Using GAMs to Analyze Plankton Comunity NMDS Data

Curtis C. Bohlen, Casco Bay Estuary Partnership

5/17/2022

Contents

Introduction

This notebook takes the output of an NMDS analyses of plankton community composition and the relationship of the community composition to major environmental variables.

The flow of analyses is as follows:

1. Conduct the NMDS analysis (mimicking Ambrose's original NMDS plot)
2. Plot the results, color coded by different environmental variables to examine major relationships with predictors on a graphic basis.
3. Conduct a linear analysis of the relationship between GAM output and each predictor, using the `envfit()` function from `vegan`. This looks for the best linear combination of the NMDS axes for predicting the environmental variables, not the other way around, but it is a fairly standard method.
4. Conduct GAM analyses of the NMDS axis scores, based on the FULL data. This analysis includes both GAM models based on quantitative environmental predictors (mirroring the analysis of abundance and diversity in the "GAM-Analysis-Environmental.pdf" notebook) and the analysis based on Season and Station (as in the "GAM-Analysis-Season-and-Station.pdf" notebook). The Season by Station analysis generally perform poorly.

5. Conduct GAM analyses of the NMDS axis scores, based on the reduced data, which omits samples with salinity less than 10 PSU. This also includes both Environmental and Season by Station analysis, for completeness.

6. Generate NMDS Graphics based on the reduced data set that omits low salinity samples.

Load Libraries

```
library(tidyverse)
#> -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
#> v ggplot2 3.3.6     v purrr   0.3.4
#> v tibble  3.1.7     v dplyr   1.0.9
#> v tidyr   1.2.0     v stringr 1.4.0
#> v readr   2.1.2     v forcats 0.5.1
#> -- Conflicts ------------------------------------------ tidyverse_conflicts() --
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()    masks stats::lag()
library(vegan)
#> Loading required package: permute
#> Loading required package: lattice
#> This is vegan 2.6-2
library(readxl)
library(mgcv)      # for GAM models
#> Loading required package: nlme
#>
#> Attaching package: 'nlme'
#> The following object is masked from 'package:dplyr':
#>
#>     collapse
#> This is mgcv 1.8-40. For overview type 'help("mgcv-package")'.
library(emmeans)   # For extracting useful "marginal" model summaries
```

Set Graphics Theme

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

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

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"


dir.create(file.path(getwd(), 'figures'), showWarnings = FALSE)
```

Input Data

Environmental 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", "skip",
                                                        "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)) %>%
  filter(! (station == 8 & month == 'May' & year == 2015))
#> New names:
#> * `` -> `...60`
```

```
names(station_data)[9:11]
#> [1] "discharge_week_cftpersec" "discharg_day"
#> [3] "discharge_week_max"
```

```
names(station_data)[9:11] <- c('disch_wk', 'disch_day', 'disch_max')
```

Station names are arbitrary, and Ambrose 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)))) %>%
  mutate(season = case_when(month == 'May' ~ 'Spring',
                            month == 'July' ~ 'Summer',
                            TRUE ~ 'Fall')) %>%
  relocate(season, .after = month) %>%
  relocate(station, .after = season)
```

Here I mostly select the depth-averaged water chemistry parameters, create short names that will work in later analyses and graphics and convert some variables to factors to control later analyses.

I retain only the weekly mean river discharge. The three discharge metrics are so highly correlated that the differences can not be that important, and I had to pick one. . . .

```r
station_data <- station_data %>%
  rename(Date = date,
         Station = station,
         Year = year) %>%
  select(-c(month)) %>%
  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')),
         is_sp_up = season == 'Spring' & Station == 1,
         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,
         Fish = `___60`
         ) %>%
  select(Date, Station, Year, Yearf, Month, Season, is_sp_up, DOY, riv_km,
         disch_wk, # disch_day, disch_max,
         Temp, Sal, Turb, AvgTurb,
         DOsat, Chl, RH, Fish) %>%
  arrange(Date, Station)
head(station_data)
#> # A tibble: 6 x 18
#>   Date                Station  Year Yearf Month Season is_sp_up   DOY riv_km
#>   <dttm>              <fct>   <dbl> <fct> <fct> <fct>  <lgl>    <dbl>  <dbl>
#> 1 2013-05-28 00:00:00 1        2013 2013  May   Spring TRUE       148  22.6
#> 2 2013-05-28 00:00:00 2        2013 2013  May   Spring FALSE      148  13.9
#> 3 2013-05-28 00:00:00 3        2013 2013  May   Spring FALSE      148   8.12
#> 4 2013-05-28 00:00:00 4        2013 2013  May   Spring FALSE      148   2.78
#> 5 2013-07-25 00:00:00 1        2013 2013  Jul   Summer FALSE      206  22.6
#> 6 2013-07-25 00:00:00 2        2013 2013  Jul   Summer FALSE      206  13.9
#> # ... with 9 more variables: disch_wk <dbl>, Temp <dbl>, Sal <dbl>, Turb <dbl>,
#> #   AvgTurb <dbl>, DOsat <dbl>, Chl <dbl>, RH <dbl>, Fish <dbl>
```

Composition Data

```r
filename.in <- "Penobscot_Zooplankton and field data_EA_2.13.20.xlsx"
file_path <- file.path(data_folder, filename.in)
zoopl <- read_excel(file_path,
                    sheet = "NMDS Happy",
                    col_types = c("date",
                                  "text", "numeric", "numeric", "text",
                                  "text", "text", "text", "text", "text",
                                  "text", "numeric", "text", "text",
                                  "numeric", "numeric", "numeric",
```

```
                                  "text", "text", "text", "numeric",
                                  "numeric", "numeric", "numeric")) %>%
  select(-c(`...20`:`...24`)) %>%
  rename_with(~ gsub(" ", "_", .x))
#> New names:
#> * `` -> `...20`
#> * `` -> `...21`
#> * `` -> `...22`
#> * `` -> `...23`
#> * `` -> `...24`
```

We renumber the stations here as well. The code is similar.

```
zoopl <- zoopl %>%
  mutate(STATION = factor(as.numeric(factor(STATION))))
zoopl
#> # A tibble: 814 x 19
#>    DATE                Month  Year STATION PHYLUM CLASS `SUB-CLASS` ORDER FAMILY
#>    <dttm>              <chr> <dbl> <fct>   <chr>  <chr> <chr>       <chr> <chr>
#>  1 2015-09-16 00:00:00 Sept~  2015 4       Arthr~ Maxi~ Copepoda    <NA>  <NA>
#>  2 2014-05-02 00:00:00 May    2014 1       Arthr~ Maxi~ Copepoda    Cala~ <NA>
#>  3 2017-07-12 00:00:00 July   2017 3       Unkno~ <NA>  <NA>        <NA>  <NA>
#>  4 2016-07-20 00:00:00 July   2016 4       Unkno~ <NA>  <NA>        <NA>  <NA>
#>  5 2015-09-16 00:00:00 Sept~  2015 3       Unkno~ <NA>  <NA>        <NA>  <NA>
#>  6 2017-10-11 00:00:00 Octo~  2017 1       Unkno~ Unid~ <NA>        <NA>  <NA>
#>  7 2016-07-20 00:00:00 July   2016 3       Unkno~ <NA>  <NA>        <NA>  <NA>
#>  8 2016-05-25 00:00:00 May    2016 2       Unkno~ <NA>  <NA>        <NA>  <NA>
#>  9 2013-09-25 00:00:00 Sept~  2013 3       Unkno~ <NA>  <NA>        <NA>  <NA>
#> 10 2013-07-25 00:00:00 July   2013 4       Unkno~ <NA>  <NA>        <NA>  <NA>
#> # ... with 804 more rows, and 10 more variables: GENUS <chr>, SPECIES <chr>,
#> #   QUANTITY <dbl>, LOWEST_TAXA <chr>, NAME <chr>, `TOTAL_#_ORGANISMS` <dbl>,
#> #   CORRECTED_PERCENT_ABUNDANCE <dbl>, `NET_MESH_SIZE_(MICRONS)` <dbl>,
#> #   NOTES <chr>, Picture_number <chr>
```

Turn Data from Long to Wide

This code generates a total abundance for each taxa by site and date and pivots it to wide format. The code is more compact that what Erin used, but slightly more opaque because it relies on several options of the `pivot_wider()` function.

