# GAM Analysis of Data From Penobscot Plankton Study, Including Year

Curtis C. Bohlen, Casco Bay Estuary Partnership

1/28/2022

## Contents

# Introduction

This notebook is a follow up on my preliminary effort to explore changes in data analysis. I inadvertently forgot to include Year as a predictor variable in the GAM models in my previous analysis.

What I've done here is SLIGHTLY modify the analysis to add Year as a predictor.

As before, this Notebook looks at:

1. Non-linear fits between zooplankton community metrics and possible environmental drivers, and
2. Examination of responses of one individual species to those same drivers.

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

Adding Year results in some changes in conclusions when compared to analyses that leave Year out of the models. This is not entirely unexpected. The problem is that Years vary both in plankton composition and abundance and in various predictors. For example, some years are warmer than others. So, if you fit both "year" and "Temperature" in a model, the model ends up partitioning variation between the two predictors.

This highlights some of the challenges the reviewers raised. Throwing all the predictors at the problem can actually obscure, rather than clarify relationships. But I'm not sure what a better approach looks like. . . .

# Load Libraries

```
library(tidyverse)
library(readxl)
library(mgcv)      # for GAM models
library(emmeans)   # For extracting useful "marginal" model summaries

theme_set(theme_classic())
```

# 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
#>   <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.

```
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)
```

## "Long" Version of the Data

Here I only use this for graphic review or how predictors vary by year.

```
long_data <- base_data %>%
  select(-c(H:Temora,  AvgTurb, DOsat)) %>%
  pivot_longer(-c(Date:riv_km, combined_density),
               names_to = 'Predictor', values_to = 'Values') %>%
  mutate(Predictor = factor(Predictor,
                            levels = c('Temp', 'Sal', 'Turb',
```

```
                                                'Chl', 'RH'),
                                labels = c("Temperature", "Salinity", "Turbidity",
                                           "Chlorophyll", "River Herring")))
```

**Add Transformed Predictors**

```
new_values <- base_data %>%
  select(-c(H:Temora,  AvgTurb, DOsat)) %>%
  mutate(Turb = log(Turb),
         Chl = log(Chl),
         RH = log1p(RH)) %>%
  pivot_longer(-c(Date:riv_km, combined_density),
               names_to = 'Predictor', values_to = 'Values_2') %>%
  pull('Values_2')

long_data <- long_data %>%
  mutate(Transformed_Values = new_values)
```

# Modeling with GAMs

Overall, I'm trying to mimic the linear models from the manuscript, only using GAMs instead of linear models.

In practice I've made three big changes:

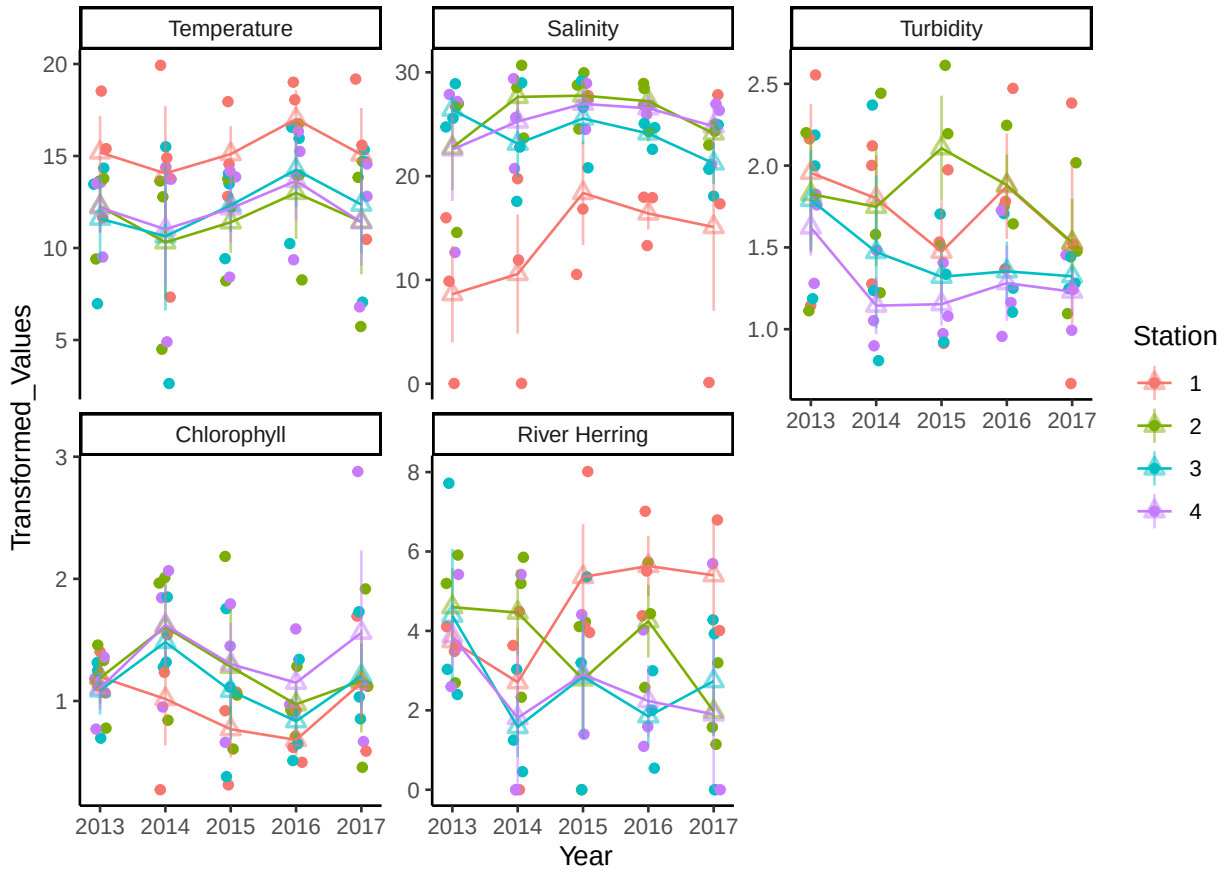
1. I transformed some predictor variables. This was to ease model fitting, but may not be the right call if the transformed variables make no sense, are hard to explain to readers, or or if there are scientific reasons to not look at transformed predictor variables.
2. I have shifted to 'GAM' models These allow fitting fairly arbitrary smooth relationships between predictors and response variables.
3. Since I'm using GAMs, I also explore alternatives to assumptions of normally distributed errors. I think alternative model specifications (other than Gaussian models), makes better sense.

# Data Review by Year

It's worth looking at how values vary year to year, since we are adding Year to the models.

```
long_data %>%
  ggplot(aes(x = Year, y = Transformed_Values, color = Station)) +
  geom_jitter(height = 0, width = .1) +
  stat_summary(shape = 2, alpha = 0.5, fun.data = 'mean_se') +
  stat_summary(geom = 'line', fun = 'mean') +
  facet_wrap(~Predictor, scales = 'free_y')
#> Warning: Removed 2 rows containing non-finite values (stat_summary).

#> Warning: Removed 2 rows containing non-finite values (stat_summary).
#> Warning: Removed 2 rows containing missing values (geom_point).
```




## General Modeling Logic and Considerations

I'm building on the thinking in the other notebook here, so I'm skipping over background on GAMS.

### Repeated Measures / Mixed models?

We need to represent BOTH sample locations and Years in the model somehow.

Each station was sampled repeatedly. Formally, this is "repeated measures" analysis. That can be modeled either as a hierarchical model (with Station as a "Random Factor"), or by fitting Stations as a factor in the model. We can't leave station out of the model entirely.

We face a similar choice regarding how to fit a "year" term, but with one more choice:

- Fit Year as a numerical predictor (a bad idea here, since we have few years, and no reason to think the relationship is linear, or even smooth)
- Fit Year s a factor, and include it as a full "fixed effects" term in the model
- Fit Year as a "random factor" term in the model.

The choice of strategy (fixed effects of random effects) should reflect a *scientific*, not statistical judgment of what question we want to ask and how best to evaluate causal hypotheses.

Are we interested in evaluating or interpreting station to station differences (yes!). Are we interested in interpreting Year to Year differences (ambiguous).

Looking at this in terms of impact of dam removal on zooplankton, then the year matters, as abundance of river herring in the system is increasing. But are we really interested in Year, or is Year functioning as a surrogate for a better predictor, say adult river herring returns?

My tendency is to want to treat year as a random factor, effectively treating it as uninteresting year to year variation that we expect to occur, but that we are uninterested in on its own.

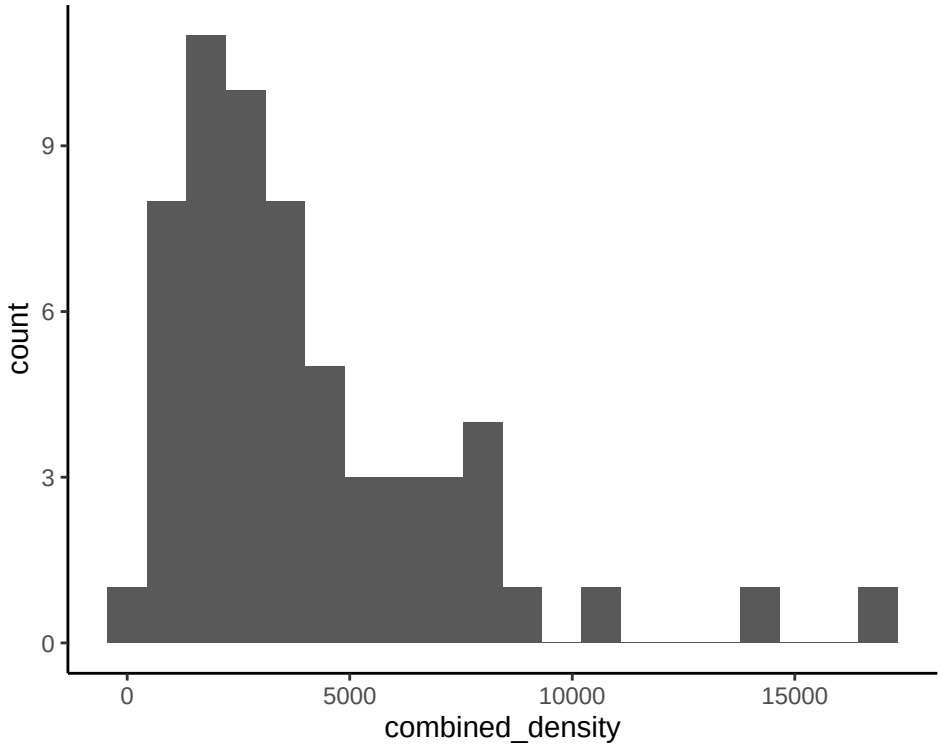
But that's not how this was handled in the manuscript, so I leave "Year" in as a factor.

