

GAMs to Analyze Plankton Community NMDS Data – Final Additions

Curtis C. Bohlen, Casco Bay Estuary Partnership

5/17/2022

Contents

Introduction	2
Load Libraries	3
Set Graphics Theme	3
Input Data	3
Folder References	3
Load Data	3
Environmental Predictors	5
Temperature	6
Salinity	8
Turbidity	10
Chlorophyll	11
Dissolved Oxygen Percent Saturation	13
Discussion	15
Total Zooplankton Density	15
Summary and Anova	15
Plot the GAM	17
Station and Season	17
Model Diagnostics	19
Discussion	20

Shannon Diversity	20
Histogram	20
Summary and Anova	20
Station (Season is Not Significant by ANOVA)	22
Plot the GAM	22
Model of River Herring Abundance	25
Summary and ANOVA	25
Station (Season is Not Significant by ANOVA)	26
Plot GAM results	27
Model Diagnostics	27
Single Species Models	29
Model Choice	29
Automating Analysis of Separate Species	29
Acartia	30
Balanus	35
Eurytemora	39
Polychaete	46
Pseudocal	50
Temora	54
Why Did the Polychaete Model fail to Converge?	58
And does it matter?	58

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(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))) %>%

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()`.

```

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
#>   <dtm>              <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.

```

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
#>   <dtm>                <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 than 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.

```
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 must have data columns with the correct names, and all variables transformed before we call it.

```
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.

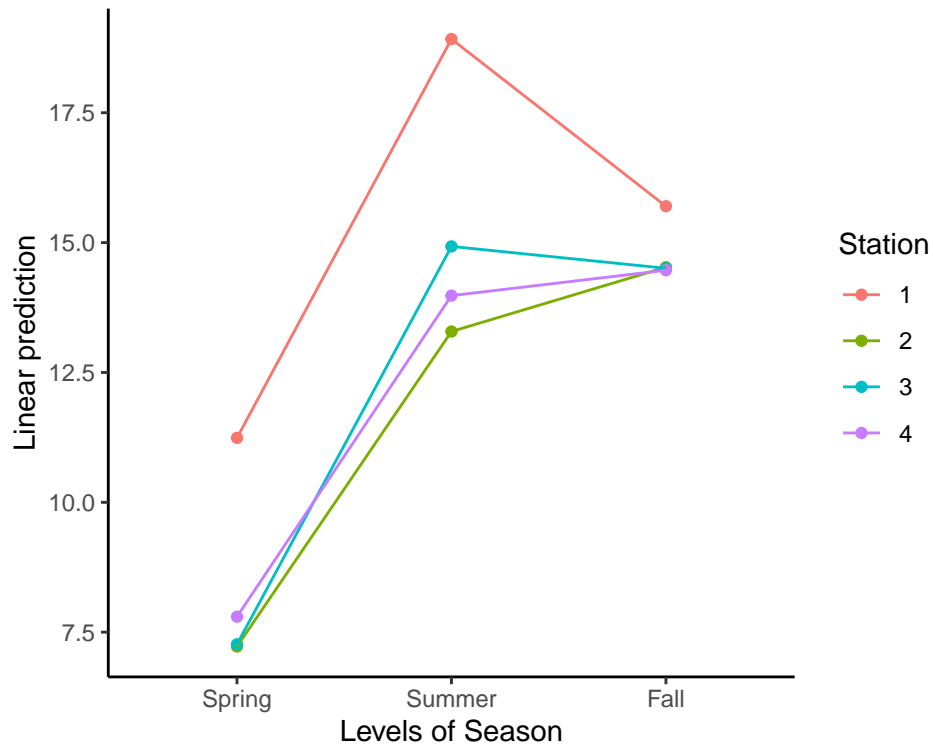
```
env_analysis <- env_data %>%
  group_by(Parameter) %>%
  nest() %>%
  mutate(lme_mods = map(data, my_lme))
```

Temperature

```
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.

```
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

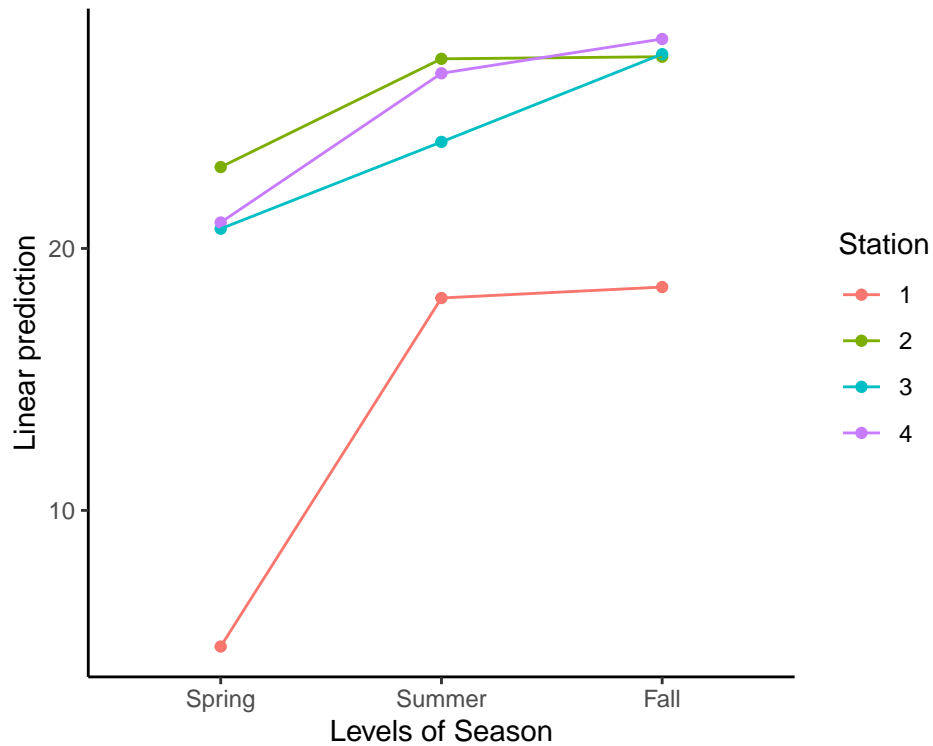
```

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.

```

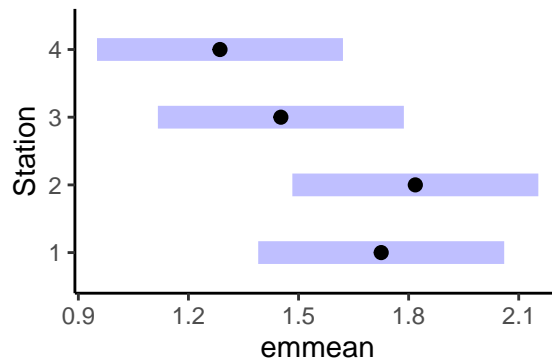
tmp <- env_analysis$data[env_analysis$Parameter == parm][[1]]
test <- lme(Value ~ Station,
  random = list(Yearf = ~ 1, sample_event = ~ 1),
  data = tmp, na.action = na.omit)
(emm <- emmeans(test, pairwise~ Station))

```

```

#> $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

```

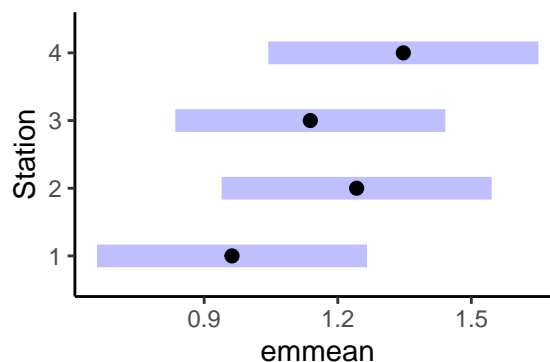
The interaction term is not significant, but this time both main effects are significant.

```

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)

```



```

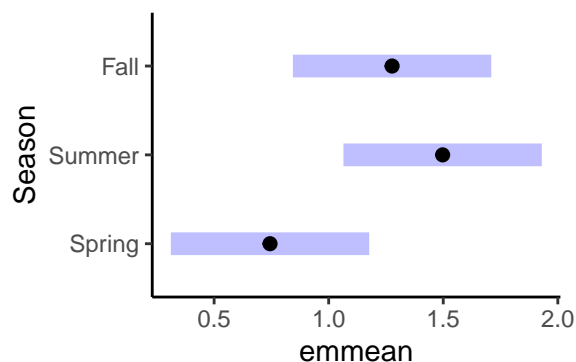
(emm_seas <- emmeans(test, pairwise~ Season))
#> $emmeans
#> Season emmean SE df lower.CL upper.CL
#> Spring 0.744 0.156 4 0.311 1.18
#> Summer 1.497 0.156 4 1.064 1.93
#> Fall 1.277 0.156 4 0.844 1.71
#>
#> Results are averaged over the levels of: Station

```

```

#> 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)

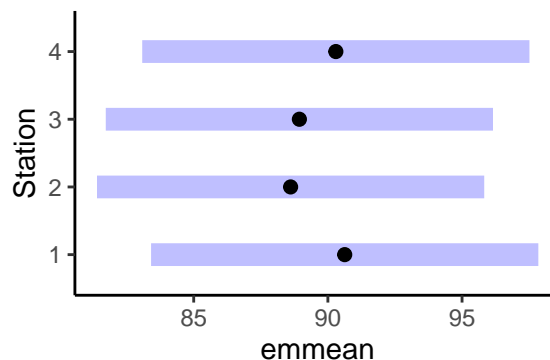
(emm_stat <- emmeans(test, pairwise~ Station))

```

```

#> $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)

```

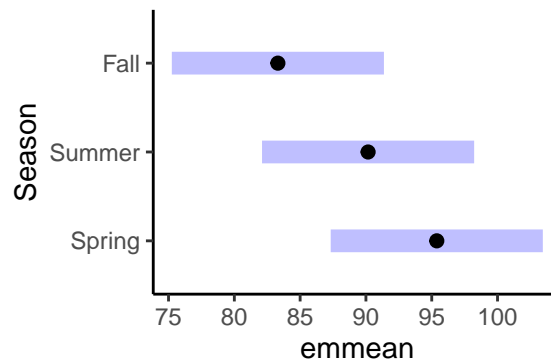


```

(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

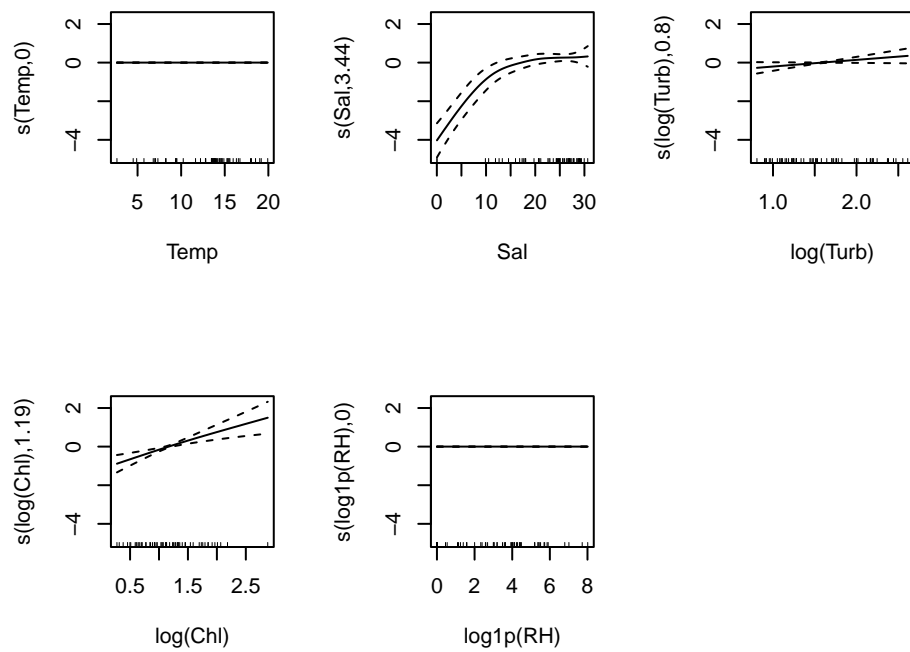
```



```
#> 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
```

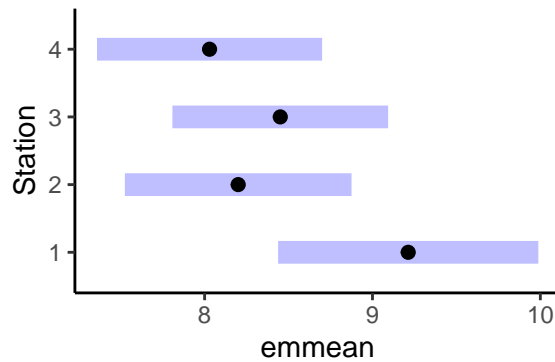
Plot the GAM

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



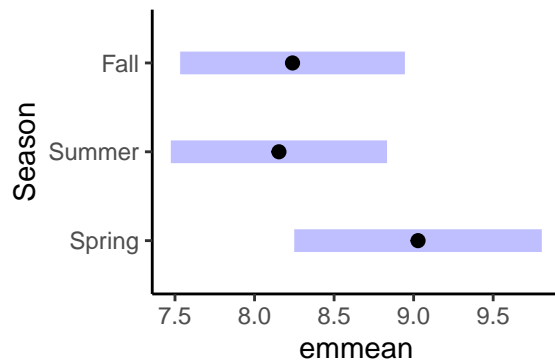
Station and Season

