Example GAM Analysis of Data From Penobscot Plankton Study

Curtis C. Bohlen, Casco Bay Estuary Partnership

1/28/2022

Contents

Introduction

This notebook is an effort to explore some of the changes in data analysis suggested by the reviewers.

It 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.

The notebook provides examples of "full" analysis workflow, and explanation of the why and the how for major data analysis decisions. The goal is to provide examples and logic to guide remaining analyses.

Load Libraries

```
library(tidyverse)
library(readxl)
library(mgcv)      # for GAM models
#library(lme4)      # For negative binomial generalized linear model
library(emmeans)   # For extracting useful "marginal" model summaries
library(GGally)    # For convenient Pairs plotting

theme_set(theme_classic())
```

Folder References

I use folder references to allow limited indirection, thus making code from GitHub repositories more likely to run "out of the box".

```
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),
```

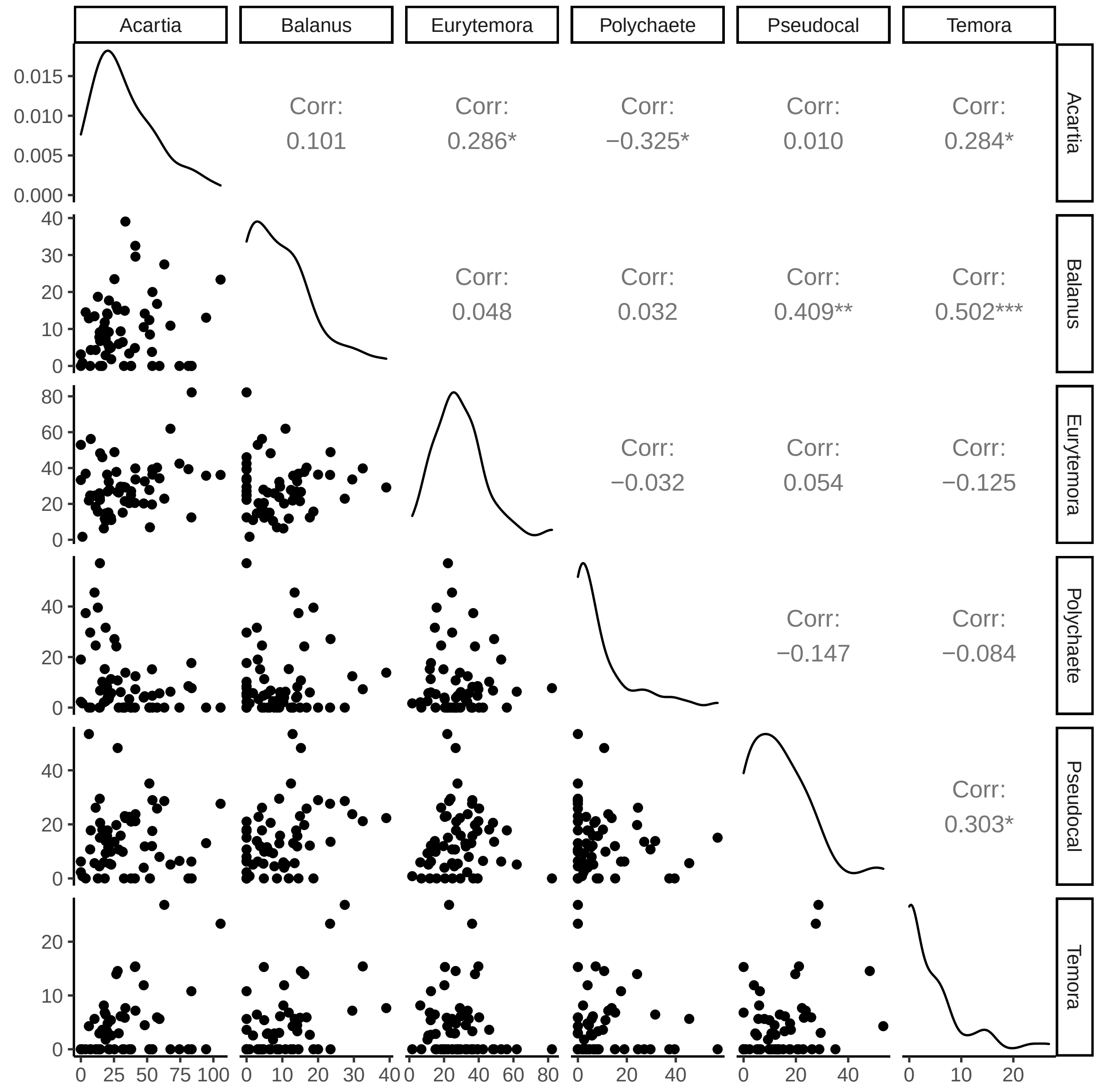
```r
         DOY = as.numeric(format(Date,format = '%j'))) %>%
  mutate(season = factor(season, levels = c('Spring', 'Summer', 'Fall'))) %>%
  rename(Season = season,
         Temp = ave_temp_c,
         Sal = ave_sal_psu,
         Turb = sur_turb,
         AvgTurb = ave_turb_ntu,
         DOsat = ave_DO_Saturation,
         Chl = ave_chl_microgperl,
         RH = Herring
         ) %>%
  select(Date, Station, Year, 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 23
#>   Date                Station  Year Month Season   DOY riv_km  Temp     Sal
#>   <dttm>              <fct>   <dbl> <fct> <fct>  <dbl>  <dbl> <dbl>   <dbl>
#> 1 2013-05-28 00:00:00 1        2013 May   Spring   148  22.6  11.7   0.0200
#> 2 2013-05-28 00:00:00 2        2013 May   Spring   148  13.9   9.40 14.6
#> 3 2013-05-28 00:00:00 3        2013 May   Spring   148   8.12  6.97 24.7
#> 4 2013-05-28 00:00:00 4        2013 May   Spring   148   2.78  9.51 12.7
#> 5 2013-07-25 00:00:00 1        2013 Jul   Summer   206  22.6  18.5  16.0
#> 6 2013-07-25 00:00:00 2        2013 Jul   Summer   206  13.9  13.6  27.0
#> # ... 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>
```

```r
rm(station_data)
```

Quick Data Review

Species Abundances

```r
base_data %>%
  select(Acartia:Temora) %>%
  mutate(across(.fns = sqrt)) %>%
  ggpairs(progress = FALSE)
```

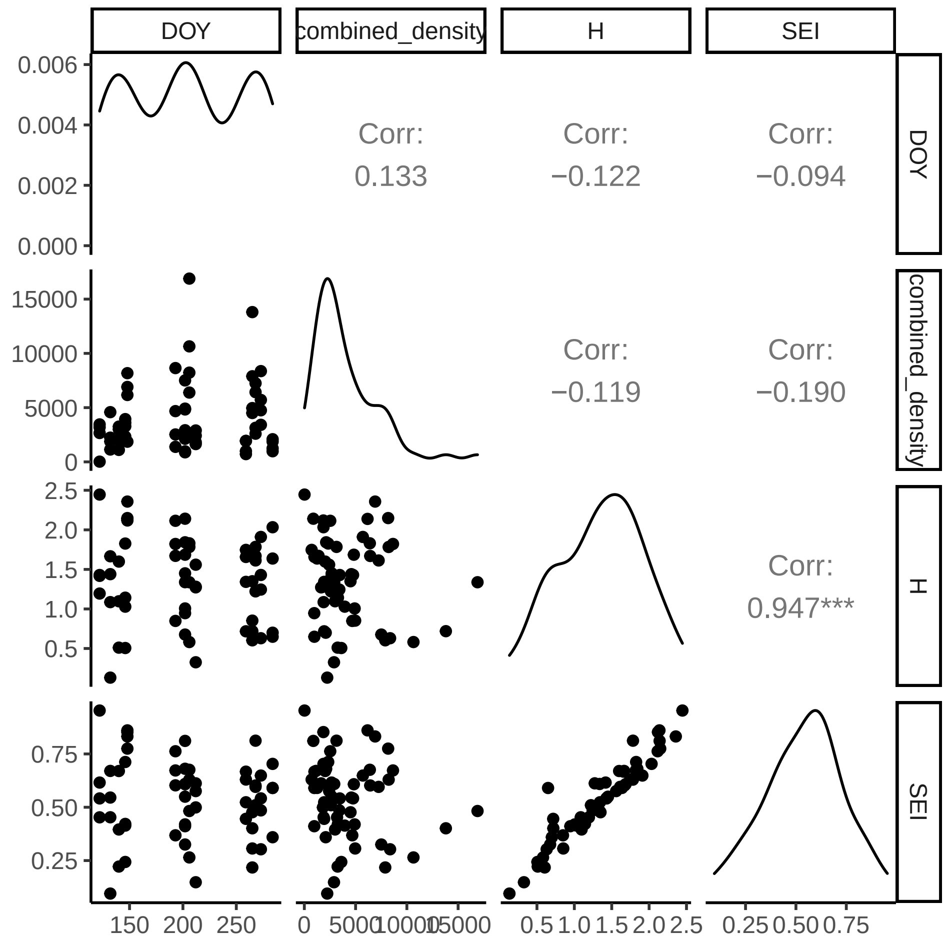


I note the abundant "zero" values for some species. That may pose challenges for model development.

Abundance and Diversity Metrics

It would be nice to include Species richness here as well, for completeness,. Given how highly correlated H and SEI are, there is little point in modeling both.

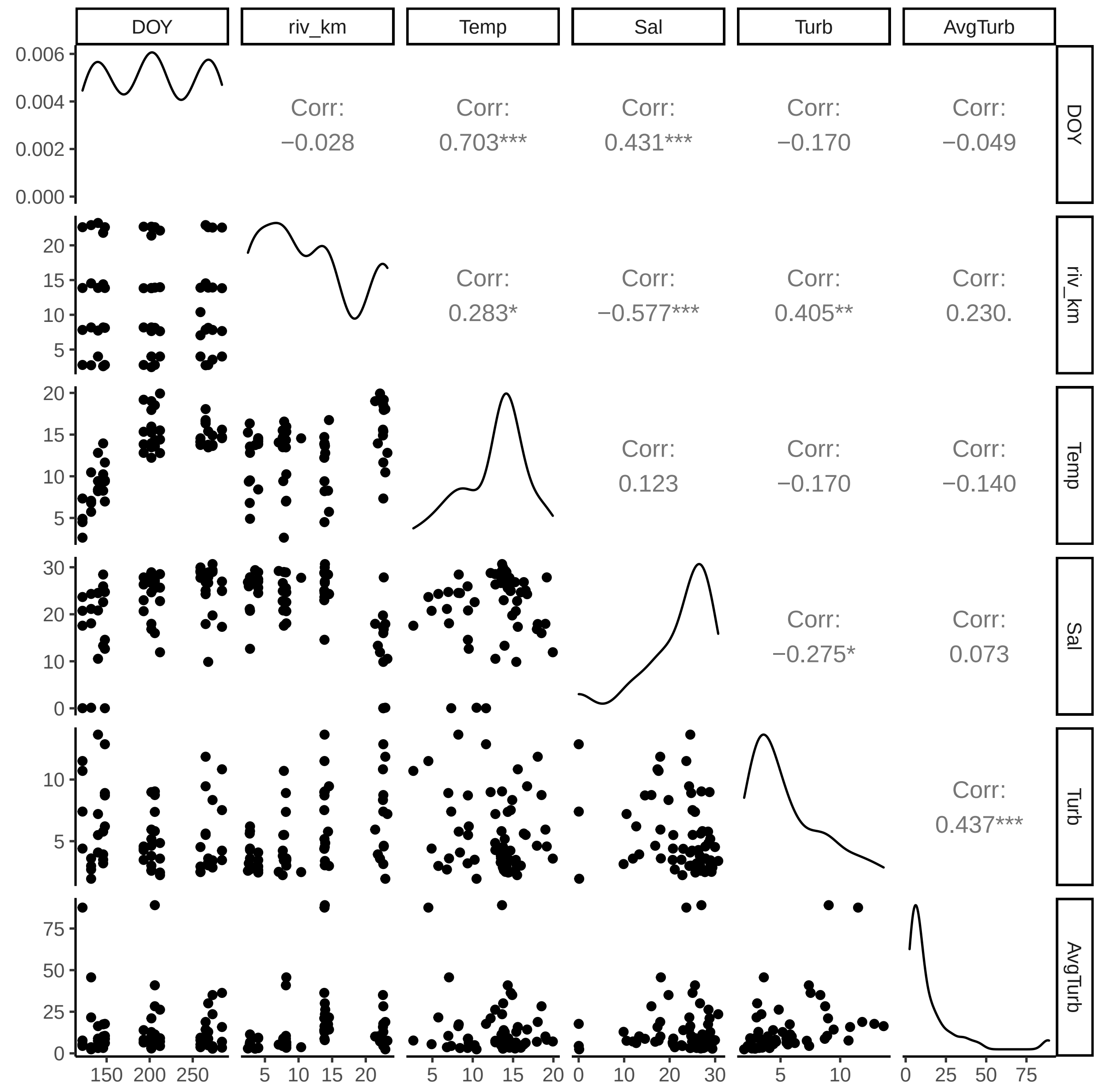
```
base_data %>%
  select(DOY, combined_density:SEI) %>%
  ggpairs(progress = FALSE)
```



There are no obvious correlations with time of year, but the diversity metrics are correlated. H and SEI are measuring nearly the same thing.