# Total Density Data

## Data Review

### Histogram

```
ggplot(base_data, aes(combined_density)) +
  geom_histogram(bins = 20)
```

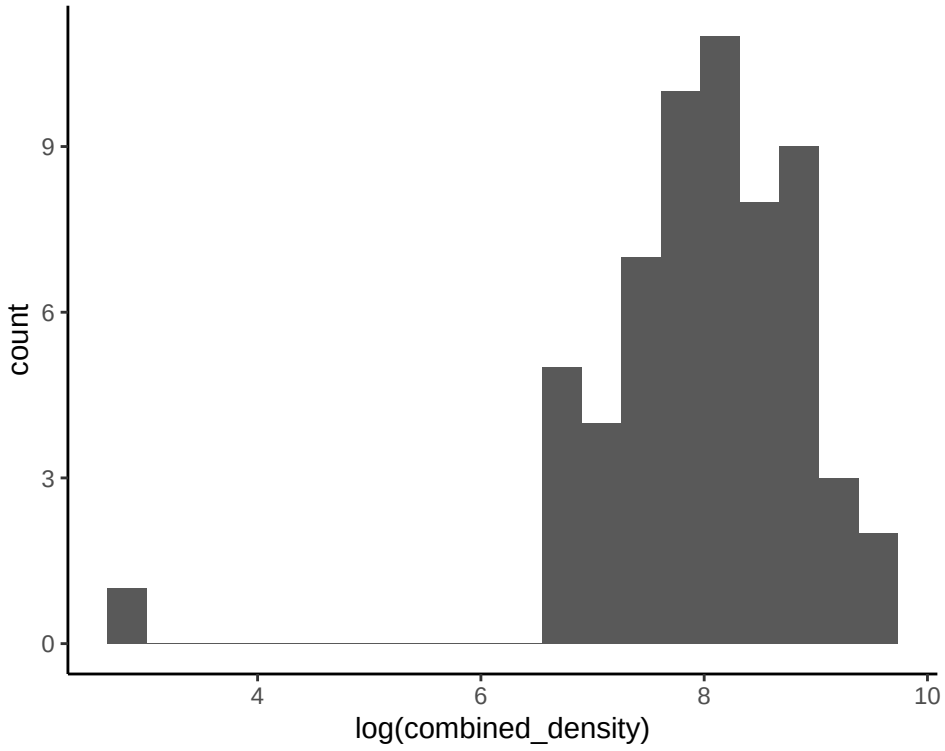



Looking at observed values can only give a hint at appropriate model strategies, since the random component of a model should match the residuals, not the raw observations. If predictors are skewed, the observations may be skewed even if errors are not. This certainly does not look like a normal distribution, so we should consider it likely that something other than a standard least squares regression may do a better job.

### Histogram of Log Transform

A log transform helps, but it may make one very low abundance value have greater impact on model fits than we would like. The low value may have n inordinate impact on any model that fits logs.

```
ggplot(base_data, aes(log(combined_density))) +
  geom_histogram(bins = 20)
```




### The Low Abundance Sample

Lets look at that low abundance sample. . .

```
low_sample <- which(base_data$combined_density ==
       min(base_data$combined_density, na.rm = TRUE))
base_data[low_sample,]
#> # A tibble: 1 x 24
#>   Date                Station  Year Yearf Month Season   DOY riv_km  Temp   Sal
#>   <dttm>              <fct>   <dbl> <fct> <fct> <fct>  <dbl>  <dbl> <dbl> <dbl>
#> 1 2014-05-02 00:00:00 1        2014 2014  May   Spring   122   22.6  7.33  0.03
#> # ... with 14 more variables: 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>
```

It is one of our low salinity samples, so it will – and probably should – have a big effect on our models at low salinity. This reminds us that we have relatively little data from low salinity, spring samples, so no matter what the models are telling us, we don't have a lot of information to go on, and should not over interpret model output.

## Basic Model Structure

### Selection of Predictors

All the following models use the same predictor variables. These are based on the linear models reported in the manuscript.

In particular, ISTILL did not try to fit any interaction terms between predictors, although from a scientific perspective some interactions might be enlightening.

I have transformed several predictors to ensure predictors are ore evenly distributed across the model space. This is for purely statistical reasons, and one could argue on scientific grounds against any of these choices.

GAM models are built on a combination of zero or more "linear" predictors and one or more "smooth" predictors. For all of the models that follow, I use the same suite of predictors, so the ONLY thing that differs is the model form.

- Linear Terms
  - Station. This fits a mean value for each Station, which adjusts for conditions at that site. This is not necessarily a very robust way of addressing location in the estuary, since I would not be surprised if those differences themselves show seasonal patterns, especially based on the location of the turbidity maximum, but it's a reasonable starting point.
  - Year. I added Year as a factor. This means the model fits a separate parameter for each year, rather then, for example, trying to fit some sort of a trend.
- Smoothed Terms
  - Temperature (untransformed)
  - Salinity (untransformed) but as you will see, there are still problems
  - log(Turbidity)
  - log(Chlorophyll)
  - log(River Herring + 1 ). I had to add one to deal with zero counts.

**Shrinkage Estimators** Each of these models use "Shrinkage" estimates of the smoothing terms, which allow certain terms to be "shrunk" out of the model. It's an alternative to AIC based model selection. See the help file for `step.gam`, which explains why `mgcv` does not include a step procedure.

## Model Alternatives

We are looking at values (density) over the positive number line. So we should principally look at models that assign zero probability (or negligibly small probability) to negative numbers.

Also, we expect our estimates of "density" to be pretty precise when density is low, but not if density is high. If we only count 5 zooplankton, it's unlikely that a replicate count would find 55, so an error of 50 is highly unlikely. On the other hand, if we count 3000, we might not be too surprised if a replicate count had 3050.

Our best model choices should reflect those two features of our data. I considered several model strategies (see the other notebook for deeper discussion).

1. A simple Gaussian GAM. On first principals, a Gaussian model is likely to be a fairly poor model for these data, because any Gaussian model expects deviations (observed - predicted) on large counts to be similar to on small counts. That's not reasonable for these data. If all our densities were of similar magnitude and relatively high (compared to their standard errors) than we might get away with a Gaussian model. That's not the case here.

2. Model assuming normally distributed errors, but with some sort of a transform applied to the dependent variable. Common choices for transforms include the log and square root transforms. These models naturally assume higher deviations for higher counts, only slightly so for the square root, but strongly so for the log transform. The log transform assumes deviations are proportional to zooplankton density, which feels like a reasonable assumption. But there's that wonky outlier in the log transformed data....

3. Use a Gamma family GLM / GAM – this implies standard error is roughly proportional to the mean. Again, not an unreasonable choice. A Gamma model cannot handle a value of zero, but we don't have any zero densities. A gamma model can be fit with:

   - an identity link
   - a log link
   - an inverse link (related to the "canonical link")

   It's not obvious to me which link function to use. In general, the selection of link functions is determined by the functional form of response expected, usually on scientific grounds. The identity link yields an "additive" model, the log link, a "multiplicative model" and the inverse link some sort of a "rate" or "harmonic mean" model. Here, I've come to believe the Log link model is probably most appropriate, and I focus on that exclusively....

On first principals, I like the transformed Gaussian and Gamma models. In general, it can be very hard to choose between a Gaussian model on log transformed data and the Gamma model with a log link. They are close cousins.

## Testing Different Models

### Log Transform (lognormal model)

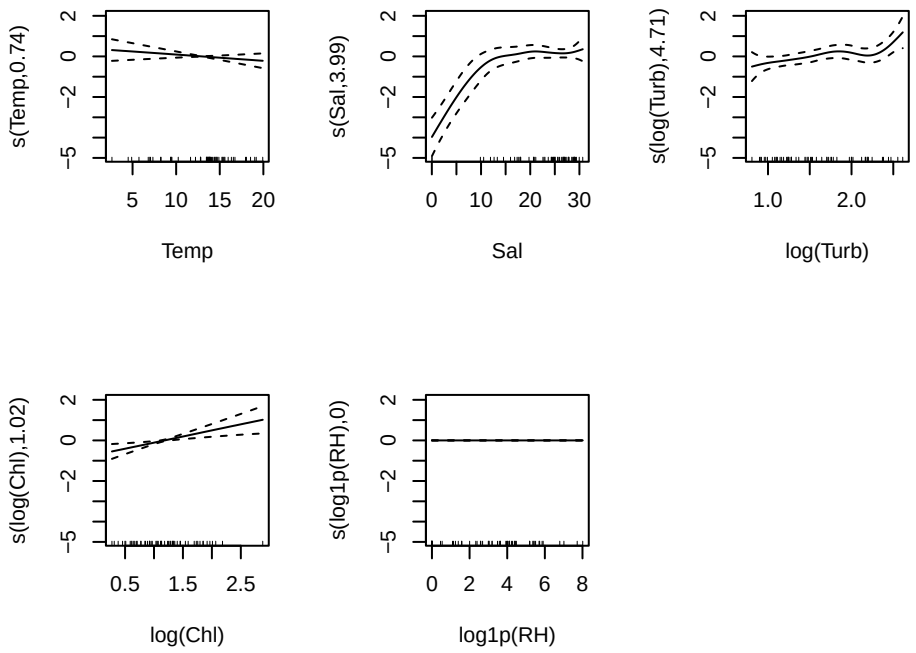
We next try a "lognormal" model. This is equivalent to assuming the underlying errors are drawn from a lognormal distribution.

```
combined_density_gam_l <- gam(log(combined_density) ~
                              Station +
                              Yearf +
                              s(Temp, bs="ts") +
                              s(Sal, bs="ts") +
                              s(log(Turb), bs="ts") +
                              s(log(Chl), bs="ts") +
                              s(log1p(RH),bs="ts"),
                            data = base_data, family = 'gaussian')
summary(combined_density_gam_l)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log(combined_density) ~ Station + Yearf + 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)   9.4757     0.3070  30.865  < 2e-16 ***
```

```
#> Station2     -0.7959     0.3250  -2.449   0.0189 *
#> Station3     -0.4924     0.3081  -1.598   0.1180
#> Station4     -0.8892     0.3323  -2.676   0.0108 *
#> Yearf2014    -1.2320     0.2514  -4.901 1.66e-05 ***
#> Yearf2015    -1.6219     0.2480  -6.541 8.65e-08 ***
#> Yearf2016    -0.4951     0.2471  -2.004   0.0520 .
#> Yearf2017    -1.2837     0.2662  -4.822 2.13e-05 ***
#> ---
#> 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)      7.352e-01      9 0.160 0.15307
#> s(Sal)       3.994e+00      9 8.844 < 2e-16 ***
#> s(log(Turb)) 4.706e+00      9 1.620 0.01254 *
#> s(log(Chl))  1.022e+00      9 1.107 0.00127 **
#> s(log1p(RH)) 1.431e-07      9 0.000 0.40248
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.718   Deviance explained = 80.4%
#> GCV = 0.39234  Scale est. = 0.26748   n = 58
```

Note that this model identifies SALINITY as an important predictor, not temperature. It also fits more complex relationships even for some of the "not significant" terms, but the fact that those terms were not "shrunk" to nothing suggests there is useful predictive information from all of the predictors – jut not very CLEAR information....