```
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
```

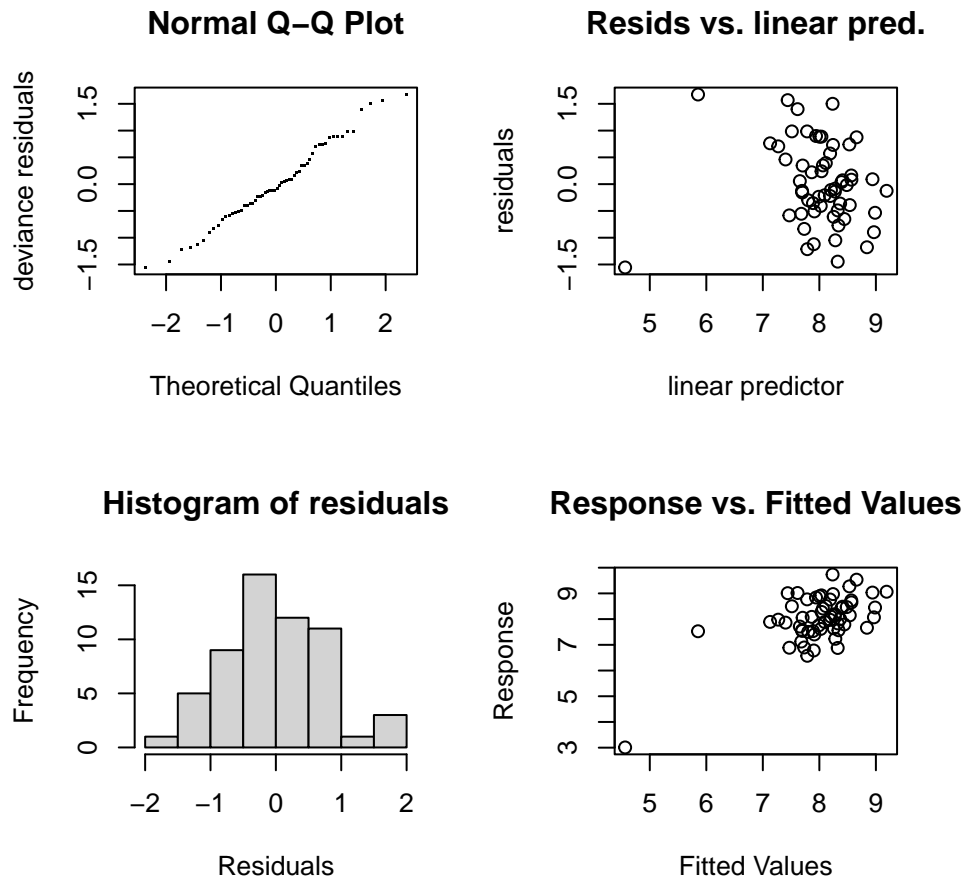
```
Seas_emms <- emmeans(density_gam, ~Season, type = 'response',
                     data = base_data)
plot(Seas_emms)
```



```
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  0.69
#> s(Sal)     9.00e+00 3.44e+00  1.03  0.58
#> s(log(Turb)) 9.00e+00 8.03e-01  0.89  0.14
#> s(log(Chl))  9.00e+00 1.19e+00  0.94  0.26
#> 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

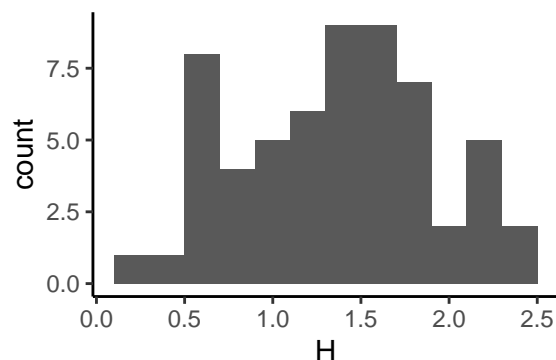
- The Station differences are significant by ANOVA F test. Pairwise comparisons show that Station 1 (upstream) shows the highest combined density, which is significantly higher than for Stations 2 and 4, but not different from Station 3 (by multiple comparisons test anyway). There are no meaningful differences 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).
```



Summary and Anova

```
shannon_gam <- gamm(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"),
```

```

        random = list(Yearf = ~ 1, sample_event = ~ 1),
        data = base_data, family = 'gaussian')
#> Warning in lme.formula(y ~ X - 1, random = rand, data = strip.offset(mf), : nlminb problem, convergence
#> message = singular convergence (7)
summary(shannon_gam$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)    0.90968    0.26455   3.439  0.00119 **
#> Station2       0.58077    0.24431   2.377  0.02134 *
#> Station3       0.63064    0.22383   2.818  0.00693 **
#> Station4       0.50504    0.23573   2.142  0.03709 *
#> SeasonSummer   0.07522    0.21251   0.354  0.72487
#> SeasonFall    -0.05613    0.22538  -0.249  0.80434
#> ---
#> 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)        7.667e-09     9 0.00 0.49594
#> s(Sal)         2.339e+00     9 1.57 0.00544 **
#> s(log(Turb))   4.107e-09     9 0.00 0.91443
#> s(log(Chl))    1.112e-08     9 0.00 0.87925
#> s(log1p(RH))   6.777e-09     9 0.00 0.29071
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.241
#> Scale est. = 0.16929    n = 58

```

```

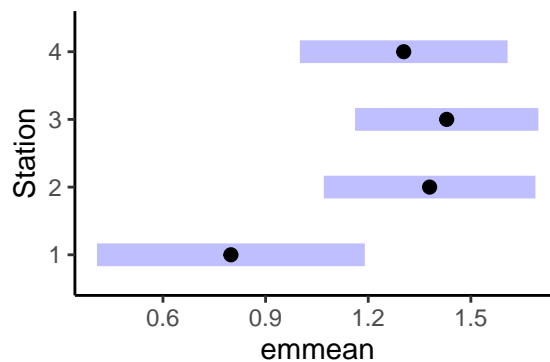
anova(shannon_gam$gam)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> H ~ Station + Season + s(Temp, bs = "ts") + s(Sal, bs = "ts") +
#>       s(log(Turb), bs = "ts") + s(log(Chl), bs = "ts") + s(log1p(RH),
#>       bs = "ts")
#>
#> Parametric Terms:
#>               df      F p-value
#> Station      3 2.699  0.0557
#> Season       2 0.250  0.7795
#>

```

```
#> Approximate significance of smooth terms:
#>           edf    Ref.df    F p-value
#> s(Temp)      7.667e-09 9.000e+00 0.00 0.49594
#> s(Sal)       2.339e+00 9.000e+00 1.57 0.00544
#> s(log(Turb)) 4.107e-09 9.000e+00 0.00 0.91443
#> s(log(Chl))  1.112e-08 9.000e+00 0.00 0.87925
#> s(log1p(RH)) 6.777e-09 9.000e+00 0.00 0.29071
```

Station (Season is Not Significant by ANOVA)

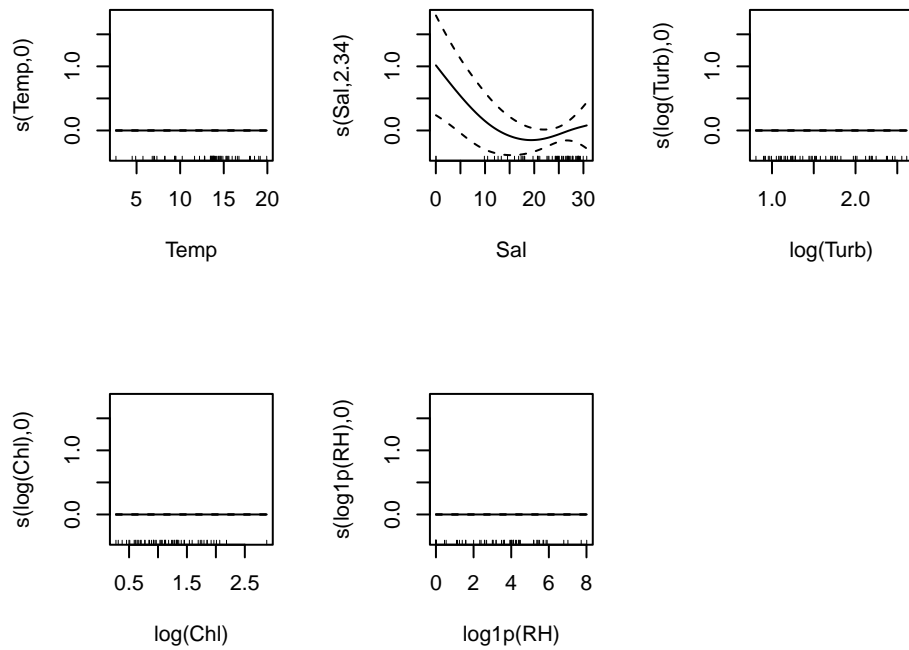
```
Sta_emms <- emmeans(shannon_gam, ~Station, type = 'response',
                    data = base_data)
plot(Sta_emms)
```



```
pairs(Sta_emms, adjust = 'bonferroni')
#> contrast      estimate    SE   df t.ratio p.value
#> Station1 - Station2 -0.5808 0.244 49.7 -2.377 0.1281
#> Station1 - Station3 -0.6306 0.224 49.7 -2.818 0.0416
#> Station1 - Station4 -0.5050 0.236 49.7 -2.142 0.2225
#> Station2 - Station3 -0.0499 0.165 49.7 -0.303 1.0000
#> Station2 - Station4  0.0757 0.163 49.7  0.464 1.0000
#> Station3 - Station4  0.1256 0.165 49.7  0.761 1.0000
#>
#> Results are averaged over the levels of: Season
#> P value adjustment: bonferroni method for 6 tests
```

Plot the GAM

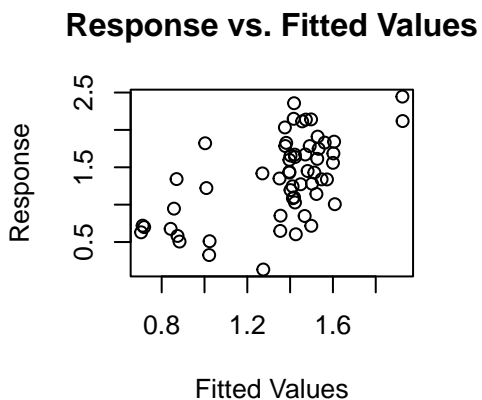
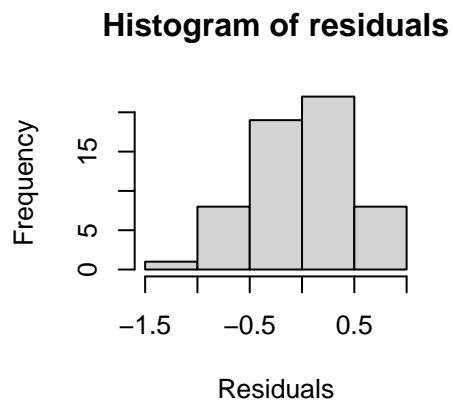
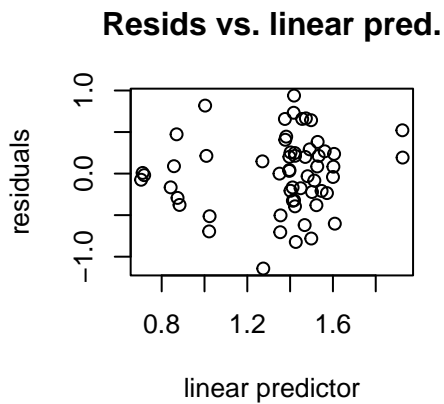
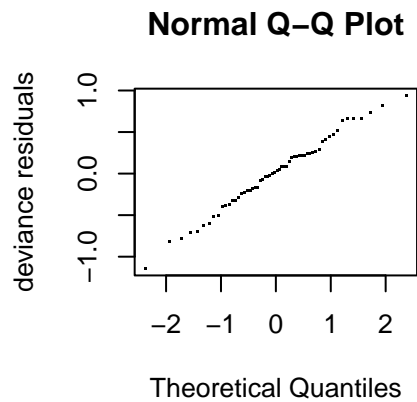
```
oldpar <- par(mfrow = c(2,3))
plot(shannon_gam$gam)
par(oldpar)
```



Only the relationship with salinity is retained in the model as statistically significant. It appears much, but perhaps not all, of that pattern is driven by a couple of low salinity samples.

Diagnostic Plots

```
oldpar <- par(mfrow = c(2,2))
gam.check(shannon_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 7.67e-09   1.00   0.44
#> s(Sal)     9.00e+00 2.34e+00   1.01   0.48
#> s(log(Turb)) 9.00e+00 4.11e-09   1.30   0.99
#> s(log(Chl))  9.00e+00 1.11e-08   1.06   0.64
#> s(log1p(RH)) 9.00e+00 6.78e-09   1.08   0.71
par(oldpar)
```

Not a bad model from a diagnostics point of view.

Model of River Herring Abundance

Summary and ANOVA

```
herring_gam <- gamm(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$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)    6.7389      1.0496   6.420 4.67e-08 ***
#> Station2      -2.6913      0.8822  -3.051 0.003635 **
#> Station3      -3.3433      0.8049  -4.153 0.000126 ***
#> Station4      -3.4724      0.8652  -4.014 0.000198 ***
#> SeasonSummer  -1.1009      0.9761  -1.128 0.264686
#> SeasonFall    -1.8403      1.0077  -1.826 0.073724 .
#> ---
#> 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)                5.829e-09      9 0.000 0.29550
#> s(Sal)                  9.508e-01      9 0.863 0.00893 **
#> s(log(Turb))            5.551e-09      9 0.000 0.76455
#> s(log(Chl))             5.527e-09      9 0.000 0.74994
#> s(log1p(combined_density)) 5.946e-01      9 0.166 0.15100
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.149
#>   Scale est. = 2.2083    n = 58
```

```
anova(herring_gam$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 6.660 0.000709
#> Season   2 1.671 0.198381
#>
#> Approximate significance of smooth terms:
#>      edf   Ref.df      F p-value
#> s(Temp)      5.829e-09 9.000e+00 0.000 0.29550
#> s(Sal)      9.508e-01 9.000e+00 0.863 0.00893
#> s(log(Turb)) 5.551e-09 9.000e+00 0.000 0.76455
#> s(log(Chl))  5.527e-09 9.000e+00 0.000 0.74994
#> s(log1p(combined_density)) 5.946e-01 9.000e+00 0.166 0.15100

```

Note that overall, Station matters, but Season does not.

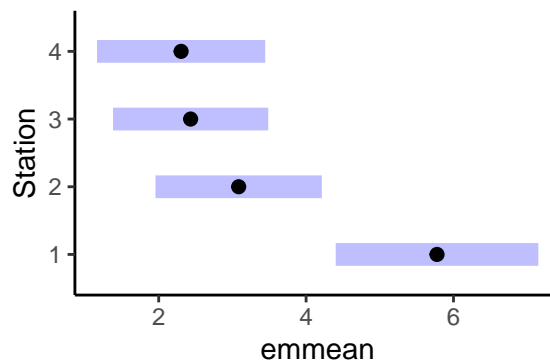
Only the smoothed relationship with Salinity in the model is statistically significant. The relationship is essentially linear (EDF = 0.95).

Station (Season is Not Significant by ANOVA)