```
zoopl2 <- zoopl %>%
  pivot_wider(c(DATE, Month, Year, STATION),
              names_from = NAME,
              names_sort = TRUE,
              values_from = CORRECTED_PERCENT_ABUNDANCE,
  values_fn = sum,
  values_fill = 0)
zoopl2
#> # A tibble: 59 x 53
#>    DATE                Month  Year STATION Acartia Amphipod `Arrow worm` Balanus
#>    <dttm>              <chr> <dbl> <fct>      <dbl>    <dbl>        <dbl>   <dbl>
#>  1 2015-09-16 00:00:00 Sept~  2015 4          43.4   0            0         3.28
```

```
#>  2 2014-05-02 00:00:00 May    2014 1          6.85  0            0         3.43
#>  3 2017-07-12 00:00:00 July   2017 3         32.5   0            0.0219   22.6
#>  4 2016-07-20 00:00:00 July   2016 4         49.8   0            0.00463   0.422
#>  5 2015-09-16 00:00:00 Sept~  2015 3         49.1   0.00942      4.96      7.66
#>  6 2017-10-11 00:00:00 Octo~  2017 1         68.6   0            0         0
#>  7 2016-07-20 00:00:00 July   2016 3         49.3   0            0         0
#>  8 2016-05-25 00:00:00 May    2016 2          6.45  0.00466      0         1.38
#>  9 2013-09-25 00:00:00 Sept~  2013 3         45.6   0            0.00236   3.88
#> 10 2013-07-25 00:00:00 July   2013 4         48.2   0            0         9.16
#> # ... with 49 more rows, and 45 more variables: Bivalve <dbl>,
#> #   `Brittle Star` <dbl>, Bryozoan <dbl>, `Calanoid spp` <dbl>, Calanus <dbl>,
#> #   Caligus <dbl>, Centropages <dbl>, Cladoceran <dbl>, `Crab larvae` <dbl>,
#> #   Crangon <dbl>, Ctenophore <dbl>, Cumacean <dbl>, Decapod <dbl>,
#> #   Diacyclops <dbl>, Eucyclops <dbl>, Eurytemora <dbl>, `Fish larvae` <dbl>,
#> #   Gastropod <dbl>, `Halicyclops fosteri` <dbl>, Harpacticoid <dbl>,
#> #   Hermit <dbl>, Hydrozoan <dbl>, Isopod <dbl>, Leptodiaptomus <dbl>, ...
```

Check for Dropped Sample

Erin Ambrose dropped 5/20/15 Station 8 from both datasheets. Environmental and zooplankton data should each have 59 rows, and they do.

Erin notes that there was no zooplankton "sample" (?) only nekton for that sample. I'm not sure if that means no sample was collected or there were no zooplankton in the sample. Anyway, she noted that this sample "threw off calculation of percent abundances."

Note that that sample is one of the Spring "washout" samples that cause trouble on our other analyses as well.

```
sum(! is.na((zoopl2 %>%
  filter((STATION == 4 & Month == 'May' & Year == 2015))))) == 0
#> [1] TRUE
sum(! is.na(station_data %>%
  filter((Station == 4 & Month == 'May' & Year == 2015)))) == 0
#> [1] TRUE
```

Correct Sample Row Alignment

I had some funny artifacts popping up in my initial (re) analyses. I finally tracked down the cause. My data was in a different order from Ambrose's version, apparently because I used different tools that have different default ordering. Since NMDS is determined only by a "distance" metric, the NMDS solution is unique only to rotations and reflections. In fact, after the NMDS is fit, the default behavior is to rotate the solution to align the first axis with the largest apparent axis of variation, so the solution returned is unique only to reflections. It looks like ordering of samples can affect which solution the algorithm presents. Only by ensuring uniform ordering can we ensure that output is similar to Ambrose's prior analysis.

Align Data Tables

```
zoopl2 <- zoopl2 %>%
  arrange(DATE, STATION)
```

```
station_data<- station_data %>%
  arrange(Date, Station)

head(zoopl2[,c(1,4)])
#> # A tibble: 6 x 2
#>   DATE                STATION
#>   <dttm>              <fct>
#> 1 2013-05-28 00:00:00 1
#> 2 2013-05-28 00:00:00 2
#> 3 2013-05-28 00:00:00 3
#> 4 2013-05-28 00:00:00 4
#> 5 2013-07-25 00:00:00 1
#> 6 2013-07-25 00:00:00 2
head(station_data[,c(1,2)])
#> # A tibble: 6 x 2
#>   Date                Station
#>   <dttm>              <fct>
#> 1 2013-05-28 00:00:00 1
#> 2 2013-05-28 00:00:00 2
#> 3 2013-05-28 00:00:00 3
#> 4 2013-05-28 00:00:00 4
#> 5 2013-07-25 00:00:00 1
#> 6 2013-07-25 00:00:00 2
```

Matrix of Species for `vegan`

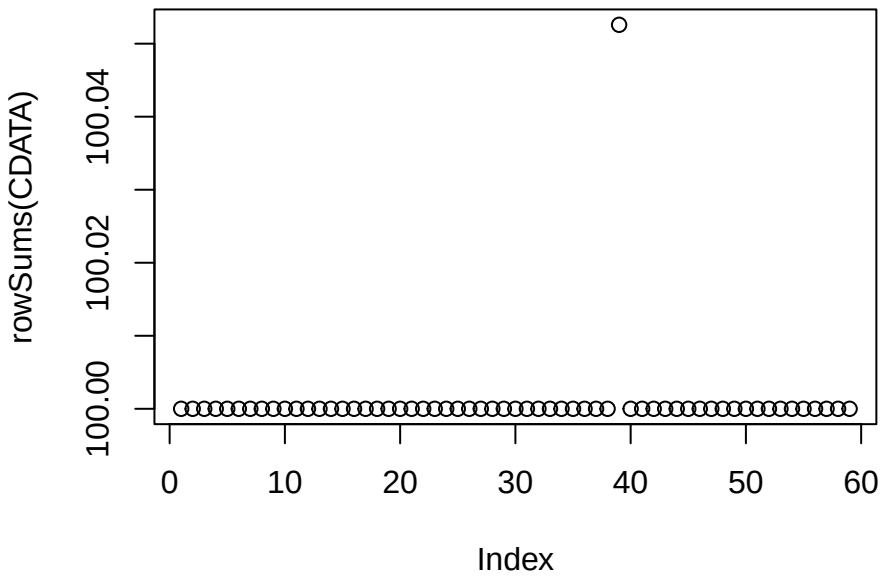
The `vegan` package likes to work with a matrix of species occurrences. Although the matrix can have row names that provide sample identifiers, that was not done here. The "matrix" I produce here is really a data frame with nothing but numeric values. While those are different data structures internally, `vegan` handles the conversion in the background.

```
CDATA <- zoopl2[,-c(1:4)]
```

Data Sanity Checks

We should have no `NA`s, and row sums should all be 1 (100%), at least within reasonable rounding error.

```
anyNA(CDATA)
#> [1] FALSE
plot(rowSums(CDATA))
```



One sample is slightly off the calculation of totals, but the deviation is tiny, so not of any interest.

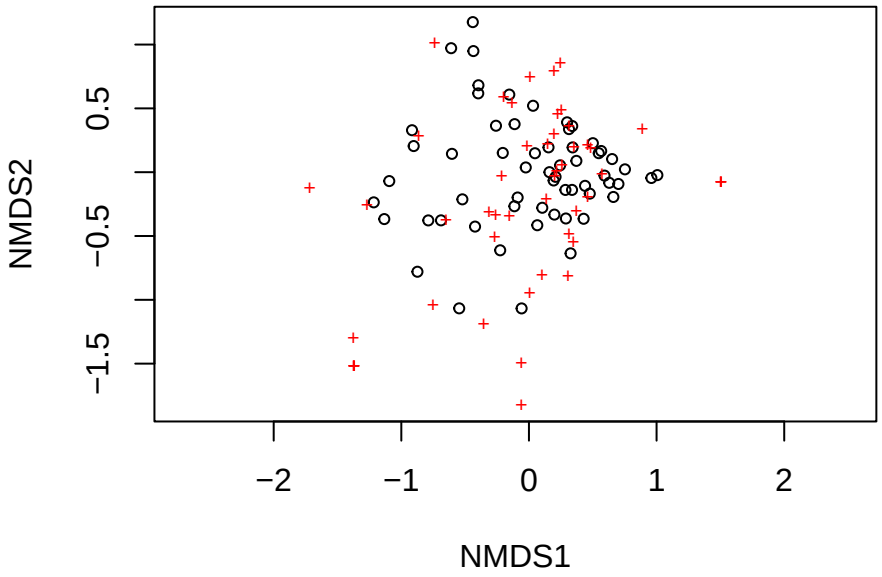
NMDS Analyses (Full Data)