Predictors

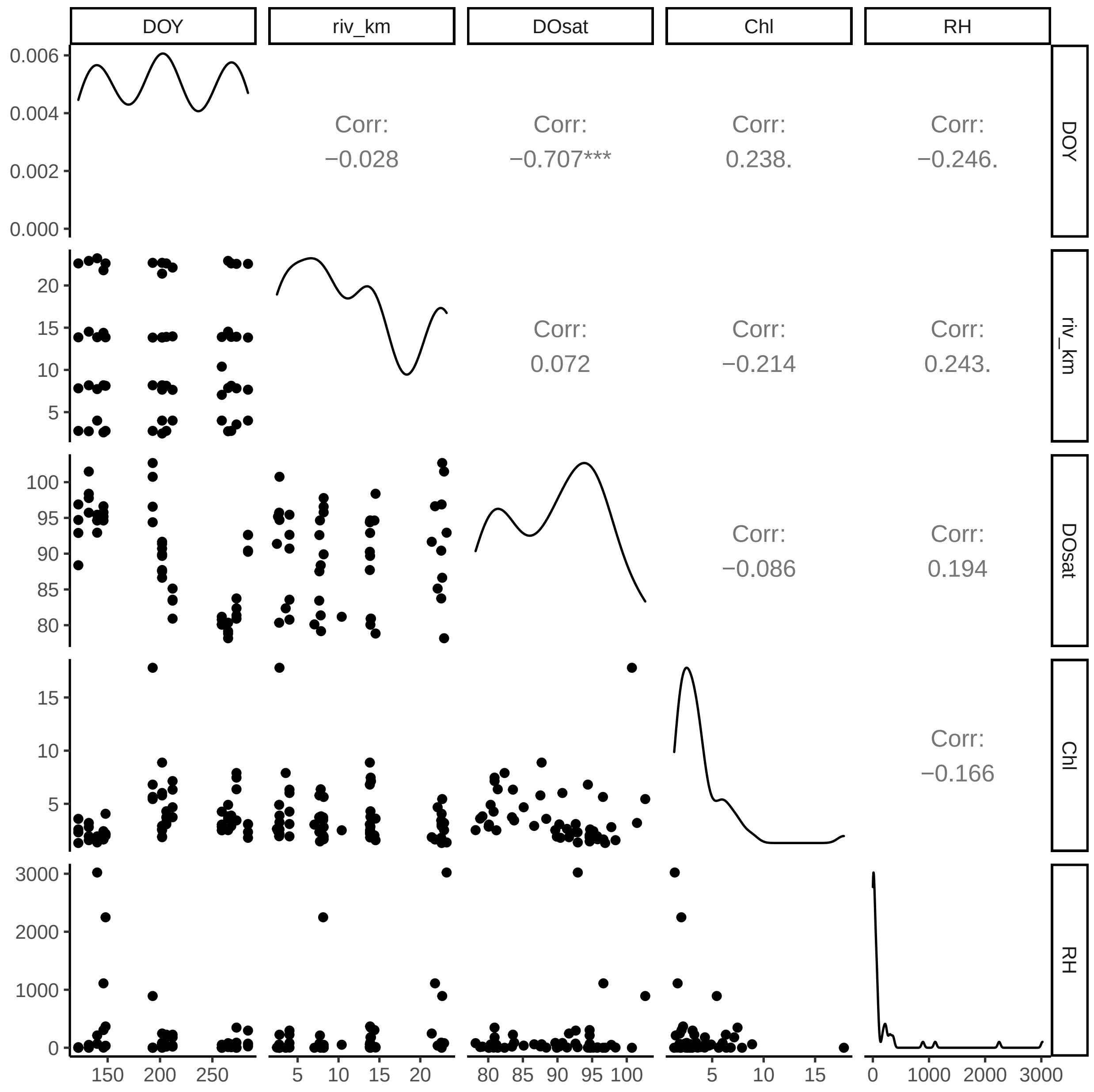
```
base_data %>%
  select(DOY:AvgTurb) %>%
  ggpairs(progress = FALSE)
```



Note the following:

1. Salinity is dominated by a handful of early spring low salinity observations.
2. As expected, river kilometer is negatively correlated with salinity and positively correlated with temperature.
3. Correlations with time of year also reflect expected patterns in Maine estuaries, with warmer fresh water interacting with cooler, saltier ocean waters.
4. Surface Turbidity is, in fact, much less highly skewed than average turbidity. The two are only moderately correlated. Both will benefit from transformation.

```
base_data %>%
  select(DOY, riv_km, DOsat:RH) %>%
  ggpairs(progress = FALSE)
```



There is one very high chlorophyll value.... River herring data is HIGHLY skewed, so it may actually be better modeled via discrete categories (e.g., absent, low, high).

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 found that helpful, in a "belts and suspenders" kind of way and in some cases, I think an alternative model specification (other than Gaussian models), makes better sense.

General Modeling Logic and Considerations

What are GAM models?

GAM models are a generalization of linear models. Traditional linear models are based on "least squares", which is equivalent to an assumption that the random components of the model are normally distributed. Linear models can include a variety of non-linear predictors, so it is possible to fit curved relationships of certain classes.

The first generalization of linear models is the "Generalized Linear Model". GLMs essentially allow extension of linear model logic to data where the random component of the model follows alternative random distributions. Actually, there is even more flexibility possible via GLMs that that suggests, but the details are confusing.

The second generalization of linear models allows fitting what are essentially arbitrary smooth non-linear relationships. It's the addition of this technology on top of GLMs that leads to GAMS.

So, to use GAMs wisely, we need to use all of the usual thinking about multiple regression models, and then in addition, think about

1. The form of the random component of the models (family, link etc.);
2. Which terms in the model enter as "linear" terms and which as "smoothed" terms;
3. How "wiggly" we want any smoothed terms to be.

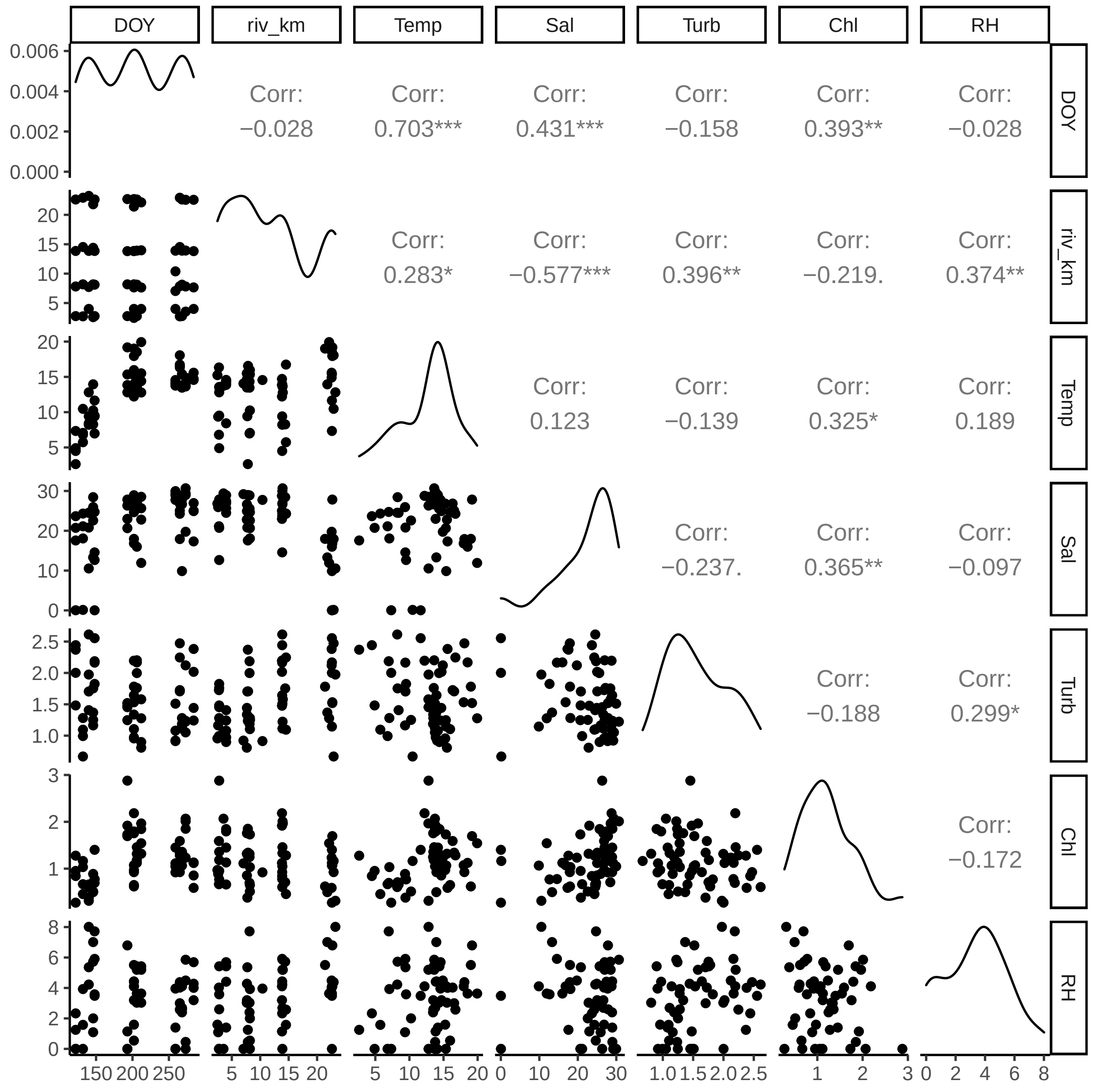
Examining Predictors

Because GAM models fit non-linear terms based on local curve fitting (using splines), areas with sparse data can be very poorly fit. When those areas of the model are near the exterior of the model space (e.g., very high or low values for one predictor), they can have outsized influence on model fit. As a result, it is often good practice to transform highly skewed predictors so that observations are more evenly scattered throughout the model space.

(That may or may not make scientific sense, and scientific criteria should always be given precedence to mere statistical convenience.)

I present transformed predictors, so I can look at the way these terms would actually enter into the GAM models. The goal here is to evaluate how well the transformed values may serve as predictors in a GAM model. The choice of transforms is somewhat arbitrary. There are methods to automate selection of transforms, but I just squint at the data and pick one from my bag of tricks. . . .

```
base_data %>%
  mutate(Turb = log(Turb),
         Chl = log(Chl),
         RH = log1p(RH)) %>%
  select(DOY, riv_km, Temp, Sal, Turb, Chl, RH) %>%
ggpairs( progress = FALSE)
```



Salinity is still a potential problem because we have relatively few low or intermediate salinity observations. Any non-linear fit is going to be extrapolating from limited information. I don't think any biologically reasonable transform will solve the problem.

Several potential predictors are correlated, which may complicate modeling and interpretation. We need to consider which predictors are most useful for sharing results with readers.

Repeated Measures / Mixed models?

We need to represent sample locations in the model somehow, as 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 and a factor in the model. We can't leave station out of the model entirely.

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

If our primary interest is in the biophysical predictors, Station should be included as random factors, as they are not intrinsically of interest. I would follow that strategy if I included river kilometer in the model. Otherwise, I would include Station in the models, because one way or the other, we are interested in the effect of location in the estuary.

For all the models that follow, I keep location in the model, rather then specify them as random factors.

Modeling Interactions

I'm not sure there is any reason we should believe that predictors behave in similar ways across the estuary. For example, a temperature of 18 C could be spring in the upper estuary or summer in the lower estuary. Plankton communities show strong seasonal patterns, so temperature ends up in some weird, complex interaction with position that could confound any temperature signal we might be looking for.

So, we should be thinking about ways to depict or model that complexity. But it turns out it is fairly complex to model interactions in GAM models. For now, I follow the structure of the models in the manuscript, by **not** including any interaction terms.