```

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

```



```

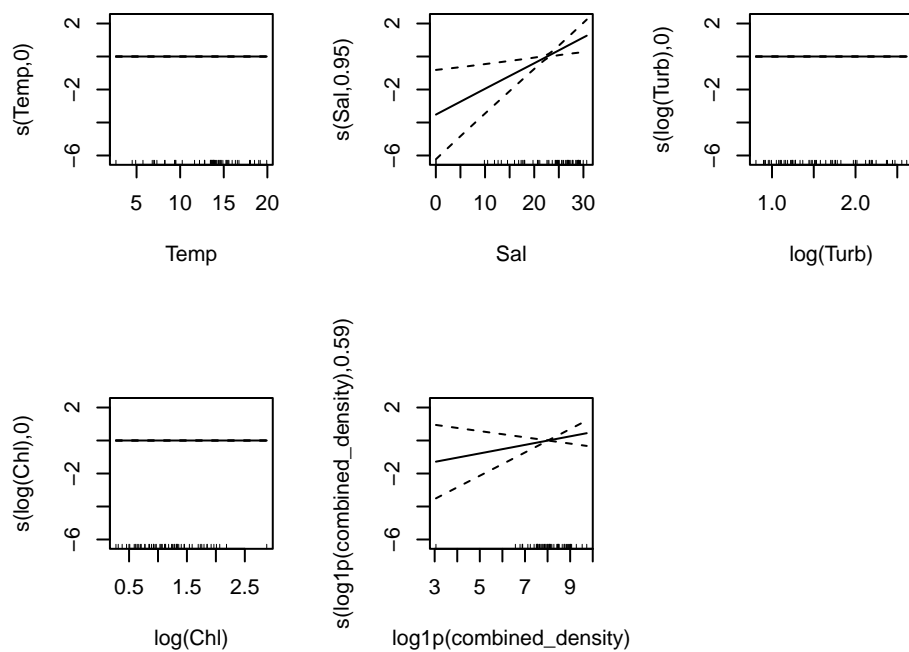
pairs(Sta_emms, adjust = 'bonferroni')
#> contrast      estimate      SE    df t.ratio p.value
#> Station1 - Station2    2.691 0.882 50.5   3.051 0.0218
#> Station1 - Station3    3.343 0.805 50.5   4.153 0.0008
#> Station1 - Station4    3.472 0.865 50.5   4.014 0.0012
#> Station2 - Station3    0.652 0.584 50.5   1.116 1.0000
#> Station2 - Station4    0.781 0.589 50.5   1.325 1.0000
#> Station3 - Station4    0.129 0.594 50.5   0.217 1.0000
#>

```

```
#> Results are averaged over the levels of: Season  
#> P value adjustment: bonferroni method for 6 tests
```

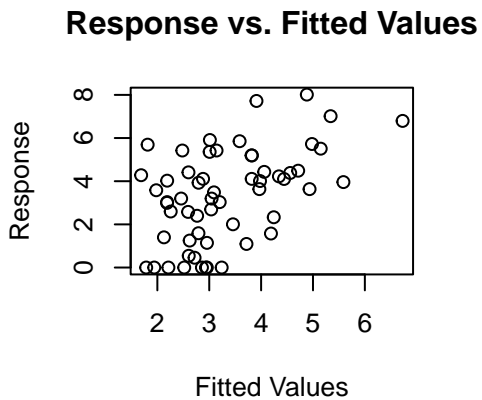
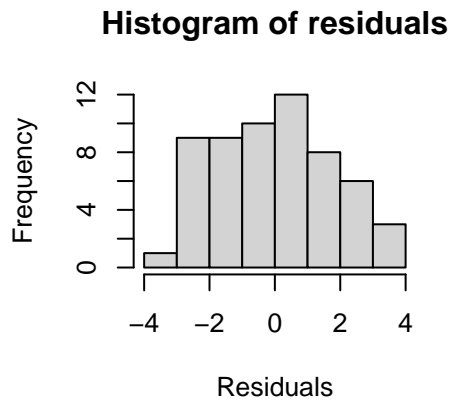
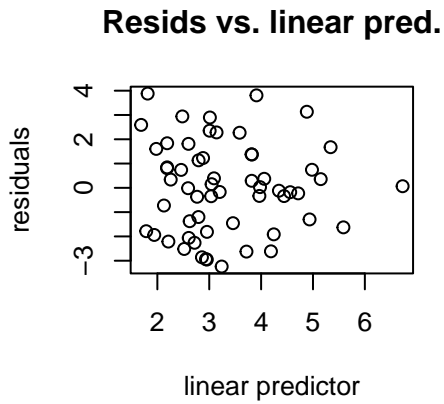
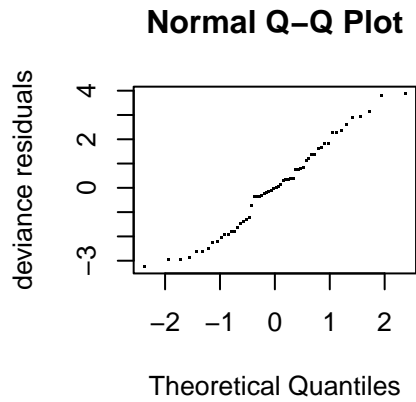
Plot GAM results

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



Model Diagnostics

```
oldpar <- par(mfrow = c(2,2))  
gam.check(herring_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 5.83e-09  0.87  0.115
#> s(Sal)     9.00e+00 9.51e-01  0.83  0.055 .
#> s(log(Turb)) 9.00e+00 5.55e-09  0.94  0.315
#> s(log(Chl))  9.00e+00 5.53e-09  1.16  0.840
#> s(log1p(combined_density)) 9.00e+00 5.95e-01  1.16  0.840
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
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 $\log(0) = -\text{Inf}$; $1/0 = \text{Inf}$, $1 / 0*0 = \text{Inf}$. The easiest solution is to add some finite small quantity to the density data, and predict that. Here we predict $\log(\text{Density} + 1)$ 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 must 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`. Low numbers constrain the GAM to fit smoother lines.

```
my_gamm <- function(.dat) {
  gamm(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))
#> Warning in lme.formula(y ~ X - 1, random = rand, data = strip.offset(mf), : nlminb problem, convergence
#> message = singular convergence (7)
```

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

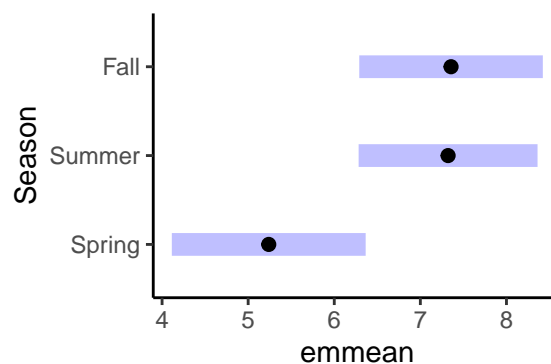
```
spp = 'Acartia'
mod <- spp_analysis$gam_mods[spp_analysis$Species == spp][[1]]
summary(mod$gam)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 4) +
#>      s(Sal, bs = "ts", k = 4) + s(log(Turb), bs = "ts", k = 4) +
#>      s(log(Chl), bs = "ts", k = 4) + s(log1p(RH), bs = "ts", k = 4)
#>
#> Parametric coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    5.6273      0.7413   7.591 7.78e-10 ***
#> Station2       -1.1691      0.5359  -2.182 0.033934 *
#> Station3       -0.5781      0.4975  -1.162 0.250803
#> Station4       -0.6917      0.5471  -1.264 0.212052
#> SeasonSummer    2.0831      0.5633   3.698 0.000546 ***
#> SeasonFall      2.1174      0.5695   3.718 0.000514 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>              edf Ref.df      F p-value
#> s(Temp)        5.216e-09      3 0.000 0.48620
#> s(Sal)          1.610e+00      3 4.780 0.00205 **
#> s(log(Turb))    8.159e-01      3 1.165 0.06015 .
#> s(log(Chl))     2.606e-01      3 0.156 0.22530
#> s(log1p(RH))    1.020e-08      3 0.000 0.76434
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.436
#>   Scale est. = 0.75339    n = 58
cat('\n')
anova(mod$gam)
#>
#> 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 1.764 0.166243
```

```
#> Season    2 8.361 0.000747
#>
#> Approximate significance of smooth terms:
#>          edf   Ref.df     F p-value
#> s(Temp)    5.216e-09 3.000e+00 0.000 0.48620
#> s(Sal)     1.610e+00 3.000e+00 4.780 0.00205
#> s(log(Turb)) 8.159e-01 3.000e+00 1.165 0.06015
#> s(log(Chl))  2.606e-01 3.000e+00 0.156 0.22530
#> s(log1p(RH)) 1.020e-08 3.000e+00 0.000 0.76434
```

Comparison Season and Station (Station is not significant)

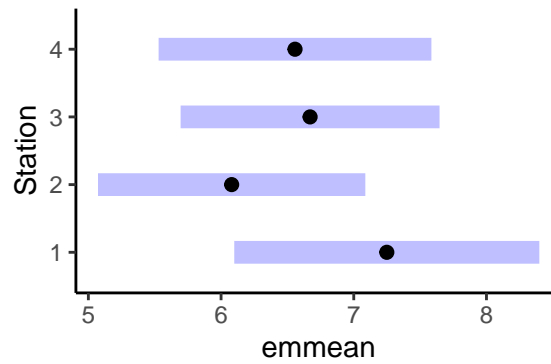
I'm showing “marginal” means – essentially means adjusted for the other predictors, at their mean values.

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



```
pairs(Seas_emms, adjust = 'bonferroni')
#> contrast      estimate    SE   df t.ratio p.value
#> Spring - Summer -2.0831 0.563 49.3 -3.698 0.0016
#> Spring - Fall   -2.1174 0.570 49.3 -3.718 0.0015
#> Summer - Fall   -0.0344 0.478 49.3 -0.072 1.0000
#>
#> Results are averaged over the levels of: Station
#> P value adjustment: bonferroni method for 3 tests
```

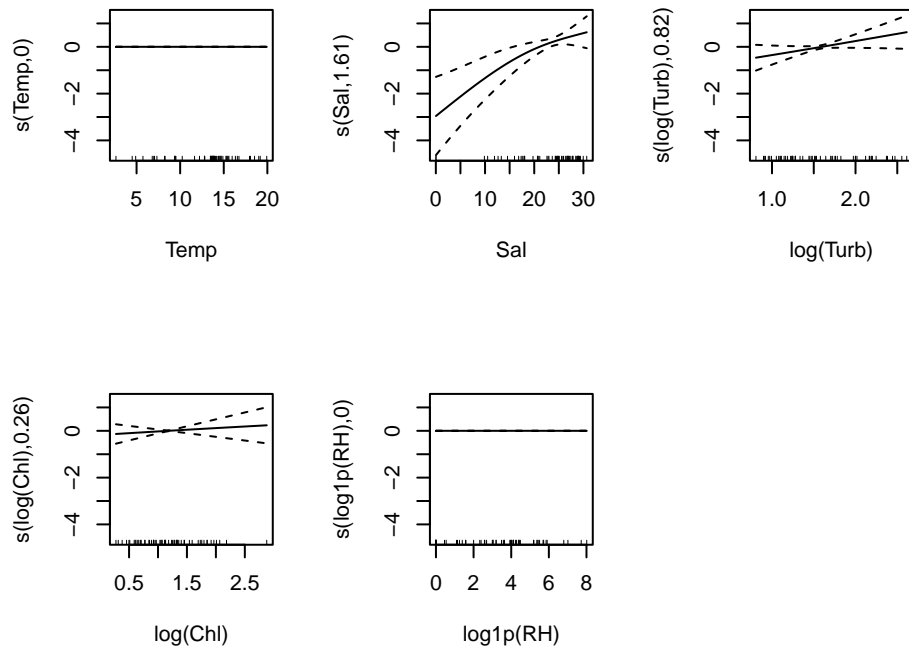
```
Sta_emms <- emmeans(mod, ~Station, type = 'response',
                    data = spp_analysis$data[spp_data$Species == spp][[1]])
plot(Sta_emms)
```



```
pairs(Sta_emms, adjust = 'bonferroni')
#> contrast      estimate    SE   df t.ratio p.value
#> Station1 - Station2    1.169 0.536 49.3   2.182  0.2036
#> Station1 - Station3    0.578 0.498 49.3   1.162  1.0000
#> Station1 - Station4    0.692 0.547 49.3   1.264  1.0000
#> Station2 - Station3   -0.591 0.370 49.3  -1.599  0.6975
#> Station2 - Station4   -0.477 0.390 49.3  -1.224  1.0000
#> Station3 - Station4    0.114 0.353 49.3    0.322  1.0000
#>
#> Results are averaged over the levels of: Season
#> P value adjustment: bonferroni method for 6 tests
```