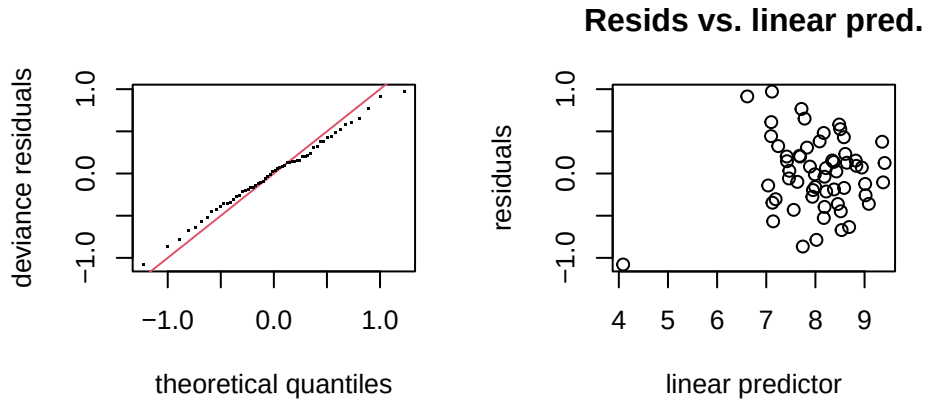
```
oldpar <- par(mfrow = c(2,3))
plot(combined_density_gam_l)
par(oldpar)
```

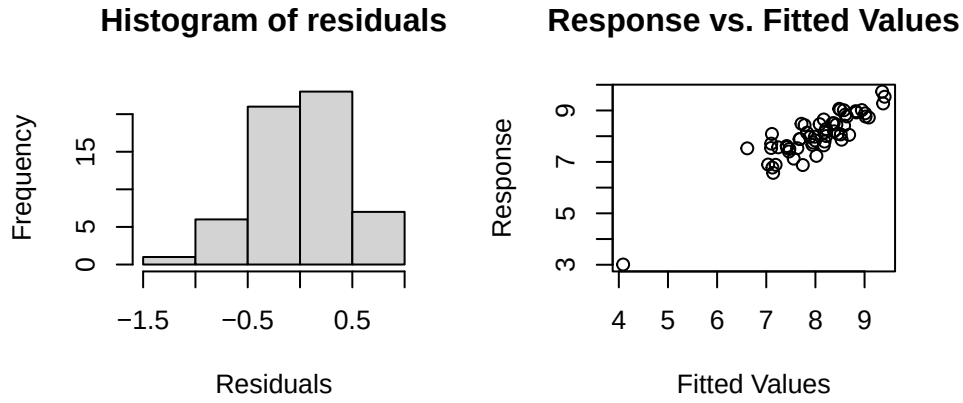



This model suggests:

1. Low salinity observation is different
2. Abundance increases with turbidity
3. Chlorophyll has some additional effect (linear in log of chlorophyll).

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




```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 18 iterations.
#> The RMS GCV score gradient at convergence was 7.407307e-08 .
#> The Hessian was positive definite.
#> Model rank =  53 / 53
#>
#> 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 7.35e-01    1.07    0.68
#> s(Sal)       9.00e+00 3.99e+00    0.96    0.40
#> s(log(Turb)) 9.00e+00 4.71e+00    0.95    0.28
#> s(log(Chl))  9.00e+00 1.02e+00    1.14    0.77
#> s(log1p(RH)) 9.00e+00 1.43e-07    0.94    0.29
par(oldpar)
```

The residuals are slightly skewed the other way.... But the extreme low fitted value is highly problematic. This just matches what we saw in the log-transformed histogram.

Just the way regression models work, the GAM probably "had" to fit that point pretty closely, so it is going to have a pretty big effect on any model. That low abundance under low salinity conditions now has an outsized impact on the model fit, especially regarding the salinity predictor.

### Gamma Model, Log Link

GLMs and GAMs allow you to select many different link functions (although many would make little sense). I find selection of link functions confusing, so I like to stick with the "canonical" link for most GLMs. But that's not the only reasonable choice.

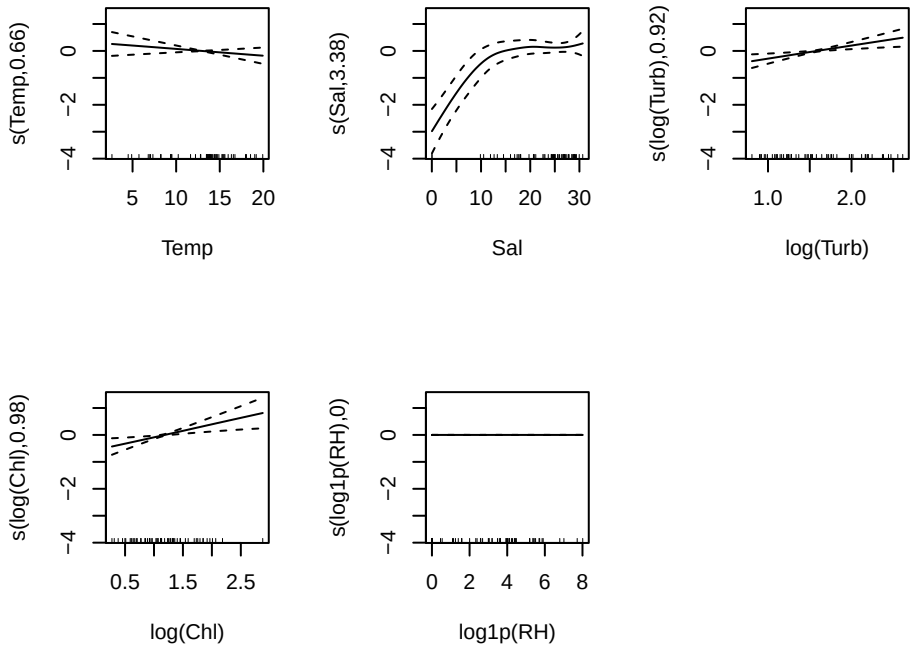
A Gamma GLM is often selected because it can model a wide range of positive valued random variables, where variability increases with expected value. So, Gamma GLMs are are often run with identity or log links, in addition to the (canonical) inverse link.

( I also looked at inverse and identify link gamma models, but after doing a little more reading and thinking, I settled on this one. . . .).

```
combined_density_gam_g2 <- gam(combined_density ~
                               Station +
                               Yearf +
                               s(Temp, bs="ts") +
                               s(Sal, bs="ts") +
                               s(log(Turb), bs="ts") +
                               s(log(Chl), bs="ts") +
                               s(log1p(RH),bs="ts"),
                               data = base_data, family = Gamma(link = 'log'))
summary(combined_density_gam_g2)
#>
#> Family: Gamma
#> Link function: log
#>
#> Formula:
#> combined_density ~ Station + Yearf + 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)   9.4212     0.2680  35.151  < 2e-16 ***
#> Station2     -0.7596     0.2849  -2.666  0.01068 *
#> Station3     -0.4233     0.2672  -1.584  0.12030
#> Station4     -0.8093     0.2875  -2.815  0.00727 **
#> Yearf2014    -1.0983     0.2182  -5.034 8.59e-06 ***
#> Yearf2015    -1.4038     0.2148  -6.537 5.46e-08 ***
#> Yearf2016    -0.4371     0.2145  -2.038  0.04763 *
#> Yearf2017    -1.1147     0.2275  -4.900 1.34e-05 ***
#> ---
#> 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)      6.634e-01      9 0.157 0.14012
#> s(Sal)       3.384e+00      9 6.624 < 2e-16 ***
#> s(log(Turb)) 9.226e-01      9 1.034 0.00174 **
#> s(log(Chl))  9.840e-01      9 0.977 0.00229 **
#> s(log1p(RH)) 4.454e-05      9 0.000 0.34760
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
```

```
#> R-sq.(adj) =  0.568   Deviance explained = 66.8%
#> GCV = 0.36464  Scale est. = 0.22577   n = 58
```

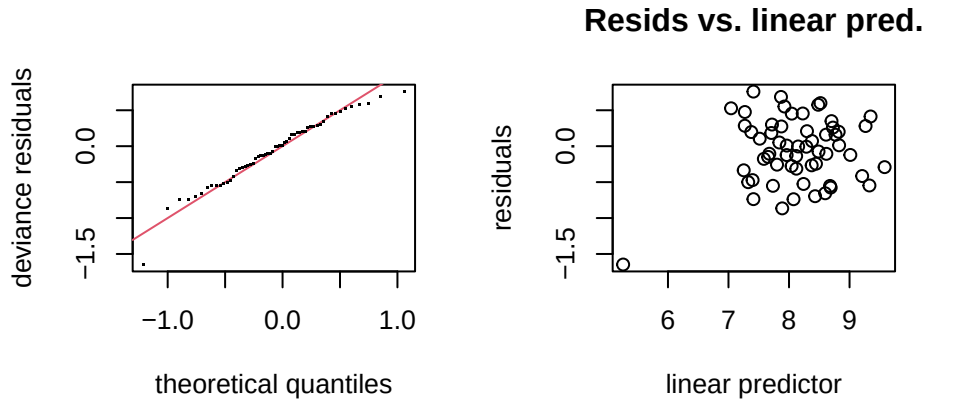
```
oldpar <- par(mfrow = c(2,3))
plot(combined_density_gam_g2)
par(oldpar)
```

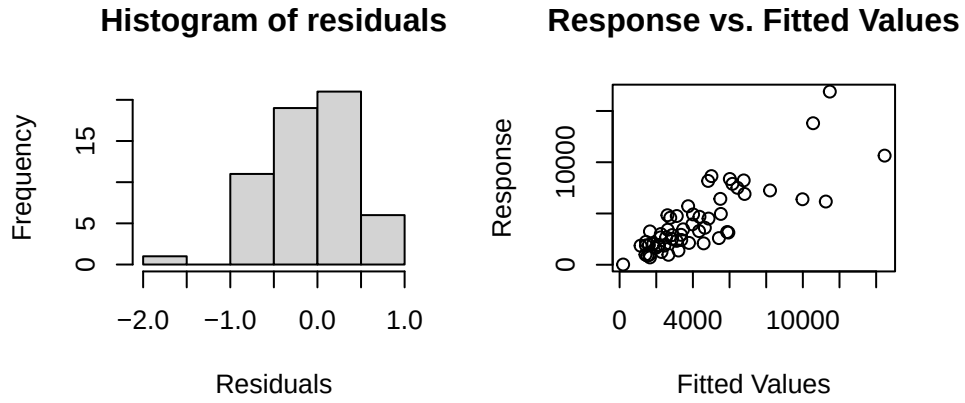



So,

1. Low salinity samples are different. After that the effect of salinity is relatively small.
2. Abundance increases with turbidity.
3. Abundance increases with Chlorophyll

```
oldpar <- par(mfrow = c(2,2))
gam.check(combined_density_gam_g2)
```




```
#>
#> Method: GCV   Optimizer: outer newton
#> full convergence after 13 iterations.
#> Gradient range [-2.914338e-07,1.915833e-08]
#> (score 0.3646405 & scale 0.2257712).
#> Hessian positive definite, eigenvalue range [2.914246e-07,0.005015266].
#> Model rank =  53 / 53
#>
#> 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 6.63e-01    1.14    0.90
#> s(Sal)       9.00e+00 3.38e+00    0.92    0.34
#> s(log(Turb)) 9.00e+00 9.23e-01    0.86    0.15
#> s(log(Chl))  9.00e+00 9.84e-01    1.06    0.61
#> s(log1p(RH)) 9.00e+00 4.45e-05    1.00    0.47
par(oldpar)
```

This model has OK residual structure, but it still has that one extreme low predicted value....

### Conclusions

1. Now that we added a "Year" term, the temperature signal is no longer identified as statistically meaningful, in any of the models. Again, this just highlights the problems of fitting models with correlated predictors.
2. That one very low abundance, low salinity sample dominates the relationship of abundance and salinity. Without that low value, I doubt there would be much of a pattern to point to. Nevertheless, that relationship is real.
3. Abundance increases strongly with turbidity.
4. In most models, there is also a positive association between zooplankton abundance and chlorophyll.

