GAMs to Analyze Plankton Comunity NMDS Data – Final Additions

Curtis C. Bohlen, Casco Bay Estuary Partnership

5/17/2022

Contents

Introduction

This notebook is a summary of my efforts to explore approaches to the analysis of plankton data from the Penobscot Estuary. Here I omit most exploratory data analysis and most alternative model formulations, and include only final models.

This Notebook looks at:

1. ANOVA models predicting environmental variables based on Season and Station
2. Non-linear fits between zooplankton density and possible environmental drivers;
3. Links between Shannon Diversity and environmental drivers
4. A GAM model looking at environmental drivers of River Herring abundance.
5. Responses of individual species to those same drivers.

I've trimmed down the analysis workflow, since I looked at the data distributions, autocorrelation structure, etc. previously, but the major steps remain the same.

Note that explicit modeling of correlation groups using hierarchical models proves to be fairly important in modelling these data.

Load Libraries

```
library(tidyverse)
#> -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
#> v ggplot2 3.3.6     v purrr   0.3.4
#> v tibble  3.1.7     v dplyr   1.0.9
#> v tidyr   1.2.0     v stringr 1.4.0
```

```
#> v readr   2.1.2     v forcats 0.5.1
#> -- Conflicts ------------------------------------------ tidyverse_conflicts() --
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()    masks stats::lag()
library(vegan)
#> Loading required package: permute
#> Loading required package: lattice
#> This is vegan 2.6-2
library(readxl)
library(mgcv)      # for GAM models
#> Loading required package: nlme
#>
#> Attaching package: 'nlme'
#> The following object is masked from 'package:dplyr':
#>
#>     collapse
#> This is mgcv 1.8-40. For overview type 'help("mgcv-package")'.
library(emmeans)   # For extracting useful "marginal" model summaries
```

Set Graphics Theme

This sets `ggplot()`graphics for no background, no grid lines, etc. in a clean format suitable for (some) publications.

```
theme_set(theme_classic())
```

Input Data

Folder References

```
data_folder <- "Original_Data"
```

Load Data

```
filename.in <- "penob.station.data EA 3.12.20.xlsx"
file_path <- file.path(data_folder, filename.in)
station_data <- read_excel(file_path,
                           sheet="Final", col_types = c("skip", "date",
                                          "numeric", "text", "numeric",
                                          "text", "skip", "skip",
                                          "skip",
                                          rep("numeric", 10),
                                          "text",
                                          rep("numeric", 47),
                                          "text",
                                          rep("numeric", 12))) %>%
```

```r
  rename_with(~ gsub(" ", "_", .x)) %>%
  rename_with(~ gsub("\\.", "_", .x)) %>%
  rename_with(~ gsub("\\?", "", .x)) %>%
  rename_with(~ gsub("%", "pct", .x)) %>%
  rename_with(~ gsub("_Abundance", "", .x)) %>%
  filter(! is.na(date))
#> New names:
#> * `` -> `...61`
```

Station names are arbitrary, and Erin previously expressed interest in renaming them from Stations 2, 4, 5 and 8 to Stations 1,2,3,and 4.

The `factor()` function by default sorts levels before assigning numeric codes, so a convenient way to replace the existing station codes with sequential numbers is to create a factor and extract the numeric indicator values with `as.numeric()`.

```r
station_data <- station_data %>%
  mutate(station = factor(as.numeric(factor(station))))
head(station_data)
#> # A tibble: 6 x 76
#>   date                 year month month_num season riv_km station station_num
#>   <dttm>              <dbl> <chr>     <dbl> <chr>   <dbl> <fct>         <dbl>
#> 1 2013-05-28 00:00:00  2013 May           5 Spring  22.6  1                 1
#> 2 2013-05-28 00:00:00  2013 May           5 Spring  13.9  2                 2
#> 3 2013-05-28 00:00:00  2013 May           5 Spring   8.12 3                 3
#> 4 2013-05-28 00:00:00  2013 May           5 Spring   2.78 4                 4
#> 5 2013-07-25 00:00:00  2013 July          7 Summer  22.6  1                 1
#> 6 2013-07-25 00:00:00  2013 July          7 Summer  13.9  2                 2
#> # ... with 68 more variables: depth <dbl>, discharge_week_cftpersec <dbl>,
#> #   discharg_day <dbl>, discharge_week_max <dbl>, tide_height <dbl>,
#> #   Full_Moon <dbl>, Abs_Moon <dbl>, Spring_or_Neap <chr>, ave_temp_c <dbl>,
#> #   ave_sal_psu <dbl>, ave_turb_ntu <dbl>, ave_do_mgperl <dbl>,
#> #   ave_DO_Saturation <dbl>, ave_chl_microgperl <dbl>, sur_temp <dbl>,
#> #   sur_sal <dbl>, sur_turb <dbl>, sur_do <dbl>, sur_chl <dbl>, bot_temp <dbl>,
#> #   bot_sal <dbl>, bot_turb <dbl>, bot_do <dbl>, bot_chl <dbl>, ...
```

Subsetting to Desired Data Columns

I base selection of predictor variables here on the ones used in the manuscript.

```r
base_data <- station_data %>%
  rename(Date = date,
         Station = station,
         Year = year) %>%
  select(-c(month, month_num)) %>%
  mutate(Month = factor(as.numeric(format(Date, format = '%m')),
                                          levels = 1:12,
                                          labels = month.abb),
         DOY = as.numeric(format(Date,format = '%j')),
         season = factor(season, levels = c('Spring', 'Summer', 'Fall')),
         Yearf = factor(Year)) %>%
  rename(Season = season,
         Temp = ave_temp_c,
```

```
         Sal = ave_sal_psu,
         Turb = sur_turb,
         AvgTurb = ave_turb_ntu,
         DOsat = ave_DO_Saturation,
         Chl = ave_chl_microgperl,
         RH = Herring
         ) %>%
  select(Date, Station, Year, Yearf, Month, Season, DOY, riv_km, Temp, Sal, Turb, AvgTurb,
         DOsat, Chl, RH,
         combined_density,H, SEI,
         Acartia, Balanus, Eurytemora, Polychaete, Pseudocal, Temora) %>%
  arrange(Date, Station)
head(base_data)
#> # A tibble: 6 x 24
#>   Date                Station  Year Yearf Month Season   DOY riv_km  Temp
#>   <dttm>              <fct>   <dbl> <fct> <fct> <fct>  <dbl>  <dbl> <dbl>
#> 1 2013-05-28 00:00:00 1        2013 2013  May   Spring   148  22.6  11.7
#> 2 2013-05-28 00:00:00 2        2013 2013  May   Spring   148  13.9   9.40
#> 3 2013-05-28 00:00:00 3        2013 2013  May   Spring   148   8.12  6.97
#> 4 2013-05-28 00:00:00 4        2013 2013  May   Spring   148   2.78  9.51
#> 5 2013-07-25 00:00:00 1        2013 2013  Jul   Summer   206  22.6  18.5
#> 6 2013-07-25 00:00:00 2        2013 2013  Jul   Summer   206  13.9  13.6
#> # ... with 15 more variables: Sal <dbl>, Turb <dbl>, AvgTurb <dbl>,
#> #   DOsat <dbl>, Chl <dbl>, RH <dbl>, combined_density <dbl>, H <dbl>,
#> #   SEI <dbl>, Acartia <dbl>, Balanus <dbl>, Eurytemora <dbl>,
#> #   Polychaete <dbl>, Pseudocal <dbl>, Temora <dbl>
```

```
rm(station_data)
```

Add Transformed Predictors

We can treat the sampling history as "spring", "summer" and "fall" observations each year from 2013 through 2017. This breaks the temporal pattern down into integer valued time, generating a "quasi regular" time series, and allowing us to simplify the analysis of temporal autocorrelation. The "real world" time difference across the winter is longer that between seasons, but I could not find a ready way to address that.

We need both the numerical sequence and a factor later, for different purposes.

```
base_data <- base_data %>%
  mutate(sample_seq = as.numeric(Season) + (Year-2013)*3,
         sample_event = factor(sample_seq))
```

Environmental Predictors

First, we look at simple linear models to predict our environmental predictors. this gives us a way to understand how the predictors are related to location and season in the estuary.

I automate the analysis using a nested tibble.

First I create a "Long" data source.

```r
env_data <- base_data %>%
  select(Yearf, Month, Season, sample_event, Station, Temp,
          Sal, Turb, Chl, DOsat) %>%
  mutate(Turb = log(Turb),
         Chl = log(Chl)) %>%
  pivot_longer(-c(Yearf:Station), names_to = 'Parameter', values_to = 'Value')
```

Next, I create a function to run the analysis. This function takes a data frame or tibble as an argument. The tibble mush have data columns with the correct names, and all variables transformed before we call it.

```r
my_lme <- function(.dat) {

  lme(Value ~ Station * Season,
      random = list(Yearf = ~ 1, sample_event = ~ 1),
      data = .dat, na.action = na.omit)
}
```

Finally, We run the analysis on the nested tibble.

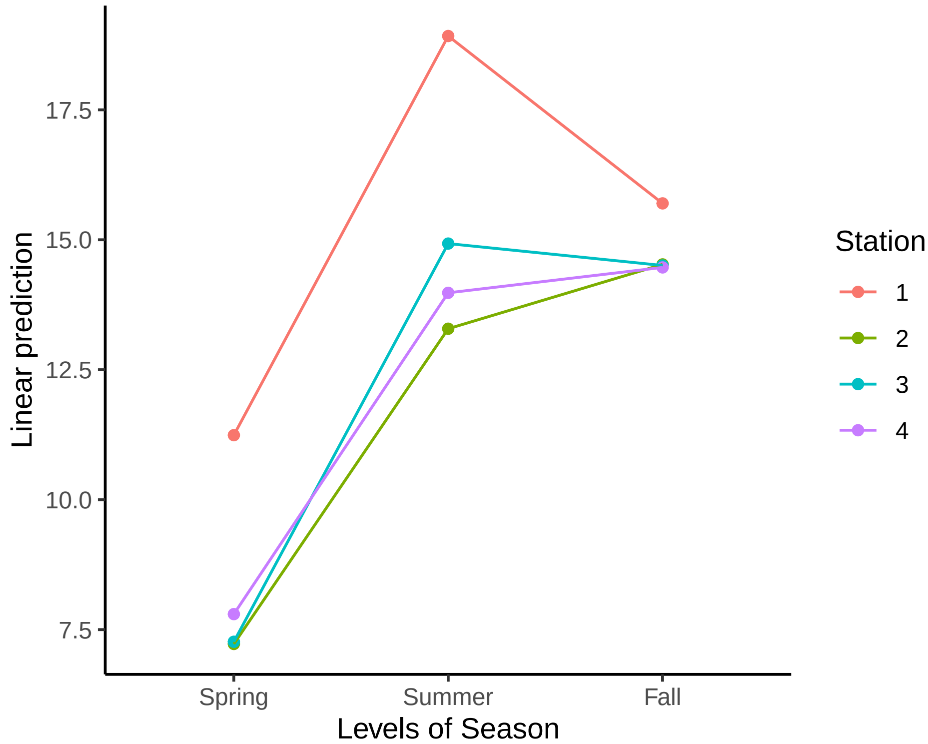
```r
env_analysis <- env_data %>%
  group_by(Parameter) %>%
  nest() %>%
  mutate(lme_mods = map(data, my_lme))
```

Temperature

```r
parm = 'Temp'
mod <- env_analysis$lme_mods[env_analysis$Parameter == parm][[1]]
anova(mod)
#>                numDF denDF  F-value p-value
#> (Intercept)        1    36 723.7796  <.0001
#> Station            3    36  85.7503  <.0001
#> Season             2     8  40.9458   1e-04
#> Station:Season     6    36  11.0275  <.0001
```

Temperature is affected by Season, Station, and their interaction.

```r
emmip(mod, Station ~ Season)
```




```
emmeans(mod, pairwise ~ Station | Season)
#> $emmeans
#> Season = Spring:
#>  Station emmean    SE df lower.CL upper.CL
#>  1        11.24 0.735  4     9.20    13.28
#>  2         7.22 0.735  4     5.18     9.27
#>  3         7.27 0.735  4     5.22     9.31
#>  4         7.80 0.735  4     5.76     9.84
#>
#> Season = Summer:
#>  Station emmean    SE df lower.CL upper.CL
#>  1        18.92 0.735  4    16.88    20.96
#>  2        13.29 0.735  4    11.25    15.33
#>  3        14.93 0.735  4    12.88    16.97
#>  4        13.98 0.735  4    11.94    16.02
#>
#> Season = Fall:
#>  Station emmean    SE df lower.CL upper.CL
#>  1        15.70 0.735  4    13.66    17.74
#>  2        14.53 0.735  4    12.48    16.57
#>  3        14.51 0.735  4    12.46    16.55
#>  4        14.47 0.735  4    12.43    16.51
#>
#> Degrees-of-freedom method: containment
#> Confidence level used: 0.95
#>
#> $contrasts
#> Season = Spring:
#>  contrast            estimate    SE df t.ratio p.value
```

```
#>  Station1 - Station2   4.0170 0.439 36   9.143  <.0001
#>  Station1 - Station3   3.9747 0.439 36   9.047  <.0001
#>  Station1 - Station4   3.4429 0.439 36   7.836  <.0001
#>  Station2 - Station3  -0.0423 0.439 36  -0.096  0.9997
#>  Station2 - Station4  -0.5741 0.439 36  -1.307  0.5648
#>  Station3 - Station4  -0.5318 0.439 36  -1.210  0.6244
#>
#> Season = Summer:
#>  contrast            estimate    SE df t.ratio p.value
#>  Station1 - Station2   5.6316 0.439 36  12.818  <.0001
#>  Station1 - Station3   3.9934 0.439 36   9.089  <.0001
#>  Station1 - Station4   4.9407 0.439 36  11.245  <.0001
#>  Station2 - Station3  -1.6382 0.439 36  -3.729  0.0035
#>  Station2 - Station4  -0.6909 0.439 36  -1.573  0.4066
#>  Station3 - Station4   0.9473 0.439 36   2.156  0.1552
#>
#> Season = Fall:
#>  contrast            estimate    SE df t.ratio p.value
#>  Station1 - Station2   1.1739 0.439 36   2.672  0.0525
#>  Station1 - Station3   1.1942 0.439 36   2.718  0.0472
#>  Station1 - Station4   1.2318 0.439 36   2.804  0.0387
#>  Station2 - Station3   0.0203 0.439 36   0.046  1.0000
#>  Station2 - Station4   0.0579 0.439 36   0.132  0.9992
#>  Station3 - Station4   0.0376 0.439 36   0.086  0.9998
#>
#> Degrees-of-freedom method: containment
#> P value adjustment: tukey method for comparing a family of 4 estimates
```

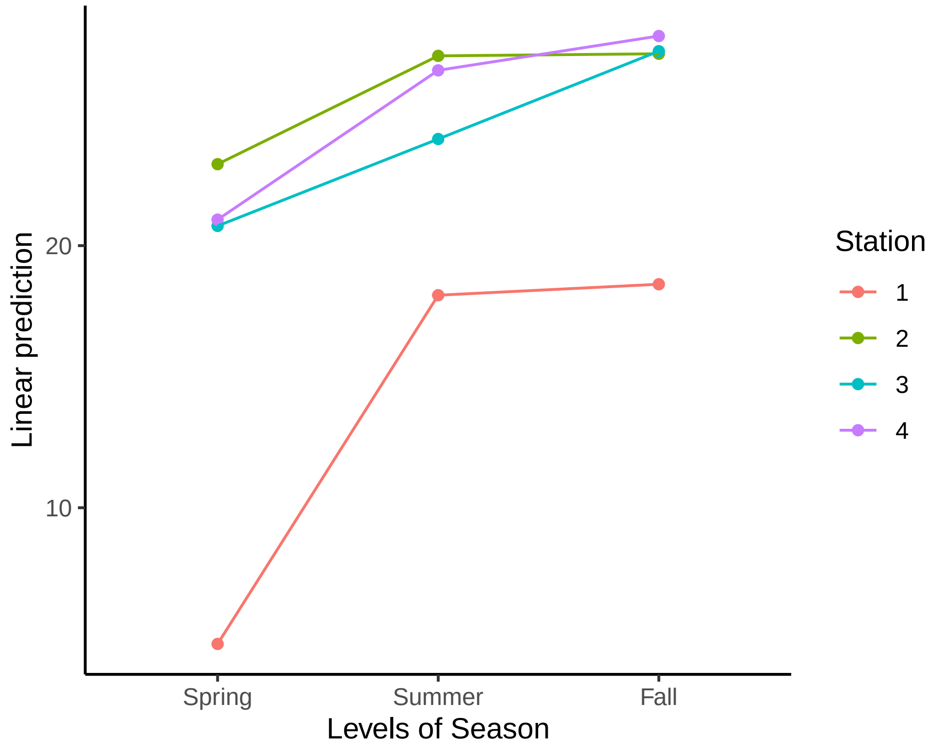
Stations 2, 3 and 4 pretty much all work the same way, with Spring significantly cooler than summer and fall. But water temperatures upstream begin to drop in the fall at Station 1, perhaps because of lower freshwater inflows, perhaps because waters on land begin to cool before ocean waters.