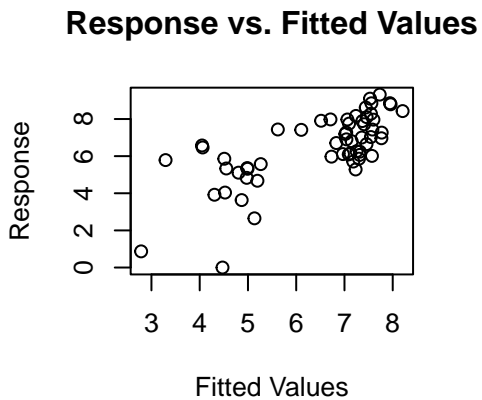
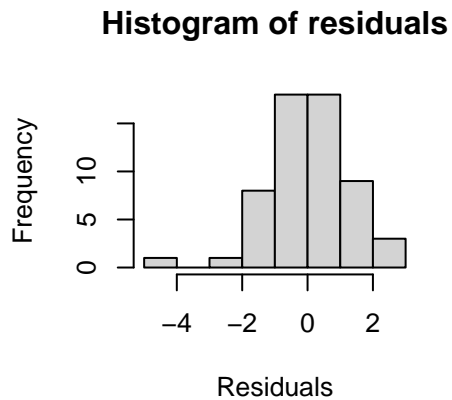
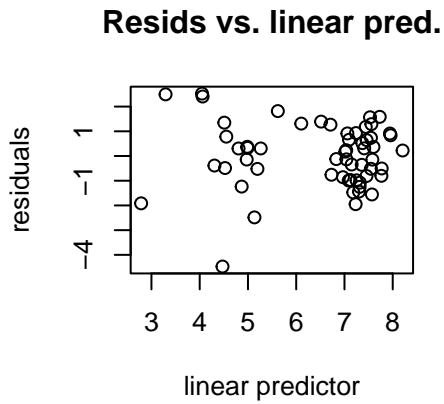
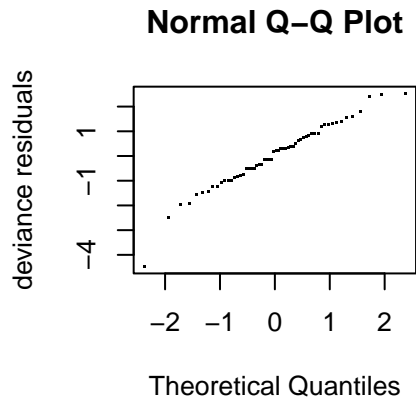
Plot GAM

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

Model Diagnostics

```
oldpar <- par(mfrow = c(2,2))
gam.check(mod$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)    3.00e+00 5.22e-09   1.03  0.630
#> s(Sal)     3.00e+00 1.61e+00   1.21  0.950
#> s(log(Turb)) 3.00e+00 8.16e-01   1.08  0.655
#> s(log(Chl))  3.00e+00 2.61e-01   0.83  0.075 .
#> s(log1p(RH)) 3.00e+00 1.02e-08   0.85  0.075 .
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
par(oldpar)
```

Balanus

Summary and ANOVA

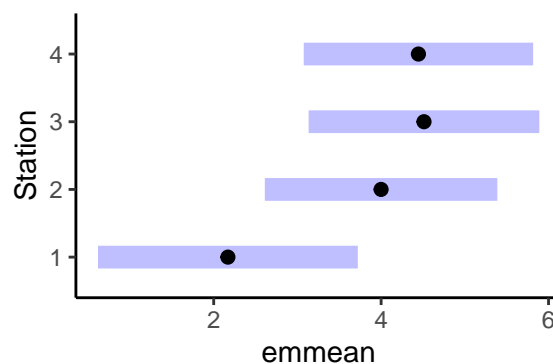
```
spp = 'Balanus'
mod <- spp_analysis$gam_mods[spp_analysis$Species == spp][[1]]
summary(mod$gam)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 4) +
#>      s(Sal, bs = "ts", k = 4) + s(log(Turb), bs = "ts", k = 4) +
#>      s(log(Chl), bs = "ts", k = 4) + s(log1p(RH), bs = "ts", k = 4)
#>
#> Parametric coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)   3.5909      1.0033   3.579 0.000778 ***
#> Station2      1.8299      0.6384   2.866 0.006062 **
#> Station3      2.3424      0.5804   4.036 0.000187 ***
#> Station4      2.2761      0.6246   3.644 0.000638 ***
#> SeasonSummer -2.0651      1.3804  -1.496 0.140942
#> SeasonFall    -2.8044      1.2945  -2.166 0.035080 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>              edf Ref.df      F p-value
#> s(Temp)       9.544e-01    3 1.428 0.04590 *
#> s(Sal)        5.554e-08    3 0.000 0.27144
#> s(log(Turb))  3.022e-10    3 0.000 0.69121
#> s(log(Chl))   1.033e+00    3 3.043 0.00628 **
#> s(log1p(RH))  1.159e-09    3 0.000 0.70543
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.205
#>   Scale est. = 1.1471    n = 58
cat('\n')
anova(mod$gam)
#>
#> 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 6.032 0.00137
```

```
#> Season    2 2.390 0.10205
#>
#> Approximate significance of smooth terms:
#>           edf    Ref.df      F p-value
#> s(Temp)      9.544e-01 3.000e+00 1.428 0.04590
#> s(Sal)       5.554e-08 3.000e+00 0.000 0.27144
#> s(log(Turb)) 3.022e-10 3.000e+00 0.000 0.69121
#> s(log(Chl))  1.033e+00 3.000e+00 3.043 0.00628
#> s(log1p(RH)) 1.159e-09 3.000e+00 0.000 0.70543
```

Comparison of Station and Season (Season is not significant)

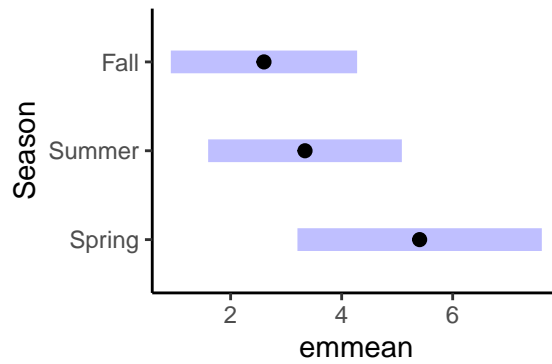
I'm showing “marginal” means – essentially means adjusted for the other predictors, at their mean values.

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



```
pairs(Sta_emms, adjust = 'bonferroni')
#> contrast      estimate    SE df t.ratio p.value
#> Station1 - Station2 -1.8299 0.638 50 -2.866 0.0364
#> Station1 - Station3 -2.3424 0.580 50 -4.036 0.0011
#> Station1 - Station4 -2.2761 0.625 50 -3.644 0.0038
#> Station2 - Station3 -0.5124 0.423 50 -1.213 1.0000
#> Station2 - Station4 -0.4461 0.432 50 -1.033 1.0000
#> Station3 - Station4  0.0663 0.440 50  0.151 1.0000
#>
#> Results are averaged over the levels of: Season
#> P value adjustment: bonferroni method for 6 tests
```

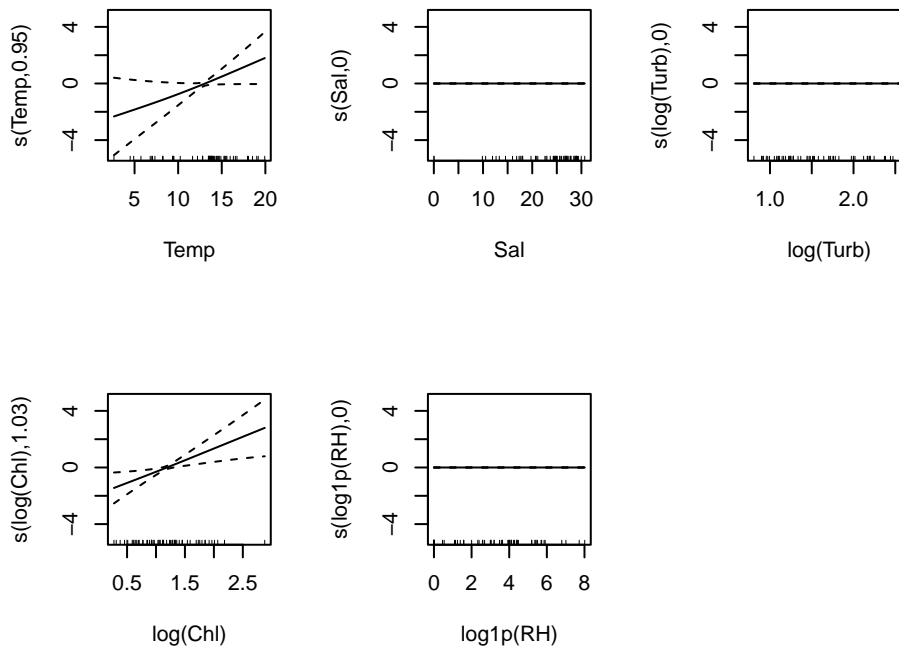
```
Seas_emms <- emmeans(mod, ~Season, type = 'response',
                    data = spp_analysis$data[spp_data$Species == spp][[1]])
plot(Seas_emms)
```



```
pairs(Seas_emms, adjust = 'bonferroni')
#> contrast      estimate    SE df t.ratio p.value
#> Spring - Summer    2.065 1.380 50   1.496  0.4228
#> Spring - Fall      2.804 1.295 50   2.166  0.1052
#> Summer - Fall      0.739 0.899 50   0.823  1.0000
#>
#> Results are averaged over the levels of: Station
#> P value adjustment: bonferroni method for 3 tests
```

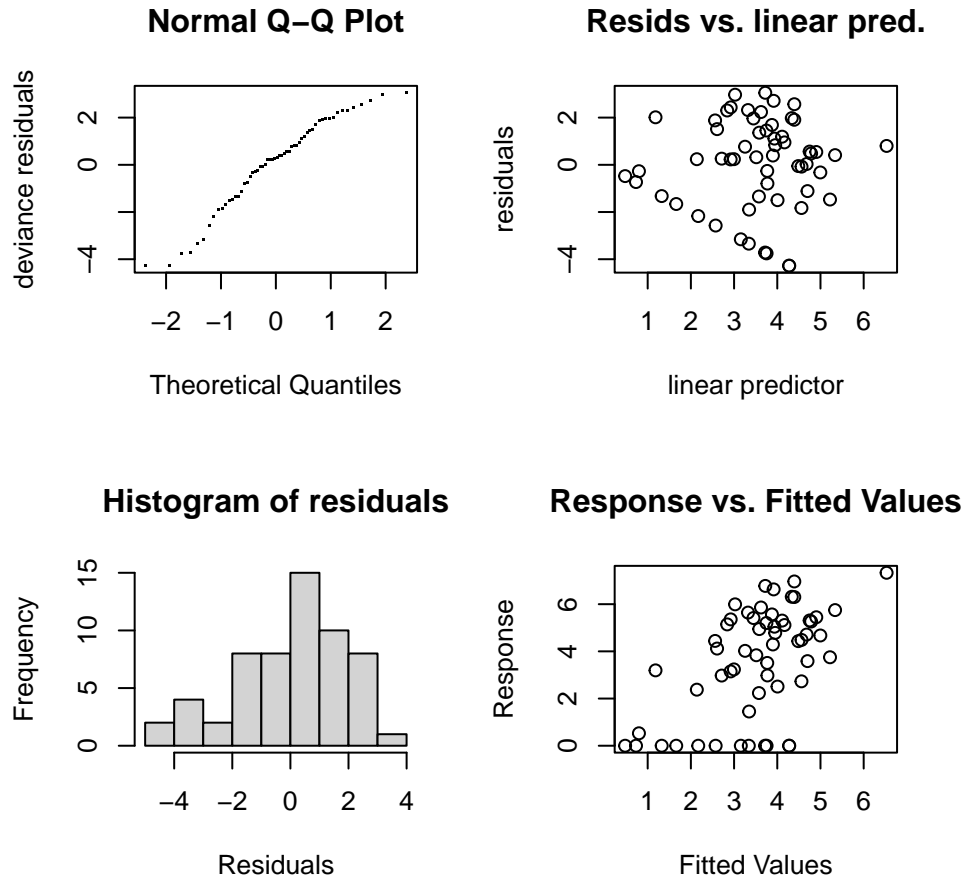
Plot GAM

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



Model Diagnostics

```
oldpar <- par(mfrow = c(2,2))
gam.check(mod$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)    3.00e+00 9.54e-01  0.87  0.18
#> s(Sal)     3.00e+00 5.55e-08  0.89  0.19
#> s(log(Turb)) 3.00e+00 3.02e-10  0.99  0.38
#> s(log(Chl))  3.00e+00 1.03e+00  1.10  0.72
#> s(log1p(RH)) 3.00e+00 1.16e-09  1.10  0.78
par(oldpar)
```

Eurytemora

Summary and ANOVA

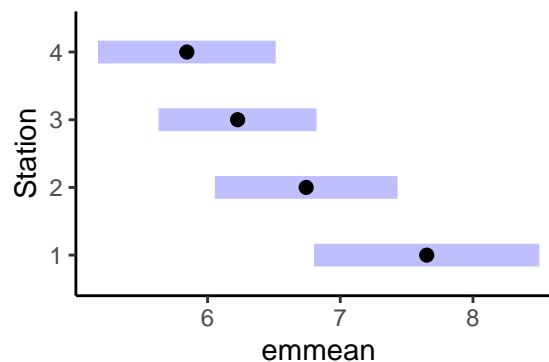
```
spp = "Eurytemora"
mod <- spp_analysis$gam_mods[spp_analysis$Species == spp][[1]]
summary(mod$gam)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 4) +
#>      s(Sal, bs = "ts", k = 4) + s(log(Turb), bs = "ts", k = 4) +
#>      s(log(Chl), bs = "ts", k = 4) + s(log1p(RH), bs = "ts", k = 4)
#>
#> Parametric coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    8.0329     0.5290  15.186 < 2e-16 ***
#> Station2       -0.9082     0.4638  -1.958  0.05599 .
#> Station3       -1.4252     0.4343  -3.281  0.00192 **
#> Station4       -1.8077     0.4651  -3.886  0.00031 ***
#> SeasonSummer  -0.8402     0.3983  -2.110  0.04009 *
#> SeasonFall    -0.8912     0.4046  -2.203  0.03240 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>              edf Ref.df      F p-value
#> s(Temp)        9.973e-09     3  0.000  0.2810
#> s(Sal)         2.550e+00     3 17.702 <2e-16 ***
#> s(log(Turb))   8.089e-01     3  1.190  0.0571 .
#> s(log(Chl))    3.536e-01     3  0.239  0.2012
#> s(log1p(RH))   1.442e-09     3  0.000  0.7533
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.45
#>   Scale est. = 0.59061    n = 58
cat('\n')
anova(mod$gam)
#>
#> 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 6.034 0.00142
```

```
#> Season    2 2.757 0.07351
#>
#> Approximate significance of smooth terms:
#>           edf    Ref.df      F p-value
#> s(Temp)      9.973e-09 3.000e+00  0.000 0.2810
#> s(Sal)       2.550e+00 3.000e+00 17.702 <2e-16
#> s(log(Turb)) 8.089e-01 3.000e+00  1.190 0.0571
#> s(log(Chl))  3.536e-01 3.000e+00  0.239 0.2012
#> s(log1p(RH)) 1.442e-09 3.000e+00  0.000 0.7533
```

Comparison of Station and Season (Season marginally significant)

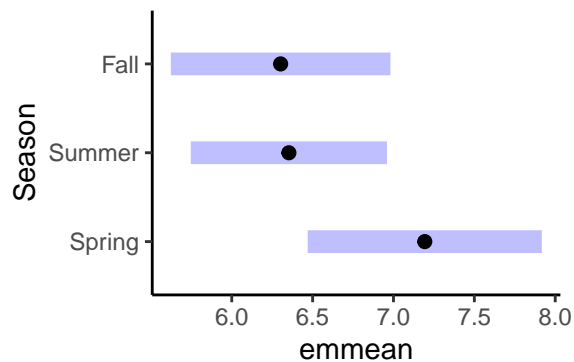
I'm showing “marginal” means – essentially means adjusted for the other predictors, at their mean values.