```
NMDSE <- metaMDS(CDATA, autotransform = FALSE, k = 2, trymax = 75)
#> Run 0 stress 0.1509449
#> Run 1 stress 0.1684932
#> Run 2 stress 0.1508906
#> ... New best solution
#> ... Procrustes: rmse 0.0084205  max resid 0.04932192
#> Run 3 stress 0.1850759
#> Run 4 stress 0.1628329
#> Run 5 stress 0.1505672
#> ... New best solution
#> ... Procrustes: rmse 0.01330423  max resid 0.08694448
#> Run 6 stress 0.1729591
#> Run 7 stress 0.1863239
#> Run 8 stress 0.1684932
#> Run 9 stress 0.1918348
#> Run 10 stress 0.1580207
#> Run 11 stress 0.2052702
#> Run 12 stress 0.15446
#> Run 13 stress 0.1658234
#> Run 14 stress 0.1810245
#> Run 15 stress 0.1493367
#> ... New best solution
```

```
#> ... Procrustes: rmse 0.0226232  max resid 0.1564459
#> Run 16 stress 0.2246299
#> Run 17 stress 0.1729591
#> Run 18 stress 0.2280086
#> Run 19 stress 0.2220988
#> Run 20 stress 0.15446
#> Run 21 stress 0.1996918
#> Run 22 stress 0.1835908
#> Run 23 stress 0.168418
#> Run 24 stress 0.1505672
#> Run 25 stress 0.168418
#> Run 26 stress 0.1505044
#> Run 27 stress 0.180874
#> Run 28 stress 0.1658234
#> Run 29 stress 0.1577084
#> Run 30 stress 0.1995611
#> Run 31 stress 0.2244388
#> Run 32 stress 0.1496383
#> ... Procrustes: rmse 0.006247454  max resid 0.03524392
#> Run 33 stress 0.1549458
#> Run 34 stress 0.1505044
#> Run 35 stress 0.168418
#> Run 36 stress 0.1658234
#> Run 37 stress 0.1546152
#> Run 38 stress 0.1628329
#> Run 39 stress 0.168545
#> Run 40 stress 0.1508906
#> Run 41 stress 0.1493367
#> ... New best solution
#> ... Procrustes: rmse 3.581936e-06  max resid 1.55318e-05
#> ... Similar to previous best
#> *** Solution reached
NMDSE
#>
#> Call:
#> metaMDS(comm = CDATA, k = 2, trymax = 75, autotransform = FALSE)
#>
#> global Multidimensional Scaling using monoMDS
#>
#> Data:     CDATA
#> Distance: bray
#>
#> Dimensions: 2
#> Stress:     0.1493367
#> Stress type 1, weak ties
#> Two convergent solutions found after 41 tries
#> Scaling: centring, PC rotation, halfchange scaling
#> Species: expanded scores based on 'CDATA'
```

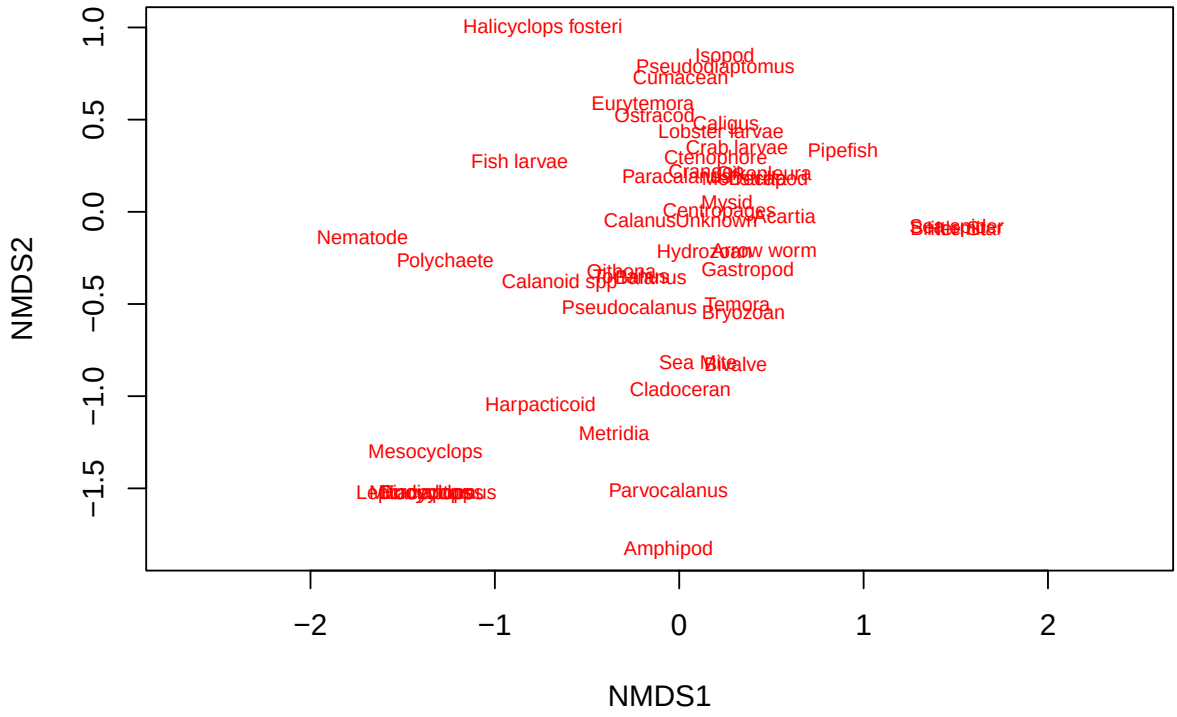
Plot