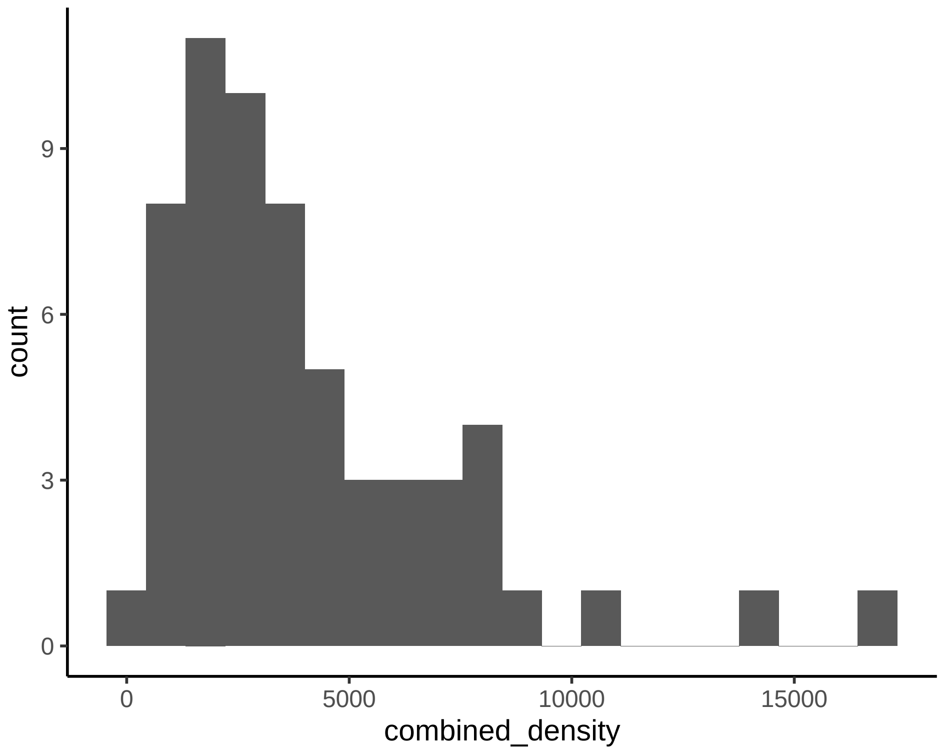
Total Density Data

I'm going to go through a thorough analysis of alternative models for this data, mostly to demonstrate the logic I use under these circumstances. This is probably one of the most complicated response variables to model, because of it's form and distribution. Count data is **never** normally distributed. Abundance data seldom is. . . .

Data Review

Histogram

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

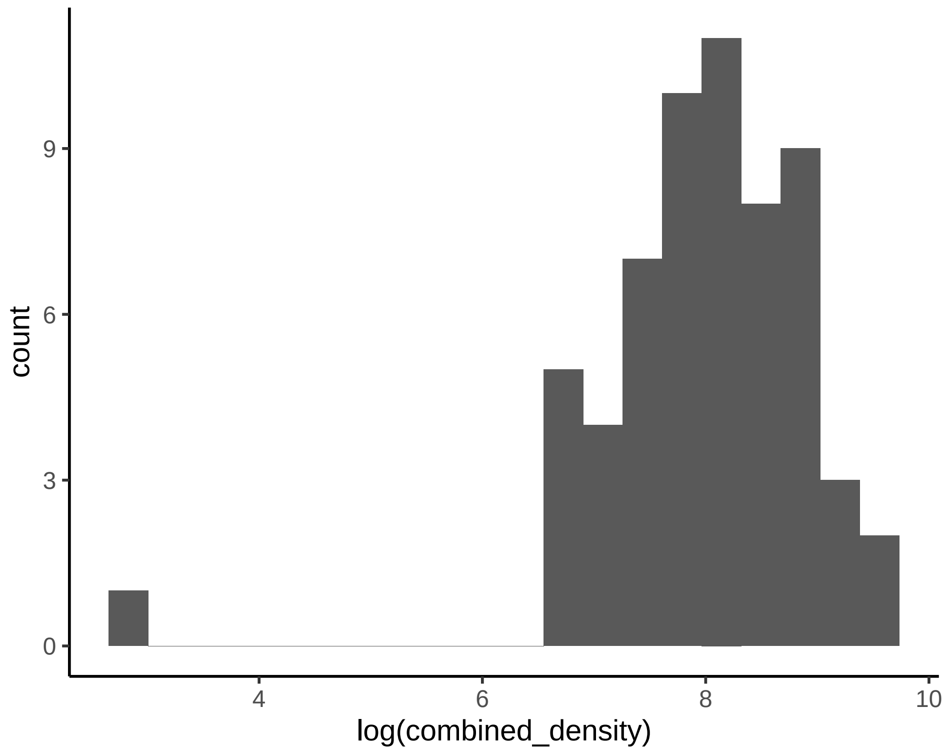


Looking at observed values can only give a hint at appropriate model strategies, since the random component of a model should match the residuals, not the raw observations. If predictors are skewed, the observations may be skewed even if errors are not. This certainly does not look like a normal distribution, 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 23
#>   Date                Station  Year Month Season   DOY riv_km  Temp   Sal  Turb
#>   <dttm>              <fct>   <dbl> <fct> <fct>  <dbl>  <dbl> <dbl> <dbl> <dbl>
#> 1 2014-05-02 00:00:00 1        2014 May   Spring   122   22.6  7.33  0.03   7.4
#> # ... with 13 more variables: 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 Predictrs

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

In particular, I did not try to fit any interaction terms between predictors, although from a scientific perspective some interactions – like Salinity - Temperature and Chlorophyll - Turbidity might be enlightening, in the context of estuary processes. (I'm not ready to go there yet, until we have had a chance to think through the way to revise discussion of estuary processes in the paper.)

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 Term
  - 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.
- Smoothed Terms
  - Temperature (untransformed)
  - Salinity (untransformed) but as you will see, there are still problems
  - log(Turbidity)
  - log(Chlorophyll)
  - s(log(River Herring + 1). I had to add one to deal with zero counts.

The choice of transforms here DO alter the models. They are not "neutral" choices. For example, as the models reduce to linear terms, the terms are linear in the transformed predictors, not linear in the underlying predictors.
It is appropriate to ask the scientific question of whether the untransformed values would be better predictors –or scientifically more relevant predictors – of zooplankton than the transformed ones. For example, predation intensity is presumably a function of total herring abundance, not log of herring abundance.

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. Shrinkage estimators play a similar role, by providing an automated way to evaluate the importance of each predictor (without just asking if they are statistically significant).

Each smoothed term enters the GAM model through a smoothing function. In it's simplest form this is just `s(x)`. To trigger a "shrinkage estimator", you modify this slightly, by specifying the "basis function" on which the smooths are calculated. Here I do that with `s(x, bs = 'ts'). The`bs = 'ts'' specifies shrinkage estimates using "thin plate" splines. There are several other reasonable choices for basis functions. I have not explored them, because the thin plate splines are widely used, tend to fit relatively simple curves (especially when fit as shrinkage estimators).

Specification of "wiggliness" While we are talking about ways we can control the smoothers, there is one more function parameter I used once or twice. Smoothers have to be told just how "wiggly" we want them to be. The mathematical details are a bit complex, because the mathematics of splines is complex (and different basis functions do things slightly differently). But you choose "wiggliness" by specifying how many "knots" are used in calculating the splines, and pass that info the the `s(x)` function via the parameter `k`.

It appears `mgcv` defaults to asking for ten knots (or equivalently, a maximum of nine degrees of freedom for each smoother). The actual splines, especially the shrinkage splines, will fit less complicated relationships because "wiggliness" is itself penalized during model fitting. Even so, in my experience, the default sometimes fits "wiggly" relationships that are almost uninterpretable in ecological terms. We expect relatively simple smooth relationships between most predictors and zooplankton abundance.

So, I take a brute force approach, and if the default gives me a fit that strikes me as too wiggly, I specify a lower value for K. That's not very objective. So it may be better at the start to choose some fixed upper bound for `k`, say `k = 5` and use that consistently. But I like to see what the default shows, because it often points to unusual observations that are having a large effect on model fitting. That helps me evaluate whether the model is being "deceived" by one or two observations.

Model Alternatives

Count data is based on discrete probability distributions – that is integer values only. Your "Density" data is not integer valued, so the methods I discussed on the call (negative binomial and quasiposson GAMs) are not going to be directly applicable.

We are, however, looking at values (abundance) over the positive number line. So we should principally look at models that also 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 principal model strategies:

1. A simple Gaussian GAM. Gaussian distributions technically span the entire real number line, so there is always SOME probability mass below zero. But If mean densities are high enough, and the standard error low enough, the effective probability of observations below zero is so tiny as to not matter. But on first principals, it's 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 low counts. That's not reasonable for these data.
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, log(x+ 1), square root, or a "Box-Cox" transform (Which is really a recipe for choosing a transform). Other than the log transform, other likely transforms also produce models with non-zero probability mass associated with negative numbers, so we need to make sure they are "good enough". Here I look at log transforms and square root transforms. Both 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.
3. We can use a GAM with Gaussian error and the log link. This is similar to, but not identical to a Gaussian model on log-transformed data. This model is **also** a poor choice, because it assumes deviations are roughly the same size for low and high densities.
4. 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, fitting arithmetic means
    - a log link, effectively fitting geometric means
    - an inverse link (related to the "canonical link"), fitting harmonic means

   I only test the inverse link here, as it is the default in `mcgv`. But the alternaives might be interesting too. There are just too many damn choices in GAMs. . . .

5. I also considered more exotic forms of a GLM / GAM. Statisticians have gotten very clever about devising alternative error distributions. I won't go into details, but a one could consider "inverse Gaussian", "tweedie", or "zero inflated" models of various kinds, none of which look all that likely to be helpful here, although some may be useful modeling individual species.

So, on first principals, I like the transformed Gaussian and Gamma models.

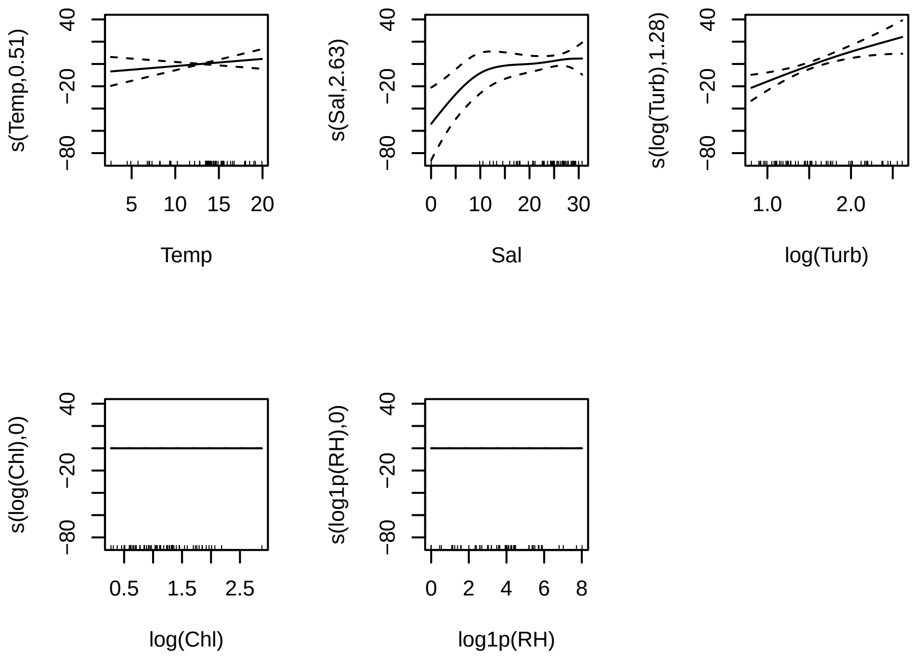
Testing Different Models

Square Root Transform

This is an *ad hoc* transform that just often happens to work well with abundance data. No great theoretical justification. . . .

```
combined_density_gam_sr <- gam(sqrt(combined_density) ~
                                 Station +
                                 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_sr)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> sqrt(combined_density) ~ Station + 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)  64.7773     7.5310   8.601 2.12e-11 ***
#> Station2    -13.0556    10.7292  -1.217    0.229
#> Station3      0.5599     9.9426   0.056    0.955
#> Station4     -4.9895    10.4530  -0.477    0.635
#> ---
#> 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)      5.053e-01      9 0.119 0.149144
#> s(Sal)       2.630e+00      9 1.563 0.006421 **
#> s(log(Turb)) 1.276e+00      9 1.780 0.000233 ***
#> s(log(Chl))  6.022e-10      9 0.000 0.362732
#> s(log1p(RH)) 3.913e-10      9 0.000 0.511160
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.279   Deviance explained = 37.3%
#> GCV = 457.48  Scale est. = 391.14    n = 58
```