Salinity

```
parm = 'Sal'
mod <- env_analysis$lme_mods[env_analysis$Parameter == parm][[1]]
anova(mod)
#>                numDF denDF  F-value p-value
#> (Intercept)        1    36 735.1896  <.0001
#> Station            3    36  37.2132  <.0001
#> Season             2     8  14.2467  0.0023
#> Station:Season     6    36   2.5453  0.0370
```

Salinity is also affected by Season, Station, and their interaction.

```
emmip(mod, Station ~ Season)
```




```
emmeans(mod, pairwise ~ Station | Season)
#> $emmeans
#> Season = Spring:
#>  Station emmean   SE df lower.CL upper.CL
#>  1          4.8 1.86  4   -0.365     9.97
#>  2         23.1 1.86  4   17.943    28.28
#>  3         20.8 1.86  4   15.586    25.92
#>  4         21.0 1.86  4   15.826    26.16
#>
#> Season = Summer:
#>  Station emmean   SE df lower.CL upper.CL
#>  1         18.1 1.86  4   12.943    23.28
#>  2         27.2 1.86  4   22.076    32.41
#>  3         24.1 1.86  4   18.903    29.24
#>  4         26.7 1.86  4   21.523    31.86
#>
#> Season = Fall:
#>  Station emmean   SE df lower.CL upper.CL
#>  1         18.5 1.86  4   13.361    23.69
#>  2         27.3 1.86  4   22.155    32.49
#>  3         27.4 1.86  4   22.253    32.59
#>  4         28.0 1.86  4   22.833    33.17
#>
#> Degrees-of-freedom method: containment
#> Confidence level used: 0.95
#>
#> $contrasts
#> Season = Spring:
#>  contrast              estimate   SE df t.ratio p.value
```

```
#>  Station1 - Station2 -18.3076 2.28 36  -8.032  <.0001
#>  Station1 - Station3 -15.9512 2.28 36  -6.999  <.0001
#>  Station1 - Station4 -16.1910 2.28 36  -7.104  <.0001
#>  Station2 - Station3   2.3564 2.28 36   1.034  0.7309
#>  Station2 - Station4   2.1166 2.28 36   0.929  0.7897
#>  Station3 - Station4  -0.2398 2.28 36  -0.105  0.9996
#>
#> Season = Summer:
#>  contrast            estimate   SE df t.ratio p.value
#>  Station1 - Station2  -9.1327 2.28 36  -4.007  0.0016
#>  Station1 - Station3  -5.9600 2.28 36  -2.615  0.0597
#>  Station1 - Station4  -8.5802 2.28 36  -3.765  0.0032
#>  Station2 - Station3   3.1726 2.28 36   1.392  0.5124
#>  Station2 - Station4   0.5524 2.28 36   0.242  0.9949
#>  Station3 - Station4  -2.6202 2.28 36  -1.150  0.6618
#>
#> Season = Fall:
#>  contrast            estimate   SE df t.ratio p.value
#>  Station1 - Station2  -8.7943 2.28 36  -3.858  0.0025
#>  Station1 - Station3  -8.8925 2.28 36  -3.902  0.0022
#>  Station1 - Station4  -9.4718 2.28 36  -4.156  0.0011
#>  Station2 - Station3  -0.0982 2.28 36  -0.043  1.0000
#>  Station2 - Station4  -0.6776 2.28 36  -0.297  0.9907
#>  Station3 - Station4  -0.5794 2.28 36  -0.254  0.9941
#>
#> Degrees-of-freedom method: containment
#> P value adjustment: tukey method for comparing a family of 4 estimates
```

Station 1 has lower salinity all year long, but the effect is MUCH larger in spring. The other three stations show very similar patterns, with no clear differences, but slightly lower salinities in Spring.

Turbidity

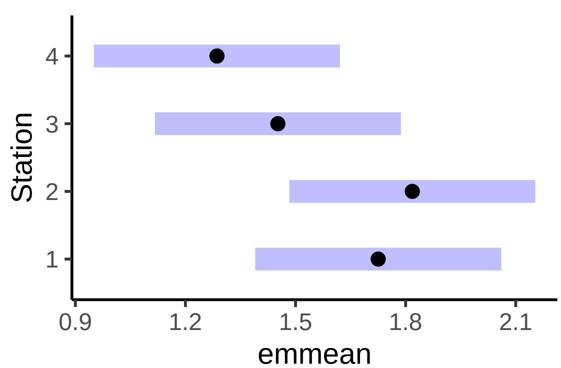
(Turbidity was analysed as a log transform)

```
parm = 'Turb'
mod <- env_analysis$lme_mods[env_analysis$Parameter == parm][[1]]
anova(mod)
#>                numDF denDF   F-value p-value
#> (Intercept)        1    36 215.35527  <.0001
#> Station            3    36  11.67827  <.0001
#> Season             2     8   0.45620  0.6492
#> Station:Season     6    36   1.27337  0.2939
```

Turbidity does NOT show a significant effect of Season or of the Season by Station interaction, so we need only consider the Station predictor. To handle this carefully, I refit a model omitting those terms.

```
tmp <- env_analysis$data[env_analysis$Parameter == parm][[1]]
test <- lme(Value ~ Station,
       random = list(Yearf = ~ 1, sample_event = ~ 1),
      data = tmp, na.action = na.omit)
(emm <- emmeans(test, pairwise~ Station))
```

```
#> $emmeans
#>  Station emmean    SE df lower.CL upper.CL
#>  1         1.73 0.121  4    1.390     2.06
#>  2         1.82 0.121  4    1.483     2.15
#>  3         1.45 0.121  4    1.117     1.79
#>  4         1.29 0.121  4    0.951     1.62
#>
#> Degrees-of-freedom method: containment
#> Confidence level used: 0.95
#>
#> $contrasts
#>  contrast            estimate    SE df t.ratio p.value
#>  Station1 - Station2   -0.093 0.103 42  -0.899  0.8053
#>  Station1 - Station3    0.273 0.103 42   2.643  0.0539
#>  Station1 - Station4    0.439 0.103 42   4.247  0.0007
#>  Station2 - Station3    0.366 0.103 42   3.542  0.0053
#>  Station2 - Station4    0.532 0.103 42   5.146  <.0001
#>  Station3 - Station4    0.166 0.103 42   1.604  0.3873
#>
#> Degrees-of-freedom method: containment
#> P value adjustment: tukey method for comparing a family of 4 estimates
plot(emm)
```



Generally, Stations 1 and 2 are associated with higher Turbidity compared to stations 3 and 4.

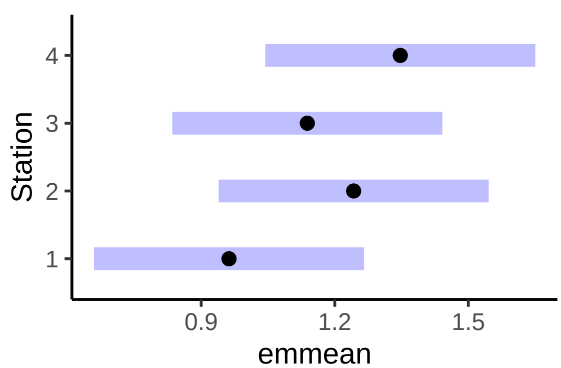
Chlorophyll

(Also log transformed for analysis)

```
parm = 'Chl'
mod <- env_analysis$lme_mods[env_analysis$Parameter == parm][[1]]
anova(mod)
#>                numDF denDF   F-value p-value
#> (Intercept)        1    36 169.60802  <.0001
#> Station            3    36   5.74446  0.0026
#> Season             2     8   6.16751  0.0240
#> Station:Season     6    36   1.61562  0.1712
```

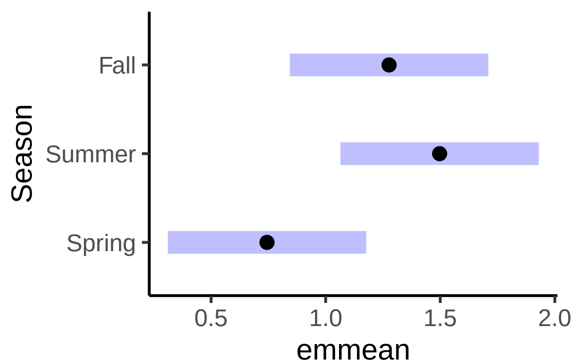
The interaction term is not significant, but this time both main effects are significant.

```r
tmp <- env_analysis$data[env_analysis$Parameter == parm][[1]]
test <- lme(Value ~ Station + Season,
       random = list(Yearf = ~ 1, sample_event = ~ 1),
      data = tmp, na.action = na.omit)

(emm_stat <- emmeans(test, pairwise~ Station))
#> $emmeans
#>  Station emmean    SE df lower.CL upper.CL
#>  1        0.963 0.109  4    0.659     1.27
#>  2        1.242 0.109  4    0.939     1.55
#>  3        1.138 0.109  4    0.835     1.44
#>  4        1.347 0.109  4    1.044     1.65
#>
#> Results are averaged over the levels of: Season
#> Degrees-of-freedom method: containment
#> Confidence level used: 0.95
#>
#> $contrasts
#>  contrast            estimate    SE df t.ratio p.value
#>  Station1 - Station2   -0.280 0.101 42  -2.774  0.0396
#>  Station1 - Station3   -0.176 0.101 42  -1.743  0.3151
#>  Station1 - Station4   -0.385 0.101 42  -3.812  0.0024
#>  Station2 - Station3    0.104 0.101 42   1.031  0.7324
#>  Station2 - Station4   -0.105 0.101 42  -1.038  0.7282
#>  Station3 - Station4   -0.209 0.101 42  -2.069  0.1799
#>
#> Results are averaged over the levels of: Season
#> Degrees-of-freedom method: containment
#> P value adjustment: tukey method for comparing a family of 4 estimates
plot(emm_stat)
```




```r
(emm_seas<- emmeans(test, pairwise~ Season))
#> $emmeans
#>  Season emmean    SE df lower.CL upper.CL
#>  Spring  0.744 0.156  4    0.311     1.18
#>  Summer  1.497 0.156  4    1.064     1.93
#>  Fall    1.277 0.156  4    0.844     1.71
#>
#> Results are averaged over the levels of: Station
```

```
#> Degrees-of-freedom method: containment
#> Confidence level used: 0.95
#>
#> $contrasts
#>  contrast         estimate    SE df t.ratio p.value
#>  Spring - Summer    -0.753 0.221  8  -3.416  0.0222
#>  Spring - Fall      -0.533 0.221  8  -2.415  0.0958
#>  Summer - Fall       0.221 0.221  8   1.000  0.5970
#>
#> Results are averaged over the levels of: Station
#> Degrees-of-freedom method: containment
#> P value adjustment: tukey method for comparing a family of 3 estimates
plot(emm_seas)
```



Generally, Station 1 and Spring are associated with lower chlorophyll.

The only statistically significant differences in Station show Station 1 is different from Station 2 and 4 (but not 3).

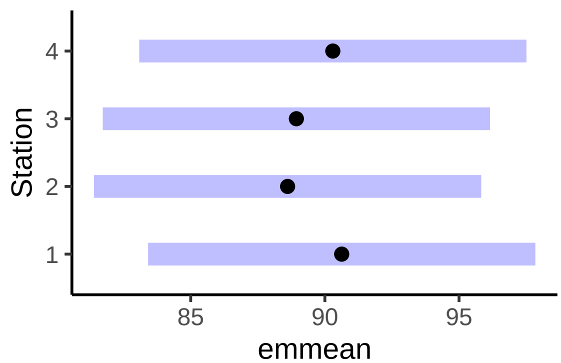
Spring is different from Summer and ALMOST different from fall.

Dissolved Oxygen Percent Saturation

```
parm = 'DOsat'
mod <- env_analysis$lme_mods[env_analysis$Parameter == parm][[1]]
anova(mod)
#>                numDF denDF   F-value p-value
#> (Intercept)        1    27 1624.5672  <.0001
#> Station            3    27    3.7837  0.0219
#> Season             2     6   16.6267  0.0036
#> Station:Season     6    27    1.0556  0.4127
```

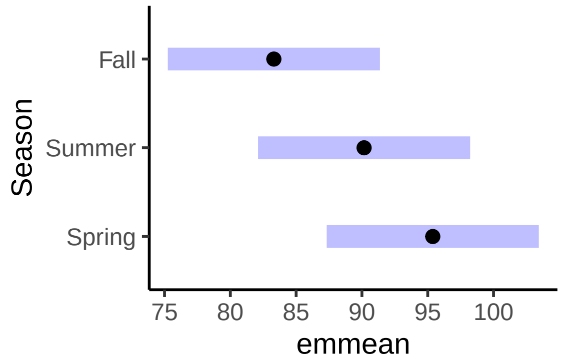
```
tmp <- env_analysis$data[env_analysis$Parameter == parm][[1]]
test <- lme(Value ~ Station + Season,
       random = list(Yearf = ~ 1, sample_event = ~ 1),
      data = tmp, na.action = na.omit)

(emm_stat <- emmeans(test, pairwise~ Station))
```

```
#> $emmeans
#>  Station emmean   SE df lower.CL upper.CL
#>  1         90.6 2.27  3     83.4     97.8
#>  2         88.6 2.27  3     81.4     95.8
#>  3         88.9 2.27  3     81.7     96.2
#>  4         90.3 2.27  3     83.1     97.5
#>
#> Results are averaged over the levels of: Season
#> Degrees-of-freedom method: containment
#> Confidence level used: 0.95
#>
#> $contrasts
#>  contrast            estimate    SE df t.ratio p.value
#>  Station1 - Station2    2.016 0.725 33   2.780  0.0421
#>  Station1 - Station3    1.689 0.725 33   2.329  0.1120
#>  Station1 - Station4    0.330 0.725 33   0.455  0.9681
#>  Station2 - Station3   -0.327 0.725 33  -0.451  0.9690
#>  Station2 - Station4   -1.686 0.725 33  -2.325  0.1129
#>  Station3 - Station4   -1.359 0.725 33  -1.874  0.2587
#>
#> Results are averaged over the levels of: Season
#> Degrees-of-freedom method: containment
#> P value adjustment: tukey method for comparing a family of 4 estimates
plot(emm_stat)
```




```
(emm_seas<- emmeans(test, pairwise~ Season))
#> $emmeans
#>  Season emmean   SE df lower.CL upper.CL
#>  Spring   95.4 2.53  3     87.3    103.4
#>  Summer   90.2 2.53  3     82.1     98.2
#>  Fall     83.3 2.53  3     75.2     91.4
#>
#> Results are averaged over the levels of: Station
#> Degrees-of-freedom method: containment
#> Confidence level used: 0.95
#>
#> $contrasts
#>  contrast        estimate  SE df t.ratio p.value
#>  Spring - Summer     5.22 2.1  6   2.485  0.1040
```

```
#>  Spring - Fall      12.07 2.1  6   5.749  0.0029
#>  Summer - Fall       6.85 2.1  6   3.264  0.0394
#>
#> Results are averaged over the levels of: Station
#> Degrees-of-freedom method: containment
#> P value adjustment: tukey method for comparing a family of 3 estimates
plot(emm_seas)
```



Differences by station are significant, but small, with the only meaningful pairwise comparison comparing Station 1 different from Station 2. Seasonal patterns are easier to interpret, with lower DO Saturation in the fall.

Discussion

Most of the environmental variables show patterns that can be readily explained in terms of estuarine processes, especially circulation and seasonal input of freshwater into the upper estuary.