```
plot(NMDSE, type = 'p')
```



Plot Species

```
plot(NMDSE, 'species', type = 't')
```



Combining the NMDS Results with Environmental Data

I want to use the names of these variables as labels in graphics later. I capitalize variable names here, so they will appear capitalized in graphics without further action on my part.

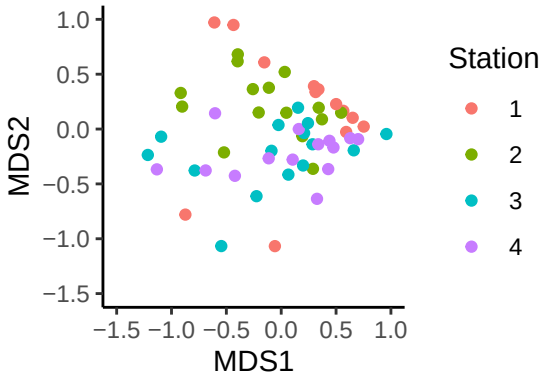
```
envNMDS <- station_data %>%
  select(-Date, -Month, -DOY, -riv_km, -AvgTurb) %>%
  mutate(Turb2 = log(Turb),
         Chl2 = log(Chl),
         RH2 = log1p(RH),
         Fish2 = log1p(Fish)) %>%
  mutate(sample_seq = as.numeric(Season) + (Year-2013)*3,
         sample_event = factor(sample_seq)) %>%
  cbind(as_tibble(NMDSE$points))
```

Graphic Exploration

These plots are intended principally to help us understand the NMDS from a more intuitive perspective. The idea is to plot the ordination, but colored by various predictor variables.

By Station

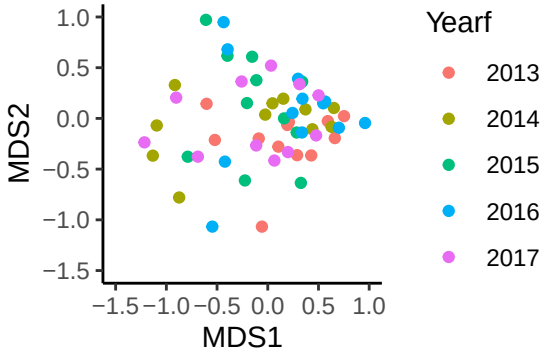
```
ggplot(envNMDS, aes(MDS1, MDS2)) +
    geom_point(aes(color=Station)) +
  xlim(c(-1.5,1)) +
  ylim(c(-1.5,1)) +
  theme(aspect.ratio=1)
#> Warning: Removed 2 rows containing missing values (geom_point).
```



Note that station 1 is split into a group along the upper edge and two points along the lower edge. The stations don't segregate fully, but there are trends. Other than those two spring samples, Station 1 is upper edge. Station 2 is upper zone as well. I suspect those two samples are "washout" event samples.

By Year

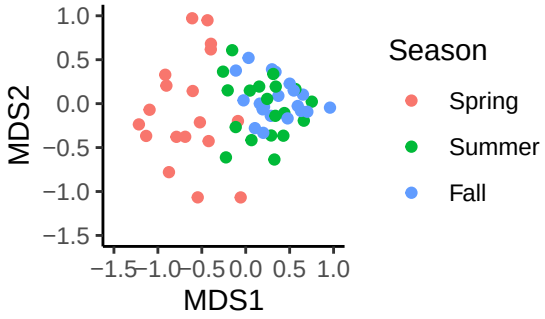
```
ggplot(envNMDS, aes(MDS1, MDS2)) +
  geom_point(aes(color=Yearf)) +
  xlim(c(-1.5,1)) +
  ylim(c(-1.5,1)) +
  theme(aspect.ratio=1)
#> Warning: Removed 2 rows containing missing values (geom_point).
```



MAYBE 2016 is towards the upper edge, but it's not clear at all. I don't see a robust pattern here.

By Season

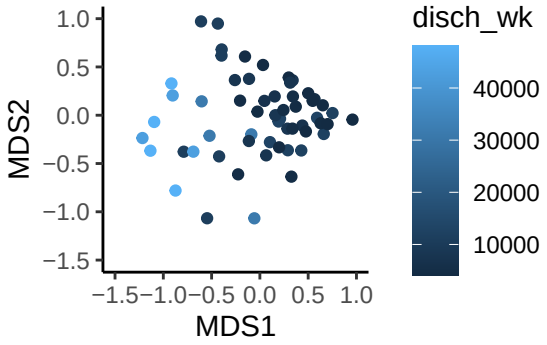
```
ggplot(envNMDS, aes(MDS1, MDS2)) +
  geom_point(aes(color=Season)) +
  xlim(c(-1.5,1)) +
  ylim(c(-1.5,1)) +
  theme(aspect.ratio=1)
#> Warning: Removed 2 rows containing missing values (geom_point).
```



Note the VERY strong association here, with Spring samples all to the left on the plot. Summer and Fall plots are fairly mixed up, but all to the left. That means Axis 1 can be interpreted as largely a "season" signal.

By River Discharge

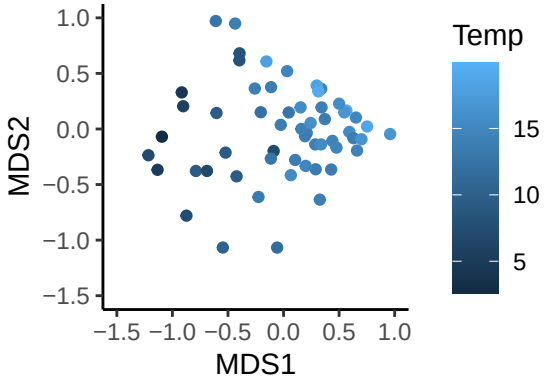
```
ggplot(envNMDS, aes(MDS1, MDS2)) +
  geom_point(aes(color=disch_wk)) +
  xlim(c(-1.5,1)) +
  ylim(c(-1.5,1)) +
  theme(aspect.ratio=1)
#> Warning: Removed 2 rows containing missing values (geom_point).
```



That's a very similar pattern the the prior one, based on season. The two predictors are highly collinear, which may cause problems with estimation later.

By Temperature

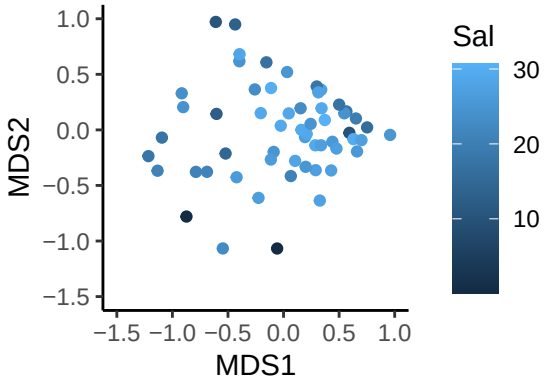
```
ggplot(envNMDS, aes(MDS1, MDS2)) +
  geom_point(aes(color=Temp)) +
  xlim(c(-1.5,1)) +
  ylim(c(-1.5,1)) +
  theme(aspect.ratio=1)
#> Warning: Removed 2 rows containing missing values (geom_point).
```



This reveals the same pattern as the last two graphics, only via the correlation between season and temperature. Cool temperatures, high river discharge in spring to the left.

By Salinity

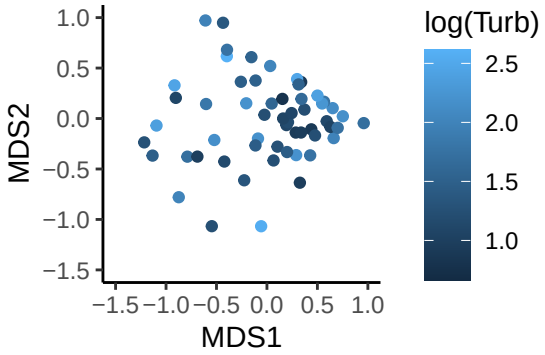
```
ggplot(envNMDS, aes(MDS1, MDS2)) +
  geom_point(aes(color=Sal)) +
  xlim(c(-1.5,1)) +
  ylim(c(-1.5,1)) +
  theme(aspect.ratio=1)
#> Warning: Removed 2 rows containing missing values (geom_point).
```



This one is hard to interpret. What jumps out at me here is the two VERY low salinity sites at the bottom, and the tendency for other lower salinity samples to fall to the left (spring) and along the upper edge (Station 1).

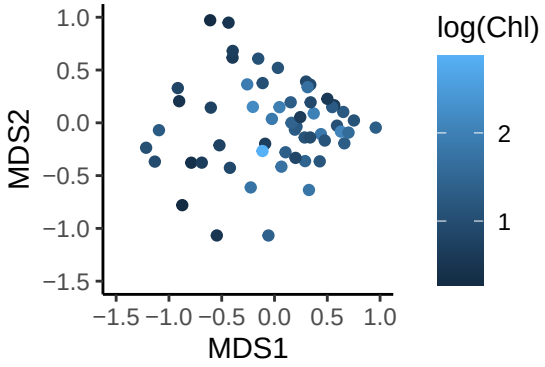
By Turbidity

```
ggplot(envNMDS, aes(MDS1, MDS2)) +
  geom_point(aes(color=log(Turb))) +
  xlim(c(-1.5,1)) +
  ylim(c(-1.5,1)) +
  theme(aspect.ratio=1)
#> Warning: Removed 2 rows containing missing values (geom_point).
```



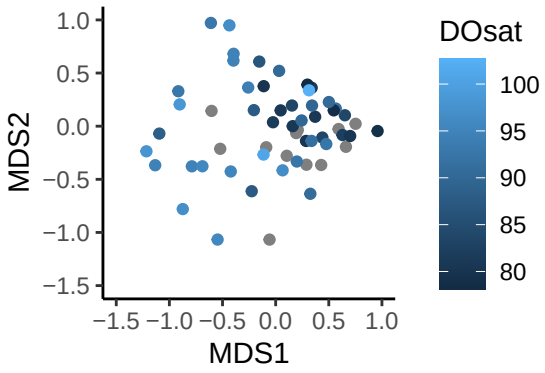
By Chlorophyll

```
ggplot(envNMDS, aes(MDS1, MDS2)) +
  geom_point(aes(color=log(Chl))) +
  xlim(c(-1.5,1)) +
  ylim(c(-1.5,1)) +
  theme(aspect.ratio=1)
#> Warning: Removed 2 rows containing missing values (geom_point).
```



By Oxygen Saturation

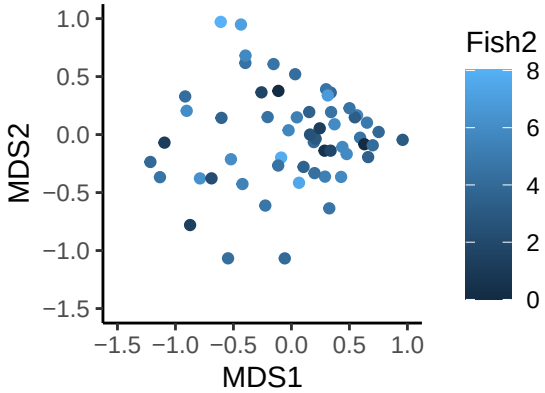
```
ggplot(envNMDS, aes(MDS1, MDS2)) +
  geom_point(aes(color=DOsat)) +
  xlim(c(-1.5,1)) +
  ylim(c(-1.5,1)) +
  theme(aspect.ratio=1)
#> Warning: Removed 2 rows containing missing values (geom_point).
```



Note that highest DO is to the left. This relationship provides an alternate "explanation" to considering axis 1 based on season, temperature, or river discharge.

By Fish