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



```
pairs(Sta_emms, adjust = 'bonferroni')
#> contrast      estimate    SE   df t.ratio p.value
#> Station1 - Station2    0.908 0.464 48.3   1.958 0.3360
#> Station1 - Station3    1.425 0.434 48.3   3.281 0.0115
#> Station1 - Station4    1.808 0.465 48.3   3.886 0.0019
#> Station2 - Station3    0.517 0.331 48.3   1.563 0.7481
#> Station2 - Station4    0.899 0.338 48.3   2.662 0.0632
#> Station3 - Station4    0.382 0.313 48.3   1.221 1.0000
#>
#> Results are averaged over the levels of: Season
#> P value adjustment: bonferroni method for 6 tests
```

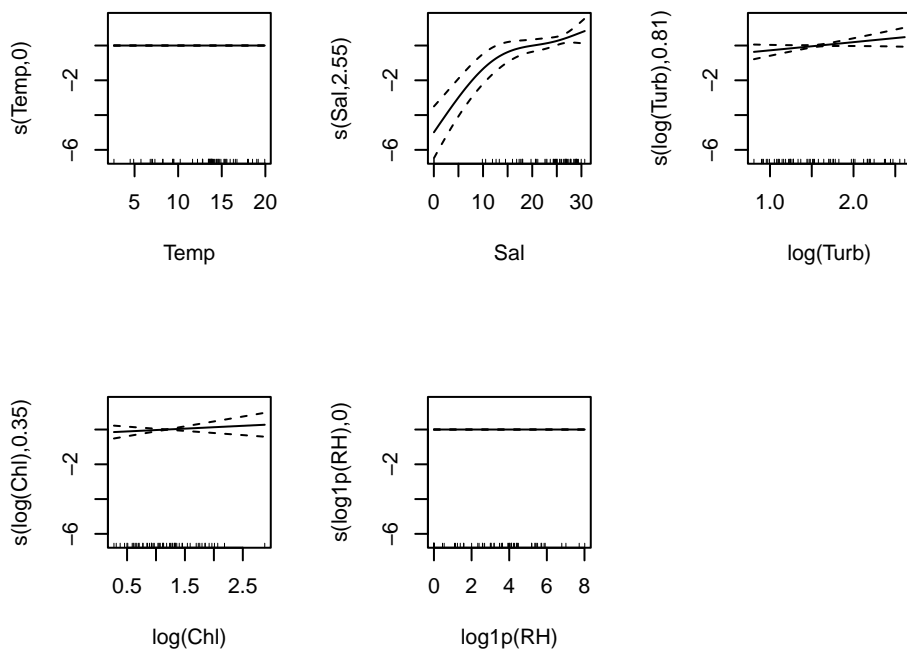
```
Seas_emms <- emmeans(mod, ~Season, type = 'response',
                    data = spp_analysis$data[spp_data$Species == spp][[1]])
plot(Seas_emms)
```

```
pairs(Seas_emms, adjust = 'bonferroni')
#> contrast      estimate    SE   df t.ratio p.value
#> Spring - Summer    0.840 0.398 48.3   2.110  0.1203
#> Spring - Fall      0.891 0.405 48.3   2.203  0.0972
#> Summer - Fall      0.051 0.316 48.3   0.162  1.0000
#>
#> Results are averaged over the levels of: Station
#> P value adjustment: bonferroni method for 3 tests
```

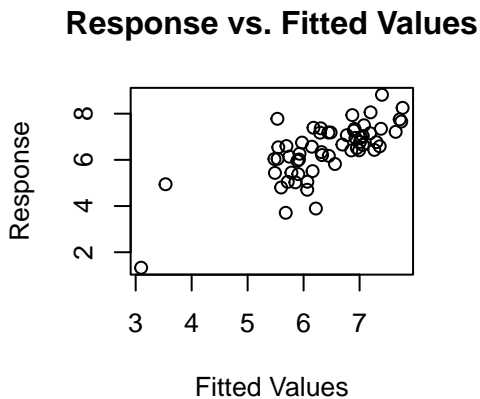
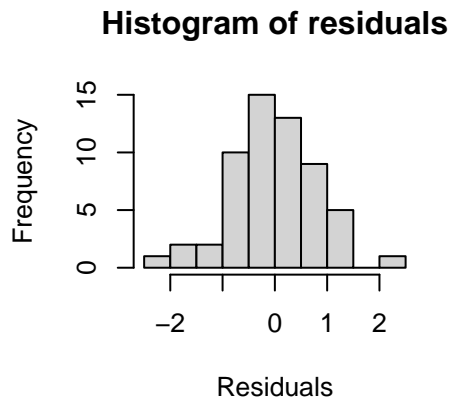
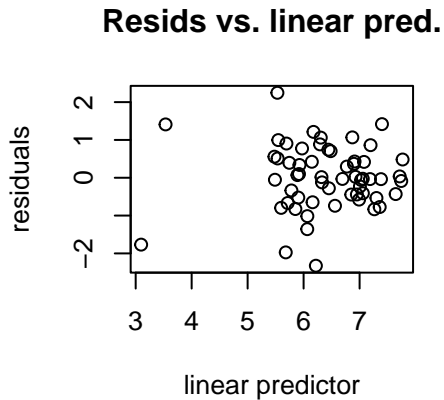
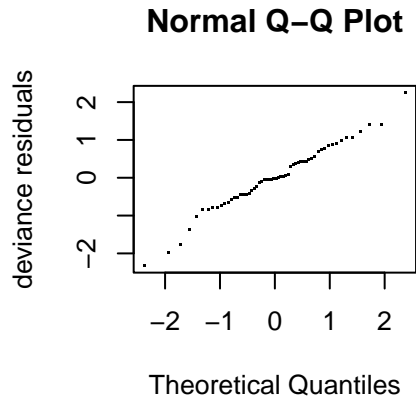
Plot GAM

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



Model Diagnostics

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

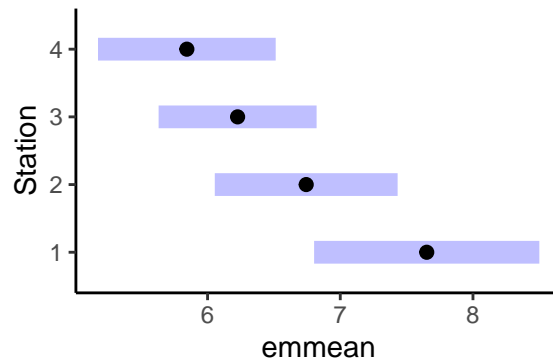


```
#>
#> 'gamm' based fit - care required with interpretation.
#> Checks based on working residuals may be misleading.
#> Basis dimension (k) checking results. Low p-value (k-index<1) may
#> indicate that k is too low, especially if edf is close to k'.
#>
#>           k'      edf k-index p-value
#> s(Temp)    3.00e+00 9.97e-09   1.06   0.68
#> s(Sal)     3.00e+00 2.55e+00   1.19   0.88
#> s(log(Turb)) 3.00e+00 8.09e-01   0.90   0.14
#> s(log(Chl))  3.00e+00 3.54e-01   1.04   0.63
#> s(log1p(RH)) 3.00e+00 1.44e-09   0.91   0.23
par(oldpar)
```

Comparison of Station and Season

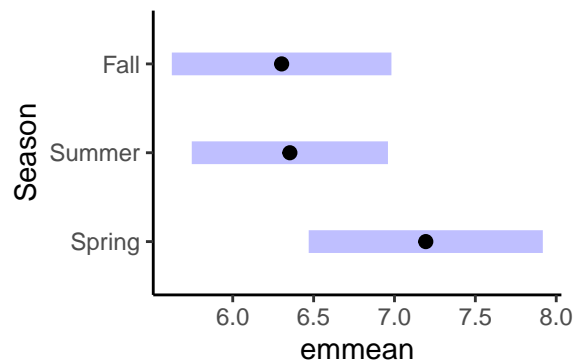
I'm showing “marginal” means – essentially means adjusted for the other predictors, at their mean values.

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



```
pairs(Sta_emms, adjust = 'bonferroni')  
#> contrast      estimate    SE   df t.ratio p.value  
#> Station1 - Station2    0.908 0.464 48.3   1.958 0.3360  
#> Station1 - Station3    1.425 0.434 48.3   3.281 0.0115  
#> Station1 - Station4    1.808 0.465 48.3   3.886 0.0019  
#> Station2 - Station3    0.517 0.331 48.3   1.563 0.7481  
#> Station2 - Station4    0.899 0.338 48.3   2.662 0.0632  
#> Station3 - Station4    0.382 0.313 48.3   1.221 1.0000  
#>  
#> Results are averaged over the levels of: Season  
#> 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')
#> contrast      estimate      SE    df t.ratio p.value
#> Spring - Summer    0.840 0.398 48.3    2.110 0.1203
#> Spring - Fall      0.891 0.405 48.3    2.203 0.0972
#> Summer - Fall      0.051 0.316 48.3    0.162 1.0000
#>
#> Results are averaged over the levels of: Station
#> P value adjustment: bonferroni method for 3 tests

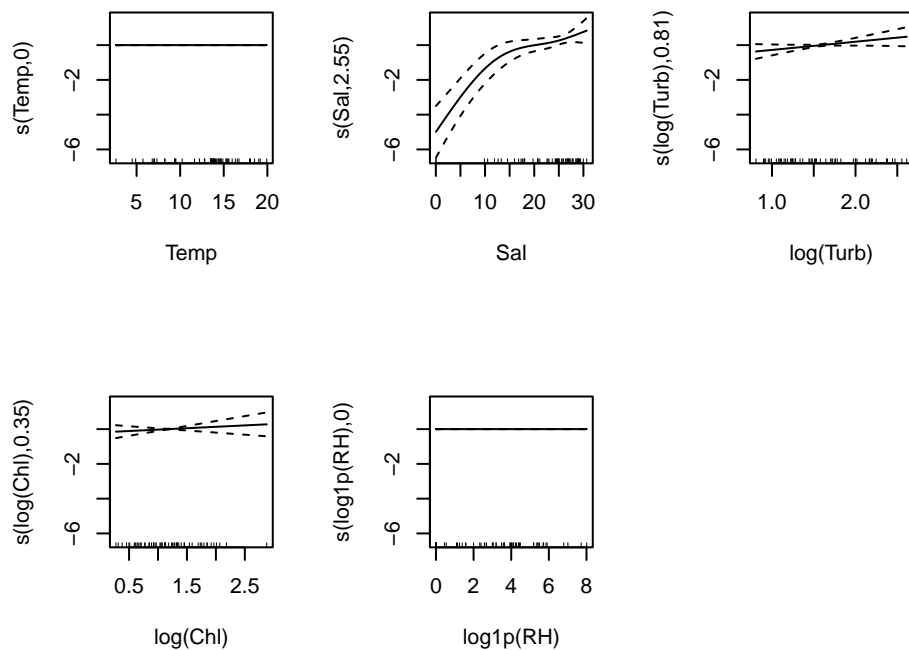
```

Plot GAM

```

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

```

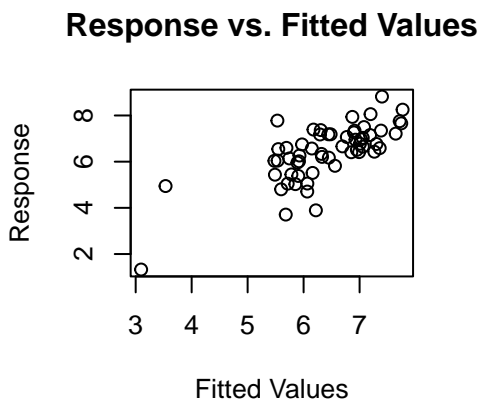
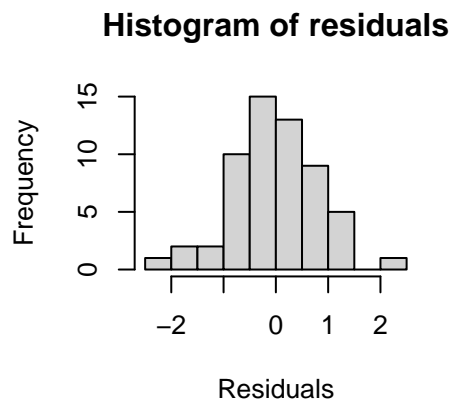
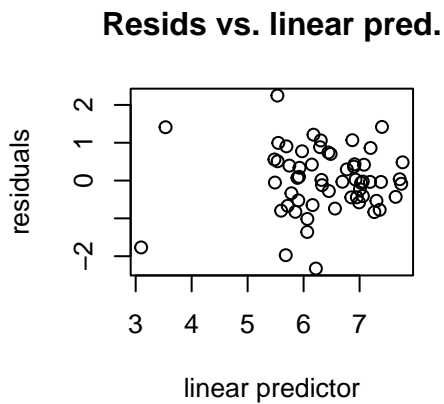
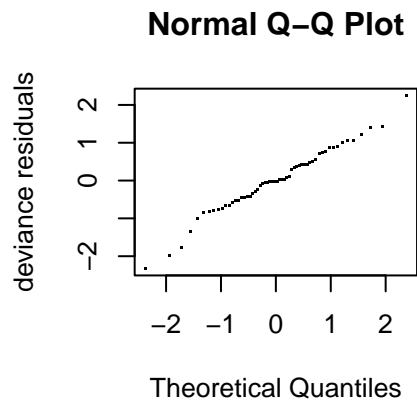