The three of the five environmental variables – Temperature, Salinity and Chlorophyll show an important similar pattern: The spring is different from the other two seasons and Station one is different from the other three stations. This presumably reflects hydrodynamics and mixing processes.

Turbidity shows higher values at the two upstream stations, presumably because those stations are associated with the location of the turbidity maximum in this estuary. Dissolved oxygen saturation declines over the course of the year.

Total Zooplankton Density

Summary and Anova

```
density_gam <- gamm(log(combined_density) ~
                          Station +
                          Season +
                          s(Temp, bs="ts") +
                          s(Sal, bs="ts") +
                          s(log(Turb), bs="ts") +
                          s(log(Chl), bs="ts") +
                          s(log1p(RH),bs="ts"),
```

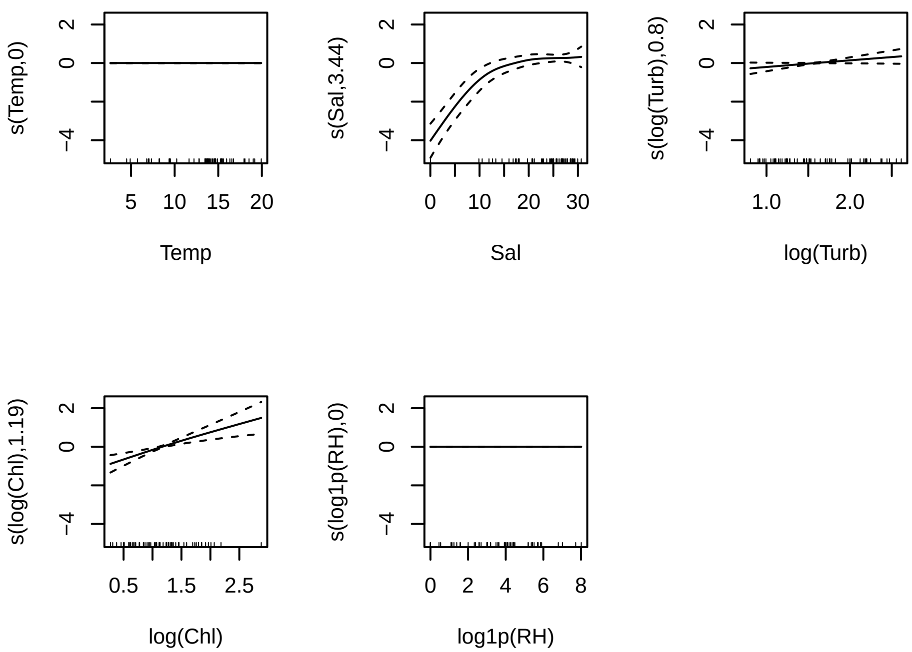
```
                        random = list(Yearf = ~ 1, sample_event = ~ 1),
                        data = base_data, family = 'gaussian')
summary(density_gam$gam)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log(combined_density) ~ Station + Season + s(Temp, bs = "ts") +
#>     s(Sal, bs = "ts") + s(log(Turb), bs = "ts") + s(log(Chl),
#>     bs = "ts") + s(log1p(RH), bs = "ts")
#>
#> Parametric coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    9.3298     0.4471  20.869  < 2e-16 ***
#> Station2      -1.0127     0.2760  -3.669 0.000624 ***
#> Station3      -0.7621     0.2627  -2.900 0.005672 **
#> Station4      -1.1834     0.2943  -4.020 0.000211 ***
#> SeasonSummer  -0.8743     0.3377  -2.589 0.012798 *
#> SeasonFall    -0.7889     0.3203  -2.463 0.017533 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>                    edf Ref.df      F  p-value
#> s(Temp)      3.688e-05      9  0.000 0.112923
#> s(Sal)       3.437e+00      9 12.044  < 2e-16 ***
#> s(log(Turb)) 8.029e-01      9  0.420 0.049332 *
#> s(log(Chl))  1.186e+00      9  2.021 0.000268 ***
#> s(log1p(RH)) 1.269e-05      9  0.000 0.800262
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.258
#>   Scale est. = 0.17578   n = 58
```

```
anova(density_gam$gam)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log(combined_density) ~ Station + Season + s(Temp, bs = "ts") +
#>     s(Sal, bs = "ts") + s(log(Turb), bs = "ts") + s(log(Chl),
#>     bs = "ts") + s(log1p(RH), bs = "ts")
#>
#> Parametric Terms:
#>         df     F p-value
#> Station  3 6.020 0.00149
#> Season   2 3.802 0.02955
#>
#> Approximate significance of smooth terms:
#>                    edf    Ref.df      F  p-value
```

```
#> s(Temp)       3.688e-05 9.000e+00  0.000 0.112923
#> s(Sal)        3.437e+00 9.000e+00 12.044  < 2e-16
#> s(log(Turb)) 8.029e-01 9.000e+00  0.420 0.049332
#> s(log(Chl))  1.186e+00 9.000e+00  2.021 0.000268
#> s(log1p(RH)) 1.269e-05 9.000e+00  0.000 0.800262
```

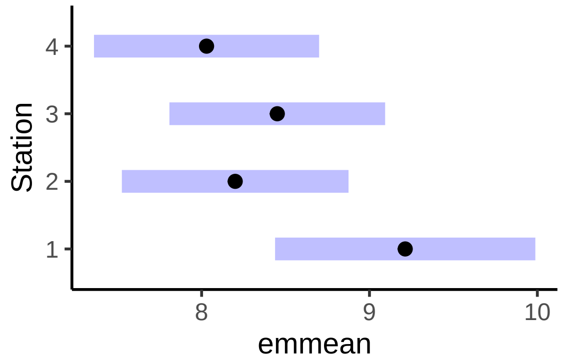
Plot the GAM

```
oldpar <- par(mfrow = c(2,3))
plot(density_gam$gam)
par(oldpar)
```



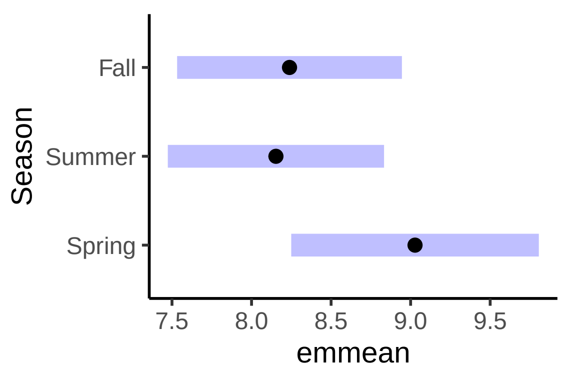
Station and Season

```
Sta_emms <- emmeans(density_gam, ~Station, type = 'response',
                    data = base_data)
plot(Sta_emms)
```




```
pairs(Sta_emms, adjust ='bonferroni')
#>  contrast            estimate    SE   df t.ratio p.value
#>  Station1 - Station2    1.013 0.276 46.6   3.669  0.0037
#>  Station1 - Station3    0.762 0.263 46.6   2.900  0.0340
#>  Station1 - Station4    1.183 0.294 46.6   4.020  0.0013
#>  Station2 - Station3   -0.251 0.188 46.6  -1.331  1.0000
#>  Station2 - Station4    0.171 0.201 46.6   0.851  1.0000
#>  Station3 - Station4    0.421 0.179 46.6   2.357  0.1362
#>
#> Results are averaged over the levels of: Season
#> P value adjustment: bonferroni method for 6 tests
```

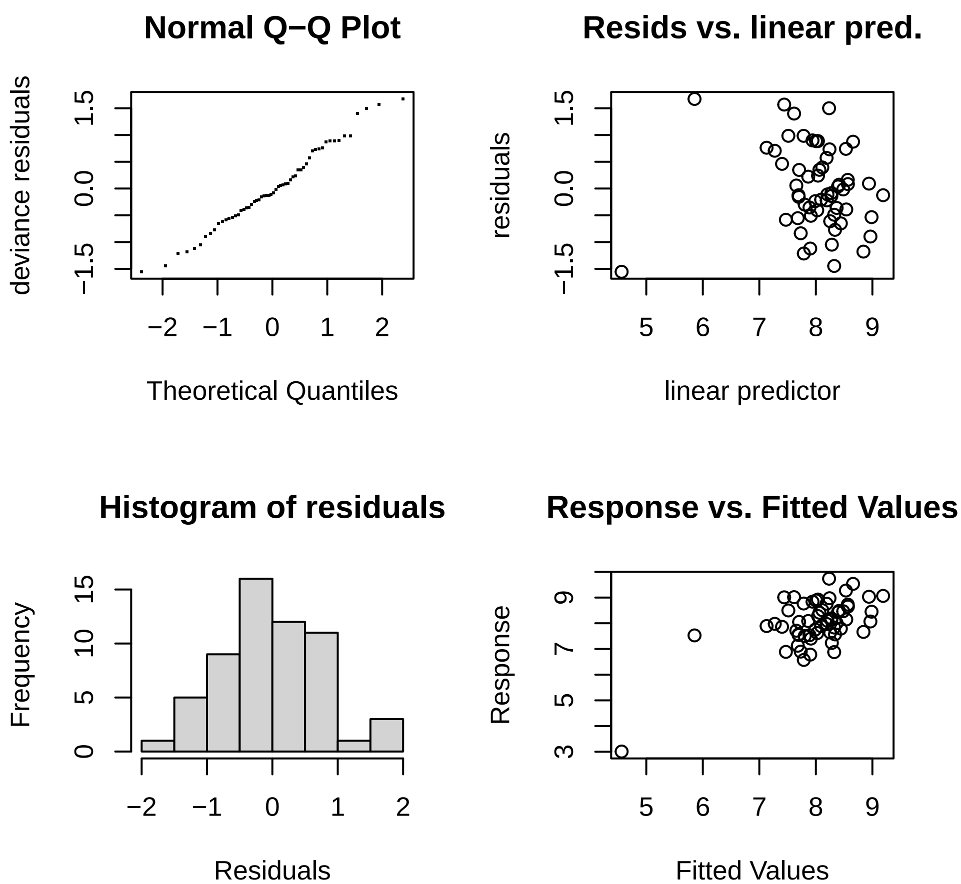
```
Seas_emms <- emmeans(density_gam, ~Season, type = 'response',
                     data = base_data)
plot(Seas_emms)
```




```
pairs(Seas_emms, adjust ='bonferroni')
#>  contrast        estimate    SE   df t.ratio p.value
#>  Spring - Summer   0.8743 0.338 46.6   2.589  0.0384
#>  Spring - Fall     0.7889 0.320 46.6   2.463  0.0526
#>  Summer - Fall    -0.0854 0.262 46.6  -0.326  1.0000
#>
#> Results are averaged over the levels of: Station
#> P value adjustment: bonferroni method for 3 tests
```

Model Diagnostics

```
oldpar <- par(mfrow = c(2,2))
gam.check(density_gam$gam)
```




```
#>
#> 'gamm' based fit - care required with interpretation.
#> Checks based on working residuals may be misleading.
#> 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(Temp)       9.00e+00 3.69e-05    1.07    0.65
#> s(Sal)        9.00e+00 3.44e+00    1.03    0.50
#> s(log(Turb))  9.00e+00 8.03e-01    0.89    0.16
#> s(log(Chl))   9.00e+00 1.19e+00    0.94    0.24
#> s(log1p(RH))  9.00e+00 1.27e-05    0.88    0.15
par(oldpar)
```

One low value is a serious outlier – it corresponds to one of the spring "washout" events. Those "washout" events have a large impact on model fit, especially the substantial non-linearity in the Salinity response.

Discussion

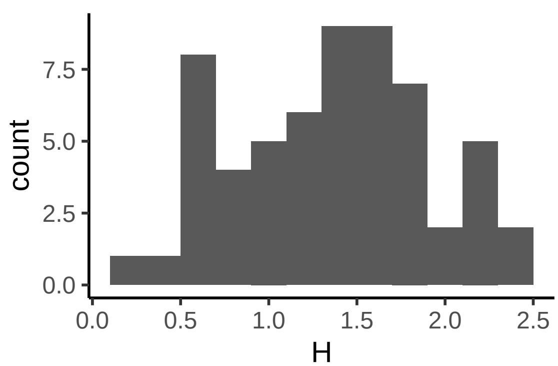
- The Station differences are significant by ANOVA F test. Pairwise comparisons show that Station 1 (upstream) shows the highest combined density, which is significantly higher than for Stations 2 and 4, but not different from Station 3 (by multiple comparisons test anyway). There are no meaningful differences among the three downstream Stations.
- While zooplankton density varies by season, only the Spring-Summer pairwise comparisons of marginal means is individually significant. Densities are somewhat higher in the spring than later in the year.
- Salinity Shows a highly significant curved (~ 3 edf) pattern, driven largely by a couple of very low salinity, low density samples from Station 1 in the Spring.
- Turbidity and Chlorophyll both fit close to linear (~ 1 edf) relationships that appear fairly robust to model specification. Zooplankton abundance is correlated with higher chlorophyll and higher turbidity. (it's not unreasonable to test for a significant interaction there, but I have not done so.)

Shannon Diversity

Histogram

To decide whether we can proceed with analysis of untransformed values.

```
base_data %>%
  ggplot(aes(x = H))+
  geom_histogram(binwidth = 0.2)
#> Warning: Removed 1 rows containing non-finite values (stat_bin).
```



Gaussian GAM with an Identity Link

Summary and Anova

```
shannon_gam <- gam(H ~ Station +
                     Season +
                     s(Temp, bs="ts") +
                     s(Sal, bs="ts") +
                     s(log(Turb), bs="ts") +
```

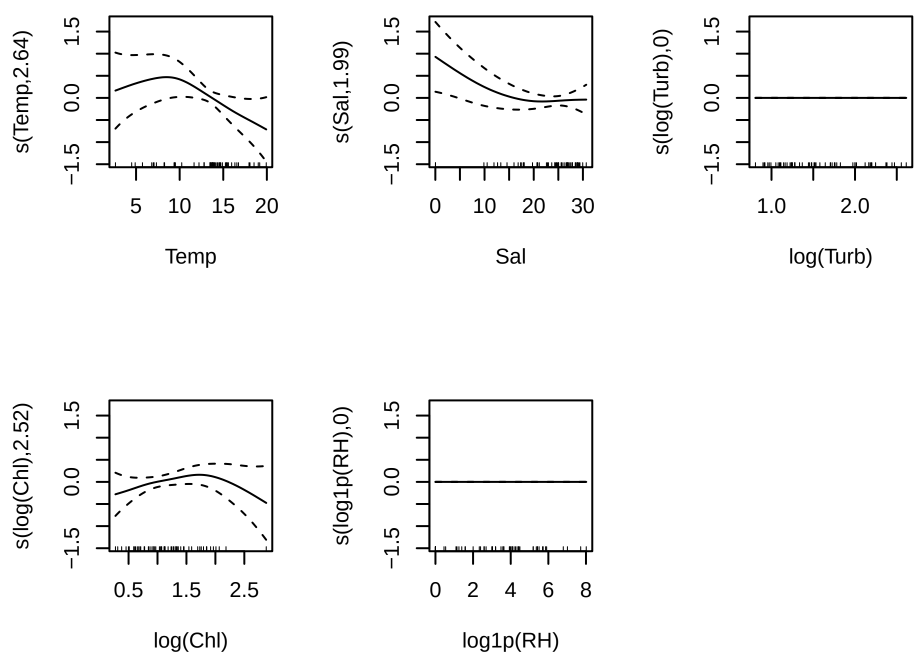
```r
                     s(log(Chl), bs="ts") +
                     s(log1p(RH),bs="ts"),
                   random = list(Yearf = ~ 1, sample_event = ~ 1),
                   data = base_data, family = 'gaussian')
summary(shannon_gam)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> H ~ Station + Season + s(Temp, bs = "ts") + s(Sal, bs = "ts") +
#>     s(log(Turb), bs = "ts") + s(log(Chl), bs = "ts") + s(log1p(RH),
#>     bs = "ts")
#>
#> Parametric coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    0.8738     0.3338   2.618    0.012 *
#> Station2       0.2543     0.2935   0.866    0.391
#> Station3       0.3433     0.2698   1.272    0.210
#> Station4       0.2077     0.2865   0.725    0.472
#> SeasonSummer   0.4842     0.4026   1.203    0.235
#> SeasonFall     0.3203     0.3957   0.810    0.422
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>                    edf Ref.df     F p-value
#> s(Temp)      2.642e+00      9 0.654  0.0753 .
#> s(Sal)       1.994e+00      9 0.757  0.0261 *
#> s(log(Turb)) 5.537e-08      9 0.000  0.5331
#> s(log(Chl))  2.517e+00      9 0.510  0.1321
#> s(log1p(RH)) 3.173e-08      9 0.000  0.3568
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.319   Deviance explained = 46.4%
#> GCV = 0.26465  Scale est. = 0.20463   n = 58
```

```r
anova(shannon_gam)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> H ~ Station + Season + s(Temp, bs = "ts") + s(Sal, bs = "ts") +
#>     s(log(Turb), bs = "ts") + s(log(Chl), bs = "ts") + s(log1p(RH),
#>     bs = "ts")
#>
#> Parametric Terms:
#>         df     F p-value
#> Station  3 0.622   0.605
#> Season   2 1.010   0.372
#>
```

```
#> Approximate significance of smooth terms:
#>                     edf    Ref.df     F p-value
#> s(Temp)       2.642e+00 9.000e+00 0.654  0.0753
#> s(Sal)        1.994e+00 9.000e+00 0.757  0.0261
#> s(log(Turb)) 5.537e-08 9.000e+00 0.000  0.5331
#> s(log(Chl))  2.517e+00 9.000e+00 0.510  0.1321
#> s(log1p(RH)) 3.173e-08 9.000e+00 0.000  0.3568
```