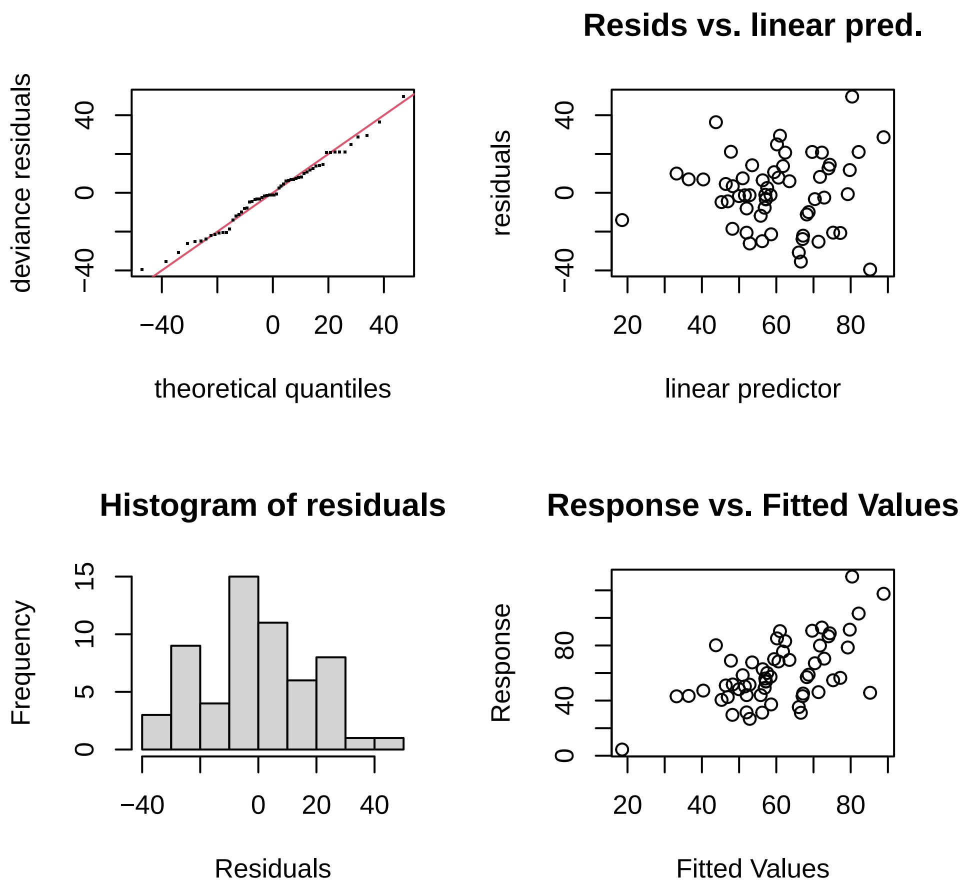
```r
oldpar <- par(mfrow = c(2,3))
plot(combined_density_gam_sr)
par(oldpar)
```



This model suggests:

1. Low salinity sample has low abundance. otherwise, salinity does not much matter.
2. Abundance increases with Turbidity.

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



```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 23 iterations.
#> The RMS GCV score gradient at convergence was 2.250976e-05 .
#> The Hessian was positive definite.
#> Model rank =  49 / 49
#>
#> 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 5.05e-01    0.95    0.32
#> s(Sal)       9.00e+00 2.63e+00    0.96    0.37
#> s(log(Turb)) 9.00e+00 1.28e+00    1.03    0.57
#> s(log(Chl))  9.00e+00 6.02e-10    1.03    0.53
#> s(log1p(RH)) 9.00e+00 3.91e-10    0.92    0.20
par(oldpar)
```

This model has excellent model diagnostics. Residuals are well distributed. No obvious model deficiencies.

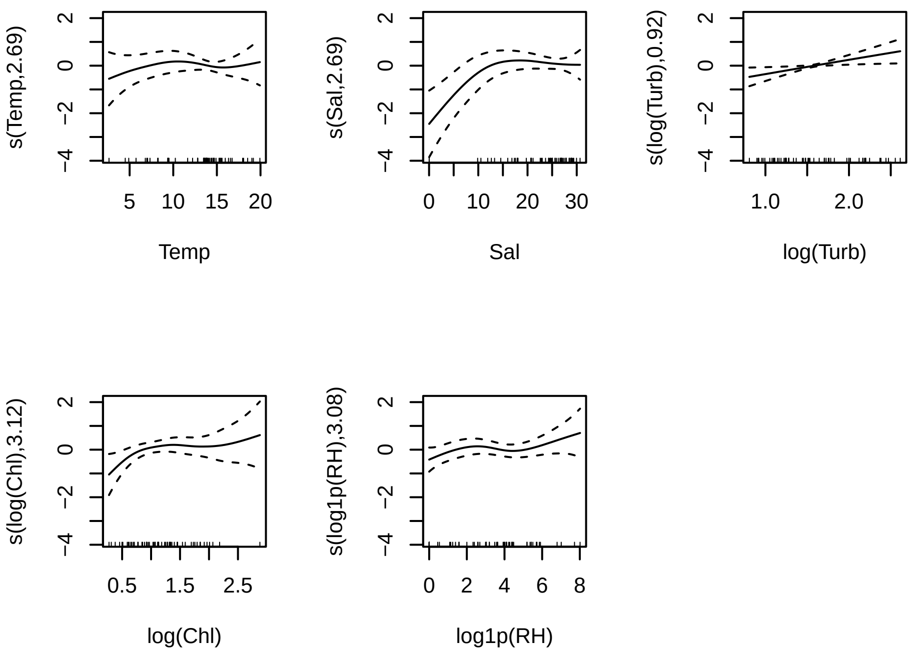
Log transform (lognormal model)

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 +
                              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 + 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)  8.09252    0.35252  22.956   <2e-16 ***
#> Station2    -0.22775    0.49259  -0.462    0.646
#> Station3     0.06739    0.45999   0.147    0.884
#> Station4    -0.13071    0.48785  -0.268    0.790
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>                 edf Ref.df     F p-value
#> s(Temp)      2.6929      9 0.235 0.49466
#> s(Sal)       2.6931      9 1.714 0.00117 **
#> s(log(Turb)) 0.9176      9 0.663 0.00967 **
#> s(log(Chl))  3.1209      9 0.738 0.07354 .
#> s(log1p(RH)) 3.0844      9 0.623 0.11339
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.461   Deviance explained = 60.8%
#> GCV = 0.71481  Scale est. = 0.51135   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)
```

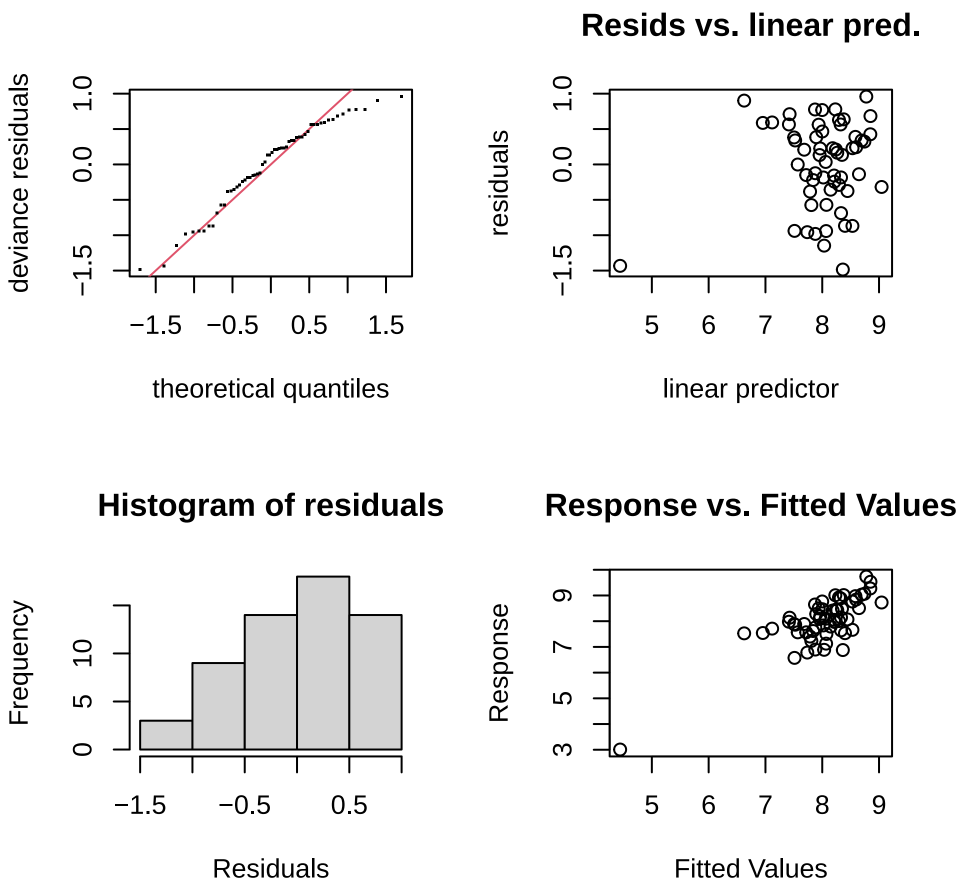


This model suggests:

1. Low salinity observation is different
2. Abundance increases with turbidity
3. Chlorophyll *may* have some additional effect, but it is only marginally significant, so we don't want to over-interpret.

It is interesting that this model did not zero out any of the predictors. That suggests ALL are useful for predicting zooplankton abundance, even if some are not "significant".

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



```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 8 iterations.
#> The RMS GCV score gradient at convergence was 1.12873e-07 .
#> The Hessian was positive definite.
#> Model rank =  49 / 49
#>
#> 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 2.693    0.91    0.21
#> s(Sal)         9.000 2.693    1.19    0.92
#> s(log(Turb)) 9.000 0.918    0.99    0.44
#> s(log(Chl))  9.000 3.121    1.04    0.56
#> s(log1p(RH)) 9.000 3.084    1.11    0.74
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 very low abundance under low salinity conditions now has an outsized impact on the model fit, especially regarding the salinity predictor.

I don't much like this model.

Gamma Regression

The canonical link of the Gamma exponential family is the negative inverse, but R's GLM and related functions use the inverse. This can make interpretation of model parameters tricky.

The model effectively estimates harmonic means based on the predictor variables, as follows::

$$\frac{1}{E(Y)} = \beta_1 X_1 + \beta_2 X_2 + \dots$$

$$E(Y) = \frac{1}{\beta_1 X_1 + \beta_2 X_1 + \dots}$$

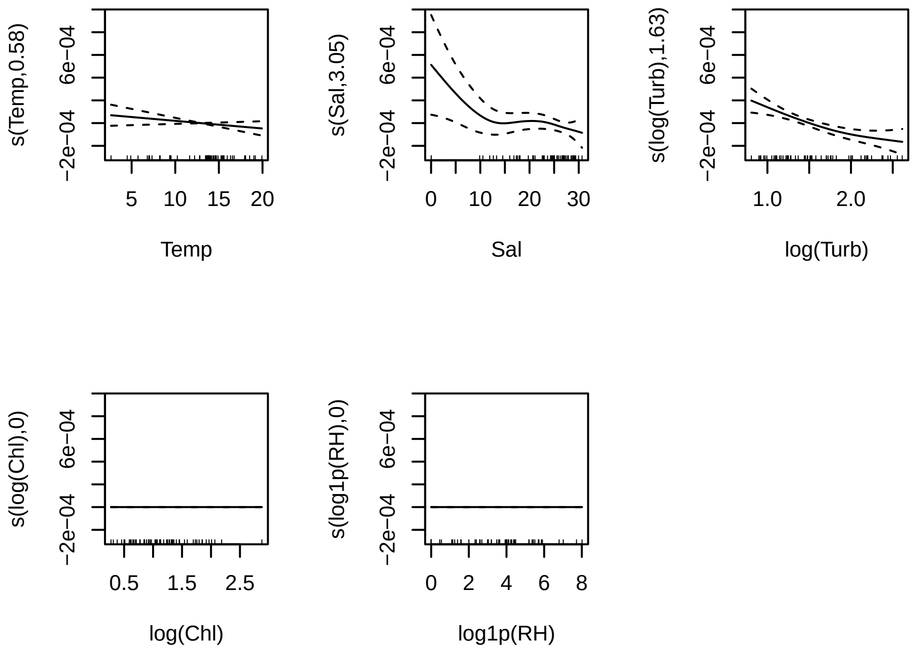
$$E(Y) = \frac{1}{\beta_1 X_1 + \beta_2 X_1 + \dots}$$

The error term is drawn from a gamma distribution. The particular deviation for each observation is dependent on the estimate, making it tricky to write out a formula for each observation. So I won't try because I'll probably get it wrong.

```
combined_density_gam_g <- gam(combined_density ~
                                Station +
                                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')
summary(combined_density_gam_g)
#>
#> Family: Gamma
#> Link function: inverse
#>
#> Formula:
#> combined_density ~ Station + 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)  2.711e-04  5.118e-05   5.296 2.81e-06 ***
#> Station2     8.833e-05  7.198e-05   1.227    0.226
#> Station3    -4.316e-05  6.567e-05  -0.657    0.514
#> Station4     1.672e-05  7.486e-05   0.223    0.824
#> ---
#> 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)      5.789e-01      9 0.249   0.0493 *
#> s(Sal)       3.049e+00      9 0.799   0.0593 .
#> s(log(Turb)) 1.629e+00      9 2.186 5.01e-05 ***
#> s(log(Chl))  1.184e-05      9 0.000   0.4354
#> s(log1p(RH)) 8.180e-06      9 0.000   0.4491
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.445   Deviance explained = 37.2%
#> GCV = 0.5638  Scale est. = 0.3217    n = 58
```

Warning! The following graphics need to be interpreted "upside down" because of of the inverse link function used by default for a Gamma GAM.