## Diversity (Shannon Index)

In my experience, **many**, but not all calculated indexes end up with error distributions that can reasonably be modeled with normal distribution errors. I start out hoping that we can just use a regular Gaussian model here. But the Shannon index is a strictly positive value, so we should consider models that also restrict our predictions to positive values. Luckily, selection of the specific model has little effect on the results.

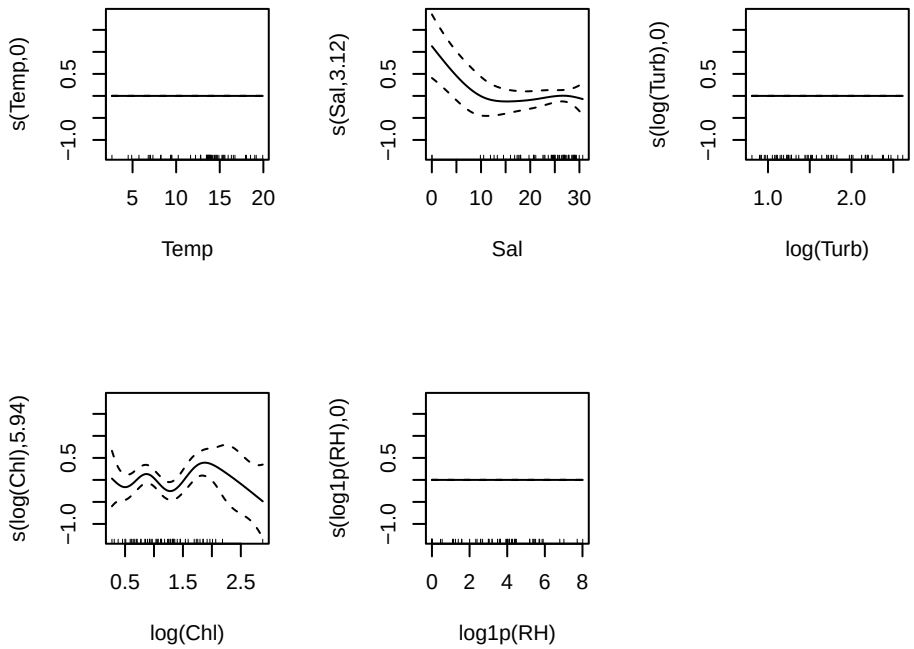
### Gaussian GAM, with Identity Link

We are using "Shrinkage" estimates of the smoothing terms again, which allow certain terms to be "shrunk" out of the model

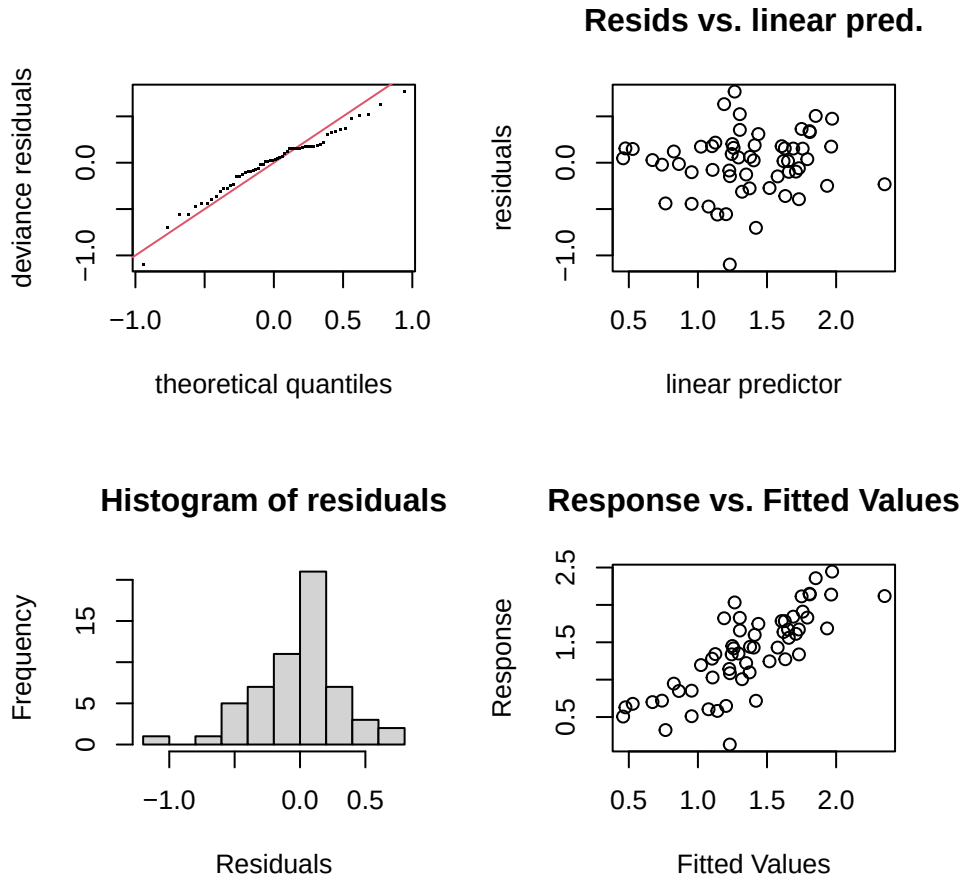
```
shannon_gam <- gam(H ~ Station +
                     Yearf +
                     s(Temp, bs="ts") +
                     s(Sal, bs="ts") +
                     s(log(Turb), bs="ts") +
                     s(log(Chl), bs="ts") +
                     s(log1p(RH),bs="ts"),
                   data = base_data, family = 'gaussian')
summary(shannon_gam)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> H ~ Station + Yearf + 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)   1.4007     0.2072   6.762  3.6e-08 ***
#> Station2      0.4880     0.2027   2.408  0.02062 *
#> Station3      0.5796     0.1906   3.041  0.00410 **
#> Station4      0.4304     0.1980   2.174  0.03554 *
#> Yearf2014    -0.5872     0.1917  -3.063  0.00386 **
#> Yearf2015    -0.3915     0.2011  -1.947  0.05840 .
```

```
#> Yearf2016    -0.6621     0.1875  -3.531  0.00104 **
#> Yearf2017    -0.4855     0.2044  -2.376  0.02228 *
#> ---
#> 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)      4.049e-06      9 0.000 0.50681
#> s(Sal)       3.116e+00      9 1.589 0.00364 **
#> s(log(Turb)) 5.712e-07      9 0.000 0.73576
#> s(log(Chl))  5.944e+00      9 1.695 0.02492 *
#> s(log1p(RH)) 2.425e-06      9 0.000 0.83489
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.477   Deviance explained = 62.4%
#> GCV = 0.22272  Scale est. = 0.15721   n = 58
```

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



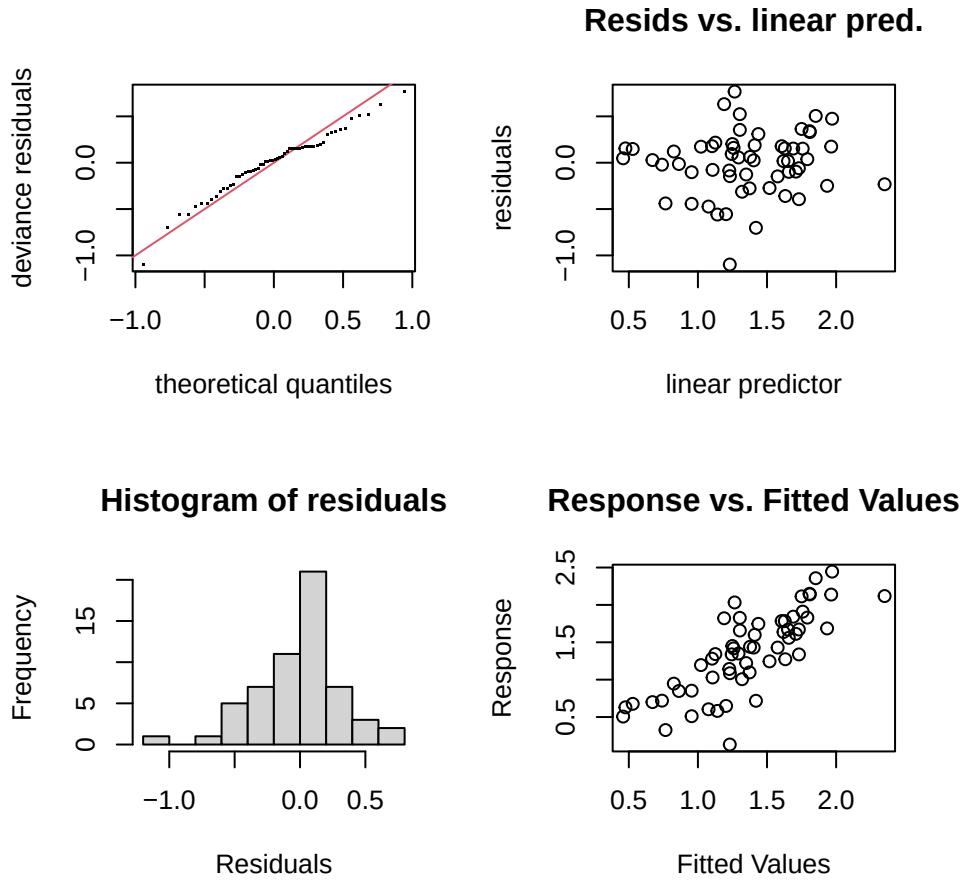

```
oldpar <- par(mfrow = c(2,2))
gam.check(shannon_gam)
```




```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 21 iterations.
#> The RMS GCV score gradient at convergence was 2.781652e-08 .
#> The Hessian was positive definite.
#> Model rank =  53 / 53
#>
#> 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 4.05e-06    0.96    0.34
#> s(Sal)       9.00e+00 3.12e+00    1.02    0.52
#> s(log(Turb)) 9.00e+00 5.71e-07    1.14    0.81
#> s(log(Chl))  9.00e+00 5.94e+00    1.13    0.76
#> s(log1p(RH)) 9.00e+00 2.43e-06    1.04    0.61
par(oldpar)
```

Not a bad model from a model diagnostics point of view.