Plot the GAM

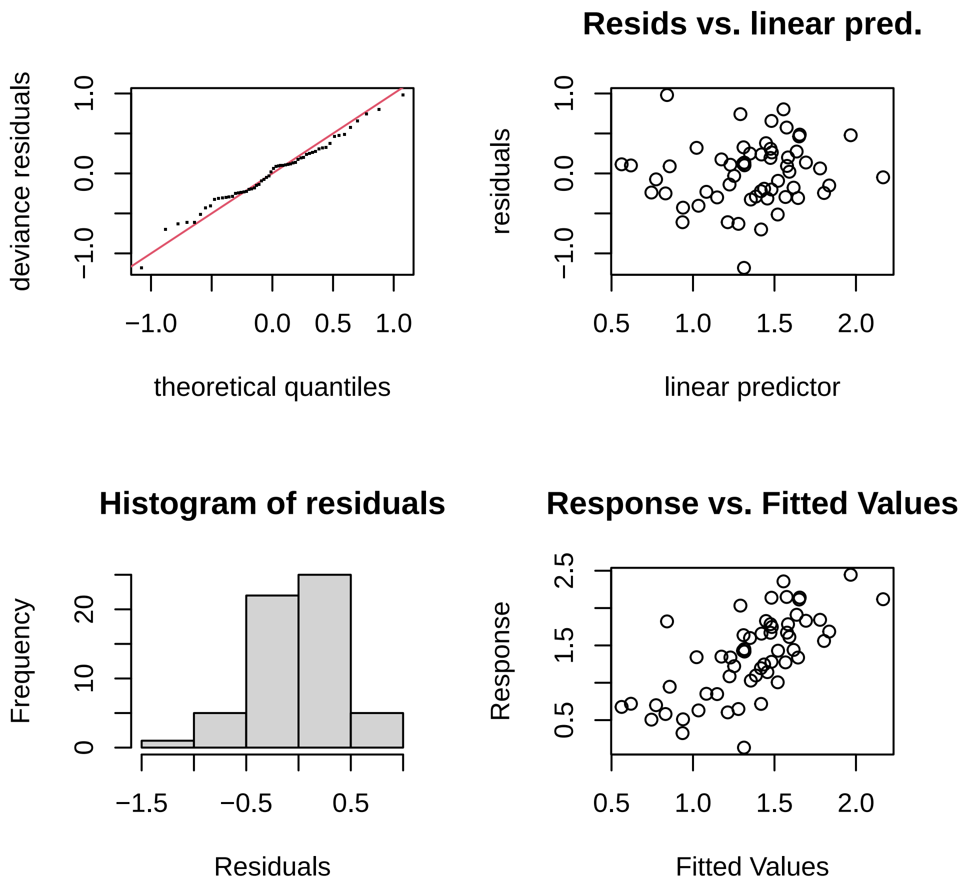
```
oldpar <- par(mfrow = c(2,3))
plot(shannon_gam)
par(oldpar)
```



While the GAMM fits curves for several predictors, only the relationship with salinity is retained in the model as statistically significant. It appears much of that pattern is driven by a couple of low salinity samples. This model is rather unsatisfying, as it finds no robust significant relationships. We may want to evaluate similar models with fewer parameters to see if we can extract more wisdom.

Diagnostic Plots

```
oldpar <- par(mfrow = c(2,2))
gam.check(shannon_gam)
```




```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 19 iterations.
#> The RMS GCV score gradient at convergence was 5.565746e-08 .
#> The Hessian was positive definite.
#> Model rank =  51 / 51
#>
#> 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(Temp)      9.00e+00 2.64e+00    1.12    0.82
#> s(Sal)       9.00e+00 1.99e+00    0.96    0.35
#> s(log(Turb)) 9.00e+00 5.54e-08    1.30    0.98
#> s(log(Chl))  9.00e+00 2.52e+00    1.13    0.80
#> s(log1p(RH)) 9.00e+00 3.17e-08    1.06    0.69
par(oldpar)
```

Not a bad model from a diagnostics point of view. If only we had learned something useful from it.

Model of River Herring Abundance

Summary and ANOVA

```
herring_gam <- gam(log1p(RH) ~ Station +
                     Season +
                     s(Temp, bs="ts") +
                     s(Sal, bs="ts") +
                     s(log(Turb), bs="ts") +
                     s(log(Chl), bs="ts") +
                     s(log1p(combined_density),bs="ts"),
                   random = list(Yearf = ~ 1, sample_event = ~ 1),
                   data = base_data, family = 'gaussian')
summary(herring_gam)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(RH) ~ Station + Season + s(Temp, bs = "ts") + s(Sal, bs = "ts") +
#>     s(log(Turb), bs = "ts") + s(log(Chl), bs = "ts") + s(log1p(combined_density),
#>     bs = "ts")
#>
#> Parametric coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)     6.470      1.338   4.837 2.16e-05 ***
#> Station2       -1.946      1.140  -1.708   0.0957 .
#> Station3       -2.316      1.054  -2.197   0.0341 *
#> Station4       -2.311      1.103  -2.096   0.0427 *
#> SeasonSummer   -1.682      1.667  -1.009   0.3192
#> SeasonFall     -2.630      1.598  -1.646   0.1080
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>                               edf Ref.df     F p-value
#> s(Temp)                     1.576      9 0.261   0.197
#> s(Sal)                      3.910      9 0.775   0.112
#> s(log(Turb))                0.869      9 0.404   0.039 *
#> s(log(Chl))                 2.486      9 0.292   0.347
#> s(log1p(combined_density)) 4.745      9 0.636   0.282
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.381   Deviance explained = 58.3%
#> GCV = 4.2624  Scale est. = 2.823     n = 58
```

```
anova(herring_gam)
#>
#> Family: gaussian
#> Link function: identity
#>
```

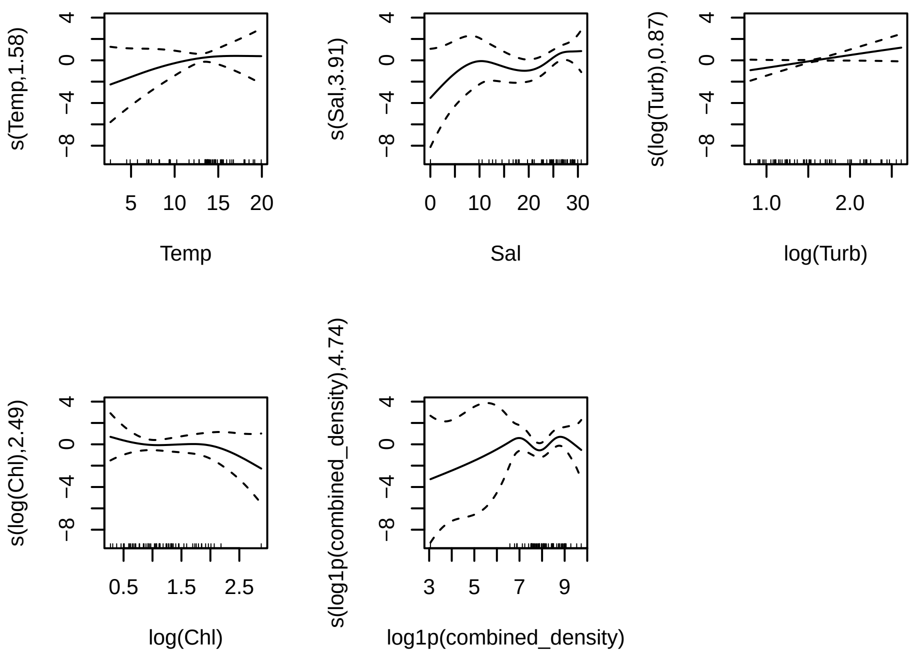
```
#> Formula:
#> log1p(RH) ~ Station + Season + s(Temp, bs = "ts") + s(Sal, bs = "ts") +
#>     s(log(Turb), bs = "ts") + s(log(Chl), bs = "ts") + s(log1p(combined_density),
#>     bs = "ts")
#>
#> Parametric Terms:
#>         df     F p-value
#> Station  3 1.737   0.176
#> Season   2 2.362   0.108
#>
#> Approximate significance of smooth terms:
#>                               edf Ref.df     F p-value
#> s(Temp)                     1.576  9.000 0.261   0.197
#> s(Sal)                      3.910  9.000 0.775   0.112
#> s(log(Turb))                0.869  9.000 0.404   0.039
#> s(log(Chl))                 2.486  9.000 0.292   0.347
#> s(log1p(combined_density)) 4.745  9.000 0.636   0.282
```

Note that overall, Station is NOT statistically significant by F test, although individual parameters ARE significant by T test. The usual statistical advice is that to avoid making claims on weak evidence, one should not interpret individual parameters within a factor unless the overall comparison is significant. Here the issue is that the Spring sample is the base case in the parameter table, and Spring is moderately different from all other seasons.

While the GAMM fits curves for several predictors, only the relationship with Turbidity in the model is statistically significant. The relationship is essentially linear (EDF = 0.87).

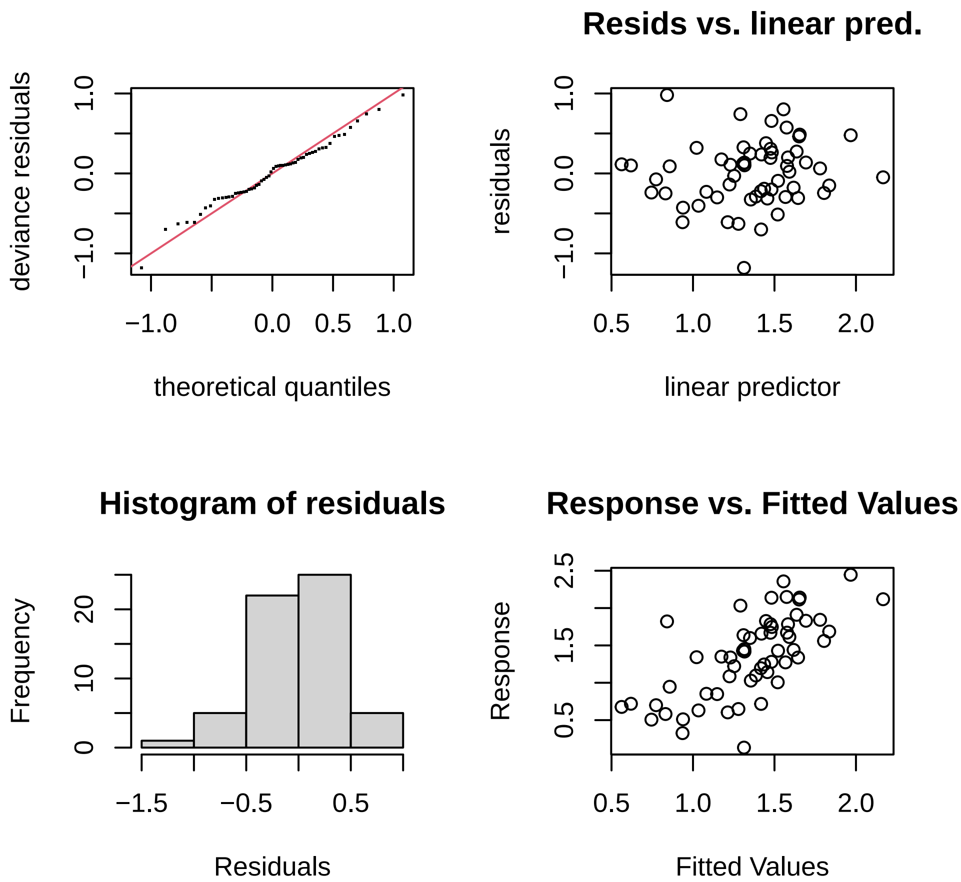
Plot GAM results

```
oldpar <- par(mfrow = c(2,3))
plot(herring_gam)
par(oldpar)
```



Model Diagnostics

```
oldpar <- par(mfrow = c(2,2))
gam.check(shannon_gam)
```




```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 19 iterations.
#> The RMS GCV score gradient at convergence was 5.565746e-08 .
#> The Hessian was positive definite.
#> Model rank =  51 / 51
#>
#> 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(Temp)       9.00e+00 2.64e+00    1.12    0.76
#> s(Sal)        9.00e+00 1.99e+00    0.96    0.37
#> s(log(Turb)) 9.00e+00 5.54e-08    1.30    0.99
#> s(log(Chl))  9.00e+00 2.52e+00    1.13    0.79
#> s(log1p(RH)) 9.00e+00 3.17e-08    1.06    0.60
par(oldpar)
```

The model is pretty good, with only slightly skewed residuals.

Single Species Models

Model Choice

Our model alternatives are similar to the choices we had for the Total Density model. The problem is, we can't use any of the continuous data distributions in GAMS with zero values (at least relying on the canonical link functions) because (log(0) = -Inf; 1/0 = Inf, 1 / 0*0 = Inf). The easiest solution is to add some finite small quantity to the density data, and predict that. Here we predict log(Density + 1) using Gaussian models.

Automating Analysis of Separate Species

I'm going to automate analysis of all selected species by using a "nested" Tibble. This is a convenient alternative to writing a "for" loop to run multiple identical analyses.

I create a "long" data source.

```
spp_data <- base_data %>%
  select(Yearf, Month, Season, sample_event, Station, Temp,
          Sal, Turb, Chl, RH,
          Acartia, Balanus, Eurytemora, Polychaete, Pseudocal, Temora) %>%
  pivot_longer(-c(Yearf:RH), names_to = 'Species', values_to = 'Density')
```

Next, I create a function to run the analysis. This function takes a data frame or tibble as an argument. The tibble mush have data columns with the correct names.

The initial model fits for some species had a lot of wiggles in them, to an extent that I thought did not make much scientific sense, so I decided to reduce the dimensionality of the GAM smoothers, by adding the parameter `k= 4`. Lowe numbers constrain the GAM to fit smoother lines.

```
my_gamm <- function(.dat) {

  gam(log1p(Density) ~ Station +
        Season +
        s(Temp, bs="ts", k = 4) +
        s(Sal, bs="ts", k = 4) +
        s(log(Turb), bs="ts", k = 4) +
        s(log(Chl), bs="ts", k = 4) +
        s(log1p(RH),bs="ts", k = 4),
      random = list(Yearf = ~ 1, sample_event = ~ 1),
      data = .dat, family = "gaussian")
}
```

Next, I create the nested tibble, and conduct the analysis on each species. . . .

```
spp_analysis <- spp_data %>%
  group_by(Species) %>%
  nest() %>%
  mutate(gam_mods = map(data, my_gamm))
```

and finally, output the model results. I can do that in a "for" loop, but it's Awkward to look through a long list of output, so I step through each species in turn.