```
ggplot(envNMDS, aes(MDS1, MDS2)) +
  geom_point(aes(color=Fish2)) +
  xlim(c(-1.5,1)) +
  ylim(c(-1.5,1)) +
  theme(aspect.ratio=1)
#> Warning: Removed 2 rows containing missing values (geom_point).
```



Using `envfit` to Estimate Correlations

The `envfit()` function, counter intuitively, predicts each environmental variable based on the two NMDS axes jointly. The related help file says "The environmental variables are the dependent variables that are explained by the ordination scores, and each dependent variable is analyzed separately." The model is always linear, which is different from our GAM models.

That means the `envfit()` output is NOT a single multivariate statistical test, but separate statistical fits for each environmental variable.

Coefficients are the coordinates of a unit-length vector that points along the "direction" in ordination space that shows maximum correlation with the NMDS scores. (Since these are unit vectors, if one coordinate goes up, the other necessarily goes down.) The R2 term "is a"goodness of fit statistic" like the one from multiple regression models. The higher the number, the better the ability of the ordination scores to predict environmental variables

These results are apparently based on randomization methods, so results change somewhat between repeated runs of the following code. The relatively high number of permutations specified here helps keep those effects small.

`envfit()` Single Variable Relationships

```
ef <- envfit(NMDSE, envNMDS[,c(1, 3:17)], permu = 9999, na.rm = TRUE)
ef
#>
#> ***VECTORS
#>
#>              NMDS1    NMDS2     r2 Pr(>r)
#> disch_wk -0.97259 -0.23254 0.6248 0.0001 ***
#> Temp      0.97606  0.21752 0.7597 0.0001 ***
#> Sal       0.98681 -0.16189 0.0978 0.1054
#> Turb     -0.33505  0.94220 0.1477 0.0309 *
#> DOsat    -0.97723 -0.21218 0.3353 0.0001 ***
#> Chl       0.72753 -0.68608 0.0782 0.1735
#> RH       -0.22970  0.97326 0.2436 0.0008 ***
#> Fish     -0.35283  0.93569 0.1317 0.0452 *
#> Turb2    -0.34248  0.93952 0.1501 0.0314 *
#> Chl2      0.85555 -0.51771 0.1660 0.0187 *
#> RH2       0.25692  0.96643 0.2386 0.0031 **
#> Fish2    -0.27062  0.96269 0.0506 0.3315
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#> Permutation: free
#> Number of permutations: 9999
#>
#> ***FACTORS:
#>
#> Centroids:
#>                 NMDS1   NMDS2
#> Station1       0.1458  0.3011
#> Station2      -0.1540  0.3187
#> Station3      -0.1657 -0.2508
#> Station4       0.0648 -0.2421
#> Yearf2014     -0.0620 -0.0441
#> Yearf2015     -0.1255  0.1201
#> Yearf2016      0.1830  0.0679
#> Yearf2017     -0.1447 -0.0129
#> SeasonSpring  -0.7443  0.0037
#> SeasonSummer   0.1803  0.0012
#> SeasonFall     0.3760  0.0872
```

```
#> is_sp_upFALSE  0.0093  0.0075
#> is_sp_upTRUE  -0.6394  0.3802
#>
#> Goodness of fit:
#>              r2 Pr(>r)
#> Station  0.1959 0.0060 **
#> Yearf    0.0441 0.6840
#> Season   0.4689 0.0001 ***
#> is_sp_up 0.0697 0.0396 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#> Permutation: free
#> Number of permutations: 9999
#>
#> 13 observations deleted due to missingness
```

A Note on theMissing Values Note 13 observations deleted due to missingness. Those are the 2013 data, which lacks DO saturation data. We can refit to include those data by dropping DO as a predictor. That might alter some fits.

```
names(envNMDS)
#>  [1] "Station"      "Year"         "Yearf"        "Season"       "is_sp_up"
#>  [6] "disch_wk"     "Temp"         "Sal"          "Turb"         "DOsat"
#> [11] "Chl"          "RH"           "Fish"         "Turb2"        "Chl2"
#> [16] "RH2"          "Fish2"        "sample_seq"   "sample_event" "MDS1"
#> [21] "MDS2"
```

`envfit()` Not Including Oxygen

When we drop oxygen and base the results on the whole data set – including 2013, the turbidity patterns vanish.

```
ef_2 <- envfit(NMDSE, envNMDS[,c(1, 3:9, 11:17)], permu = 9999, na.rm = TRUE)
ef_2
#>
#> ***VECTORS
#>
#>             NMDS1    NMDS2     r2 Pr(>r)
#> disch_wk -0.93244 -0.36133 0.5852 0.0001 ***
#> Temp      0.96695  0.25497 0.7434 0.0001 ***
#> Sal       0.88012  0.47476 0.0737 0.1257
#> Turb     -0.57811  0.81596 0.0300 0.4357
#> Chl       0.77421 -0.63293 0.0623 0.1723
#> RH       -0.28444  0.95869 0.1316 0.0240 *
#> Fish     -0.43622  0.89984 0.0847 0.0862 .
#> Turb2    -0.53315  0.84602 0.0355 0.3749
#> Chl2      0.86763 -0.49722 0.1631 0.0082 **
#> RH2       0.34358  0.93912 0.1316 0.0210 *
#> Fish2    -0.36712  0.93017 0.0268 0.4650
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#> Permutation: free
#> Number of permutations: 9999
#>
#> ***FACTORS:
#>
#> Centroids:
#>                  NMDS1   NMDS2
#> Station1        0.2065  0.1600
#> Station2       -0.1258  0.2123
#> Station3       -0.0805 -0.2292
#> Station4        0.0459 -0.2259
#> Yearf2013       0.1633 -0.2201
#> Yearf2014      -0.0620 -0.0441
#> Yearf2015      -0.1255  0.1201
#> Yearf2016       0.1830  0.0679
#> Yearf2017      -0.1447 -0.0129
#> SeasonSpring   -0.6495 -0.0713
#> SeasonSummer    0.2508 -0.0440
#> SeasonFall      0.3557  0.0494
#> is_sp_upFALSE   0.0447 -0.0231
#> is_sp_upTRUE   -0.4939  0.0183
#>
#> Goodness of fit:
#>              r2 Pr(>r)
#> Station  0.1305 0.0211 *
#> Yearf    0.0739 0.3890
#> Season   0.4338 0.0001 ***
#> is_sp_up 0.0409 0.0930 .
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#> Permutation: free
#> Number of permutations: 9999
#>
#> 1 observation deleted due to missingness
```

Note that River Herring and log(River Herring + 1) are significantly correlated with the MSDS, while total fish is at best marginally correlated.

Extracting Vector Information

The `ef` or `ef_2` object is an `envfit` S3 object, with three named slots. The vector information we need to plot the environment arrows is available in `vectors`. But that object is itself also an S3 object, with five named items. The help page for `envfit()` tells us that the information on the direction of the arrows is in the `arrows` component. We are told that arrows contain "Arrow endpoints from `vectorfit`. The arrows are scaled to unit length."