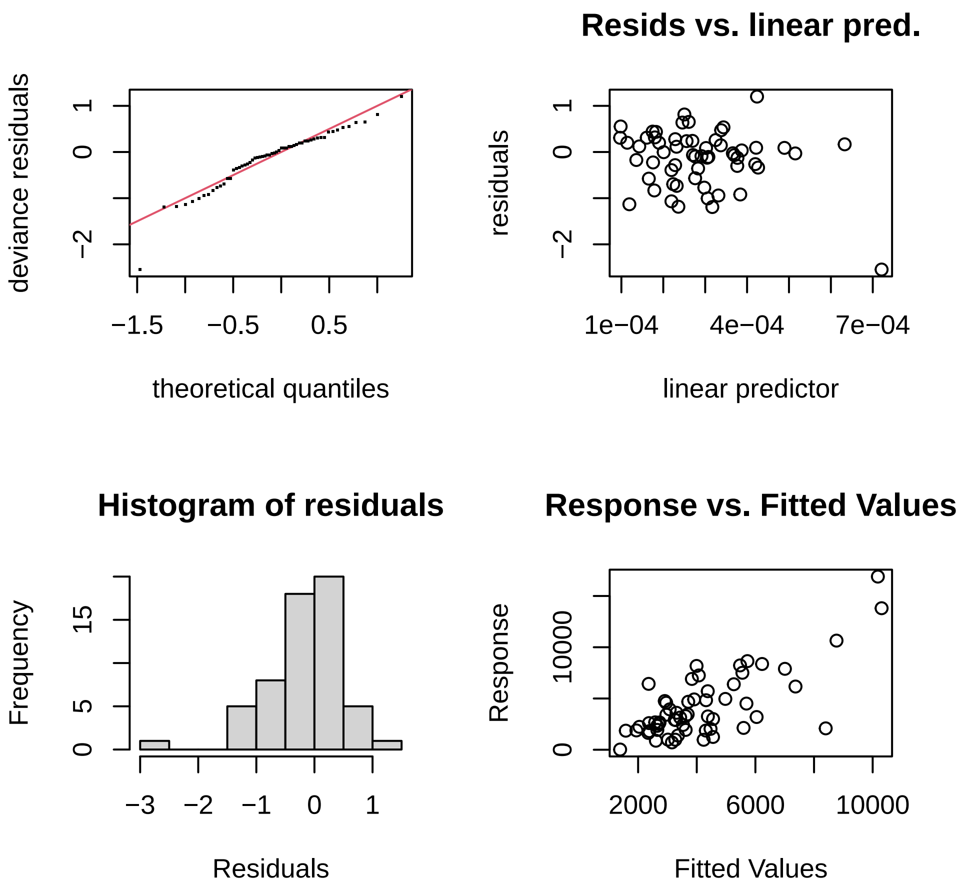
```
oldpar <- par(mfrow = c(2,3))
plot(combined_density_gam_g)
par(oldpar)
```



So, remembering that we have to think "upside down" here:

1. Abundance increases with temperature
2. (low salinity sample is different, but not quite significantly under this model)
3. Abundance increases with turbidity.

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



```
#>
#> Method: GCV   Optimizer: outer newton
#> full convergence after 12 iterations.
#> Gradient range [-1.785602e-07,1.13621e-08]
#> (score 0.5637988 & scale 0.321703).
#> Hessian positive definite, eigenvalue range [1.206943e-07,0.007855226].
#> Model rank =  49 / 49
#>
#> 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 5.79e-01    0.93    0.38
#> s(Sal)         9.00e+00 3.05e+00    0.98    0.54
#> s(log(Turb)) 9.00e+00 1.63e+00    0.96    0.47
#> s(log(Chl))  9.00e+00 1.18e-05    0.82    0.16
#> s(log1p(RH)) 9.00e+00 8.18e-06    0.77    0.04 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
par(oldpar)
```

This model has excellent distribution of (deviance) residuals. There is a tendency towards a scale-location relationship in the RAW (response - fitted) residuals, as shown in the last graphic. That reflects the nature of the gamma GAM, which assumes higher error at higher predicted values, and thus does not “try” to fit them as precisely. So, not a problem.

This is a good model for these data.

Conclusions from too many GAM Models

I don’t much like the lognormal model.

The other models are both reasonable. What is most important here, though, is that all models provide similar interpretations of the pattern in the data:

1. Abundance either is low under low temperatures, or increases with temperature.
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, if only based on a single sample.
3. Abundance increases strongly with turbidity.

Similar GLM Models?

With that background, we could fit similar GLM models to match any of the GAM models. The only advantage is that most people are familiar with interpreting regression coefficients drawn from a linear model, so it may help with communicating model results.

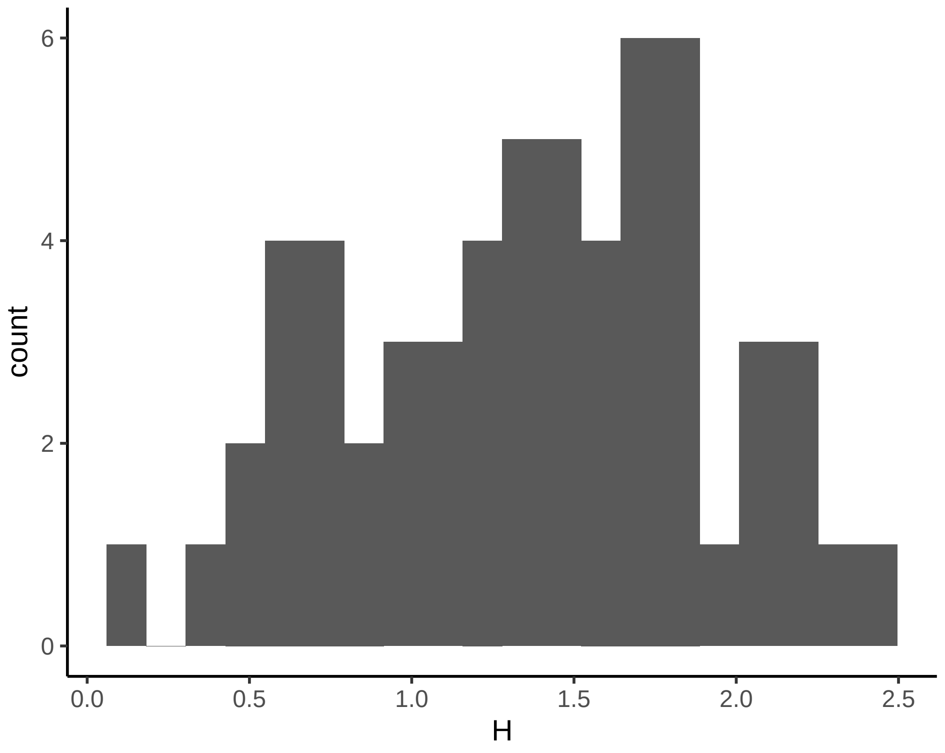
I think we are probably better off focusing on how to present results in some sort of a graphic (rather than tabular) form. I explore some possibilities in another notebook

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 family model here. But the Shannon index is a strictly positive value, so we may want to consider models that also restrict our predictions to positive values.

Histogram

```
ggplot(base_data, aes(H)) +
  geom_histogram(bins = 20)
#> Warning: Removed 1 rows containing non-finite values (stat_bin).
```



NO obvious reason to think about transforms of these data will be needed, so we forge on. But I'm feeling nervous about the possibility that some predicted values may be negative. . . .

Basic GAM

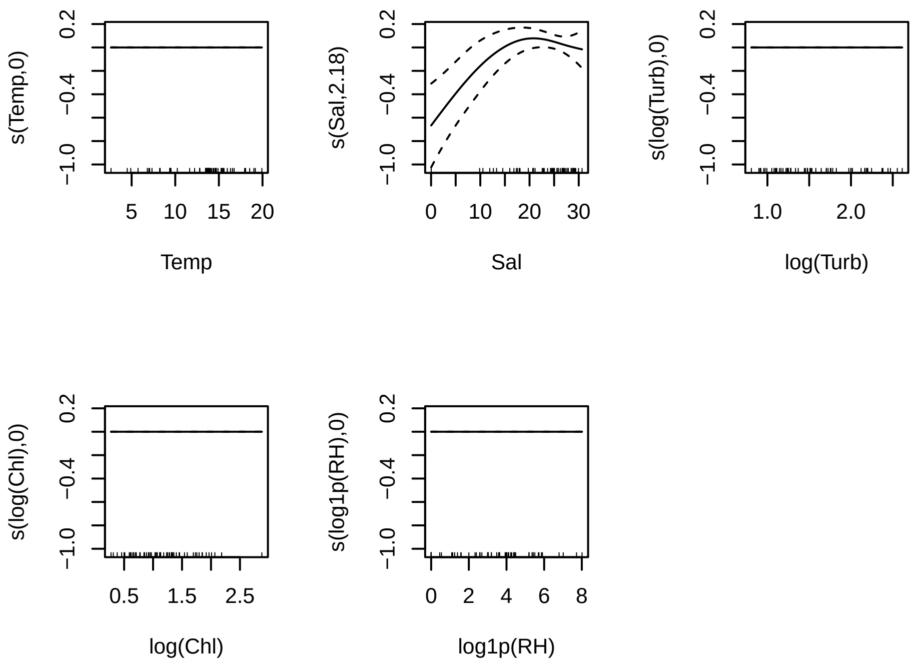
We are using "Shrinkage" estimates of the smoothing terms again, which allow certain terms to be "shrunk" out of the model

```
shannon_gam <- gam(H ~ Station +
                     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')
summary(shannon_gam)
#>
#> Family: Gamma
#> Link function: inverse
#>
#> Formula:
#> H ~ Station + 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)   1.1572     0.1339   8.642 1.28e-11 ***
#> Station2     -0.5050     0.1630  -3.098  0.00314 **
#> Station3     -0.5270     0.1587  -3.321  0.00165 **
#> Station4     -0.4687     0.1619  -2.895  0.00554 **
#> ---
#> 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)       6.137e-06      9 0.000 0.816757
#> s(Sal)        2.182e+00      9 1.865 0.000469 ***
#> s(log(Turb)) 1.357e-06      9 0.000 0.828088
#> s(log(Chl))  5.287e-06      9 0.000 0.396580
#> s(log1p(RH)) 5.597e-06      9 0.000 0.315766
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.279   Deviance explained = 26.9%
#> GCV = 0.20577  Scale est. = 0.13906   n = 58
```

Note that stations differ in diversity (coefficients are difference from Station 1, upstream). Stations did NOT differ when we looked at density.