Acartia

Summary and ANOVA

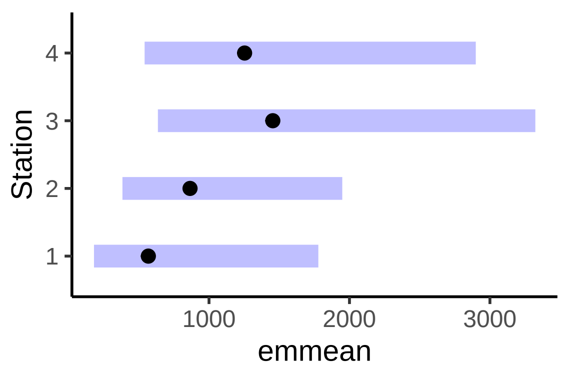
```
spp = 'Acartia'
mod <- spp_analysis$gam_mods[spp_analysis$Species == spp][[1]]
summary(mod)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 4) +
#>     s(Sal, bs = "ts", k = 4) + s(log(Turb), bs = "ts", k = 4) +
#>     s(log(Chl), bs = "ts", k = 4) + s(log1p(RH), bs = "ts", k = 4)
#>
#> Parametric coefficients:
#>               Estimate Std. Error t value Pr(>|t|)
#> (Intercept)     4.2233     0.6754   6.253  1.3e-07 ***
#> Station2        0.4199     0.5900   0.712   0.4804
#> Station3        0.9390     0.5566   1.687   0.0985 .
#> Station4        0.7905     0.6061   1.304   0.1988
#> SeasonSummer    2.3134     1.0156   2.278   0.0275 *
#> SeasonFall      2.6436     0.9928   2.663   0.0107 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>                    edf Ref.df     F p-value
#> s(Temp)      2.864e+00      3 4.239 0.00754 **
#> s(Sal)       1.424e-10      3 0.000 0.38719
#> s(log(Turb)) 6.140e-01      3 0.663 0.07251 .
#> s(log(Chl))  2.737e+00      3 3.844 0.01026 *
#> s(log1p(RH)) 6.360e-01      3 0.633 0.08469 .
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =   0.62   Deviance explained = 69.9%
#> GCV = 1.6621  Scale est. = 1.2938    n = 58
cat('\n')
anova(mod)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 4) +
#>     s(Sal, bs = "ts", k = 4) + s(log(Turb), bs = "ts", k = 4) +
#>     s(log(Chl), bs = "ts", k = 4) + s(log1p(RH), bs = "ts", k = 4)
#>
#> Parametric Terms:
#>         df     F p-value
#> Station  3 1.158   0.336
```

```
#> Season    2 3.647   0.034
#>
#> Approximate significance of smooth terms:
#>                     edf    Ref.df     F p-value
#> s(Temp)       2.864e+00 3.000e+00 4.239 0.00754
#> s(Sal)        1.424e-10 3.000e+00 0.000 0.38719
#> s(log(Turb)) 6.140e-01 3.000e+00 0.663 0.07251
#> s(log(Chl))  2.737e+00 3.000e+00 3.844 0.01026
#> s(log1p(RH)) 6.360e-01 3.000e+00 0.633 0.08469
```

Comparison of Station and Season

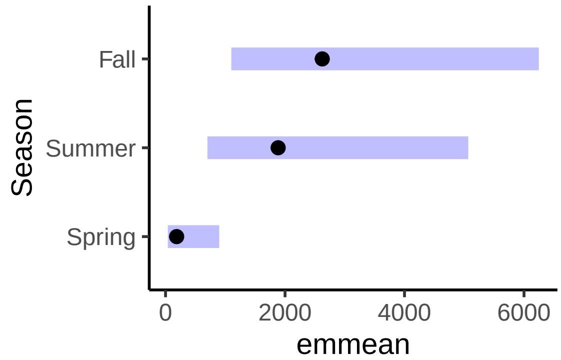
I'm showing "marginal" means – essentially means adjusted for the other predictors, at their mean values.

```
Sta_emms <- emmeans(mod, ~Station, type = 'response',
                    data = spp_analysis$data[spp_data$Species == spp][[1]])
plot(Sta_emms)
```




```
pairs(Sta_emms, adjust ='bonferroni')
#> Note: Use 'contrast(regrid(object), ...)' to obtain contrasts of back-transformed estimates
#>  contrast            estimate    SE   df t.ratio p.value
#>  Station1 - Station2   -0.420 0.590 45.1  -0.712  1.0000
#>  Station1 - Station3   -0.939 0.557 45.1  -1.687  0.5911
#>  Station1 - Station4   -0.790 0.606 45.1  -1.304  1.0000
#>  Station2 - Station3   -0.519 0.445 45.1  -1.167  1.0000
#>  Station2 - Station4   -0.371 0.465 45.1  -0.797  1.0000
#>  Station3 - Station4    0.148 0.443 45.1   0.335  1.0000
#>
#> Results are averaged over the levels of: Season
#> Note: contrasts are still on the log1p scale
#> P value adjustment: bonferroni method for 6 tests
```

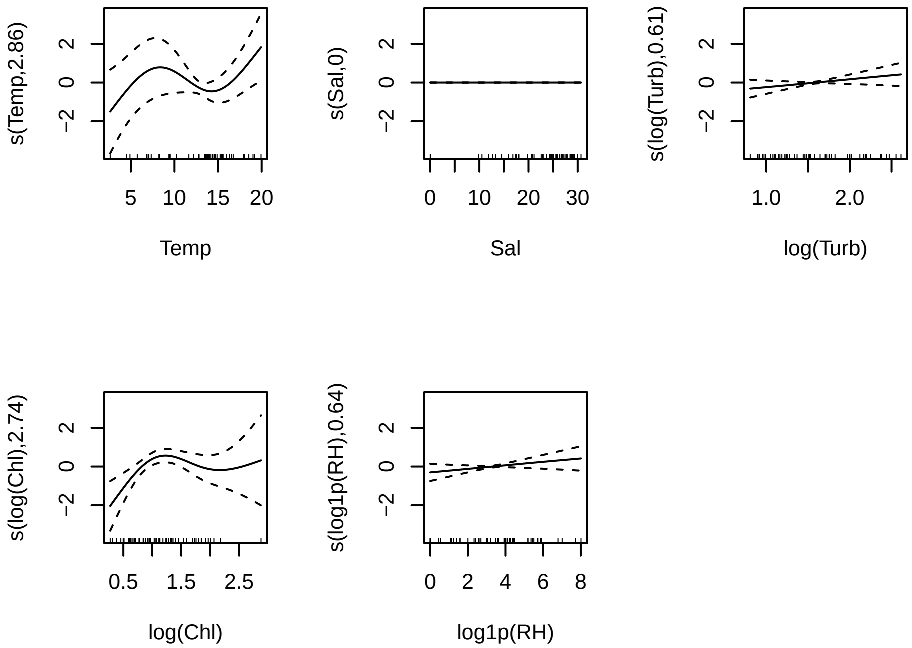
```
Seas_emms <- emmeans(mod, ~Season, type = 'response',
                     data = spp_analysis$data[spp_data$Species == spp][[1]])
plot(Seas_emms)
```




```
pairs(Seas_emms, adjust ='bonferroni')
#> Note: Use 'contrast(regrid(object), ...)' to obtain contrasts of back-transformed estimates
#>  contrast        estimate    SE   df t.ratio p.value
#>  Spring - Summer    -2.31 1.016 45.1  -2.278  0.0826
#>  Spring - Fall      -2.64 0.993 45.1  -2.663  0.0321
#>  Summer - Fall      -0.33 0.399 45.1  -0.827  1.0000
#>
#> Results are averaged over the levels of: Station
#> Note: contrasts are still on the log1p scale
#> P value adjustment: bonferroni method for 3 tests
```

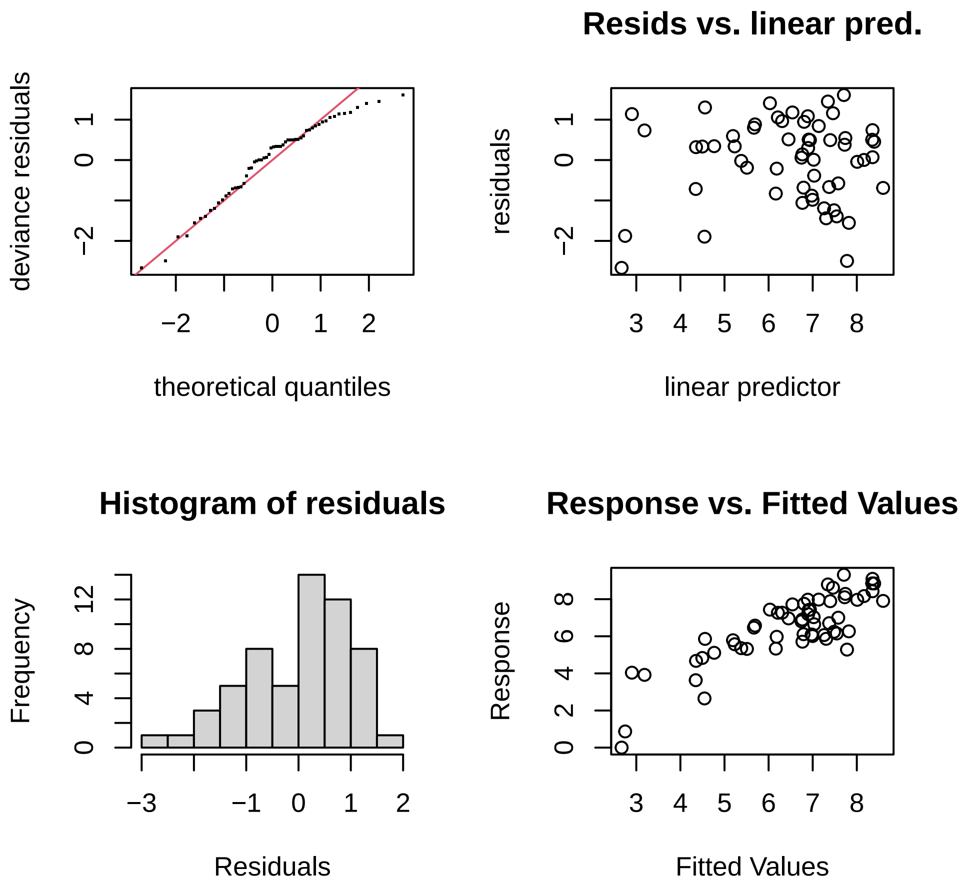
Plot GAM

```
oldpar <- par(mfrow = c(2,3))
plot(mod)
par(oldpar)
```



Model Diagnostics

```
oldpar <- par(mfrow = c(2,2))
gam.check(mod)
```




```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 24 iterations.
#> The RMS GCV score gradient at convergence was 1.066668e-07 .
#> The Hessian was positive definite.
#> Model rank =  21 / 21
#>
#> 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(Temp)      3.00e+00 2.86e+00    1.01    0.49
#> s(Sal)       3.00e+00 1.42e-10    1.20    0.90
#> s(log(Turb)) 3.00e+00 6.14e-01    1.17    0.88
#> s(log(Chl))  3.00e+00 2.74e+00    0.79    0.06 .
#> s(log1p(RH)) 3.00e+00 6.36e-01    0.98    0.39
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
par(oldpar)
```

Balanus

Summary and ANOVA

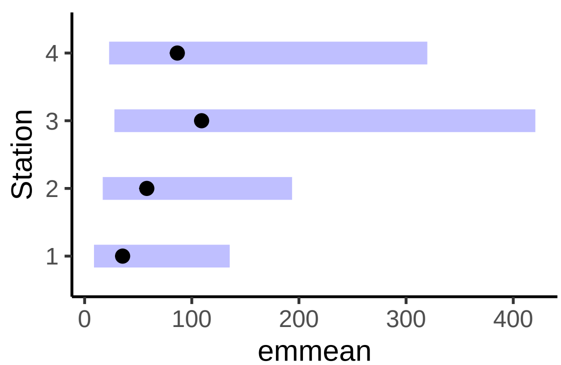
```
spp = 'Balanus'
mod <- spp_analysis$gam_mods[spp_analysis$Species == spp][[1]]
summary(mod)
#> 
#> Family: gaussian 
#> Link function: identity 
#> 
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 4) + 
#>     s(Sal, bs = "ts", k = 4) + s(log(Turb), bs = "ts", k = 4) + 
#>     s(log(Chl), bs = "ts", k = 4) + s(log1p(RH), bs = "ts", k = 4)
#> 
#> Parametric coefficients:
#>               Estimate Std. Error t value Pr(>|t|)    
#> (Intercept)    4.1730     0.8044   5.188 4.23e-06 ***
#> Station2       0.4820     0.7248   0.665  0.50926    
#> Station3       1.1060     0.7381   1.498  0.14057    
#> Station4       0.8757     0.7866   1.113  0.27115    
#> SeasonSummer  -1.3842     0.8070  -1.715  0.09273 .  
#> SeasonFall    -2.0889     0.7492  -2.788  0.00757 ** 
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#> 
#> Approximate significance of smooth terms:
#>                    edf Ref.df     F  p-value    
#> s(Temp)      2.581e-11      3 0.000    0.842    
#> s(Sal)       4.858e-11      3 0.000    0.632    
#> s(log(Turb)) 1.046e-11      3 0.000    0.689    
#> s(log(Chl))  1.794e+00      3 6.741 8.18e-05 ***
#> s(log1p(RH)) 2.159e+00      3 0.926    0.278    
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#> 
#> R-sq.(adj) =  0.348   Deviance explained =   45%
#> GCV =  4.077  Scale est. = 3.3774    n = 58
cat('\n')
anova(mod)
#> 
#> Family: gaussian 
#> Link function: identity 
#> 
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 4) + 
#>     s(Sal, bs = "ts", k = 4) + s(log(Turb), bs = "ts", k = 4) + 
#>     s(log(Chl), bs = "ts", k = 4) + s(log1p(RH), bs = "ts", k = 4)
#> 
#> Parametric Terms:
#>         df     F p-value
#> Station  3 0.817  0.4910
```

```
#> Season   2 3.936  0.0261
#>
#> Approximate significance of smooth terms:
#>                     edf    Ref.df     F  p-value
#> s(Temp)       2.581e-11 3.000e+00 0.000    0.842
#> s(Sal)        4.858e-11 3.000e+00 0.000    0.632
#> s(log(Turb)) 1.046e-11 3.000e+00 0.000    0.689
#> s(log(Chl))  1.794e+00 3.000e+00 6.741 8.18e-05
#> s(log1p(RH)) 2.159e+00 3.000e+00 0.926    0.278
```

Comparison of Station and Season

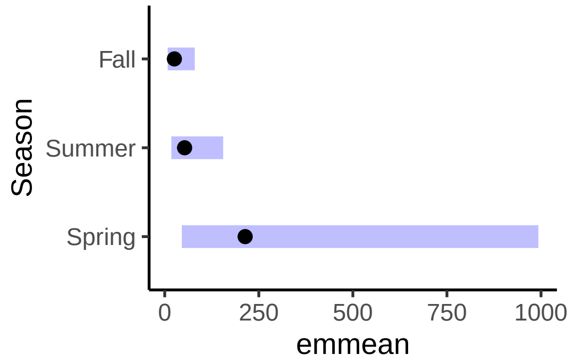
I'm showing "marginal" means – essentially means adjusted for the other predictors, at their mean values.

```
Sta_emms <- emmeans(mod, ~Station, type = 'response',
                    data = spp_analysis$data[spp_data$Species == spp][[1]])
plot(Sta_emms)
```




```
pairs(Sta_emms, adjust ='bonferroni')
#> Note: Use 'contrast(regrid(object), ...)' to obtain contrasts of back-transformed estimates
#>  contrast            estimate    SE df t.ratio p.value
#>  Station1 - Station2   -0.482 0.725 48  -0.665  1.0000
#>  Station1 - Station3   -1.106 0.738 48  -1.498  0.8434
#>  Station1 - Station4   -0.876 0.787 48  -1.113  1.0000
#>  Station2 - Station3   -0.624 0.695 48  -0.898  1.0000
#>  Station2 - Station4   -0.394 0.707 48  -0.557  1.0000
#>  Station3 - Station4    0.230 0.707 48   0.326  1.0000
#>
#> Results are averaged over the levels of: Season
#> Note: contrasts are still on the log1p scale
#> P value adjustment: bonferroni method for 6 tests
```