Model Diagnostics

```

oldpar <- par(mfrow = c(2,2))
gam.check(mod$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)    3.00e+00 9.97e-09  1.06  0.64
#> s(Sal)     3.00e+00 2.55e+00  1.19  0.92
#> s(log(Turb)) 3.00e+00 8.09e-01  0.90  0.16
#> s(log(Chl))  3.00e+00 3.54e-01  1.04  0.56
#> s(log1p(RH)) 3.00e+00 1.44e-09  0.91  0.17
par(oldpar)
```

Polychaete

Warning: Model did not converge normally! ### Summary and ANOVA

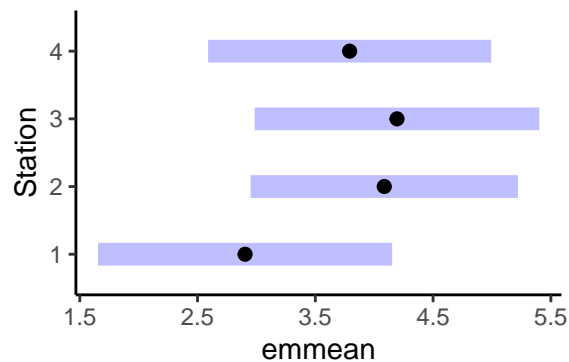
```
spp = "Polychaete"
mod <- spp_analysis$gam_mods[spp_analysis$Species == spp][[1]]
summary(mod$gam)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 4) +
#>   s(Sal, bs = "ts", k = 4) + s(log(Turb), bs = "ts", k = 4) +
#>   s(log(Chl), bs = "ts", k = 4) + s(log1p(RH), bs = "ts", k = 4)
#>
#> Parametric coefficients:
#>               Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    5.0201     0.9813   5.116 4.94e-06 ***
#> Station2        1.1820     0.5206   2.270 0.027511 *
#> Station3        1.2899     0.5172   2.494 0.015980 *
#> Station4         0.8877     0.5779   1.536 0.130755
#> SeasonSummer   -4.6656     1.2350  -3.778 0.000421 ***
#> SeasonFall     -3.3783     1.1877  -2.844 0.006422 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>               edf Ref.df    F p-value
#> s(Temp)        1.555e-10     3 0.000 0.58683
#> s(Sal)          2.065e-10     3 0.000 0.80810
#> s(log(Turb))    2.982e-07     3 0.000 0.27491
#> s(log(Chl))     1.087e+00     3 2.801 0.00974 **
#> s(log1p(RH))    7.273e-01     3 0.824 0.09951 .
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.346
#>   Scale est. = 1.4196    n = 58
cat('\n')
anova(mod$gam)
#>
#> 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 2.461 0.07330
#> Season       2 7.506 0.00141
```

```
#>
#> Approximate significance of smooth terms:
#>           edf    Ref.df      F p-value
#> s(Temp)      1.555e-10 3.000e+00 0.000 0.58683
#> s(Sal)       2.065e-10 3.000e+00 0.000 0.80810
#> s(log(Turb)) 2.982e-07 3.000e+00 0.000 0.27491
#> s(log(Chl))  1.087e+00 3.000e+00 2.801 0.00974
#> s(log1p(RH)) 7.273e-01 3.000e+00 0.824 0.09951
```

Comparison of Station and Season

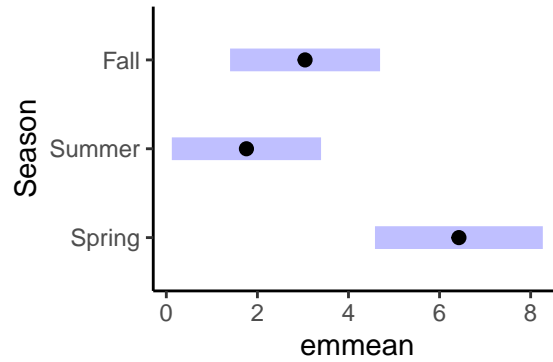
I'm showing “marginal” means – essentially means adjusted for the other predictors, at their mean values.

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



```
pairs(Sta_emms, adjust = 'bonferroni')
#> contrast      estimate    SE    df t.ratio p.value
#> Station1 - Station2   -1.182 0.521 50.2  -2.270 0.1651
#> Station1 - Station3   -1.290 0.517 50.2  -2.494 0.0959
#> Station1 - Station4   -0.888 0.578 50.2  -1.536 0.7845
#> Station2 - Station3   -0.108 0.472 50.2  -0.228 1.0000
#> Station2 - Station4    0.294 0.483 50.2   0.609 1.0000
#> Station3 - Station4    0.402 0.490 50.2   0.821 1.0000
#>
#> Results are averaged over the levels of: Season
#> P value adjustment: bonferroni method for 6 tests
```

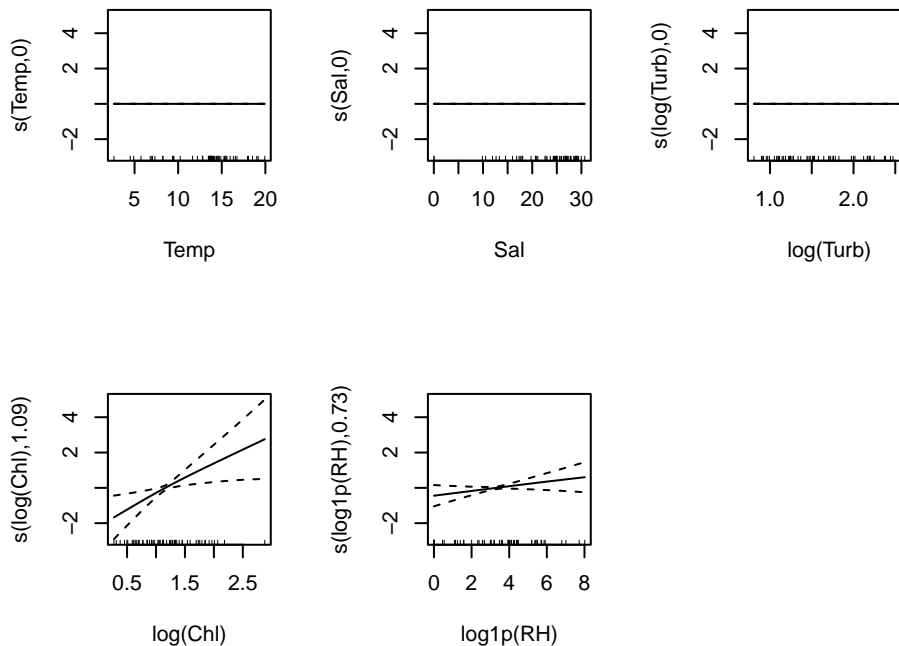
```
Seas_emms <- emmeans(mod, ~Season, type = 'response',
                    data = spp_analysis$data[spp_data$Species == spp][[1]])
plot(Seas_emms)
```



```
pairs(Seas_emms, adjust = 'bonferroni')
#> contrast      estimate    SE   df t.ratio p.value
#> Spring - Summer    4.67 1.24 50.2   3.778  0.0013
#> Spring - Fall      3.38 1.19 50.2   2.844  0.0193
#> Summer - Fall     -1.29 1.13 50.2  -1.138  0.7811
#>
#> Results are averaged over the levels of: Station
#> P value adjustment: bonferroni method for 3 tests
```

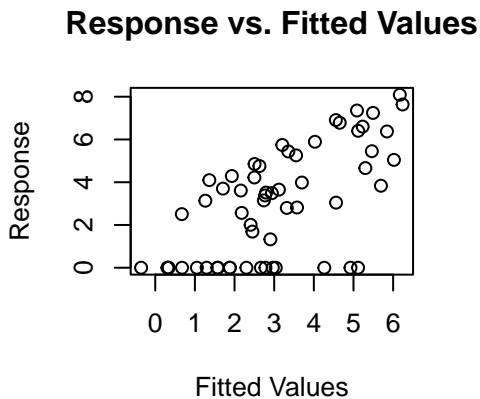
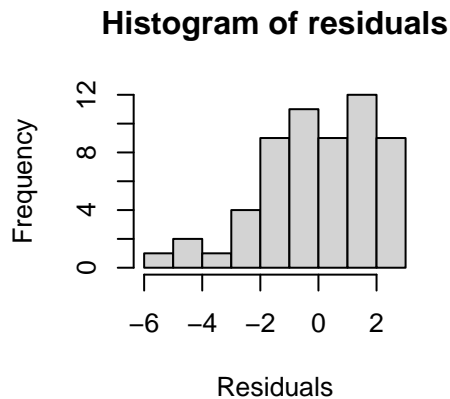
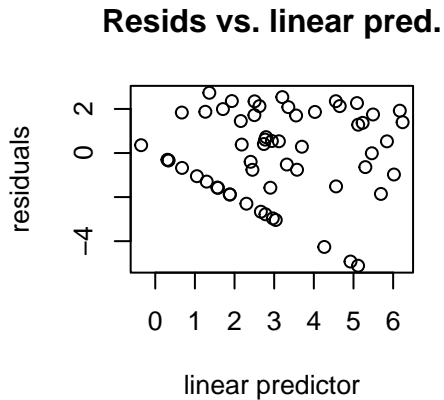
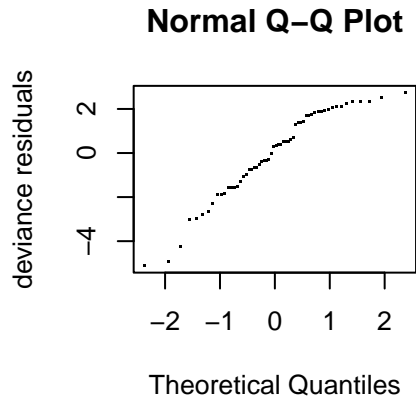
Plot GAM

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



Model Diagnostics

```
oldpar <- par(mfrow = c(2,2))
gam.check(mod$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)    3.00e+00 1.55e-10   0.91   0.22
#> s(Sal)     3.00e+00 2.06e-10   0.98   0.40
#> s(log(Turb)) 3.00e+00 2.98e-07   0.78   0.05 *
#> s(log(Chl))  3.00e+00 1.09e+00   0.92   0.20
#> s(log1p(RH)) 3.00e+00 7.27e-01   1.17   0.90
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
par(oldpar)
```

Pseudocal

Summary and ANOVA

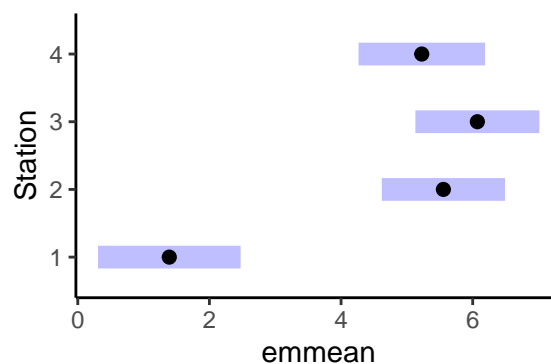
```
spp = "Pseudocal"
mod <- spp_analysis$gam_mods[spp_analysis$Species == spp][[1]]
summary(mod$gam)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 4) +
#>   s(Sal, bs = "ts", k = 4) + s(log(Turb), bs = "ts", k = 4) +
#>   s(log(Chl), bs = "ts", k = 4) + s(log1p(RH), bs = "ts", k = 4)
#>
#> Parametric coefficients:
#>               Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    2.2742     0.6585   3.453  0.00113 **
#> Station2       4.1643     0.6266   6.646 2.07e-08 ***
#> Station3       4.6803     0.5858   7.989 1.62e-10 ***
#> Station4       3.8355     0.6071   6.318 6.78e-08 ***
#> SeasonSummer  -0.8905     1.0800  -0.825  0.41352
#> SeasonFall    -2.2906     1.0411  -2.200  0.03240 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>               edf Ref.df      F p-value
#> s(Temp)        1.074e+00    3 1.664  0.0407 *
#> s(Sal)          1.580e-08    3 0.000  0.1832
#> s(log(Turb))    4.216e-10    3 0.000  0.7530
#> s(log(Chl))     2.615e-10    3 0.000  0.8183
#> s(log1p(RH))    4.782e-01    3 0.313  0.2122
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.562
#>   Scale est. = 1.2566    n = 58
cat('\n')
anova(mod$gam)
#>
#> 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 21.743 3.56e-09
```

```
#> Season    2  3.726    0.031
#>
#> Approximate significance of smooth terms:
#>           edf    Ref.df      F p-value
#> s(Temp)      1.074e+00 3.000e+00 1.664 0.0407
#> s(Sal)       1.580e-08 3.000e+00 0.000 0.1832
#> s(log(Turb)) 4.216e-10 3.000e+00 0.000 0.7530
#> s(log(Chl))  2.615e-10 3.000e+00 0.000 0.8183
#> s(log1p(RH)) 4.782e-01 3.000e+00 0.313 0.2122
```

Comparison of Station and Season

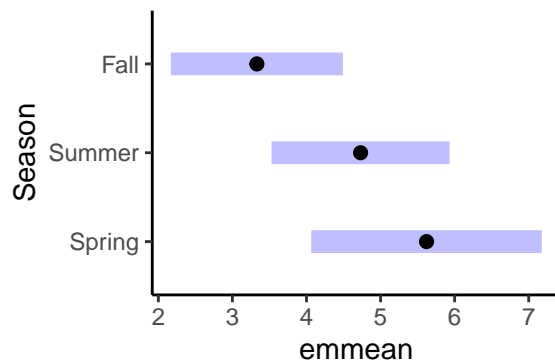
I'm showing “marginal” means – essentially means adjusted for the other predictors, at their mean values.