```
ef_2$vectors$arrows
#>                NMDS1      NMDS2
#> disch_wk -0.9324376 -0.3613310
#> Temp      0.9669488  0.2549706
#> Sal       0.8801170  0.4747568
#> Turb     -0.5781051  0.8159623
```

```
#> Chl       0.7742075 -0.6329318
#> RH       -0.2844394  0.9586940
#> Fish     -0.4362223  0.8998389
#> Turb2    -0.5331460  0.8460232
#> Chl2      0.8676262 -0.4972171
#> RH2       0.3435783  0.9391240
#> Fish2    -0.3671230  0.9301724
#> attr(,"decostand")
#> [1] "normalize"
```

The information we need to determine the magnitude of those vectors is in the `r` component of the `vectors` component, which (according to the `envfit()` help file) contains "Goodness of fit statistic: Squared correlation coefficient".

The correlation coefficient is (formally) a bivariate statistic, but here it is being used to indicate the strength of association between two predictors (NMDS Axis 1 and NMDS axis 2) and each environmental variable. I have not been able to find clear documentation of what is going on, but it appears the R squared value is (or is analogous to) the R squared value reported the implied (two predictor, one response) linear regression. If that is the case, the value of r can be (roughly) interpreted as the correlation coefficient between the best linear combination of NMDS Axis 1 and Axis 2 and each environmental variable.

It's worth pointing out that the r values are mostly pretty low. The NMDS ordination does not do a very good job of "predicting" environmental variables, except for Discharge and Temperature, which it does quite well. Here are the implied correlation coefficients:

```
sqrt(ef_2$vectors$r)
#>  disch_wk      Temp       Sal      Turb       Chl        RH      Fish     Turb2
#> 0.7650111 0.8621839 0.2715303 0.1730653 0.2496305 0.3628106 0.2911109 0.1884531
#>      Chl2       RH2     Fish2
#> 0.4038715 0.3627169 0.1637557
```

For plotting, we scale the length of each of the arrows by the associated correlation coefficient (square root of the r squared value). The higher the correlation coefficient, the longer the arrow. Thus arrow direction shows the mix of Axis 1 and Axis 2 that correlates best with each environmental variable, while the length of the arrow shows the relative ability of the ordination to predict environmental variables.

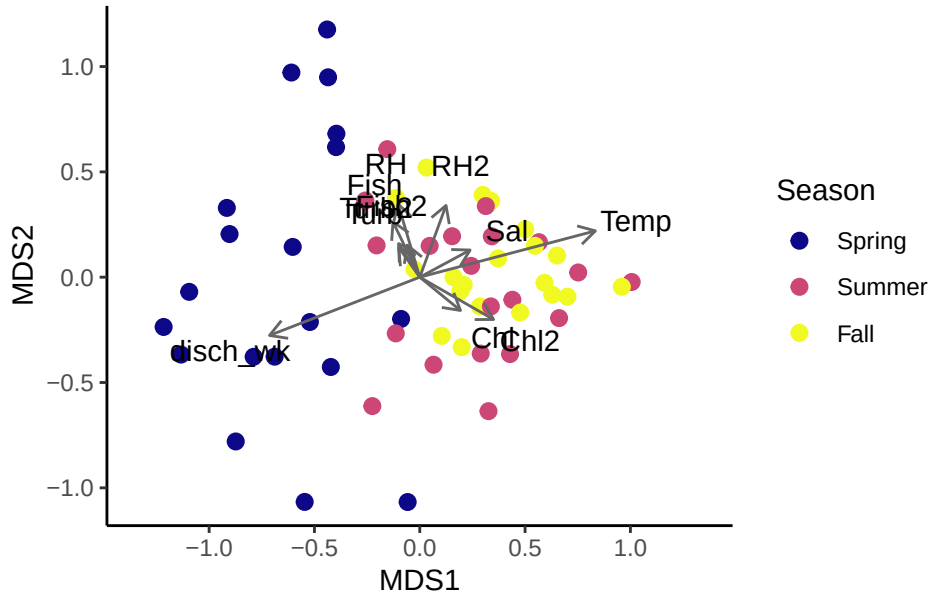
```
arrows <- ef_2$vectors$arrows
rsq    <- ef_2$vectors$r
scaled_arrows <- as_tibble(arrows*sqrt(rsq)) %>%
  mutate(parameter = rownames(arrows))
```

```
scale_factor = .2  # This scales the relative length of the arrows.
                   # it is selected largely for graphic convenience.

scaled_arrows <- scaled_arrows %>%
  mutate(ann_xpos = NMDS1 + arrows[,1]* scale_factor, #arrows is unscaled
                                                      # creates constant spacing
         ann_ypos = NMDS2 + arrows[,2] * scale_factor)
```

While we are creating vectors, we also want to place the text annotations identifying each vector. We want to space the labels so they are a fixed distance beyond the end of each vector with a little vector arithmetic.

Draft Graphic

```
plt <- ggplot(data = envNMDS, aes(MDS1, MDS2)) +
  geom_point(aes(color = Season), size = 2.5) +
  geom_segment(data=scaled_arrows,
               mapping = aes(x=0,xend=NMDS1,y=0,yend=NMDS2),
               arrow = arrow(length = unit(0.25, "cm")) ,colour="grey40") +
  geom_text(data=scaled_arrows,
            mapping = aes(x= ann_xpos,
                          y= ann_ypos,label=parameter),
            size=4, nudge_x =0, nudge_y = 0, hjust = .5)+
  scale_color_viridis_d(option = 'C', name = 'Season') +
  coord_fixed(xlim =c(-1.35, 1.35))
plt
```



That's too messy. We want to plot a smaller number of arrows. I recommend showing only the transformed variables, since that is what we use in the GAMs.

Possible Publication Graphics

The instructions to authors suggests figure widths should line up with columns, and proposes figure widths should be: 39, 84, 129, or 174 mm wide, with height not to exceed 235 mm. Presumably that corresponds to 1,2,3,or 4 columns wide?

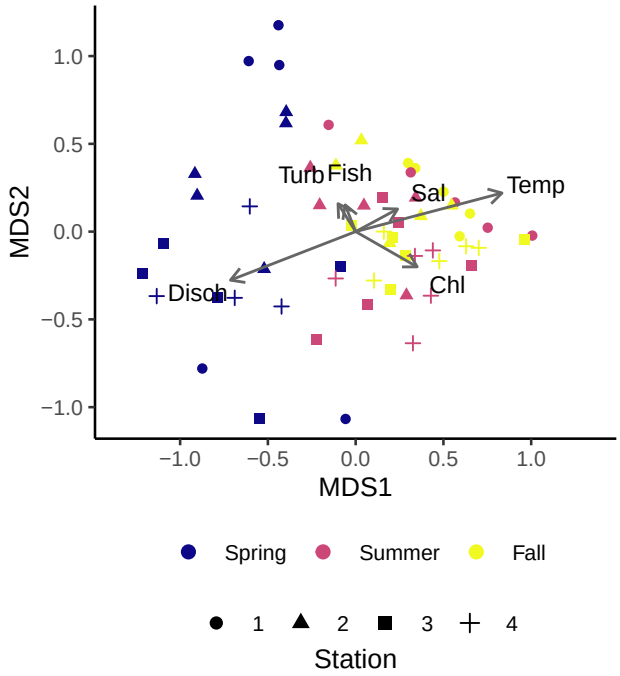
39 mm is about one and one half inches, which his quite small, so we will use the 84 mm wide option, which is about 3.3 inches.

Transformed Environmental Variables

We drop the untransformed explanatory variables. We retain Discharge here, even though we did not use it in any final models because of collinearity.

```
tmp_arrows <- scaled_arrows %>%
  filter(parameter %in% c('disch_wk', 'Temp', 'Sal', 'Turb2', 'Chl2',
                          'Fish2')) %>%
  mutate(parameter = if_else( parameter == 'disch_wk','Discharge', parameter)) %>%
  mutate(parameter = factor(parameter,
                            levels = c("Discharge", 'Temp', 'Sal',
                                       'Turb2', 'Chl2',
                                       'Fish2'),
                            labels = c('Disch', 'Temp', 'Sal',
                                       'Turb', 'Chl',
                                      'Fish'))) %>%
  mutate(ann_xpos = case_when(
    parameter == 'Turb' ~ ann_xpos - 0.1,
    parameter == 'Fish' ~ ann_xpos + 0.1,
    TRUE ~ ann_xpos))

plt <- ggplot(data = envNMDS, aes(MDS1, MDS2)) +
  geom_point(aes(color = Season, shape = Station), size = 1.5) +
  geom_segment(data=tmp_arrows,
               mapping = aes(x=0,xend=NMDS1,y=0,yend=NMDS2),
               arrow = arrow(length = unit(0.2, "cm")) ,colour="grey40") +
  geom_text(data=tmp_arrows,
            mapping = aes(x=ann_xpos,
                          y=ann_ypos,label=parameter),
            size=3, nudge_x =0, nudge_y = 0, hjust = .5)+

  scale_color_viridis_d(option = 'C', name = '') +
  scale_shape(name = 'Station') +

  coord_fixed(xlim =c(-1.35, 1.35)) +

  guides(color = guide_legend(title.position = "bottom",
                              title.hjust = 0.5,
                              override.aes = list(size = 2),
                              order = 1),
         shape = guide_legend(title.position = "bottom",
                              title.hjust = 0.5,
                              override.aes = list(size = 2),
                              order = 2))
```