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



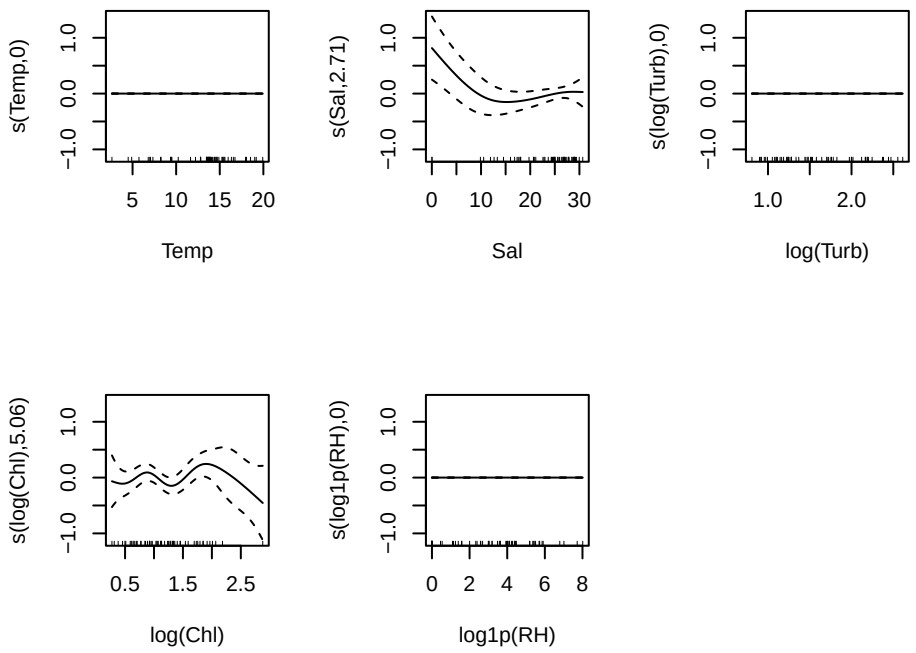

```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 21 iterations.
#> The RMS GCV score gradient at convergence was 2.781652e-08 .
#> The Hessian was positive definite.
#> Model rank =  53 / 53
#>
#> Basis dimension (k) checking results. Low p-value (k-index<1) may
#> indicate that k is too low, especially if edf is close to k'.
#>
#>                   k'      edf k-index p-value
#> s(Temp)      9.00e+00 4.05e-06    0.96    0.35
#> s(Sal)       9.00e+00 3.12e+00    1.02    0.52
#> s(log(Turb)) 9.00e+00 5.71e-07    1.14    0.76
#> s(log(Chl))  9.00e+00 5.94e+00    1.13    0.76
#> s(log1p(RH)) 9.00e+00 2.43e-06    1.04    0.62
par(oldpar)
```

That's an OK model. The biggest weakness is one extreme negative outlier.

### Gamma Gam with Log Link

```
shannon_gam_g2 <- gam(H ~ Station +
                      Yearf +
                      s(Temp, bs="ts") +
                      s(Sal, bs="ts") +
                      s(log(Turb), bs="ts") +
                      s(log(Chl), bs="ts") +
                      s(log1p(RH),bs="ts"),
                    data = base_data, family = Gamma(link = 'log'))
summary(shannon_gam_g2)
#>
#> Family: Gamma
#> Link function: log
#>
#> Formula:
#> H ~ Station + Yearf + 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)   0.1992     0.1647   1.210  0.23311
#> Station2      0.4293     0.1616   2.657  0.01109 *
#> Station3      0.5130     0.1521   3.373  0.00160 **
#> Station4      0.4026     0.1587   2.538  0.01493 *
#> Yearf2014    -0.4081     0.1514  -2.696  0.01004 *
#> Yearf2015    -0.2519     0.1589  -1.585  0.12033
#> Yearf2016    -0.4913     0.1497  -3.282  0.00207 **
#> Yearf2017    -0.3032     0.1601  -1.894  0.06512 .
#> ---
#> 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)      9.545e-07      9 0.000 0.32061
#> s(Sal)       2.715e+00      9 1.652 0.00188 **
#> s(log(Turb)) 5.309e-06      9 0.000 0.78458
#> s(log(Chl))  5.057e+00      9 1.292 0.04721 *
#> s(log1p(RH)) 3.138e-06      9 0.000 0.98960
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.435   Deviance explained = 49.6%
#> GCV = 0.21377  Scale est. = 0.10319   n = 58
```

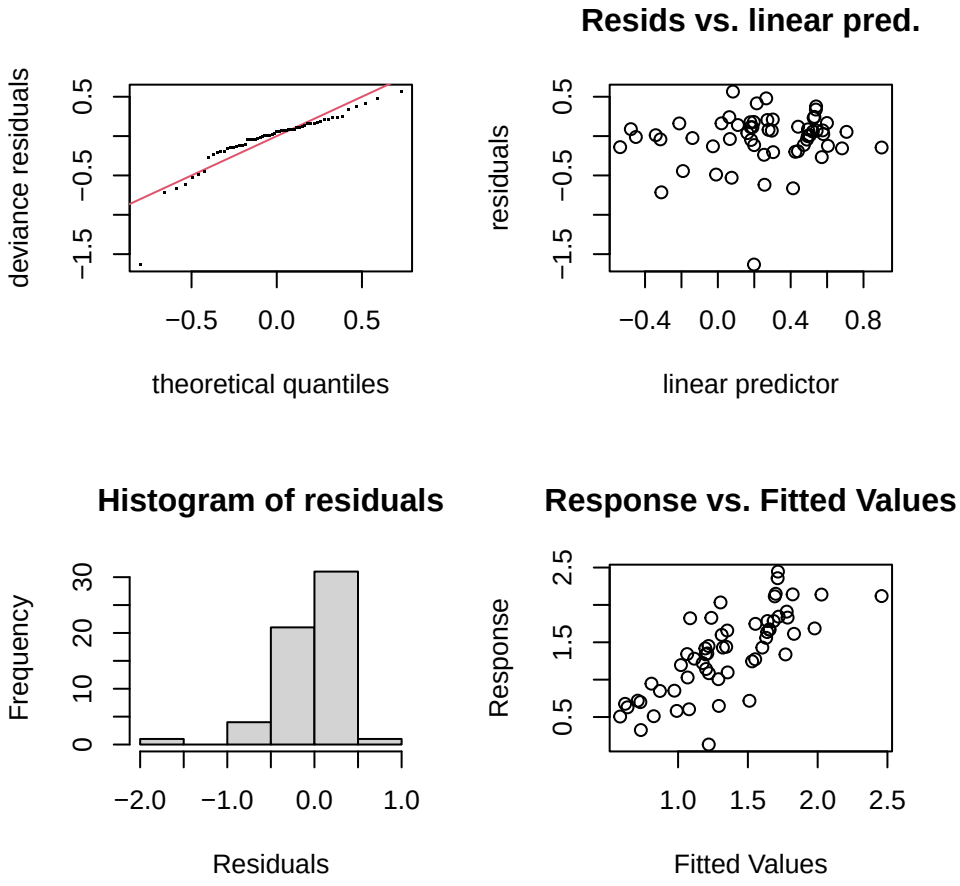
```
oldpar <- par(mfrow = c(2,3))
plot(shannon_gam_g2)
par(oldpar)
```




Fit without constraints, as here, this fits a wiggly relationship to chlorophyll, which makes little sense to me. If you constrain the model so that it won't fit such a wiggly relationship, (e.g. with `s(log(Chl), bs="ts", k = 5)`), Chlorophyll ceases to be important.

The only consistently significant relationship between diversity and any of the predictors relates to salinity. Unfortunately, most of that pattern is again due to a handful of low salinity samples.

```
oldpar <- par(mfrow = c(2,2))
gam.check(shannon_gam_g2)
```




```
#>
#> Method: GCV   Optimizer: outer newton
#> full convergence after 14 iterations.
#> Gradient range [-4.814114e-08,5.067809e-10]
#> (score 0.2137675 & scale 0.1031904).
#> Hessian positive definite, eigenvalue range [4.308634e-09,0.003614985].
#> Model rank =  53 / 53
#>
#> 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 9.55e-07    0.99    0.42
#> s(Sal)       9.00e+00 2.71e+00    1.05    0.64
#> s(log(Turb)) 9.00e+00 5.31e-06    1.13    0.86
#> s(log(Chl))  9.00e+00 5.06e+00    1.09    0.72
#> s(log1p(RH)) 9.00e+00 3.14e-06    0.98    0.44
par(oldpar)
```

That extreme negative outlier is the main weakness of this model. This model is perhaps slightly less trustworthy than the Gaussian model. Luckily, results are consistent.

## Drop the Low Salinity Observations

Let's just see what happens if we drop our handful of low salinity observations. Here we drop all samples with salinity below 5, which amounts to three samples from Station 1 in May of 2013, 2014, and 2017.

The goal here is to evaluate my guess that those low salinity observations are "really different" from the other samples, and thus dominate model fitting.

```
low_sample <- which(base_data$Sal <= 5)
base_data[low_sample,]
#> # A tibble: 3 x 24
#>   Date                Station  Year Yearf Month Season   DOY riv_km  Temp    Sal
#>   <dttm>              <fct>   <dbl> <fct> <fct> <fct>  <dbl>  <dbl> <dbl>  <dbl>
#> 1 2013-05-28 00:00:00 1        2013 2013  May   Spring   148   22.6 11.7  0.0200
#> 2 2014-05-02 00:00:00 1        2014 2014  May   Spring   122   22.6  7.33 0.03
#> 3 2017-05-12 00:00:00 1        2017 2017  May   Spring   132   22.9 10.5  0.12
#> # ... with 14 more variables: 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>
smaller_data <- base_data[-low_sample,]
```

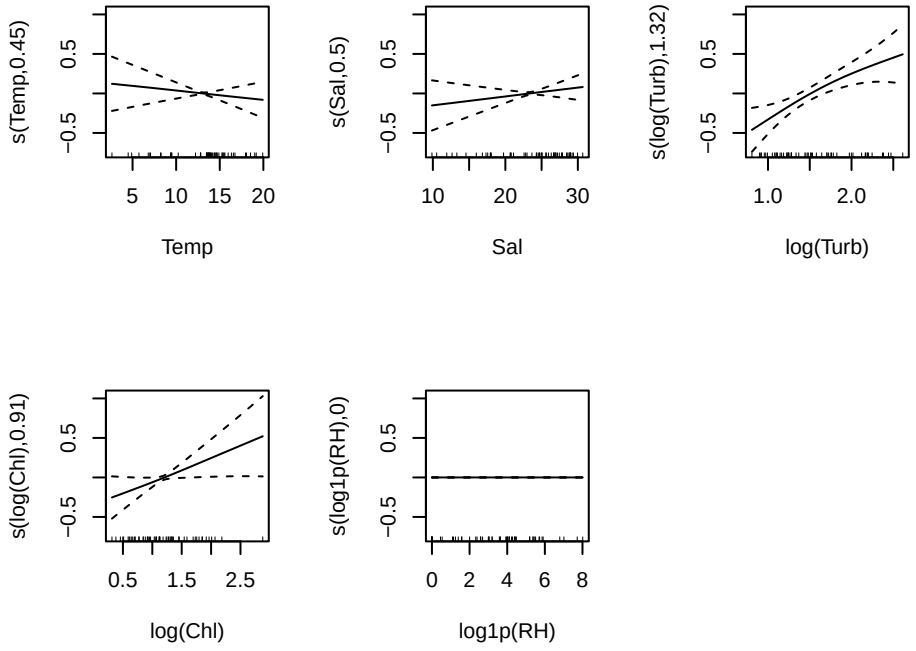
### Total Density

#### Log Transformed Model

```
drop_density_gam_l <- gam(log(combined_density) ~
                            Station +
                            Yearf +
                            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),
                          data = smaller_data, family = 'gaussian')
summary(drop_density_gam_l)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log(combined_density) ~ Station + Yearf + 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)   9.1166     0.2280  39.982  < 2e-16 ***
#> Station2     -0.5150     0.2298  -2.241  0.03000 *
#> Station3     -0.2426     0.2176  -1.115  0.27092
#> Station4     -0.5580     0.2393  -2.332  0.02427 *
```