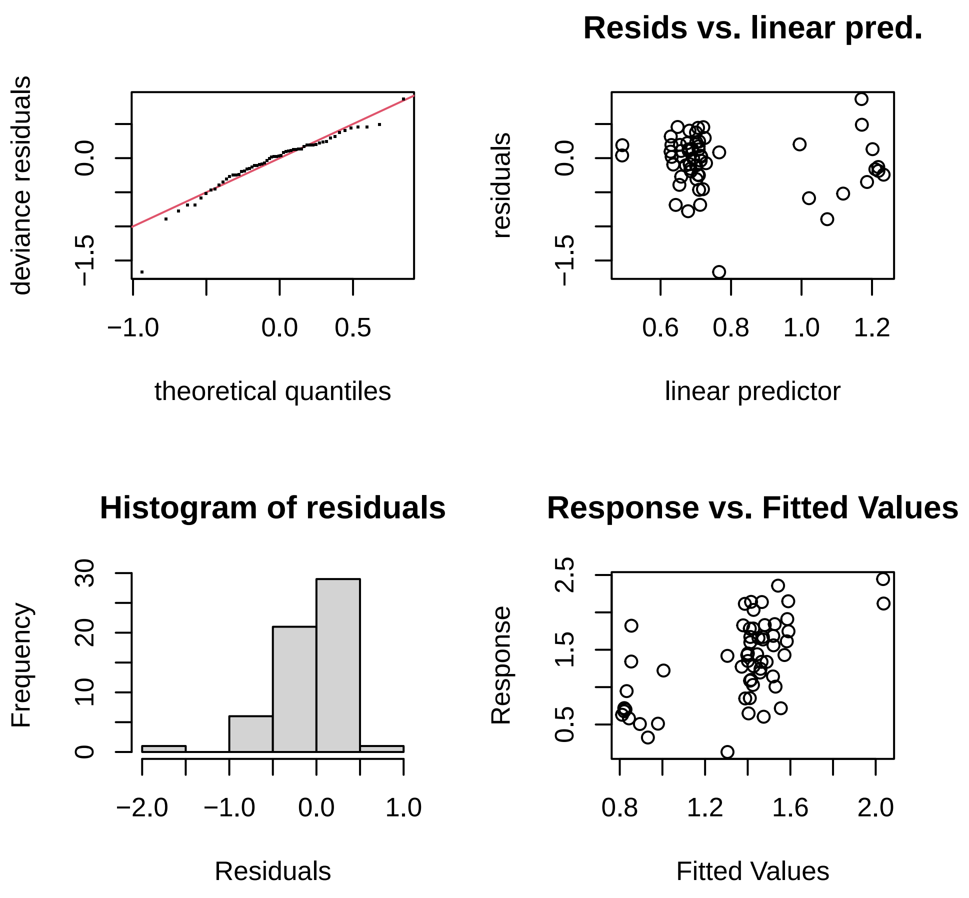
The only significant relationship between diversity and any of the predictors relates to salinity.

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



Again, I suspect the SHAPE Of that curve is highly influenced by that lowest salinity sample. I wonder if it is worthwhile dropping that sample from the analysis and seeing what impact that may have on the fit. . . .

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



```
#>
#> Method: GCV   Optimizer: outer newton
#> full convergence after 14 iterations.
#> Gradient range [-4.61012e-08,4.148135e-09]
#> (score 0.2057656 & scale 0.1390563).
#> Hessian positive definite, eigenvalue range [4.873519e-09,0.004304728].
#> Model rank =  49 / 49
#>
#> 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 6.14e-06    1.04    0.56
#> s(Sal)        9.00e+00 2.18e+00    0.94    0.37
#> s(log(Turb)) 9.00e+00 1.36e-06    1.26    0.98
#> s(log(Chl))  9.00e+00 5.29e-06    1.07    0.82
#> s(log1p(RH)) 9.00e+00 5.60e-06    1.10    0.78
par(oldpar)
```

That's a pretty good model. Again, the biggest weakness is the lack of low-salinity data, which I suspect is also the one very low residual.
The model puts a lot of weight on the few low salinity observations.

We could try more exotic models. But it's probably more informative to check predictions and look at how much they change if we drop that one very low salinity observation. I'll do that a little later in this notebook

SEI is highly correlated with H, so I don't bother showing the results of a separate analysis. They are essentially identical.

Drop the Wonky Low Salinity Observation.

Let's just see what happens if we drop that low salinity observation before modeling. This is NOT to "ignore" that sample, but to try to understand what it may be hiding in the rest of the data.

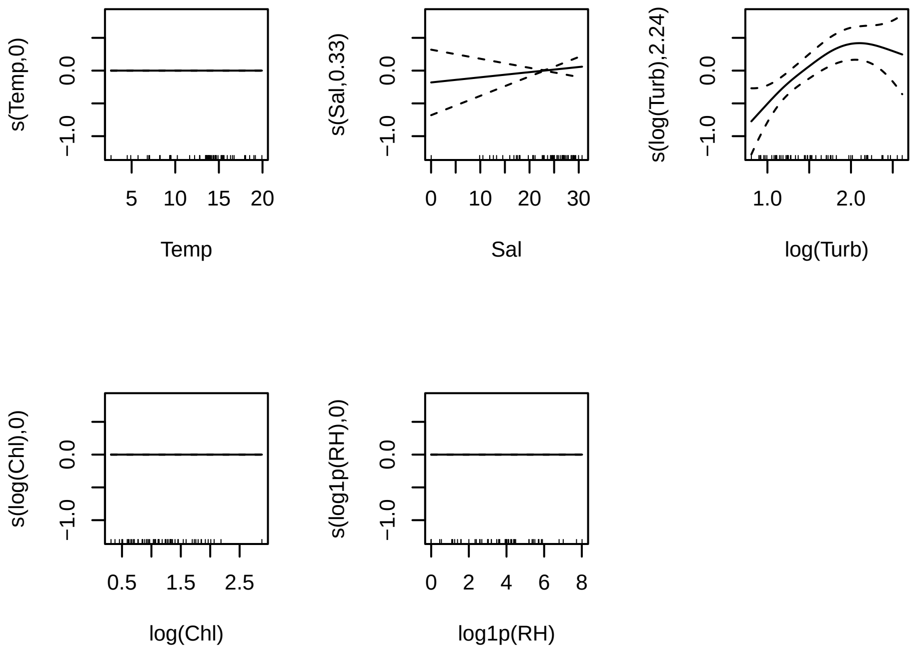
```
low_sample <- which(base_data$combined_density ==
        min(base_data$combined_density, na.rm = TRUE))
smaller_data <- base_data[-low_sample,]
```

Total Density

Without that low value, I like the log transformed analysis of TOtals...

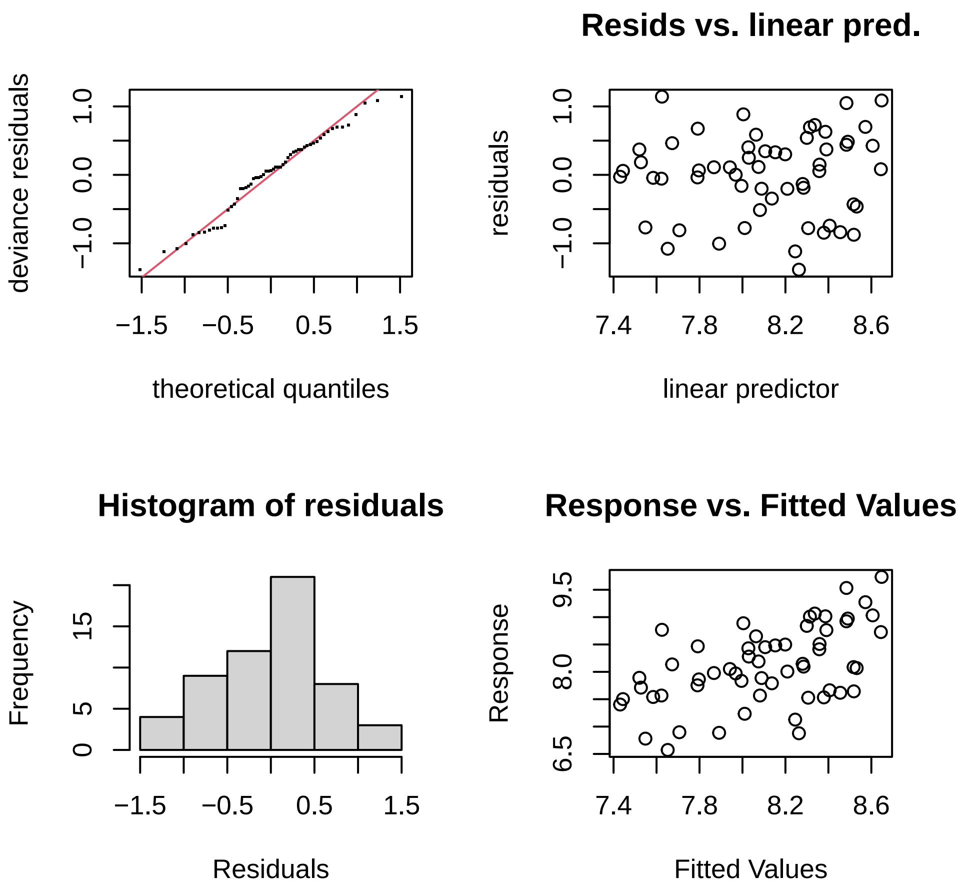
```
drop_density_gam_l <- gam(log(combined_density) ~
                              Station +
                              s(Temp, bs="ts") +
                              s(Sal, bs="ts") +
                              s(log(Turb), bs="ts") +
                              s(log(Chl), bs="ts") +
                              s(log1p(RH),bs="ts"),
                            data = smaller_data, family = 'gaussian')
summary(drop_density_gam_l)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log(combined_density) ~ Station + 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)  8.212825   0.194083  42.316   <2e-16 ***
#> Station2    -0.262049   0.266109  -0.985    0.329
#> Station3     0.006462   0.261717   0.025    0.980
#> Station4    -0.155380   0.277240  -0.560    0.578
#> ---
#> 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)      6.895e-08      9 0.000 0.391282
#> s(Sal)       3.334e-01      9 0.058 0.215422
#> s(log(Turb)) 2.240e+00      9 1.660 0.000891 ***
#> s(log(Chl))  1.550e-07      9 0.000 0.486535
#> s(log1p(RH)) 2.626e-08      9 0.000 0.679178
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.199   Deviance explained = 27.9%
#> GCV = 0.46172  Scale est. = 0.40847   n = 57
```

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



This model suggests that abundance increases with turbidity. Temperature MAY be useful as a predictor as well, but it's not "significant". Salinity no longer matters at all.

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



```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 25 iterations.
#> The RMS GCV score gradient at convergence was 5.025211e-08 .
#> The Hessian was positive definite.
#> Model rank =  49 / 49
#>
#> 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 6.89e-08    0.90    0.22
#> s(Sal)       9.00e+00 3.33e-01    0.99    0.48
#> s(log(Turb)) 9.00e+00 2.24e+00    1.18    0.90
#> s(log(Chl))  9.00e+00 1.55e-07    0.97    0.29
#> s(log1p(RH)) 9.00e+00 2.63e-08    0.95    0.31
par(oldpar)
```

That is an excellent model for the redused data set.

Diversity

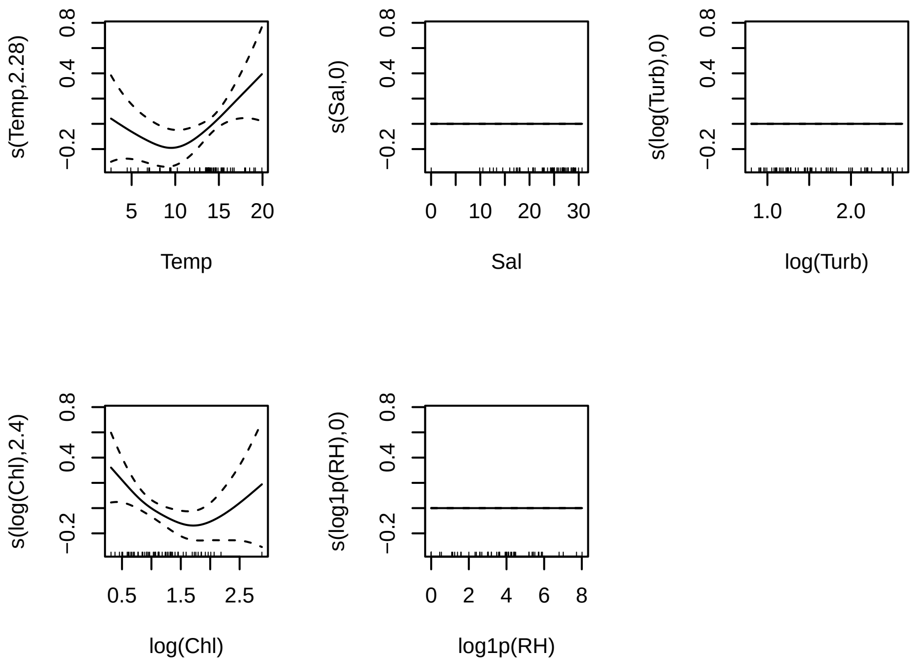
```
drop_shannon_gam <- gam(H ~ Station +
                      s(Temp, bs="ts") +
                      s(Sal, bs="ts") +
                      s(log(Turb), bs="ts") +
                      s(log(Chl), bs="ts") +
                      s(log1p(RH),bs="ts"),
                    data = smaller_data, family = 'Gamma')
summary(drop_shannon_gam)
#>
#> Family: Gamma
#> Link function: inverse
#>
#> Formula:
#> H ~ Station + 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)   0.9353     0.1159   8.069 1.66e-10 ***
#> Station2     -0.1821     0.1443  -1.262    0.213
#> Station3     -0.2186     0.1394  -1.567    0.124
#> Station4     -0.1500     0.1456  -1.030    0.308
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>                    edf Ref.df     F p-value
#> s(Temp)      2.282e+00      9 0.919  0.0179 *
#> s(Sal)       6.921e-06      9 0.000  0.5037
#> s(log(Turb)) 7.632e-06      9 0.000  0.4737
#> s(log(Chl))  2.404e+00      9 0.923  0.0175 *
#> s(log1p(RH)) 4.316e-06      9 0.000  0.2726
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.306   Deviance explained = 32.8%
#> GCV = 0.2068  Scale est. = 0.13359   n = 57
```

What is most interesting here is that now that we have dropped the one very low salinity sample, we get a completely different impression of how diversity varies with the other environmental predictors.