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



```
pairs(Sta_emms, adjust = 'bonferroni')
#> contrast      estimate    SE   df t.ratio p.value
#> Station1 - Station2 -4.164 0.627 50.5 -6.646 <.0001
#> Station1 - Station3 -4.680 0.586 50.5 -7.989 <.0001
#> Station1 - Station4 -3.836 0.607 50.5 -6.318 <.0001
#> Station2 - Station3 -0.516 0.443 50.5 -1.165 1.0000
#> Station2 - Station4  0.329 0.450 50.5  0.731 1.0000
#> Station3 - Station4  0.845 0.443 50.5  1.908 0.3729
#>
#> Results are averaged over the levels of: Season
#> P value adjustment: bonferroni method for 6 tests
```

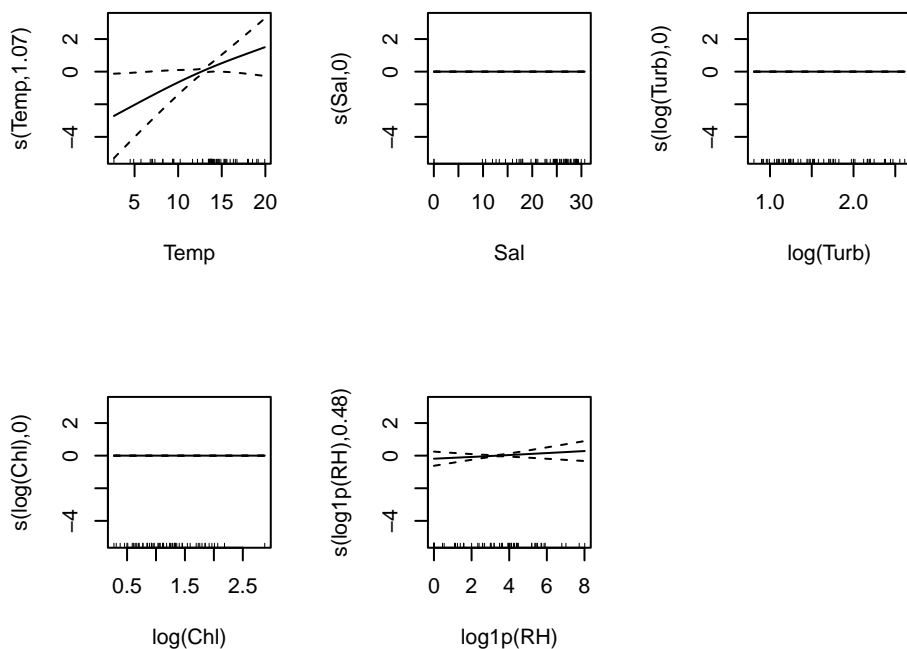
```
Seas_emms <- emmeans(mod, ~Season, type = 'response',
                    data = spp_analysis$data[spp_data$Species == spp][[1]])
plot(Seas_emms)
```



```
pairs(Seas_emms, adjust = 'bonferroni')
#> contrast      estimate    SE   df t.ratio p.value
#> Spring - Summer    0.891 1.080 50.5   0.825  1.0000
#> Spring - Fall      2.291 1.041 50.5   2.200  0.0972
#> Summer - Fall      1.400 0.659 50.5   2.125  0.1154
#>
#> Results are averaged over the levels of: Station
#> P value adjustment: bonferroni method for 3 tests
```

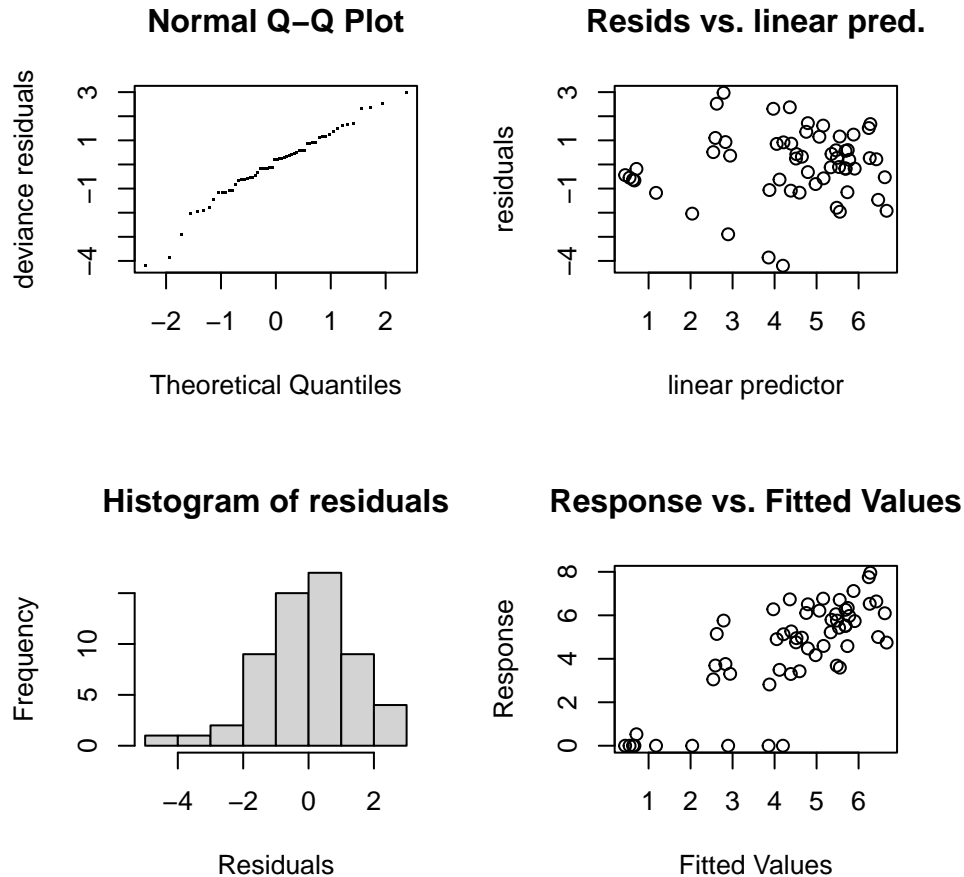
Plot GAM

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



Model Diagnostics

```
oldpar <- par(mfrow = c(2,2))
gam.check(mod$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)    3.00e+00 1.07e+00  0.72  0.01 **
#> s(Sal)      3.00e+00 1.58e-08  0.86  0.10
#> s(log(Turb)) 3.00e+00 4.22e-10  1.03  0.52
#> s(log(Chl))  3.00e+00 2.61e-10  1.29  0.99
#> s(log1p(RH)) 3.00e+00 4.78e-01  1.21  0.92
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
par(oldpar)
```

Temora

Summary and ANOVA

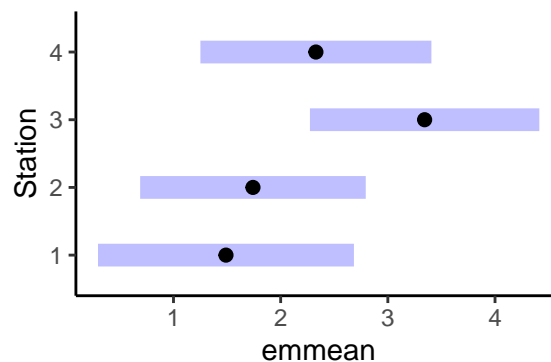
```
spp = "Temora"
mod <- spp_analysis$gam_mods[spp_analysis$Species == spp][[1]]
summary(mod$gam)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 4) +
#>      s(Sal, bs = "ts", k = 4) + s(log(Turb), bs = "ts", k = 4) +
#>      s(log(Chl), bs = "ts", k = 4) + s(log1p(RH), bs = "ts", k = 4)
#>
#> Parametric coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)   1.5639      0.7733   2.022  0.04838 *
#> Station2      0.2519      0.6882   0.366  0.71588
#> Station3      1.8536      0.6736   2.752  0.00819 **
#> Station4      0.8392      0.7228   1.161  0.25102
#> SeasonSummer -0.7134      0.7446  -0.958  0.34255
#> SeasonFall   -0.2892      0.6722  -0.430  0.66884
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>              edf Ref.df      F p-value
#> s(Temp)       3.079e-09      3 0.000 0.67355
#> s(Sal)        2.697e-08      3 0.000 0.40191
#> s(log(Turb))  3.931e-08      3 0.000 0.38103
#> s(log(Chl))   9.754e-01      3 2.997 0.00651 **
#> s(log1p(RH))  9.598e-09      3 0.000 0.73095
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.211
#>   Scale est. = 2.8181    n = 58
cat('\n')
anova(mod$gam)
#>
#> 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 3.136  0.0333
```

```
#> Season    2 0.501  0.6089
#>
#> Approximate significance of smooth terms:
#>           edf    Ref.df      F p-value
#> s(Temp)      3.079e-09 3.000e+00 0.000 0.67355
#> s(Sal)        2.697e-08 3.000e+00 0.000 0.40191
#> s(log(Turb))  3.931e-08 3.000e+00 0.000 0.38103
#> s(log(Chl))   9.754e-01 3.000e+00 2.997 0.00651
#> s(log1p(RH))  9.598e-09 3.000e+00 0.000 0.73095
```

Comparison of Station and Season

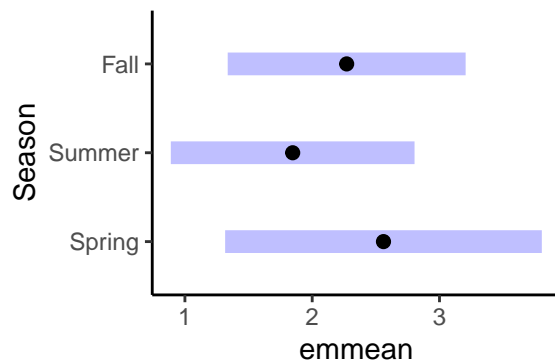
I'm showing “marginal” means – essentially means adjusted for the other predictors, at their mean values.

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



```
pairs(Sta_emms, adjust = 'bonferroni')
#> contrast      estimate    SE df t.ratio p.value
#> Station1 - Station2   -0.252 0.688 51  -0.366  1.0000
#> Station1 - Station3   -1.854 0.674 51  -2.752  0.0491
#> Station1 - Station4   -0.839 0.723 51  -1.161  1.0000
#> Station2 - Station3   -1.602 0.651 51  -2.462  0.1034
#> Station2 - Station4   -0.587 0.664 51  -0.885  1.0000
#> Station3 - Station4    1.014 0.674 51   1.506  0.8295
#>
#> Results are averaged over the levels of: Season
#> P value adjustment: bonferroni method for 6 tests
```

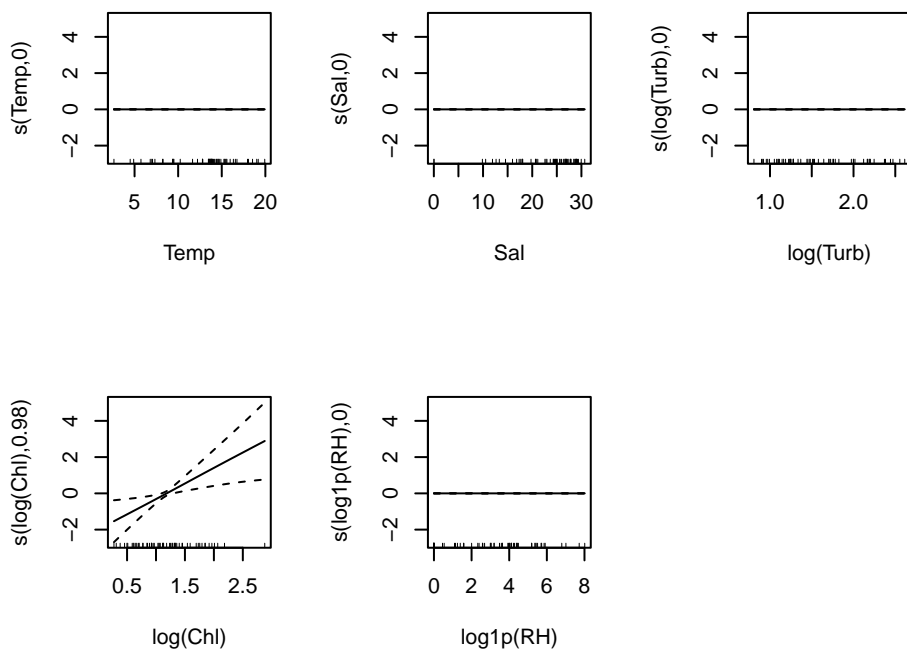
```
Seas_emms <- emmeans(mod, ~Season, type = 'response',
                    data = spp_analysis$data[spp_data$Species == spp][[1]])
plot(Seas_emms)
```



```
pairs(Seas_emms, adjust = 'bonferroni')
#> contrast      estimate    SE df t.ratio p.value
#> Spring - Summer    0.713 0.745 51   0.958  1.0000
#> Spring - Fall      0.289 0.672 51   0.430  1.0000
#> Summer - Fall     -0.424 0.577 51  -0.735  1.0000
#>
#> Results are averaged over the levels of: Station
#> P value adjustment: bonferroni method for 3 tests
```

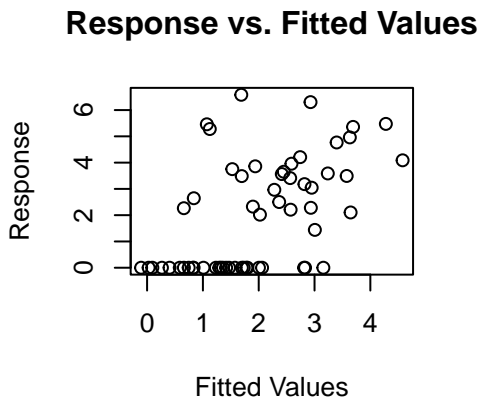
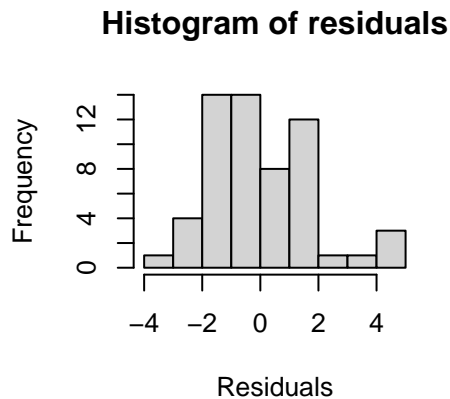
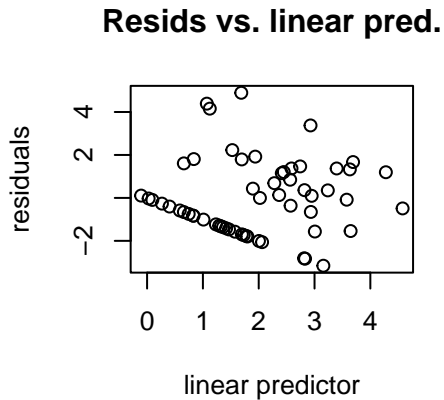
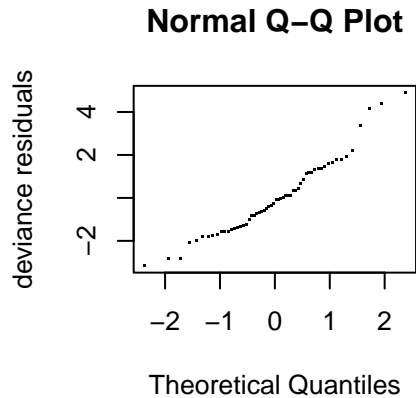
Plot GAM

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



Model Diagnostics

```
oldpar <- par(mfrow = c(2,2))
gam.check(mod$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)    3.00e+00 3.08e-09   1.08   0.68
#> s(Sal)     3.00e+00 2.70e-08   1.13   0.81
#> s(log(Turb)) 3.00e+00 3.93e-08   1.10   0.74
#> s(log(Chl))  3.00e+00 9.75e-01   0.97   0.38
#> s(log1p(RH)) 3.00e+00 9.60e-09   1.25   0.94
par(oldpar)
```

Why Did the Polychaete Model fail to Converge?

