## GCV = 0.012464 Scale est. = 0.011743 n = 78
## [1] "GAM quality"
## [1] "AIC: -118.938625733174"
##      GCV.Cp
## 0.01246433

```



```

## 
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 4 iterations.
## The RMS GCV score gradient at convergence was 6.255304e-16 .
## The Hessian was positive definite.
## Model rank = 10 / 10
## 
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
## 
##      k'    edf  k-index p-value
## s(D0) 9.00 3.52     1.18     0.99
## $mfrow
## [1] 2 2

# edf = effective degrees of freedom
# GCV = generalized cross validation, an estimate of the mean square prediction error.
# ↪ comparing between models, lower GCV is better
# Lower AIC is also better
# Higher R^2 is better

system("say Script complete")

```