In the "whole dataset" GAM model, salinity was the only statistically significant predictor of diversity. No longer. Now both temperature and chlorophyll show significant associations with elevated diversity.

That feels ever so *ad hoc* to me, so I'm not sure it's legit to report results after dropping that sample, but the modified analysis helps demonstrate the difficulties presented by this data set.

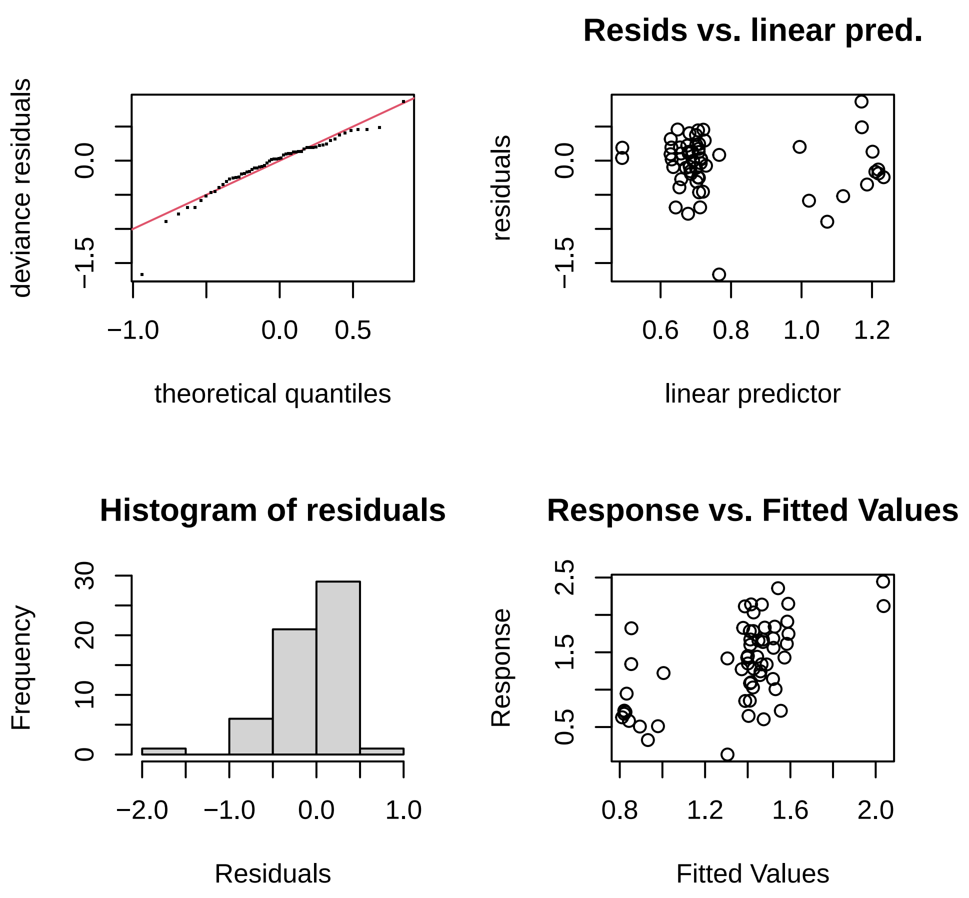
```
oldpar <- par(mfrow = c(2,3))
plot(drop_shannon_gam)
par(oldpar)
```



Temperature and chlorophyll matter, but the relationships are non-linear. The errors are broad enough at low temperature and at high chlorophyll so the appropriate interpretation is:

1. Diversity is lower at temperatures below about 10 C, but above 10 C, diversity increases in warmer waters.
2. Diversity tends to be higher at low chlorophyl levels, and drops as you approach intermediate chlorophyll levels. The data for high chlorophyll levels is too sparse to try to interpret.

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



```
#>
#> Method: GCV   Optimizer: outer newton
#> full convergence after 14 iterations.
#> Gradient range [-4.61012e-08,4.148135e-09]
#> (score 0.2057656 & scale 0.1390563).
#> Hessian positive definite, eigenvalue range [4.873519e-09,0.004304728].
#> Model rank =  49 / 49
#>
#> 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 6.14e-06    1.04    0.62
#> s(Sal)        9.00e+00 2.18e+00    0.94    0.30
#> s(log(Turb)) 9.00e+00 1.36e-06    1.26    0.97
#> s(log(Chl))  9.00e+00 5.29e-06    1.07    0.66
#> s(log1p(RH)) 9.00e+00 5.60e-06    1.10    0.81
par(oldpar)
```

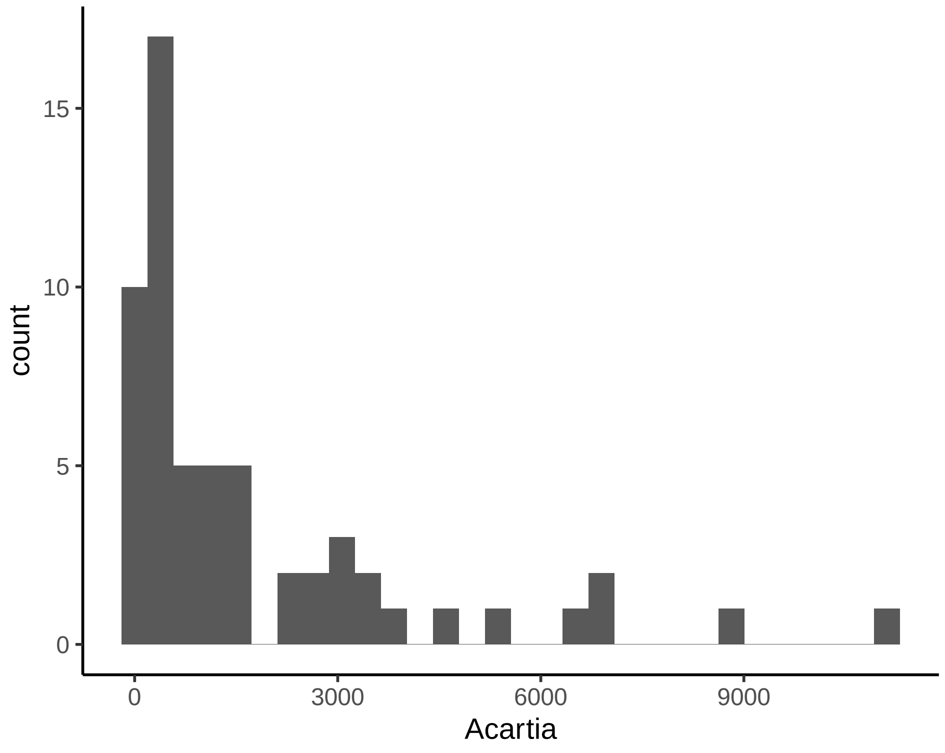
Here we still have one (low diversity) outlier. It may have unusual impact on the fit.

Example of a Single Species Model – Acartia

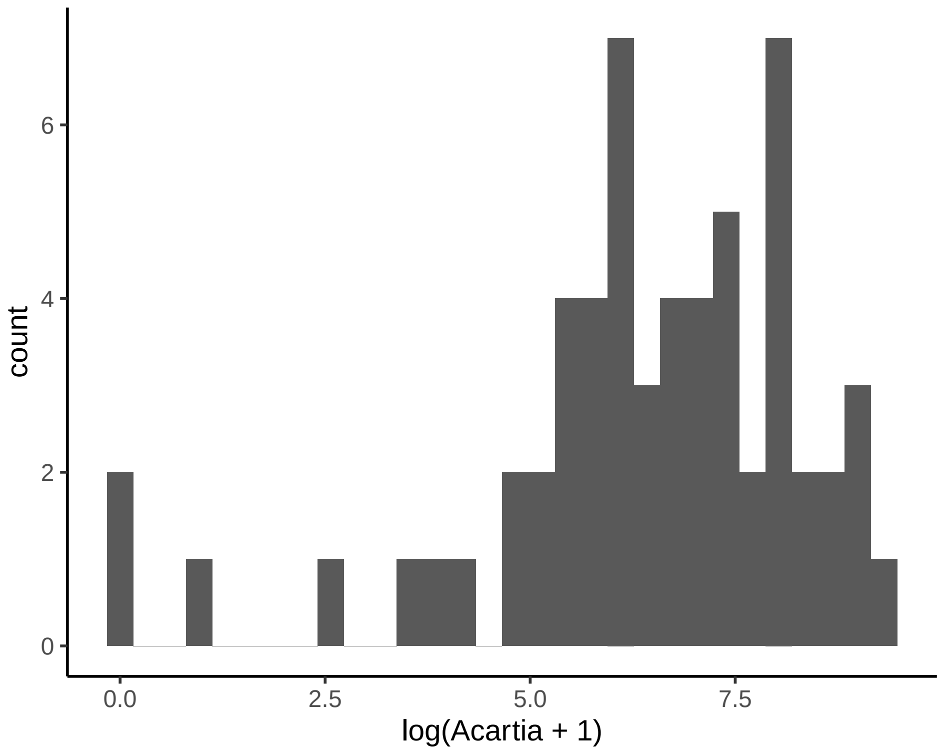
Data Review

Histogram

```
ggplot(base_data, aes(Acartia)) +
  geom_histogram()
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> Warning: Removed 1 rows containing non-finite values (stat_bin).
```



```
ggplot(base_data, aes(log(Acartia + 1))) +
  geom_histogram()
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> Warning: Removed 1 rows containing non-finite values (stat_bin).
```



Here, we have the typical skewed distribution one expects from count data, but because of scaling to densities, values are not restricted to integer values. We also have a couple of zero values and several other very low outliers. It's not yet clear whether / how they matter.

I'm curious whether it makes any sense to model counts directly? That would allow / require us to use negative binomial or quasipoisson count regressions.

Model Choices

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

Simple "count" data are sometimes modeled with a poisson distribution (although that basically never works in ecology. . . .). A Poisson distribution has the property that the variance equals the mean.

While these data are no longer raw counts, they are closely tied to count data, so we might expect our squared residuals to increase roughly linearly with predictions as well.

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.

Gamma Regression

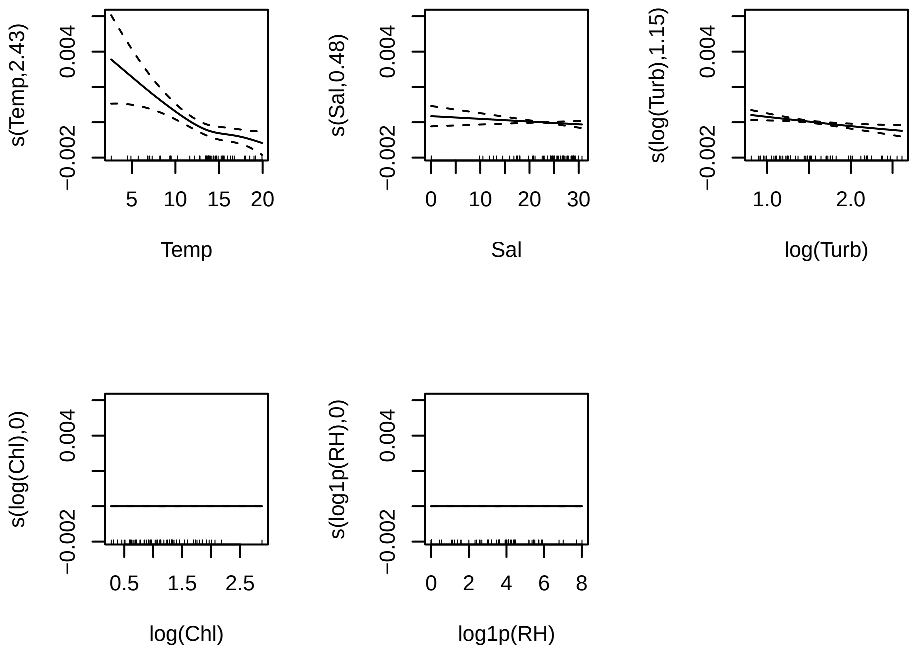
We are using “Shrinkage” estimates of the smoothing terms again, which allow certain terms to be “shrunk” out of the model.