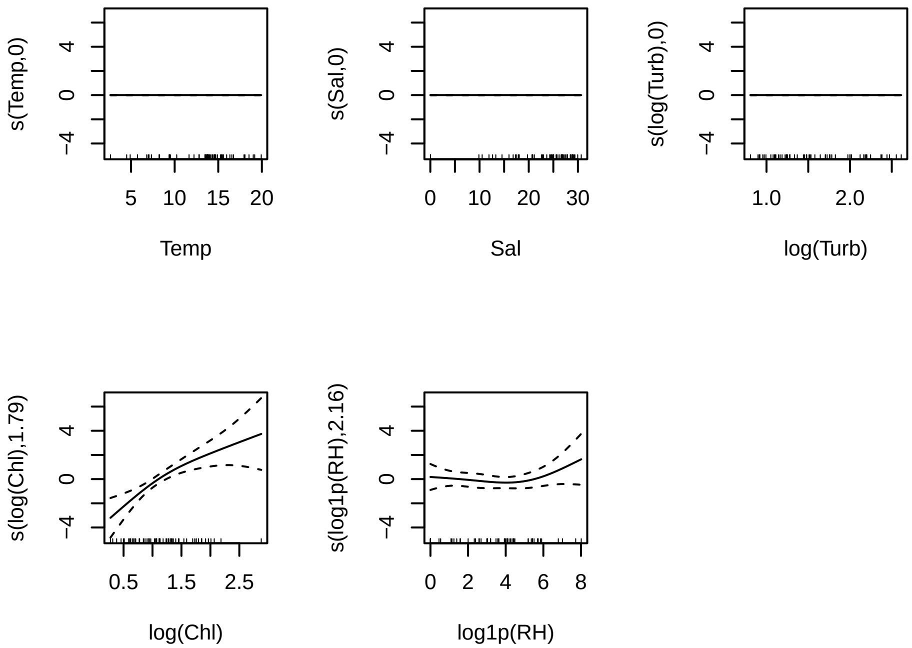
```
Seas_emms <- emmeans(mod, ~Season, type = 'response',
                     data = spp_analysis$data[spp_data$Species == spp][[1]])
plot(Seas_emms)
```




```
pairs(Seas_emms, adjust ='bonferroni')
#> Note: Use 'contrast(regrid(object), ...)' to obtain contrasts of back-transformed estimates
#>  contrast        estimate    SE df t.ratio p.value
#>  Spring - Summer    1.384 0.807 48   1.715  0.2782
#>  Spring - Fall      2.089 0.749 48   2.788  0.0227
#>  Summer - Fall      0.705 0.610 48   1.156  0.7600
#>
#> Results are averaged over the levels of: Station
#> Note: contrasts are still on the log1p scale
#> P value adjustment: bonferroni method for 3 tests
```

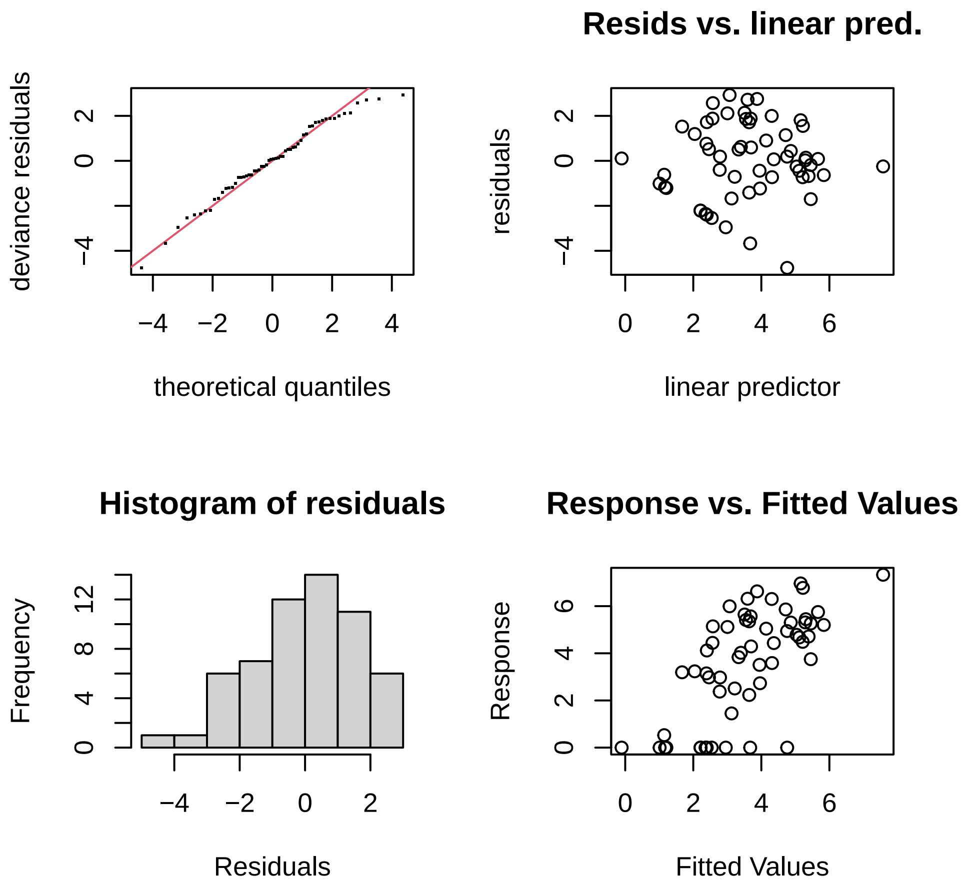
Plot GAM

```
oldpar <- par(mfrow = c(2,3))
plot(mod)
par(oldpar)
```



Model Diagnostics

```
oldpar <- par(mfrow = c(2,2))
gam.check(mod)
```




```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 20 iterations.
#> The RMS GCV score gradient at convergence was 1.94595e-07 .
#> The Hessian was positive definite.
#> Model rank =  21 / 21
#>
#> 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(Temp)       3.00e+00 2.58e-11    0.89    0.14
#> s(Sal)        3.00e+00 4.86e-11    0.90    0.14
#> s(log(Turb)) 3.00e+00 1.05e-11    1.04    0.55
#> s(log(Chl))  3.00e+00 1.79e+00    1.17    0.90
#> s(log1p(RH)) 3.00e+00 2.16e+00    1.15    0.77
par(oldpar)
```

Eurytemora

Summary and ANOVA

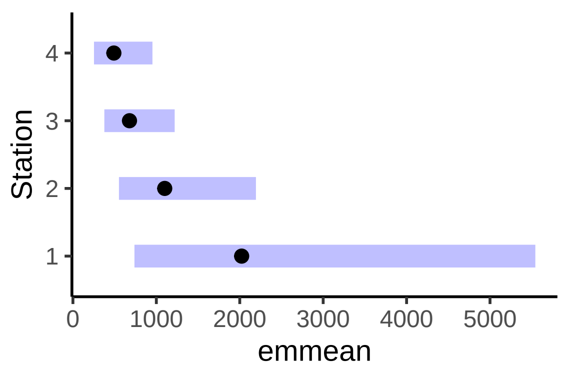
```r
spp =  "Eurytemora"
mod <- spp_analysis$gam_mods[spp_analysis$Species == spp][[1]]
summary(mod)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 4) +
#>     s(Sal, bs = "ts", k = 4) + s(log(Turb), bs = "ts", k = 4) +
#>     s(log(Chl), bs = "ts", k = 4) + s(log1p(RH), bs = "ts", k = 4)
#>
#> Parametric coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    8.3514     0.5513  15.149  < 2e-16 ***
#> Station2      -0.6082     0.5400  -1.126  0.26586
#> Station3      -1.0918     0.5006  -2.181  0.03432 *
#> Station4      -1.4144     0.5251  -2.694  0.00983 **
#> SeasonSummer  -1.6297     0.6682  -2.439  0.01864 *
#> SeasonFall    -1.7576     0.6663  -2.638  0.01134 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>                    edf Ref.df      F p-value
#> s(Temp)      2.122e+00      3  2.261 0.03666 *
#> s(Sal)       2.756e+00      3 15.226 < 2e-16 ***
#> s(log(Turb)) 9.291e-01      3  2.413 0.00556 **
#> s(log(Chl))  1.925e-01      3  0.084 0.25634
#> s(log1p(RH)) 1.243e-10      3  0.000 0.81723
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.522   Deviance explained = 61.4%
#> GCV = 0.92683  Scale est. = 0.73507   n = 58
cat('\n')
anova(mod)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 4) +
#>     s(Sal, bs = "ts", k = 4) + s(log(Turb), bs = "ts", k = 4) +
#>     s(log(Chl), bs = "ts", k = 4) + s(log1p(RH), bs = "ts", k = 4)
#>
#> Parametric Terms:
#>         df     F p-value
#> Station  3 3.422  0.0248
```

```
#> Season   2 3.485  0.0390
#>
#> Approximate significance of smooth terms:
#>                    edf    Ref.df      F p-value
#> s(Temp)      2.122e+00 3.000e+00  2.261 0.03666
#> s(Sal)       2.756e+00 3.000e+00 15.226 < 2e-16
#> s(log(Turb)) 9.291e-01 3.000e+00  2.413 0.00556
#> s(log(Chl))  1.925e-01 3.000e+00  0.084 0.25634
#> s(log1p(RH)) 1.243e-10 3.000e+00  0.000 0.81723
```

Comparison of Station and Season

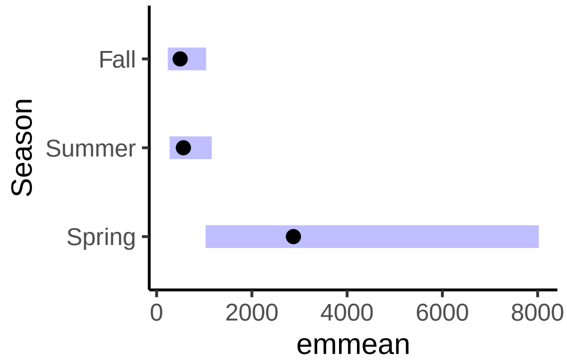
I'm showing "marginal" means – essentially means adjusted for the other predictors, at their mean values.

```
Sta_emms <- emmeans(mod, ~Station, type = 'response',
                    data = spp_analysis$data[spp_data$Species == spp][[1]])
plot(Sta_emms)
```




```
pairs(Sta_emms, adjust ='bonferroni')
#> Note: Use 'contrast(regrid(object), ...)' to obtain contrasts of back-transformed estimates
#>  contrast            estimate    SE df t.ratio p.value
#>  Station1 - Station2    0.608 0.540 46   1.126  1.0000
#>  Station1 - Station3    1.092 0.501 46   2.181  0.2059
#>  Station1 - Station4    1.414 0.525 46   2.694  0.0590
#>  Station2 - Station3    0.484 0.350 46   1.382  1.0000
#>  Station2 - Station4    0.806 0.354 46   2.280  0.1636
#>  Station3 - Station4    0.323 0.328 46   0.983  1.0000
#>
#> Results are averaged over the levels of: Season
#> Note: contrasts are still on the log1p scale
#> P value adjustment: bonferroni method for 6 tests
```

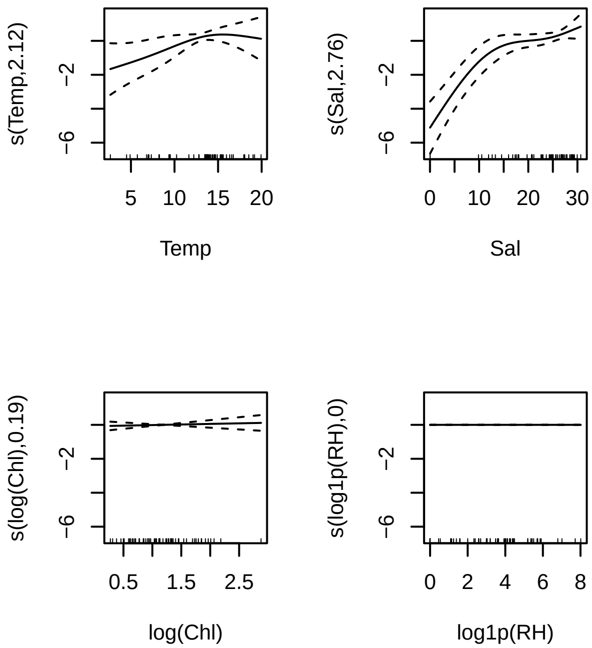
```
Seas_emms <- emmeans(mod, ~Season, type = 'response',
                     data = spp_analysis$data[spp_data$Species == spp][[1]])
plot(Seas_emms)
```




```
pairs(Seas_emms, adjust ='bonferroni')
#> Note: Use 'contrast(regrid(object), ...)' to obtain contrasts of back-transformed estimates
#>  contrast        estimate    SE df t.ratio p.value
#>  Spring - Summer    1.630 0.668 46   2.439  0.0559
#>  Spring - Fall      1.758 0.666 46   2.638  0.0340
#>  Summer - Fall      0.128 0.288 46   0.444  1.0000
#>
#> Results are averaged over the levels of: Station
#> Note: contrasts are still on the log1p scale
#> P value adjustment: bonferroni method for 3 tests
```

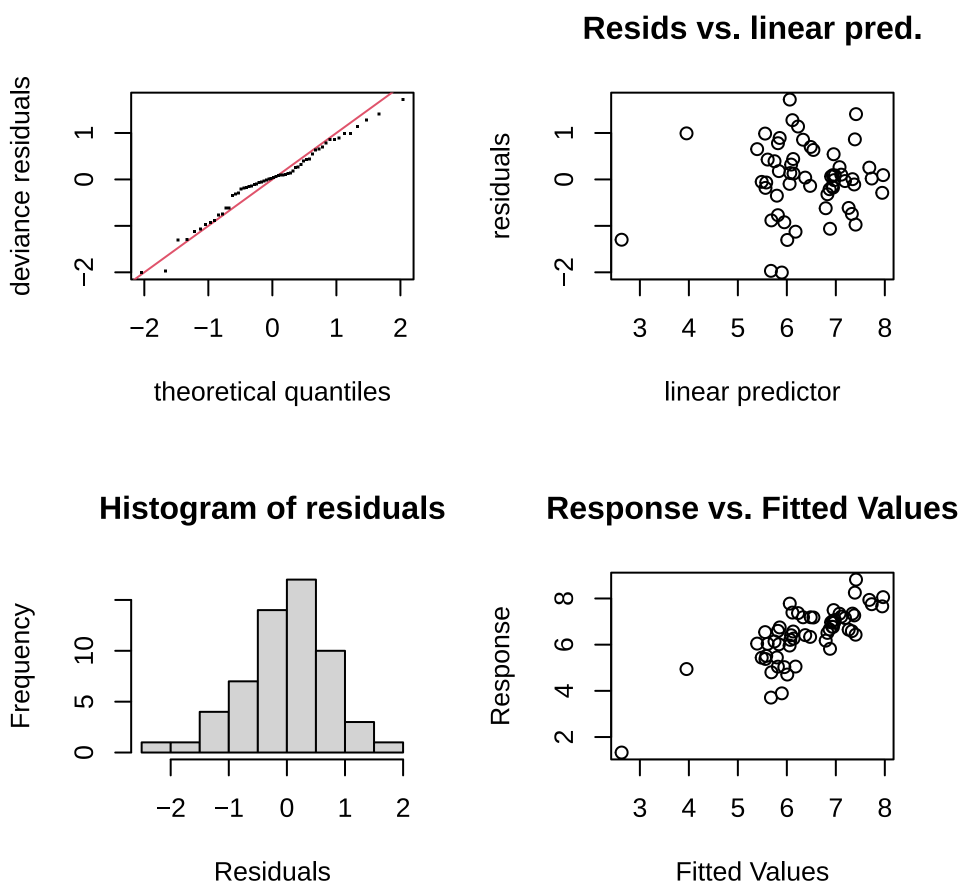
Plot GAM

```
oldpar <- par(mfrow = c(2,3))
plot(mod)
par(oldpar)
```



Model Diagnostics

```
oldpar <- par(mfrow = c(2,2))
gam.check(mod)
```




```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 53 iterations.
#> The RMS GCV score gradient at convergence was 1.105685e-07 .
#> The Hessian was positive definite.
#> Model rank =  21 / 21
#>
#> 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(Temp)      3.00e+00 2.12e+00    1.09    0.68
#> s(Sal)       3.00e+00 2.76e+00    1.16    0.84
#> s(log(Turb)) 3.00e+00 9.29e-01    0.89    0.16
#> s(log(Chl))  3.00e+00 1.92e-01    0.95    0.29
#> s(log1p(RH)) 3.00e+00 1.24e-10    1.07    0.64
par(oldpar)
```