And does it matter?

I run the same analysis again...

```
p_data <- spp_data %>%  
  filter(Species == 'Polychaete')
```

```
mod_1 <- gamm(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 = p_data, family = "gaussian")  
#> Warning in lme.formula(y ~ X - 1, random = rand, data = strip.offset(mf), : nlminb problem, convergence  
#> message = singular convergence (7)
```

```
summary(mod_1$gam)  
#>  
#> 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)    5.0201      0.9813   5.116 4.94e-06 ***  
#> Station2       1.1820      0.5206   2.270 0.027511 *  
#> Station3       1.2899      0.5172   2.494 0.015980 *  
#> Station4       0.8877      0.5779   1.536 0.130755  
#> SeasonSummer  -4.6656      1.2350  -3.778 0.000421 ***  
#> SeasonFall    -3.3783      1.1877  -2.844 0.006422 **  
#> ---  
#> 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)        1.555e-10      3 0.000 0.58683  
#> s(Sal)         2.065e-10      3 0.000 0.80810  
#> s(log(Turb))   2.982e-07      3 0.000 0.27491  
#> s(log(Chl))    1.087e+00      3 2.801 0.00974 **  
#> s(log1p(RH))   7.273e-01      3 0.824 0.09951 .  
#> ---  
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
#>
```

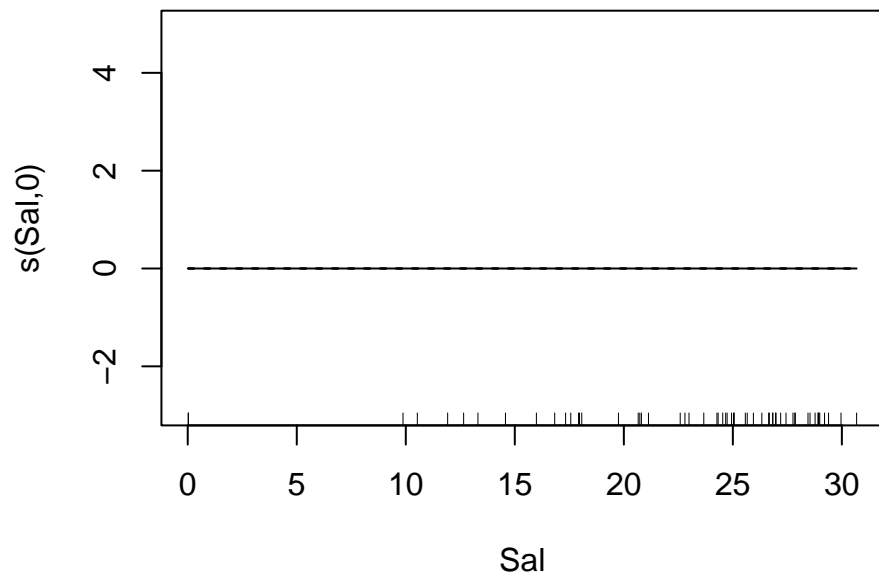
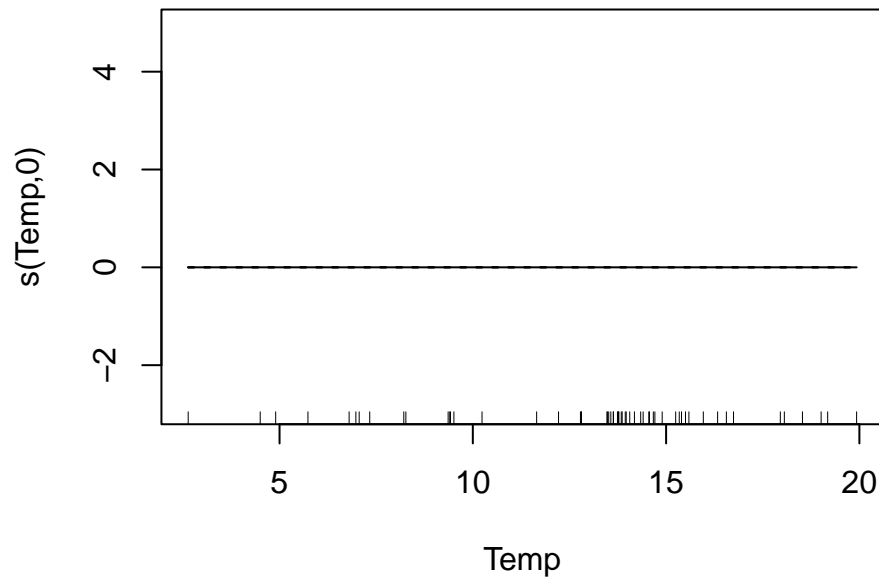
```
#> R-sq.(adj) = 0.346
#> Scale est. = 1.4196    n = 58
```

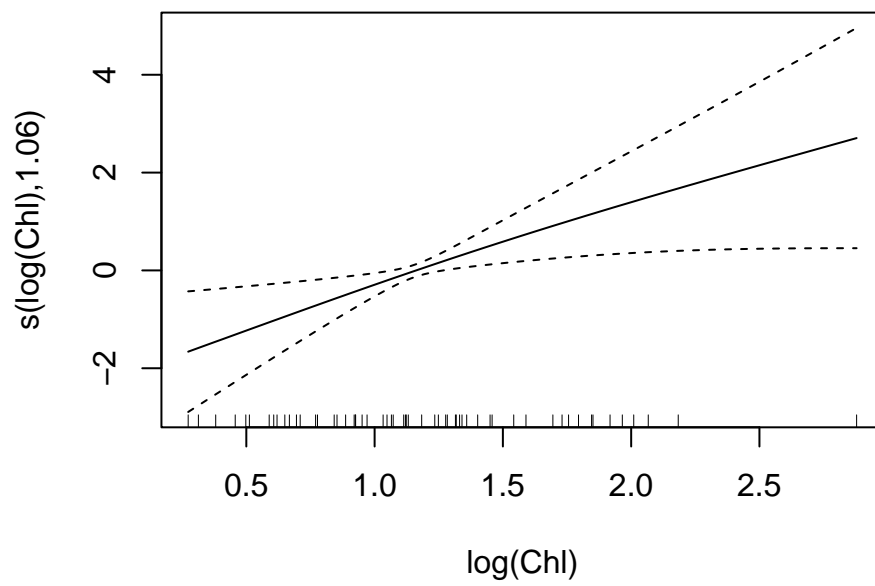
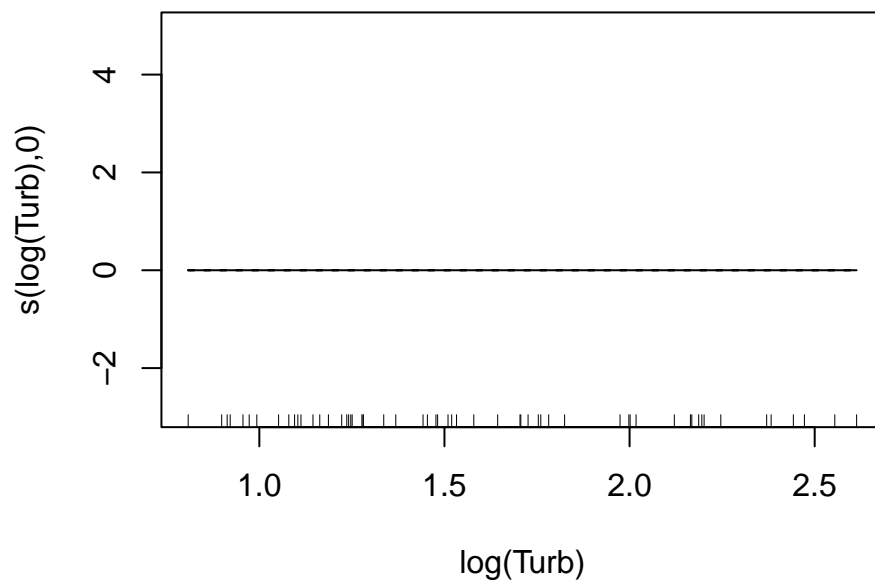
Removing ANY of the smoothers or EITHER of the random effects makes the warning disappear. Even reducing the degree of smoothing in the smoothed terms removes the problem – in spite of the fact that the actual fitted relationships are linear. Changing the degree of smoothing should have minimal impact on the model, since the smoothed terms were all reduced to nearly linear relationships. I run a slightly modified model to compare results.

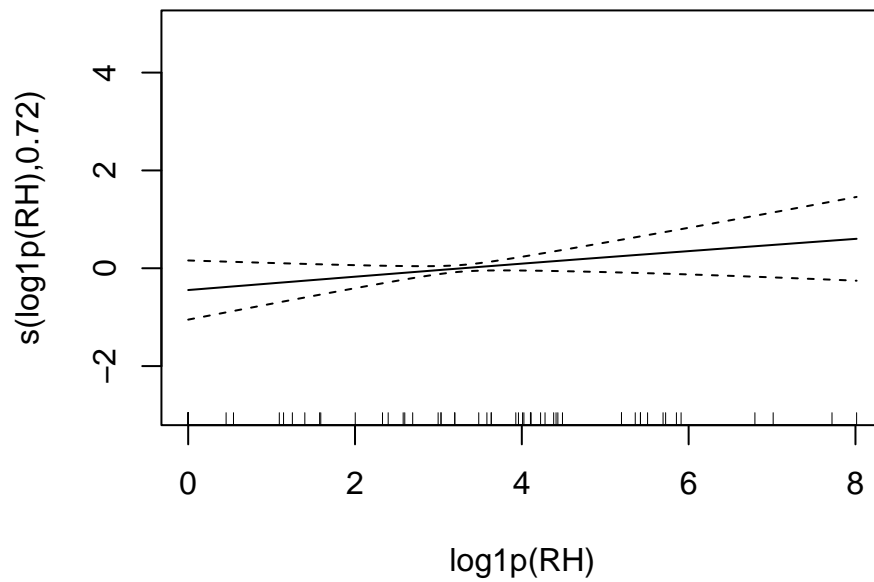
```
mod_2 <- gamm(log1p(Density) ~ Station +
  Season +
  s(Temp, bs="ts", k = 3) +
  s(Sal, bs="ts", k = 3) +
  s(log(Turb), bs="ts", k = 3) +
  s(log(Chl), bs="ts", k = 3) +
  s(log1p(RH), bs="ts", k = 3),
  random = list(Yearf = ~ 1, sample_event = ~ 1),
  data = p_data, family = "gaussian")
summary(mod_2$gam)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Density) ~ Station + Season + s(Temp, bs = "ts", k = 3) +
#>      s(Sal, bs = "ts", k = 3) + s(log(Turb), bs = "ts", k = 3) +
#>      s(log(Chl), bs = "ts", k = 3) + s(log1p(RH), bs = "ts", k = 3)
#>
#> Parametric coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    5.0197      0.9812   5.116 4.93e-06 ***
#> Station2        1.1819      0.5206   2.270 0.027528 *
#> Station3        1.2882      0.5176   2.489 0.016167 *
#> Station4        0.8911      0.5779   1.542 0.129397
#> SeasonSummer   -4.6655      1.2347  -3.779 0.000419 ***
#> SeasonFall     -3.3782      1.1881  -2.843 0.006439 **
#> ---
#> 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)        7.777e-11     2 0.000 0.5669
#> s(Sal)         5.749e-11     2 0.000 0.7981
#> s(log(Turb))   1.211e-06     2 0.000 0.2658
#> s(log(Chl))    1.055e+00     2 4.155 0.0097 **
#> s(log1p(RH))   7.230e-01     2 1.225 0.1005
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) = 0.346
#> Scale est. = 1.4219    n = 58
```

Parameter values are largely unchanged with this slightly simpler model, and we no longer get the warning.

```
plot(mod_2$gam)
```







And the graphic results are identical.

That suggests the problem is that the model has simply gotten too complex for the limited amount of data. Effectively, we ran out of degrees of freedom. Here we have only a single missing value, but a large number of zero values, which could create a singular fit.

```
sum(is.na(p_data$Density))
#> [1] 1
sum(p_data$Density == 0)
#> [1] NA
```