A reminder: The default link function for the Gamma GAM in R is the inverse, so we need to think “upside down” when interpreting model output. Higher values correspond to lower abundances.

```
acartia_gam_g <- gam(I(Acartia + 1) ~ Station +
                     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')
summary(acartia_gam_g)
#>
#> Family: Gamma
#> Link function: inverse
#>
#> Formula:
#> I(Acartia + 1) ~ Station + 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)  1.324e-03  2.607e-04   5.077  5.7e-06 ***
#> Station2     1.528e-05  2.489e-04   0.061    0.951
#> Station3    -3.462e-04  2.287e-04  -1.514    0.136
#> Station4    -1.611e-04  3.059e-04  -0.527    0.601
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>                    edf Ref.df     F  p-value
#> s(Temp)      2.430e+00      9 2.041 0.000268 ***
#> s(Sal)       4.773e-01      9 0.162 0.082062 .
#> s(log(Turb)) 1.146e+00      9 1.140 0.001156 **
#> s(log(Chl))  2.749e-06      9 0.000 0.382929
#> s(log1p(RH)) 1.572e-05      9 0.000 0.345101
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.543   Deviance explained = 45.6%
#> GCV = 1.4931  Scale est. = 0.9383    n = 58
```

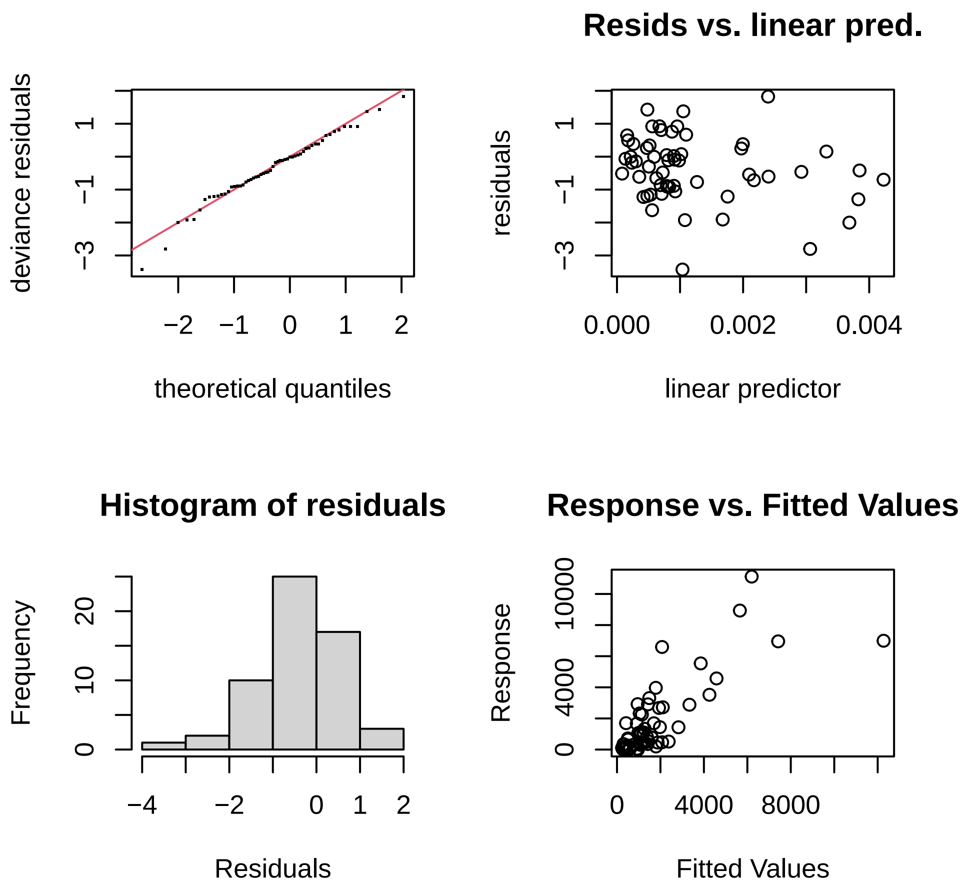
A reminder: Think “upside down”.

```
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. Very weakly, 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 waters. . . .

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



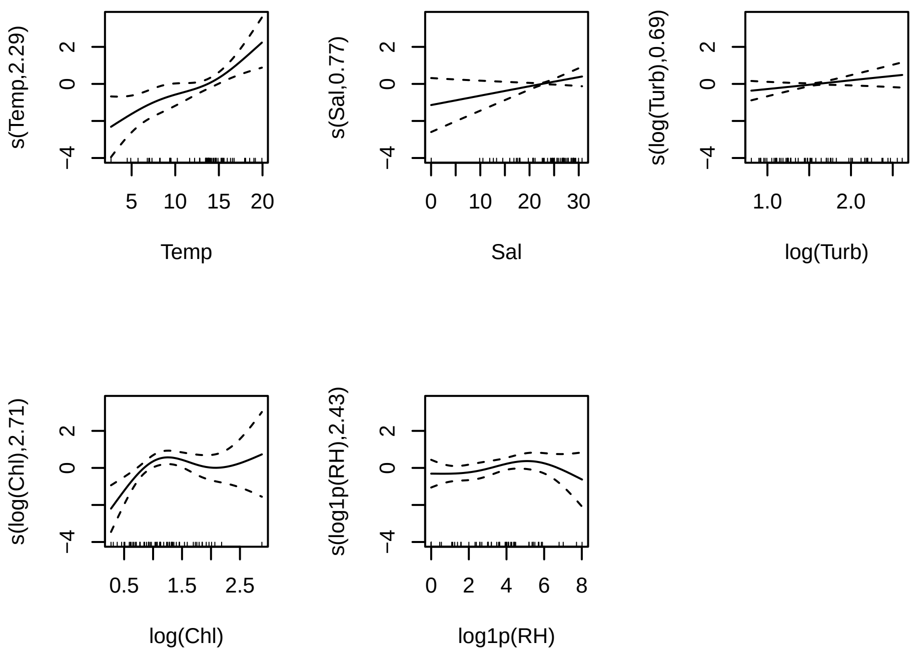
```
#>
#> Method: GCV   Optimizer: outer newton
#> full convergence after 17 iterations.
#> Gradient range [-5.036853e-07,-2.78166e-08]
#> (score 1.493072 & scale 0.9382997).
#> Hessian positive definite, eigenvalue range [1.031354e-07,0.0246352].
#> Model rank =  49 / 49
#>
#> Basis dimension (k) checking results. Low p-value (k-index<1) may
#> indicate that k is too low, especially if edf is close to k'.
#>
#>                     k'      edf k-index p-value
#> s(Temp)       9.00e+00 2.43e+00    0.89    0.49
#> s(Sal)        9.00e+00 4.77e-01    1.11    0.97
#> s(log(Turb)) 9.00e+00 1.15e+00    0.90    0.55
#> s(log(Chl))  9.00e+00 2.75e-06    0.52  <2e-16 ***
#> s(log1p(RH)) 9.00e+00 1.57e-05    0.87    0.43
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
par(oldpar)
```

We have a couple of unusually low deviance residuals, but otherwise, this looks O.K. The gamma model may have slightly over fit the scale-location pattern in the residuals.

Loglinear GAM

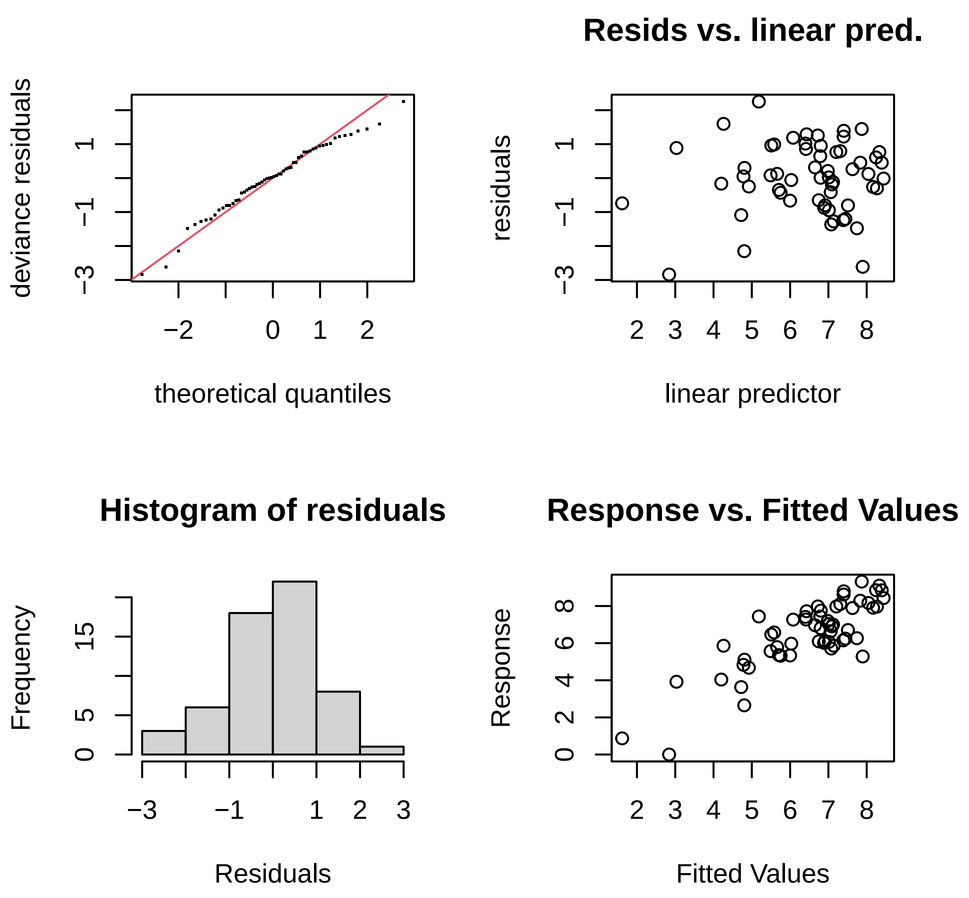
```
acartia_gam_l <- gam(log1p(Acartia) ~ Station +
                     s(Temp, bs="ts", k = 4) +
                     s(Sal, bs="ts", k = 4) +
                     s(log(Turb), bs="ts", k = 4) +
                     s(log(Chl), bs="ts", k = 4) +
                     s(log1p(RH),bs="ts", k = 4),
                   data = base_data, family = "gaussian")
summary(acartia_gam_l)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> log1p(Acartia) ~ Station + 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)   6.0969     0.5068  12.031 1.13e-15 ***
#> Station2      0.1972     0.7094   0.278    0.782
#> Station3      0.7525     0.6563   1.147    0.258
#> Station4      0.5484     0.7087   0.774    0.443
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>                 edf Ref.df     F  p-value
#> s(Temp)      2.2941      3 5.464 0.000626 ***
#> s(Sal)       0.7701      3 0.826 0.069754 .
#> s(log(Turb)) 0.6936      3 0.692 0.081766 .
#> s(log(Chl))  2.7084      3 4.149 0.005968 **
#> s(log1p(RH)) 2.4279      3 1.155 0.232763
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.602   Deviance explained = 68.5%
#> GCV = 1.7396  Scale est. = 1.3529    n = 58
```

```
oldpar <- par(mfrow = c(2,3))
plot(acartia_gam_l)
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 (driven by extreme sample? Relationship not robust)
3. They kinda like high turbidity waters, but again, the connection is not robust.
4. Acartia occurs most at higher chlorophyll numbers.

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



```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 15 iterations.
#> The RMS GCV score gradient at convergence was 2.295853e-06 .
#> The Hessian was positive definite.
#> Model rank =  19 / 19
#>
#> 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)       3.000 2.294    1.00    0.35
#> s(Sal)        3.000 0.770    1.30    0.97
#> s(log(Turb)) 3.000 0.694    1.13    0.80
#> s(log(Chl))  3.000 2.708    0.87    0.12
#> s(log1p(RH)) 3.000 2.428    1.09    0.70
par(oldpar)
```

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