Polychaete

Summary and ANOVA

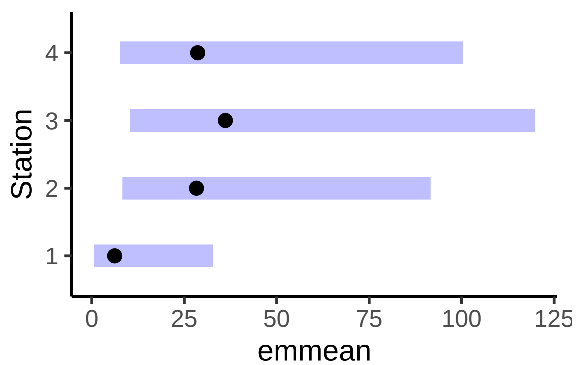
```
spp =  "Polychaete"
mod <- spp_analysis$gam_mods[spp_analysis$Species == spp][[1]]
summary(mod)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 4) +
#>     s(Sal, bs = "ts", k = 4) + s(log(Turb), bs = "ts", k = 4) +
#>     s(log(Chl), bs = "ts", k = 4) + s(log1p(RH), bs = "ts", k = 4)
#>
#> Parametric coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    4.0181     0.9547   4.209 0.000111 ***
#> Station2       1.4049     0.9529   1.474 0.146852
#> Station3       1.6412     0.9084   1.807 0.077027 .
#> Station4       1.4158     0.9844   1.438 0.156779
#> SeasonSummer  -3.6977     1.4839  -2.492 0.016177 *
#> SeasonFall    -2.3651     1.3738  -1.722 0.091510 .
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>                    edf Ref.df     F p-value
#> s(Temp)      1.846e+00      3 1.305 0.11752
#> s(Sal)       9.830e-11      3 0.000 0.56768
#> s(log(Turb)) 7.525e-01      3 0.918 0.05710 .
#> s(log(Chl))  9.075e-01      3 2.149 0.00831 **
#> s(log1p(RH)) 5.935e-11      3 0.000 0.56093
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.437   Deviance explained = 52.1%
#> GCV = 4.5041  Scale est. = 3.7659    n = 58
cat('\n')
anova(mod)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 4) +
#>     s(Sal, bs = "ts", k = 4) + s(log(Turb), bs = "ts", k = 4) +
#>     s(log(Chl), bs = "ts", k = 4) + s(log1p(RH), bs = "ts", k = 4)
#>
#> Parametric Terms:
#>         df     F p-value
#> Station  3 1.131  0.3458
```

```
#> Season   2 3.820  0.0288
#>
#> Approximate significance of smooth terms:
#>                     edf    Ref.df     F p-value
#> s(Temp)      1.846e+00 3.000e+00 1.305 0.11752
#> s(Sal)       9.830e-11 3.000e+00 0.000 0.56768
#> s(log(Turb)) 7.525e-01 3.000e+00 0.918 0.05710
#> s(log(Chl))  9.075e-01 3.000e+00 2.149 0.00831
#> s(log1p(RH)) 5.935e-11 3.000e+00 0.000 0.56093
```

Comparison of Station and Season

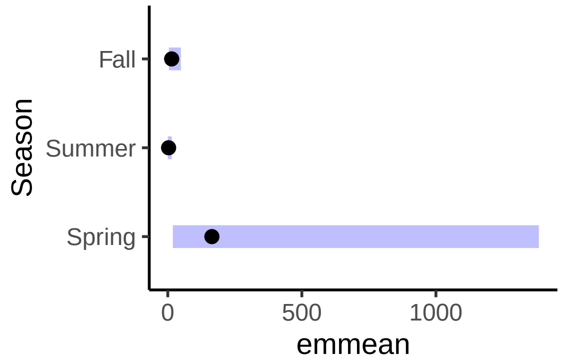
I'm showing "marginal" means – essentially means adjusted for the other predictors, at their mean values.

```
Sta_emms <- emmeans(mod, ~Station, type = 'response',
                    data = spp_analysis$data[spp_data$Species == spp][[1]])
plot(Sta_emms)
```




```
pairs(Sta_emms, adjust ='bonferroni')
#> Note: Use 'contrast(regrid(object), ...)' to obtain contrasts of back-transformed estimates
#>  contrast            estimate    SE   df t.ratio p.value
#>  Station1 - Station2  -1.4049 0.953 48.5  -1.474  0.8811
#>  Station1 - Station3  -1.6412 0.908 48.5  -1.807  0.4622
#>  Station1 - Station4  -1.4158 0.984 48.5  -1.438  0.9407
#>  Station2 - Station3  -0.2363 0.745 48.5  -0.317  1.0000
#>  Station2 - Station4  -0.0109 0.781 48.5  -0.014  1.0000
#>  Station3 - Station4   0.2254 0.742 48.5   0.304  1.0000
#>
#> Results are averaged over the levels of: Season
#> Note: contrasts are still on the log1p scale
#> P value adjustment: bonferroni method for 6 tests
```

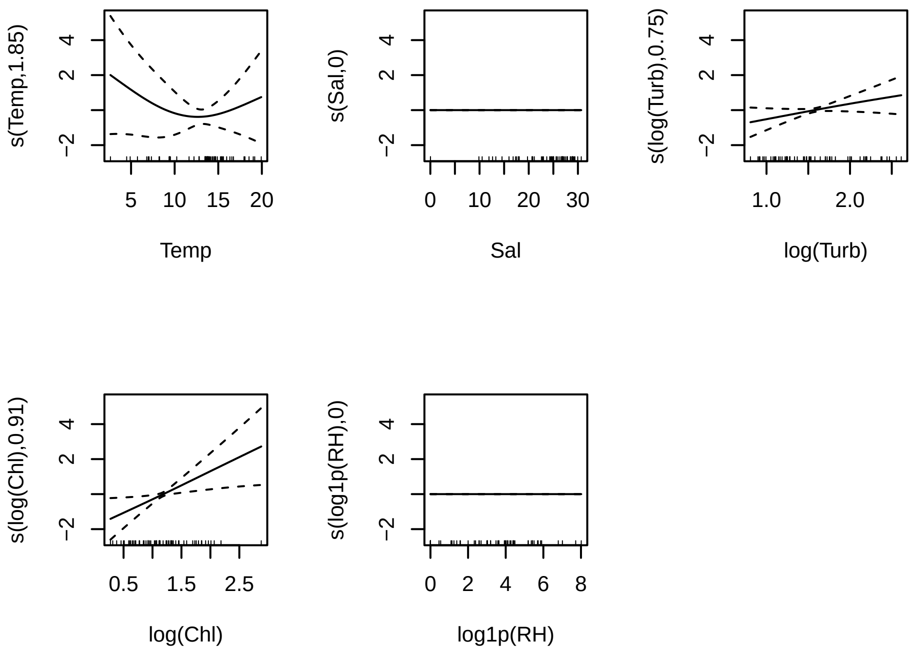
```
Seas_emms <- emmeans(mod, ~Season, type = 'response',
                     data = spp_analysis$data[spp_data$Species == spp][[1]])
plot(Seas_emms)
```




```
pairs(Seas_emms, adjust ='bonferroni')
#> Note: Use 'contrast(regrid(object), ...)' to obtain contrasts of back-transformed estimates
#>  contrast        estimate    SE   df t.ratio p.value
#>  Spring - Summer     3.70 1.484 48.5   2.492  0.0485
#>  Spring - Fall       2.37 1.374 48.5   1.722  0.2745
#>  Summer - Fall      -1.33 0.648 48.5  -2.057  0.1353
#>
#> Results are averaged over the levels of: Station
#> Note: contrasts are still on the log1p scale
#> P value adjustment: bonferroni method for 3 tests
```

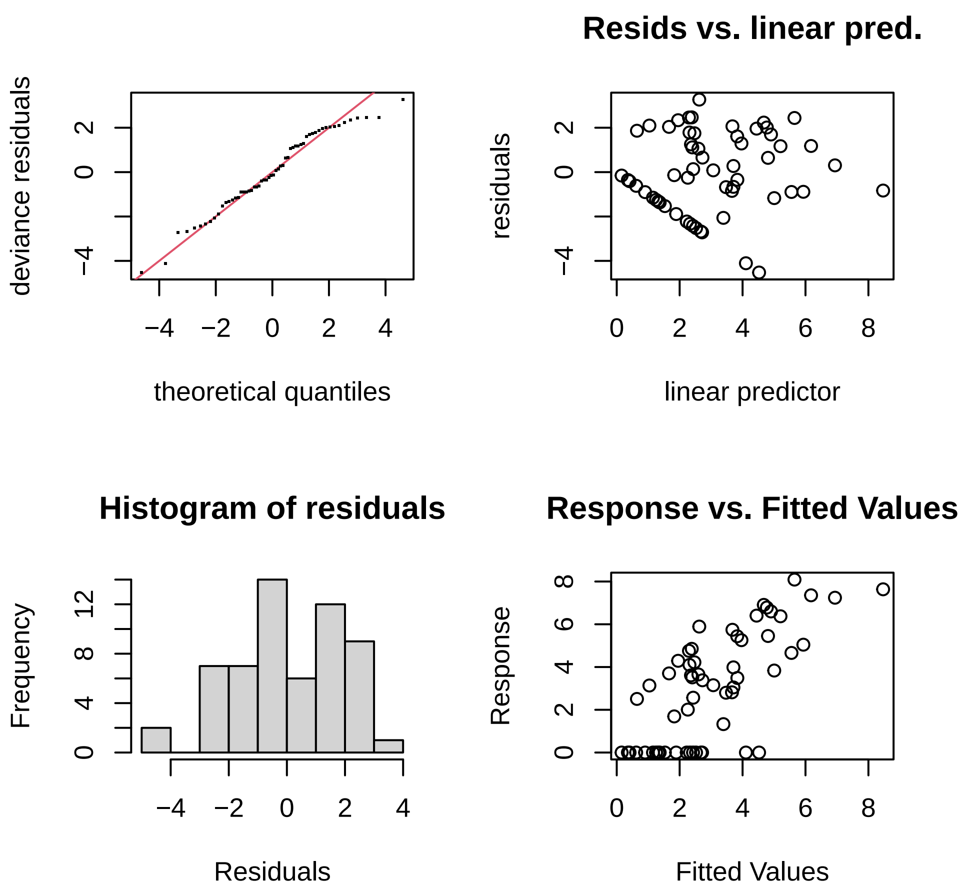
Plot GAM

```
oldpar <- par(mfrow = c(2,3))
plot(mod)
par(oldpar)
```



Model Diagnostics

```
oldpar <- par(mfrow = c(2,2))
gam.check(mod)
```




```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 20 iterations.
#> The RMS GCV score gradient at convergence was 1.922952e-07 .
#> The Hessian was positive definite.
#> Model rank =  21 / 21
#>
#> 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(Temp)      3.00e+00 1.85e+00    1.03   0.555
#> s(Sal)       3.00e+00 9.83e-11    1.11   0.745
#> s(log(Turb)) 3.00e+00 7.52e-01    0.76   0.025 *
#> s(log(Chl))  3.00e+00 9.07e-01    0.94   0.315
#> s(log1p(RH)) 3.00e+00 5.93e-11    1.09   0.655
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
par(oldpar)
```

Pseudocal

Summary and ANOVA

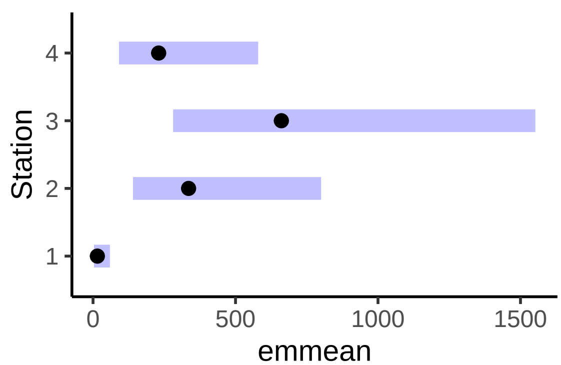
```
spp =  "Pseudocal"
mod <- spp_analysis$gam_mods[spp_analysis$Species == spp][[1]]
summary(mod)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 4) +
#>     s(Sal, bs = "ts", k = 4) + s(log(Turb), bs = "ts", k = 4) +
#>     s(log(Chl), bs = "ts", k = 4) + s(log1p(RH), bs = "ts", k = 4)
#>
#> Parametric coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    4.2268     0.7730   5.468 1.68e-06 ***
#> Station2       3.0302     0.7968   3.803 0.000410 ***
#> Station3       3.7070     0.7186   5.159 4.85e-06 ***
#> Station4       2.6556     0.7700   3.449 0.001195 **
#> SeasonSummer  -2.4303     0.9624  -2.525 0.014978 *
#> SeasonFall    -4.0359     0.9639  -4.187 0.000122 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>                    edf Ref.df     F  p-value
#> s(Temp)      2.165e+00      3 8.642 1.86e-05 ***
#> s(Sal)       9.069e-01      3 1.960   0.0109 *
#> s(log(Turb)) 1.699e+00      3 0.678   0.3018
#> s(log(Chl))  1.603e-10      3 0.000   0.5275
#> s(log1p(RH)) 2.440e-10      3 0.000   0.7785
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.675   Deviance explained =   73%
#> GCV = 2.1139  Scale est. = 1.7213    n = 58
cat('\n')
anova(mod)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 4) +
#>     s(Sal, bs = "ts", k = 4) + s(log(Turb), bs = "ts", k = 4) +
#>     s(log(Chl), bs = "ts", k = 4) + s(log1p(RH), bs = "ts", k = 4)
#>
#> Parametric Terms:
#>         df      F  p-value
#> Station  3  9.161 6.91e-05
```

```
#> Season    2 12.793 3.63e-05
#>
#> Approximate significance of smooth terms:
#>                    edf    Ref.df     F  p-value
#> s(Temp)      2.165e+00 3.000e+00 8.642 1.86e-05
#> s(Sal)       9.069e-01 3.000e+00 1.960   0.0109
#> s(log(Turb)) 1.699e+00 3.000e+00 0.678   0.3018
#> s(log(Chl))  1.603e-10 3.000e+00 0.000   0.5275
#> s(log1p(RH)) 2.440e-10 3.000e+00 0.000   0.7785
```

Comparison of Station and Season

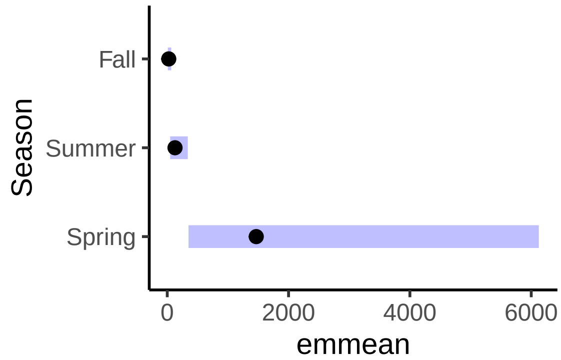
I'm showing "marginal" means – essentially means adjusted for the other predictors, at their mean values.

```
Sta_emms <- emmeans(mod, ~Station, type = 'response',
                    data = spp_analysis$data[spp_data$Species == spp][[1]])
plot(Sta_emms)
```




```
pairs(Sta_emms, adjust ='bonferroni')
#> Note: Use 'contrast(regrid(object), ...)' to obtain contrasts of back-transformed estimates
#>  contrast            estimate    SE   df t.ratio p.value
#>  Station1 - Station2   -3.030 0.797 47.2  -3.803  0.0025
#>  Station1 - Station3   -3.707 0.719 47.2  -5.159  <.0001
#>  Station1 - Station4   -2.656 0.770 47.2  -3.449  0.0072
#>  Station2 - Station3   -0.677 0.522 47.2  -1.296  1.0000
#>  Station2 - Station4    0.375 0.542 47.2   0.690  1.0000
#>  Station3 - Station4    1.051 0.496 47.2   2.120  0.2357
#>
#> Results are averaged over the levels of: Season
#> Note: contrasts are still on the log1p scale
#> P value adjustment: bonferroni method for 6 tests
```