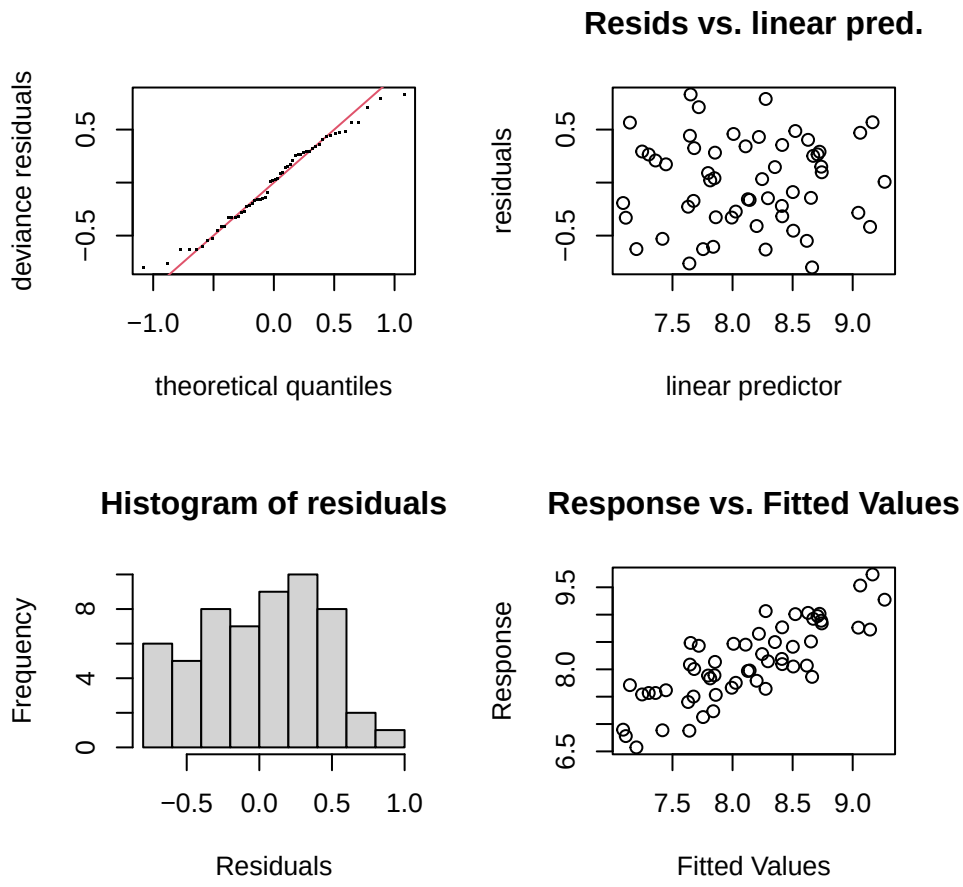
```
#> Yearf2014    -0.7355     0.2115  -3.478  0.00114 **
#> Yearf2015    -1.3155     0.2035  -6.465 6.46e-08 ***
#> Yearf2016    -0.2918     0.1988  -1.468  0.14918
#> Yearf2017    -0.9724     0.2016  -4.823 1.67e-05 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>                     edf Ref.df     F p-value
#> s(Temp)      4.470e-01      3 0.178 0.26029
#> s(Sal)       4.992e-01      3 0.308 0.17094
#> s(log(Turb)) 1.325e+00      3 4.505 0.00059 ***
#> s(log(Chl))  9.141e-01      3 1.451 0.01981 *
#> s(log1p(RH)) 1.816e-10      3 0.000 0.98993
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.592   Deviance explained = 66.8%
#> GCV = 0.2614  Scale est. = 0.20919   n = 56
```

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




This model confirms that abundance increases with turbidity and chlorophyll, and that salinity barely matters if you drop those low salinity observations.

```
oldpar <- par(mfrow = c(2,2))
gam.check(drop_density_gam_1)
```



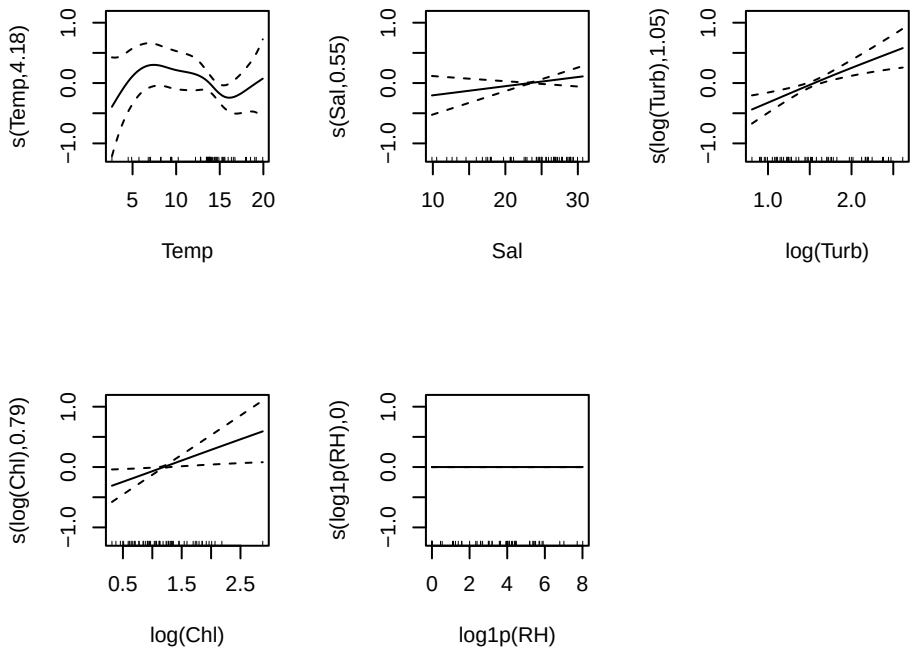

```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 16 iterations.
#> The RMS GCV score gradient at convergence was 5.538128e-08 .
#> The Hessian was positive definite.
#> Model rank =  23 / 23
#>
#> Basis dimension (k) checking results. Low p-value (k-index<1) may
#> indicate that k is too low, especially if edf is close to k'.
#>
#>                    k'      edf k-index p-value
#> s(Temp)      3.00e+00 4.47e-01    0.92    0.18
#> s(Sal)       3.00e+00 4.99e-01    0.84    0.06 .
#> s(log(Turb)) 3.00e+00 1.32e+00    0.90    0.23
#> s(log(Chl))  3.00e+00 9.14e-01    1.05    0.57
#> s(log1p(RH)) 3.00e+00 1.82e-10    1.24    0.96
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
par(oldpar)
```

That is an excellent model for the reduced data set.

### Gamma Model, Log Link

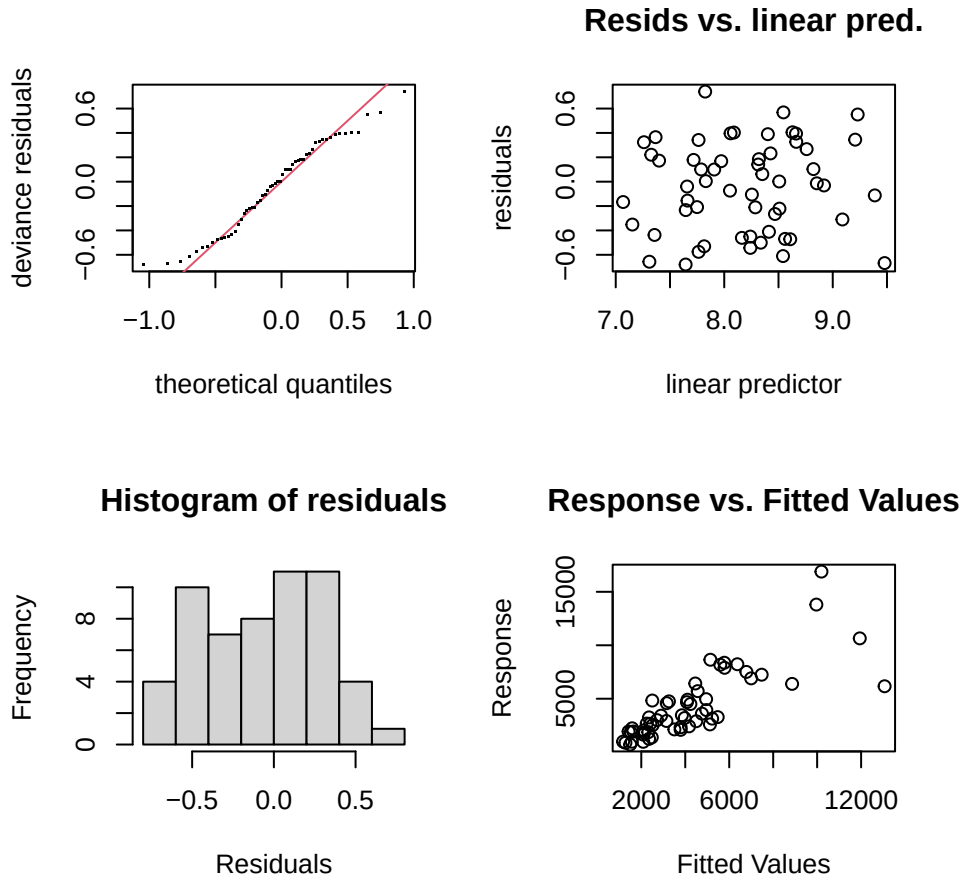
```
drop_density_gam_g2 <- gam(combined_density ~
                             Station +
                             Yearf +
                             s(Temp, bs="ts") +
                             s(Sal, bs="ts") +
                             s(log(Turb), bs="ts") +
                             s(log(Chl), bs="ts") +
                             s(log1p(RH),bs="ts"),
                           data = smaller_data, family = Gamma(link = 'log'))
summary(drop_density_gam_g2)
#>
#> Family: Gamma
#> Link function: log
#>
#> Formula:
#> combined_density ~ Station + Yearf + 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)   9.2485     0.2388  38.728  < 2e-16 ***
#> Station2     -0.6909     0.2554  -2.705  0.00986 **
#> Station3     -0.2618     0.2377  -1.102  0.27697
#> Station4     -0.6935     0.2638  -2.629  0.01197 *
#> Yearf2014    -0.6974     0.2135  -3.266  0.00219 **
#> Yearf2015    -1.3268     0.1866  -7.109 1.09e-08 ***
#> Yearf2016    -0.2552     0.1932  -1.321  0.19368
#> Yearf2017    -0.9286     0.1929  -4.814 2.00e-05 ***
#> ---
#> 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)      4.179e+00      9 0.995 0.065139 .
#> s(Sal)       5.495e-01      9 0.181 0.082700 .
#> s(log(Turb)) 1.054e+00      9 1.693 0.000207 ***
#> s(log(Chl))  7.905e-01      9 0.611 0.007816 **
#> s(log1p(RH)) 2.403e-05      9 0.000 0.733516
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.526   Deviance explained = 72.6%
#> GCV = 0.24642  Scale est. = 0.17269   n = 56
```

```
oldpar <- par(mfrow = c(2,3))
plot(drop_density_gam_g2)
par(oldpar)
```




The Gamma models models also confirms that abundance increases with turbidity and chlorophyll, while salinity only matters if you keep those low salinity observations. Results are similar for identity, log, and inverse links, except that the identity link finds a significant relationship with temperature, which here is only marginally significant.

```
oldpar <- par(mfrow = c(2,2))
gam.check(drop_density_gam_g2)
```




```
#>
#> Method: GCV   Optimizer: outer newton
#> full convergence after 12 iterations.
#> Gradient range [-2.33198e-07,1.503351e-08]
#> (score 0.2464155 & scale 0.1726891).
#> Hessian positive definite, eigenvalue range [2.331877e-07,0.003557504].
#> Model rank =  53 / 53
#>
#> 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)      9.000000 4.178653    1.06   0.665
#> s(Sal)       9.000000 0.549504    0.83   0.095 .
#> s(log(Turb)) 9.000000 1.053557    0.88   0.200
#> s(log(Chl))  9.000000 0.790504    0.98   0.385
#> s(log1p(RH)) 9.000000 0.000024    1.18   0.940
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
par(oldpar)
```

Another excellent model for the reduced data.