```
plt +
  theme_classic(base_size = 10) +
  theme(
        legend.position = 'bottom',
        legend.box = 'Vertical',
        legend.spacing.y = unit(0, 'cm'),
        legend.margin = margin(0,0,0,0))
```



So, we see that Discharge, and Temperature (and Oxygen Saturation, not shown because of missing data, and and Season, not shown because it is a factor) are strongly associated with Axis 1. Unfortunately, community response to these measured variables are highly correlated, which is likely to pose problems for estimation in the GAM models to come.

That is still too busy.

Significant Environmental Variables

```
ef_2$vectors
#>              NMDS1    NMDS2     r2 Pr(>r)
#> disch_wk -0.93244 -0.36133 0.5852 0.0001 ***
#> Temp      0.96695  0.25497 0.7434 0.0001 ***
#> Sal       0.88012  0.47476 0.0737 0.1257
#> Turb     -0.57811  0.81596 0.0300 0.4357
#> Chl       0.77421 -0.63293 0.0623 0.1723
#> RH       -0.28444  0.95869 0.1316 0.0240 *
#> Fish     -0.43622  0.89984 0.0847 0.0862 .
#> Turb2    -0.53315  0.84602 0.0355 0.3749
#> Chl2      0.86763 -0.49722 0.1631 0.0082 **
#> RH2       0.34358  0.93912 0.1316 0.0210 *
#> Fish2    -0.36712  0.93017 0.0268 0.4650
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

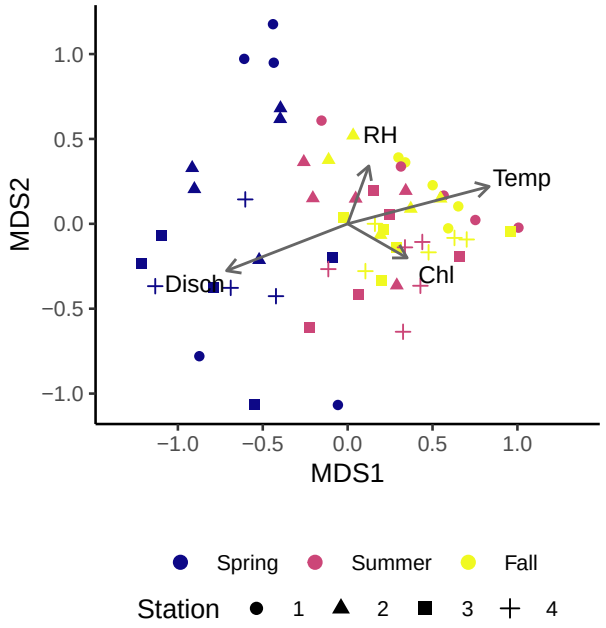
```
#> Permutation: free
#> Number of permutations: 9999
```

Several variables showed no significant link to the NMDS plot, according to the `envfit()` function output. We can simplify one notch more by dropping non-significant variables.

```
tmp_arrows <- scaled_arrows %>%
  filter(parameter %in% c('disch_wk', 'Temp',  'Chl2',  'RH2')) %>%
  mutate(parameter = factor(parameter,
                            levels = c("disch_wk", 'Temp', 'Chl2',  'RH2'),
                            labels = c('Disch', 'Temp',  'Chl', 'RH')))

plt <- ggplot(data = envNMDS, aes(MDS1, MDS2)) +
  geom_point(aes(color = Season, shape = Station), size = 1.5) +
  geom_segment(data=tmp_arrows,
               mapping = aes(x=0,xend = NMDS1,y=0,yend=NMDS2),
               arrow = arrow(length = unit(0.2, "cm")) ,colour="grey40") +
  geom_text(data=tmp_arrows,
            mapping = aes(x=ann_xpos,
                          y=ann_ypos,label=parameter),
            size=3, nudge_x =0, nudge_y = 0, hjust = .5)+
  scale_color_viridis_d(option = 'C', name = '') +
  scale_shape(name = 'Station') +

  coord_fixed(xlim =c(-1.35, 1.35)) +

  guides(color = guide_legend(title.position = "top",
                              title.hjust = 0.5,
                              override.aes = list(size = 2),
                              order = 1),
         shape = guide_legend(title.position = "left",
                              title.hjust = 0.5,
                              override.aes = list(size = 2),
                              order = 2))
```

We can't change the figure width, but we can alter the figure height. With that in mind, let's reorient the legends and juggle dimensions

```
plt +
  theme_classic(base_size = 10) +
  theme(
        legend.position = 'bottom',
        legend.box = 'Vertical',
        legend.spacing.y = unit(0, 'cm'),
        legend.box.just = 'top',
        legend.margin = margin(0,0,0,0),
        legend.box.margin = margin(0,0,0,0)
        )
```




```
ggsave('figures/nmds_env_significant.png', type='cairo',
         width = 3.3, height = 3.4)
ggsave('figures/nmds_env_significant.pdf', device = cairo_pdf,
       width = 3.3, height = 3.4)
```

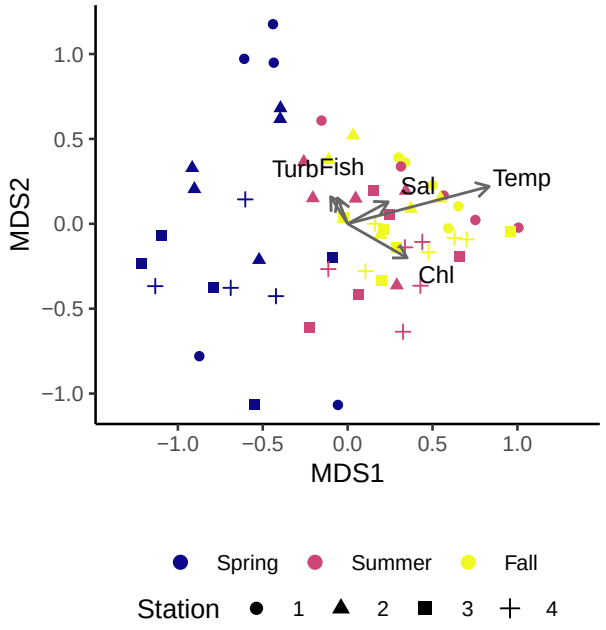
Environmental Variables Used in Models

Another way to simplify is to only show variables we use in the GAM and LMER models.

```
tmp_arrows <- scaled_arrows %>%
  filter(parameter %in% c('Temp', 'Sal', 'Turb2', 'Chl2',  'Fish2')) %>%
  mutate(parameter = factor(parameter,
                            levels = c('Temp', 'Sal', 'Turb2', 'Chl2',  'Fish2'),
                            labels = c('Temp', 'Sal', 'Turb', 'Chl',  'Fish'))) %>%
  mutate(ann_xpos = case_when(
    parameter == 'Turb' ~ ann_xpos - 0.1,
    parameter == 'Fish' ~ ann_xpos + 0.1,
    TRUE ~ ann_xpos))

plt <- ggplot(data = envNMDS, aes(MDS1, MDS2)) +
  geom_point(aes(color = Season, shape = Station), size = 1.5) +
  geom_segment(data=tmp_arrows,
               mapping = aes(x=0,xend=NMDS1,y=0,yend=NMDS2),
               arrow = arrow(length = unit(0.2, "cm")) ,colour="grey40") +
  geom_text(data=tmp_arrows,
            mapping = aes(x=ann_xpos,
                          y=ann_ypos,
                          label=parameter),
            size=3, nudge_x =0, nudge_y = 0, hjust = .5)+
  scale_color_viridis_d(option = 'C', name = '') +
  scale_shape(name = 'Station') +
```

```
  coord_fixed(xlim =c(-1.35, 1.35)) +

  guides(color = guide_legend(title.position = "top",
                              title.hjust = 0.5,
                              override.aes = list(size = 2),
                              order = 1),
         shape = guide_legend(title.position = "left",
                              title.hjust = 0.5,
                              override.aes = list(size = 2),
                              order = 2))
```

```
plt +
  theme_classic(base_size = 10) +
  theme(
        legend.position = 'bottom',
        legend.box = 'Vertical',
        legend.spacing.y = unit(0, 'cm'),
        legend.box.just = 'top',
        legend.margin = margin(0,0,0,0),
        legend.box.margin = margin(0,0,0,0)
        )
```




```
ggsave('figures/nmds_env_selected.png', type='cairo',
         width = 3.3, height = 3.4)
ggsave('figures/nmds_env_selected.pdf', device = cairo_pdf,
       width = 3.3, height = 3.4)
```

Qualitative Conclusions

- Axis 1 is highly correlated with season, and thus highly correlated with discharge, temperature and oxygen saturation (which was dropped from this graphic because of missing 2013 data).