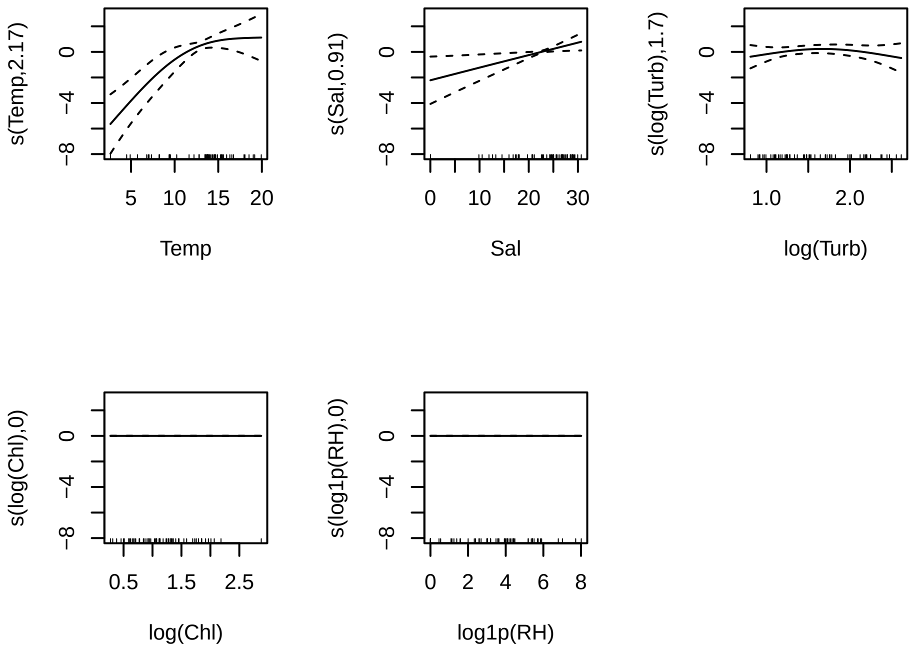
```
Seas_emms <- emmeans(mod, ~Season, type = 'response',
                     data = spp_analysis$data[spp_data$Species == spp][[1]])
plot(Seas_emms)
```




```
pairs(Seas_emms, adjust ='bonferroni')
#> Note: Use 'contrast(regrid(object), ...)' to obtain contrasts of back-transformed estimates
#>  contrast        estimate    SE   df t.ratio p.value
#>  Spring - Summer     2.43 0.962 47.2   2.525  0.0449
#>  Spring - Fall       4.04 0.964 47.2   4.187  0.0004
#>  Summer - Fall       1.61 0.432 47.2   3.716  0.0016
#>
#> Results are averaged over the levels of: Station
#> Note: contrasts are still on the log1p scale
#> P value adjustment: bonferroni method for 3 tests
```

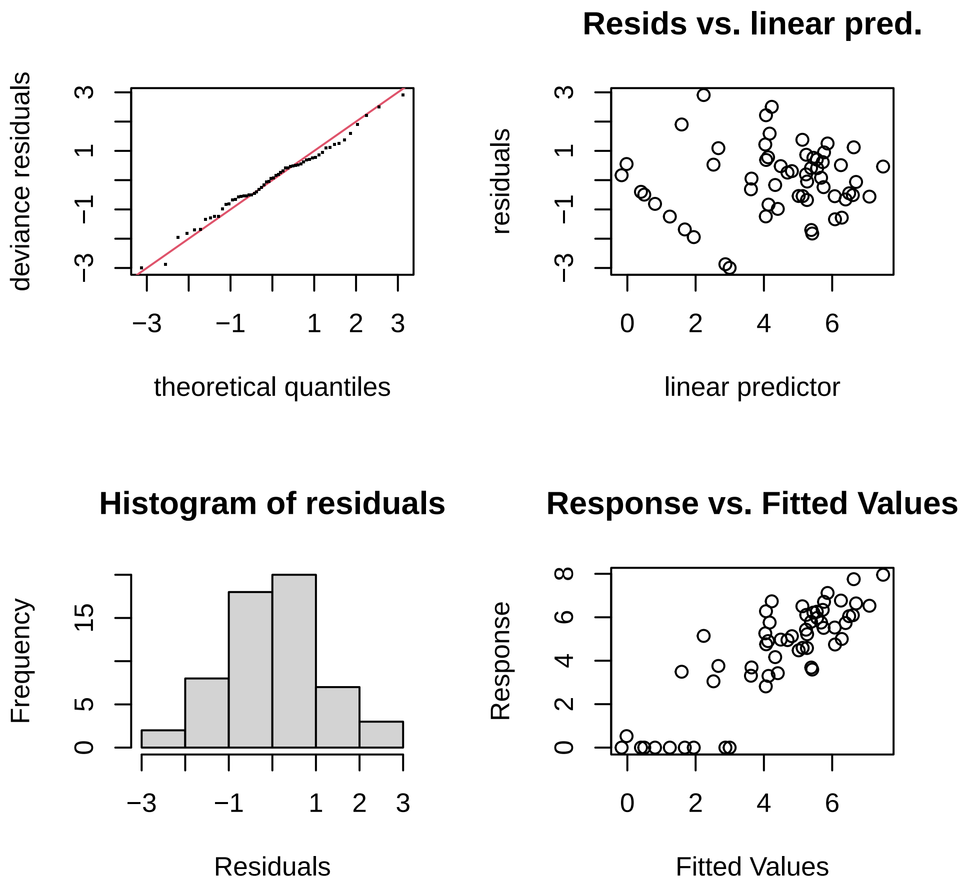
Plot GAM

```
oldpar <- par(mfrow = c(2,3))
plot(mod)
par(oldpar)
```



Model Diagnostics

```r
oldpar <- par(mfrow = c(2,2))
gam.check(mod)
```




```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 20 iterations.
#> The RMS GCV score gradient at convergence was 8.18504e-08 .
#> The Hessian was positive definite.
#> Model rank =  21 / 21
#>
#> 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(Temp)       3.00e+00 2.17e+00    0.83   0.055 .
#> s(Sal)        3.00e+00 9.07e-01    0.83   0.060 .
#> s(log(Turb)) 3.00e+00 1.70e+00    1.04   0.550
#> s(log(Chl))  3.00e+00 1.60e-10    1.34   1.000
#> s(log1p(RH)) 3.00e+00 2.44e-10    1.16   0.880
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
par(oldpar)
```

Temora

Summary and ANOVA

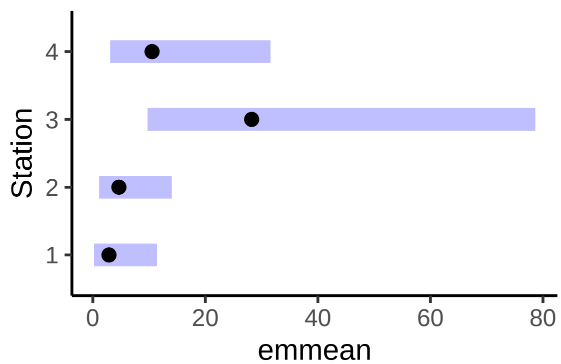
```
spp =  "Temora"
mod <- spp_analysis$gam_mods[spp_analysis$Species == spp][[1]]
summary(mod)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 4) +
#>     s(Sal, bs = "ts", k = 4) + s(log(Turb), bs = "ts", k = 4) +
#>     s(log(Chl), bs = "ts", k = 4) + s(log1p(RH), bs = "ts", k = 4)
#>
#> Parametric coefficients:
#>               Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    1.27819    0.79945   1.599  0.11607
#> Station2       0.37540    0.74772   0.502  0.61780
#> Station3       2.02093    0.73328   2.756  0.00811 **
#> Station4       1.09087    0.79124   1.379  0.17405
#> SeasonSummer -0.47093    0.76757  -0.614  0.54227
#> SeasonFall   -0.08541    0.70719  -0.121  0.90435
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>                    edf Ref.df     F p-value
#> s(Temp)      2.395e-11      3 0.000 0.63955
#> s(Sal)       1.071e-01      3 0.043 0.27433
#> s(log(Turb)) 2.998e-01      3 0.151 0.22275
#> s(log(Chl))  8.847e-01      3 2.300 0.00588 **
#> s(log1p(RH)) 1.190e-10      3 0.000 0.51267
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.219   Deviance explained = 30.5%
#> GCV = 3.8932  Scale est. = 3.4037    n = 58
cat('\n')
anova(mod)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 4) +
#>     s(Sal, bs = "ts", k = 4) + s(log(Turb), bs = "ts", k = 4) +
#>     s(log(Chl), bs = "ts", k = 4) + s(log1p(RH), bs = "ts", k = 4)
#>
#> Parametric Terms:
#>         df     F p-value
#> Station  3 3.110  0.0344
```

```
#> Season   2 0.267  0.7670
#>
#> Approximate significance of smooth terms:
#>                    edf    Ref.df     F p-value
#> s(Temp)      2.395e-11 3.000e+00 0.000 0.63955
#> s(Sal)       1.071e-01 3.000e+00 0.043 0.27433
#> s(log(Turb)) 2.998e-01 3.000e+00 0.151 0.22275
#> s(log(Chl))  8.847e-01 3.000e+00 2.300 0.00588
#> s(log1p(RH)) 1.190e-10 3.000e+00 0.000 0.51267
```

Comparison of Station and Season

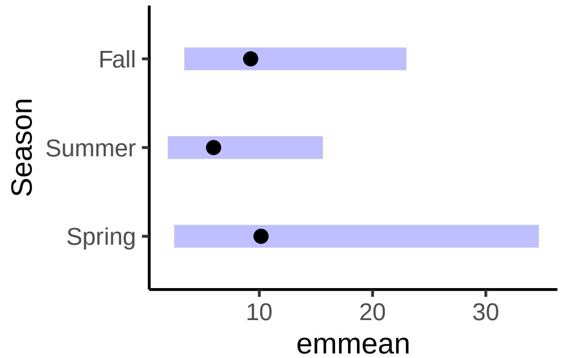
I'm showing "marginal" means – essentially means adjusted for the other predictors, at their mean values.

```
Sta_emms <- emmeans(mod, ~Station, type = 'response',
                    data = spp_analysis$data[spp_data$Species == spp][[1]])
plot(Sta_emms)
```




```
pairs(Sta_emms, adjust ='bonferroni')
#> Note: Use 'contrast(regrid(object), ...)' to obtain contrasts of back-transformed estimates
#>  contrast            estimate    SE   df t.ratio p.value
#>  Station1 - Station2   -0.375 0.748 50.7  -0.502  1.0000
#>  Station1 - Station3   -2.021 0.733 50.7  -2.756  0.0487
#>  Station1 - Station4   -1.091 0.791 50.7  -1.379  1.0000
#>  Station2 - Station3   -1.646 0.687 50.7  -2.394  0.1223
#>  Station2 - Station4   -0.715 0.711 50.7  -1.007  1.0000
#>  Station3 - Station4    0.930 0.701 50.7   1.327  1.0000
#>
#> Results are averaged over the levels of: Season
#> Note: contrasts are still on the log1p scale
#> P value adjustment: bonferroni method for 6 tests
```

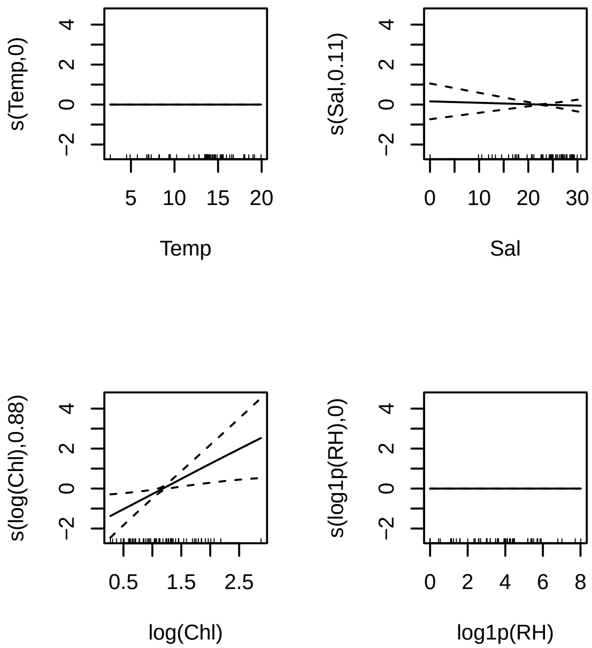
```
Seas_emms <- emmeans(mod, ~Season, type = 'response',
                     data = spp_analysis$data[spp_data$Species == spp][[1]])
plot(Seas_emms)
```




```
pairs(Seas_emms, adjust ='bonferroni')
#> Note: Use 'contrast(regrid(object), ...)' to obtain contrasts of back-transformed estimates
#>  contrast        estimate    SE   df t.ratio p.value
#>  Spring - Summer   0.4709 0.768 50.7   0.614  1.0000
#>  Spring - Fall     0.0854 0.707 50.7   0.121  1.0000
#>  Summer - Fall    -0.3855 0.598 50.7  -0.645  1.0000
#>
#> Results are averaged over the levels of: Station
#> Note: contrasts are still on the log1p scale
#> P value adjustment: bonferroni method for 3 tests
```

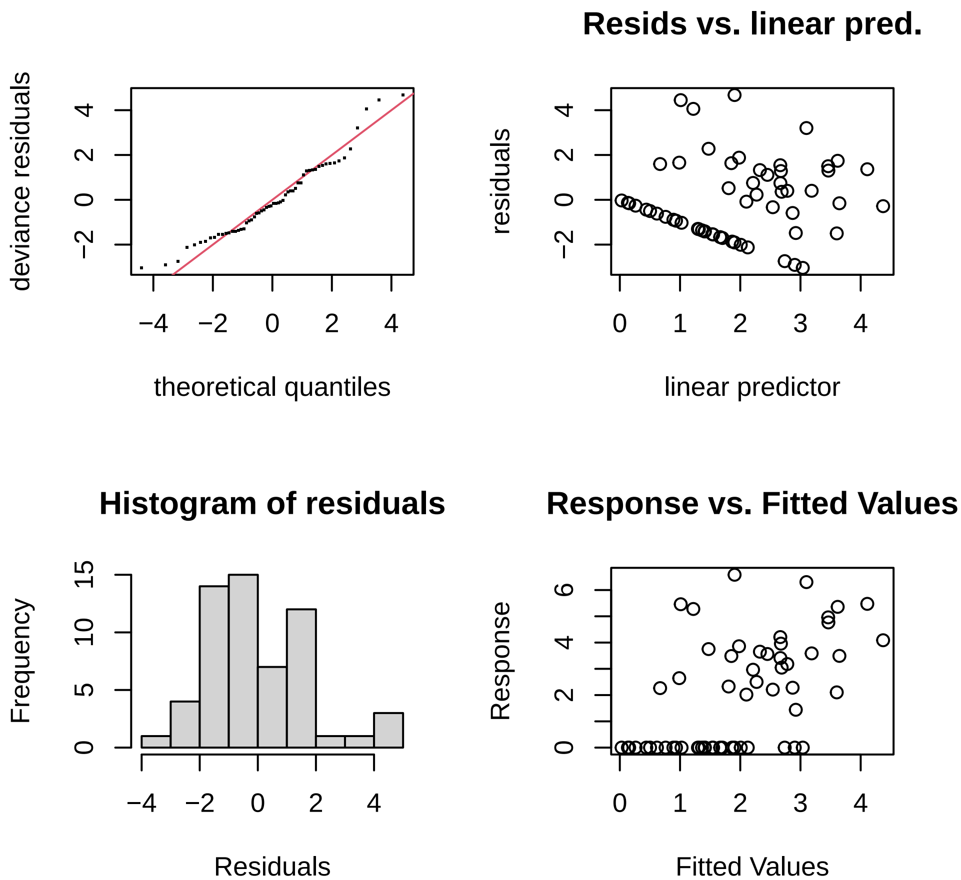
PLot GAM

```
oldpar <- par(mfrow = c(2,3))
plot(mod)
par(oldpar)
```



Model Diagnostics

```r
oldpar <- par(mfrow = c(2,2))
gam.check(mod)
```




```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 16 iterations.
#> The RMS GCV score gradient at convergence was 2.765473e-07 .
#> The Hessian was positive definite.
#> Model rank =  21 / 21
#>
#> 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(Temp)       3.00e+00 2.39e-11    1.07    0.69
#> s(Sal)        3.00e+00 1.07e-01    1.13    0.80
#> s(log(Turb)) 3.00e+00 3.00e-01    1.10    0.69
#> s(log(Chl))  3.00e+00 8.85e-01    0.97    0.34
#> s(log1p(RH)) 3.00e+00 1.19e-10    1.25    0.97
par(oldpar)
```