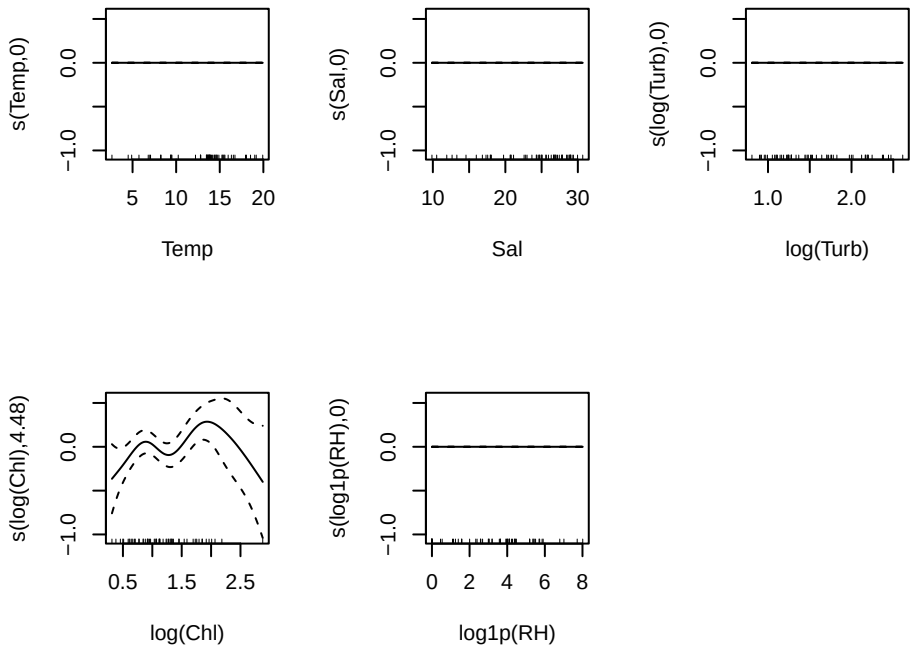
## Diversity

### Gamma Model with Log Link

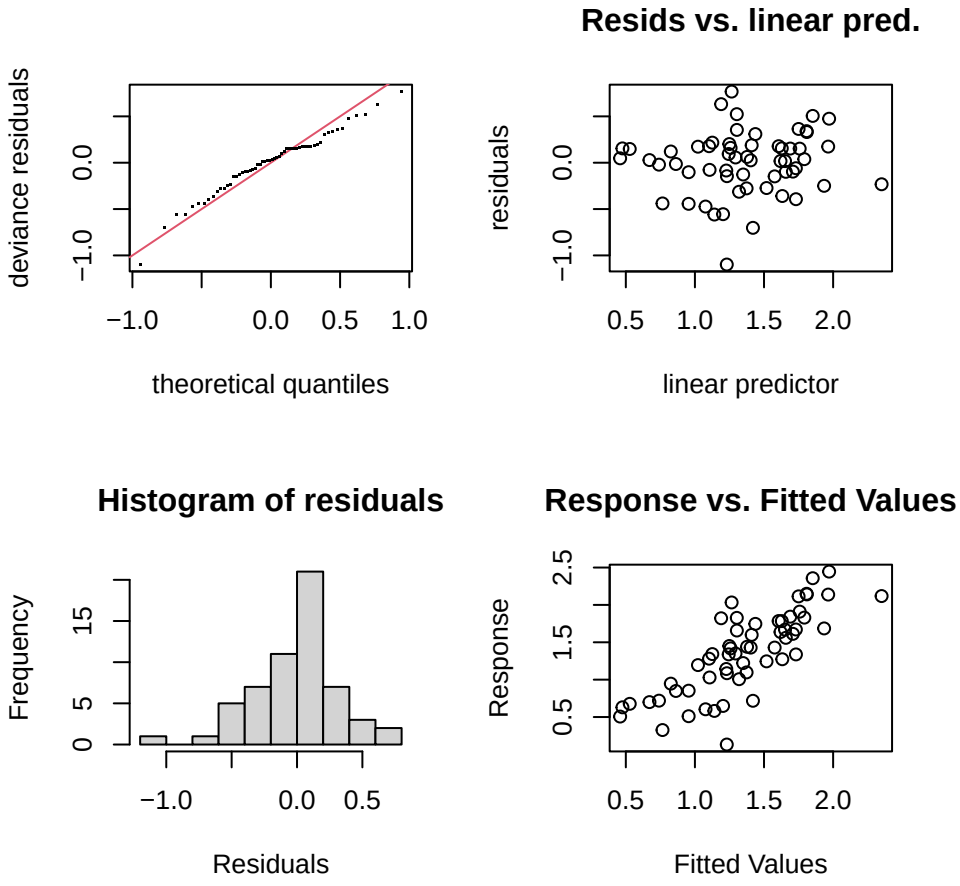
```
drop_shannon_gam <- gam(H ~ Station +
                          Yearf +
                          s(Temp, bs="ts") +
                          s(Sal, bs="ts") +
                          s(log(Turb), bs="ts") +
                          s(log(Chl), bs="ts") +
                          s(log1p(RH),bs="ts"),
                        data = smaller_data, family = Gamma(link = 'log'))
summary(drop_shannon_gam)
#>
#> Family: Gamma
#> Link function: log
#>
#> Formula:
#> H ~ Station + Yearf + 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)   0.1065     0.1349   0.789 0.434345
#> Station2      0.5042     0.1283   3.929 0.000301 ***
#> Station3      0.5756     0.1243   4.629 3.31e-05 ***
#> Station4      0.4705     0.1317   3.573 0.000878 ***
#> Yearf2014    -0.4796     0.1479  -3.243 0.002272 **
#> Yearf2015    -0.1871     0.1457  -1.285 0.205680
#> Yearf2016    -0.4519     0.1380  -3.276 0.002071 **
#> Yearf2017    -0.2938     0.1452  -2.024 0.049103 *
#> ---
#> 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)      1.132e-06      9 0.000 0.39367
#> s(Sal)       1.756e-05      9 0.000 0.23892
#> s(log(Turb)) 5.860e-06      9 0.000 0.87668
#> s(log(Chl))  4.481e+00      9 1.676 0.00971 **
#> s(log1p(RH)) 1.784e-05      9 0.000 0.41144
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.457   Deviance explained = 47.2%
#> GCV = 0.19272  Scale est. = 0.099352  n = 56
```

chlorophyll again emerges as an important predictor of diversity, but the relationship is "wiggly", and may be mostly driven by one high diversity, high chlorophyll sample. I'd prefer a simpler smoothed fit, but exploring simpler smoothers (by specifying `k = 5`) yields a similar fit. At `k = 4`, Chlorophyll does not stay in the model as an important term.

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




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




```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 21 iterations.
#> The RMS GCV score gradient at convergence was 2.781652e-08 .
#> The Hessian was positive definite.
#> Model rank =  53 / 53
#>
#> 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 4.05e-06    0.96    0.38
#> s(Sal)       9.00e+00 3.12e+00    1.02    0.52
#> s(log(Turb)) 9.00e+00 5.71e-07    1.14    0.82
#> s(log(Chl))  9.00e+00 5.94e+00    1.13    0.77
#> s(log1p(RH)) 9.00e+00 2.43e-06    1.04    0.54
par(oldpar)
```

# Example of a Single Species Model – Acartia

## Model Choices

Our model alternatives are basically similar to what we had for the Total Density models.

The Gamma distribution is a continuous-valued distribution with the property that the variance is roughly proportional to the mean. That suggests it may make sense to start by trying a gamma GAM model. Lognormal and inverse Gaussian models could also be appropriate, depending on how heavy-tailed the error distribution is, and how fast we believe the errors increase with predicted values.

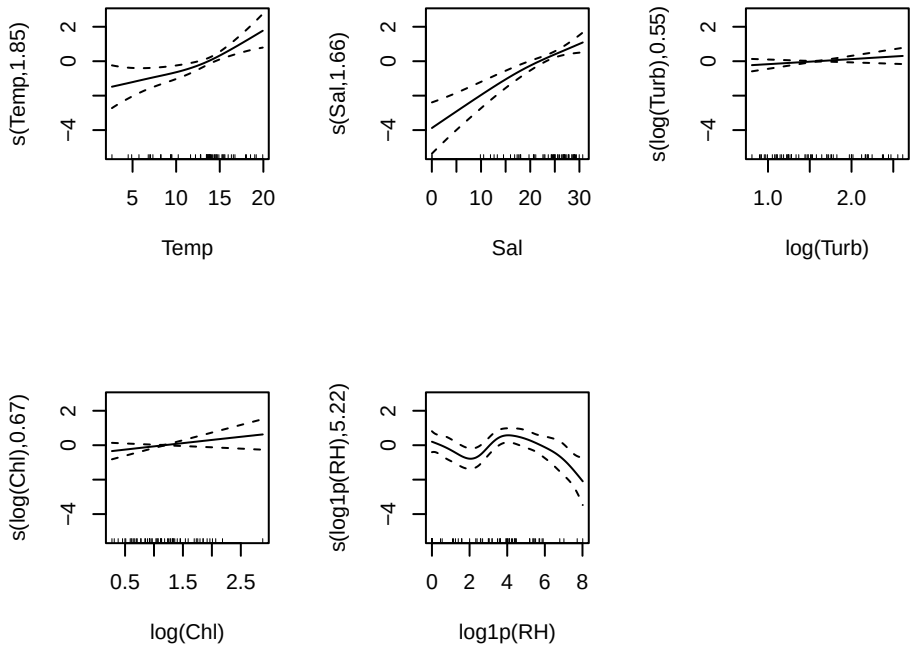
The problem is, we can't use any of the continuous data distributions in GAMS with zero values, at least relying on the canonical link functions, because (log(0) = -Inf; 1/0 = Inf, 1 / 0*0 = Inf).

The easiest solution is to add some finite small quantity to the density data, and predict that. Here we predict Density + 1. An alternative would be to use a different link function. ### Loglinear GAM

```
acartia_gam_l <- gam(log1p(Acartia) ~ Station +
                       Yearf +
                       s(Temp, bs="ts") +
                       s(Sal, bs="ts") +
                       s(log(Turb), bs="ts") +
                       s(log(Chl), bs="ts") +
                       s(log1p(RH),bs="ts"),
                     data = base_data, family = "gaussian")
summary(acartia_gam_l)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Acartia) ~ Station + Yearf + 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)   8.5770     0.5186  16.538  < 2e-16 ***
#> Station2     -0.8310     0.5877  -1.414 0.165134
#> Station3     -0.2425     0.5383  -0.450 0.654848
#> Station4     -0.6118     0.5828  -1.050 0.300096
#> Yearf2014    -1.8274     0.4096  -4.462 6.45e-05 ***
#> Yearf2015    -3.1121     0.4083  -7.621 2.58e-09 ***
#> Yearf2016    -1.4969     0.3957  -3.783 0.000507 ***
#> Yearf2017    -2.1029     0.4069  -5.168 6.90e-06 ***
#> ---
#> 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)      1.8532      9 1.821  0.00019 ***
#> s(Sal)       1.6636      9 3.325 2.29e-06 ***
#> s(log(Turb)) 0.5522      9 0.186  0.07834 .
#> s(log(Chl))  0.6737      9 0.229  0.07695 .
#> s(log1p(RH)) 5.2159      9 2.621  0.00100 **
```

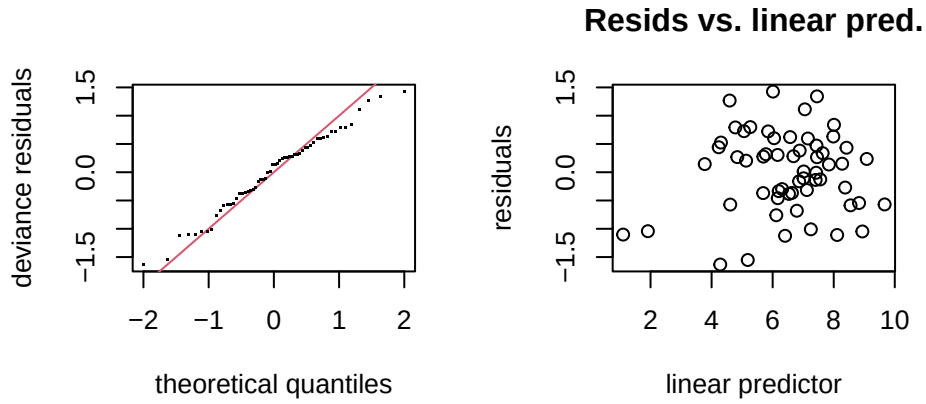
```
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.792   Deviance explained = 85.4%
#> GCV = 1.0265  Scale est. = 0.70865   n = 58
```

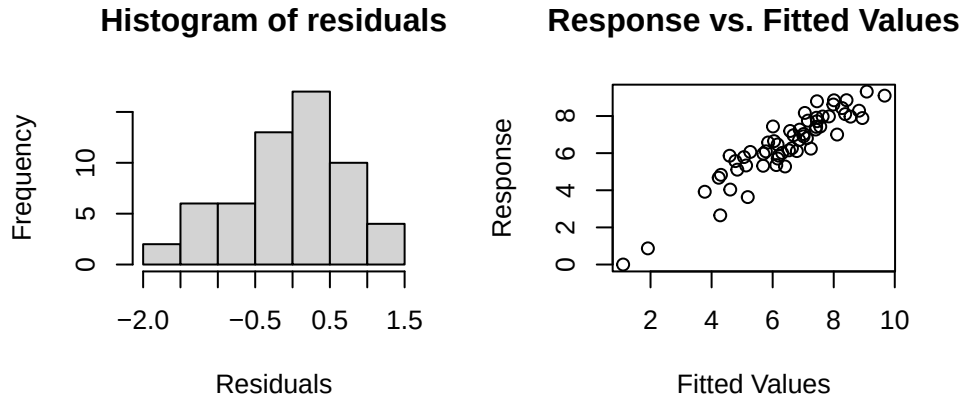
```
oldpar <- par(mfrow = c(2,3))
plot(acartia_gam_1)
par(oldpar)
```




1. Acartia are least abundant in cooler / colder water (seasonal? position in estuary?)
2. Acartia don't like the freshwater end of things as much
3. They kinda like high turbidity, high Chlorophyll waters, but here, the connection is not statistically robust.
4. Acartia is less abundant when river herring are most abundant.

```
oldpar <- par(mfrow = c(2,2))
gam.check(acartia_gam_1)
```




```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 13 iterations.
#> The RMS GCV score gradient at convergence was 6.214017e-07 .
#> The Hessian was positive definite.
#> Model rank =  53 / 53
#>
#> 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)      9.000 1.853    1.11    0.76
#> s(Sal)       9.000 1.664    1.14    0.80
#> s(log(Turb)) 9.000 0.552    1.06    0.66
#> s(log(Chl))  9.000 0.674    0.98    0.37
#> s(log1p(RH)) 9.000 5.216    1.16    0.88
par(oldpar)
```

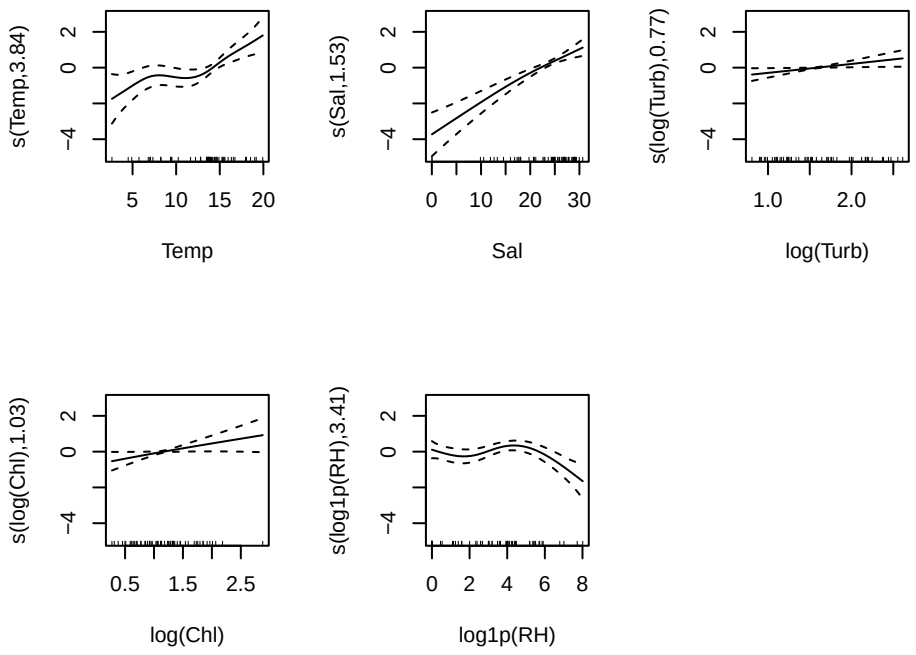
That's actually a pretty good model for these data....

### Gamma Regression with Log Link

We are using "Shrinkage" estimates of the smoothing terms again, which allow certain terms to be "shrunk" out of the model.

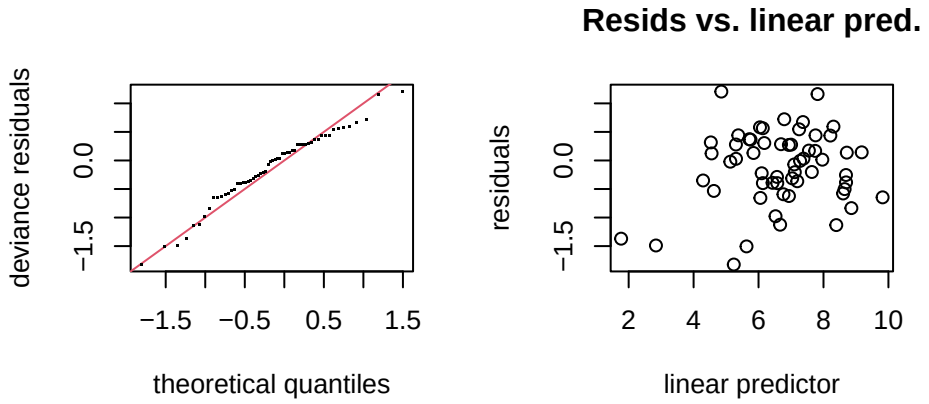
```
acartia_gam_g <- gam(I(Acartia + 1) ~ Station +
                       Yearf +
                       s(Temp, bs="ts") +
                       s(Sal, bs="ts") +
                       s(log(Turb), bs="ts") +
                       s(log(Chl), bs="ts") +
                       s(log1p(RH),bs="ts"),
                     data = base_data, family = Gamma(link = 'log'))
summary(acartia_gam_g)
#>
#> Family: Gamma
#> Link function: log
#>
#> Formula:
#> I(Acartia + 1) ~ Station + Yearf + 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)   8.8523     0.4296  20.607  < 2e-16 ***
#> Station2     -1.1077     0.4823  -2.297 0.027042 *
#> Station3     -0.5142     0.4432  -1.160 0.252965
#> Station4     -0.8168     0.4847  -1.685 0.099903 .
#> Yearf2014    -1.6067     0.3512  -4.574 4.66e-05 ***
#> Yearf2015    -2.8605     0.3252  -8.797 7.72e-11 ***
#> Yearf2016    -1.4109     0.3315  -4.256 0.000125 ***
#> Yearf2017    -2.0332     0.3361  -6.049 4.25e-07 ***
#> ---
#> 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)      3.8380      9 2.717 0.000176 ***
#> s(Sal)       1.5313      9 4.592  < 2e-16 ***
#> s(log(Turb)) 0.7718      9 0.556 0.010964 *
#> s(log(Chl))  1.0303      9 0.493 0.027402 *
#> s(log1p(RH)) 3.4076      9 1.961 0.001333 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.215   Deviance explained = 79.5%
#> GCV = 0.90209  Scale est. = 0.47209   n = 58
```

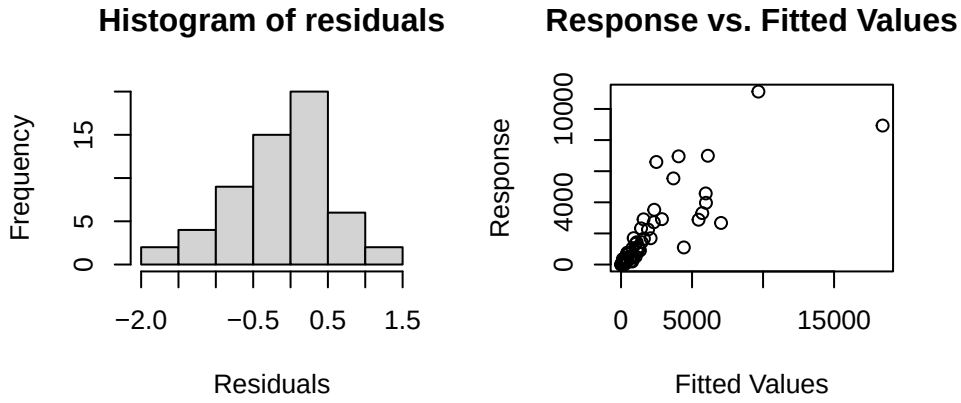
```
oldpar <- par(mfrow = c(2,3))
plot(acartia_gam_g)
par(oldpar)
```




1. Acartia are least abundant in cooler / colder water (Seasonal? position in estuary?)
2. Acartia may be less abundant in the fresher water sections of the estuary (but there's that low salinity sample to worry about again!).
3. Acartia likes those high turbidity, high chlorophyll waters
4. Acartia abundance drops when river herring abundance is high.

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




```
#>
#> Method: GCV   Optimizer: outer newton
#> full convergence after 7 iterations.
#> Gradient range [-2.898351e-09,3.600757e-10]
#> (score 0.902093 & scale 0.4720865).
#> Hessian positive definite, eigenvalue range [0.001879831,0.03402407].
#> Model rank =  53 / 53
#>
#> 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)      9.000 3.838    1.13    0.92
#> s(Sal)       9.000 1.531    1.08    0.81
#> s(log(Turb)) 9.000 0.772    1.03    0.68
#> s(log(Chl))  9.000 1.030    0.94    0.41
#> s(log1p(RH)) 9.000 3.408    0.95    0.42
par(oldpar)
```