- Axis 2 is closely related to Station, with upstream stations to the upper right and downstream stations to the lower left. Two "oddball" Station 1 samples are extreme low salinity spring samples – the same samples that cause problems fitting many of our models. The second axis is not especially correlated with ANY of the environmental variables.
- I run the GAM analysis without the low salinity oddball samples later in this notebook, but did not produce any graphics there. It would not be hard to do so.

GAM Analysis (Full Data)

Environmental Drivers

Axis 1

Note I increase the iterations to fit the model. This probably means the gradient near the solution is low, so parameter estimates may not be very good.

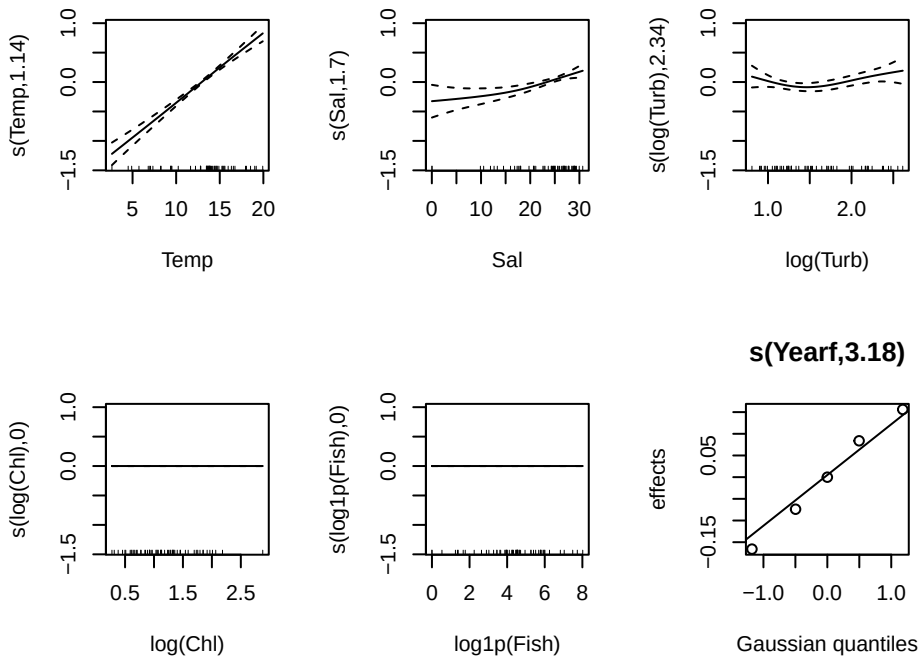
This is the "full" model. This includes all potential predictors, both experimental (Station, Season, Year, Sampling Event) and measured (everything else).

```
gam_1_1 <- gam(MDS1 ~
                s(Temp, bs="ts", k = 5) +
                s(Sal, bs="ts", k = 5) +
                s(log(Turb), bs="ts", k = 5) +
                s(log(Chl), bs="ts", k = 5) +
                s(log1p(Fish),bs="ts", k = 5) +
                s(Yearf, bs = 're'),
              data = envNMDS, family = 'gaussian')
summary(gam_1_1)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> MDS1 ~ s(Temp, bs = "ts", k = 5) + s(Sal, bs = "ts", k = 5) +
#>     s(log(Turb), bs = "ts", k = 5) + s(log(Chl), bs = "ts", k = 5) +
#>     s(log1p(Fish), bs = "ts", k = 5) + s(Yearf, bs = "re")
#>
#> Parametric coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 0.003447   0.068650    0.05     0.96
#>
#> Approximate significance of smooth terms:
#>                       edf Ref.df      F  p-value
#> s(Temp)         1.143e+00      4 67.971  < 2e-16 ***
#> s(Sal)          1.702e+00      4  4.898 0.000233 ***
#> s(log(Turb))    2.341e+00      4  2.248 0.029212 *
#> s(log(Chl))     3.732e-10      4  0.000 0.823127
#> s(log1p(Fish)) 1.040e-09      4  0.000 0.473495
#> s(Yearf)        3.178e+00      4  4.079 0.001306 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
```

```
#> R-sq.(adj) =  0.824   Deviance explained =   85%
#> GCV = 0.062477  Scale est. = 0.05239   n = 58
```

```
concurvity(gam_1_1)
#>          para   s(Temp)    s(Sal) s(log(Turb)) s(log(Chl)) s(log1p(Fish))
#> worst       1 0.6687582 0.7446731    0.4388267   0.6815834      0.4367056
#> observed    1 0.5208394 0.6578673    0.3412011   0.6323646      0.3265359
#> estimate    1 0.5372910 0.5754949    0.3616421   0.6194483      0.3402522
#>           s(Yearf)
#> worst    1.0000000
#> observed 0.3566667
#> estimate 0.5359525
```

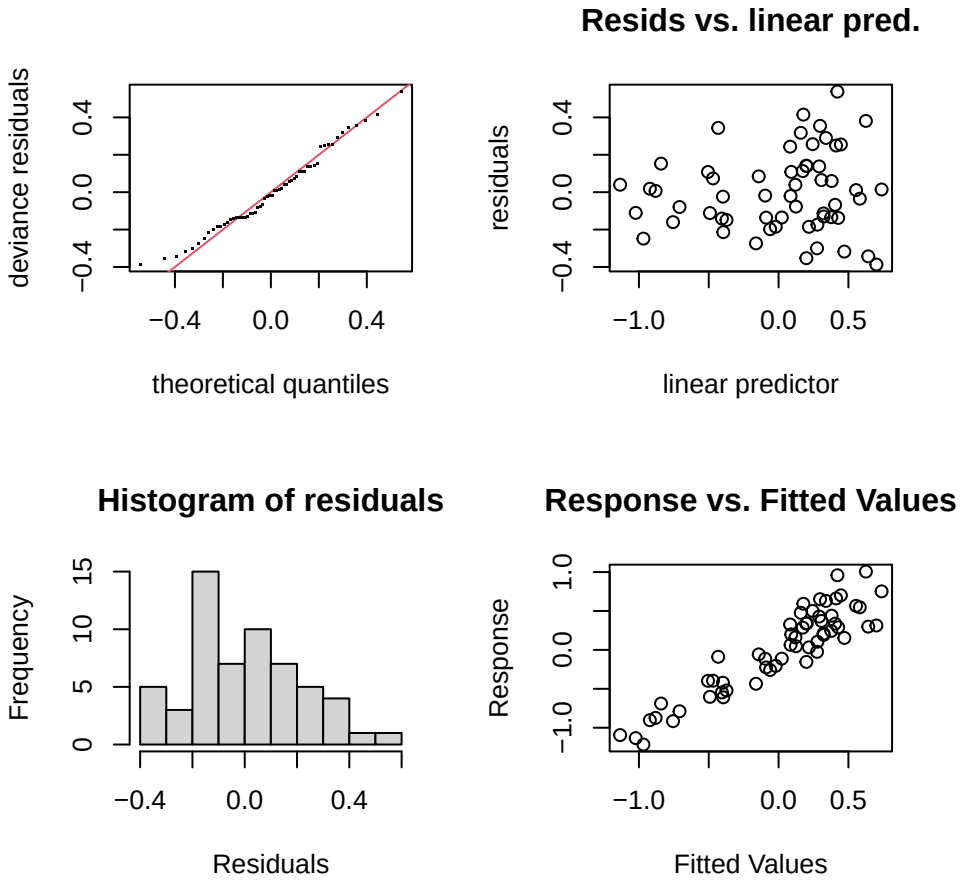
```
oldpar <- par(mfrow = c(2,3))
plot(gam_1_1)
```




```
par(oldpar)
```

As we saw from the `envfit()` analysis, Axis 1 is associated with Temperature and to a lesser extent, salinity. This is likely to be largely a seasonal pattern.

```
oldpar <- par(mfrow = c(2,2))
gam.check(gam_1_1)
```



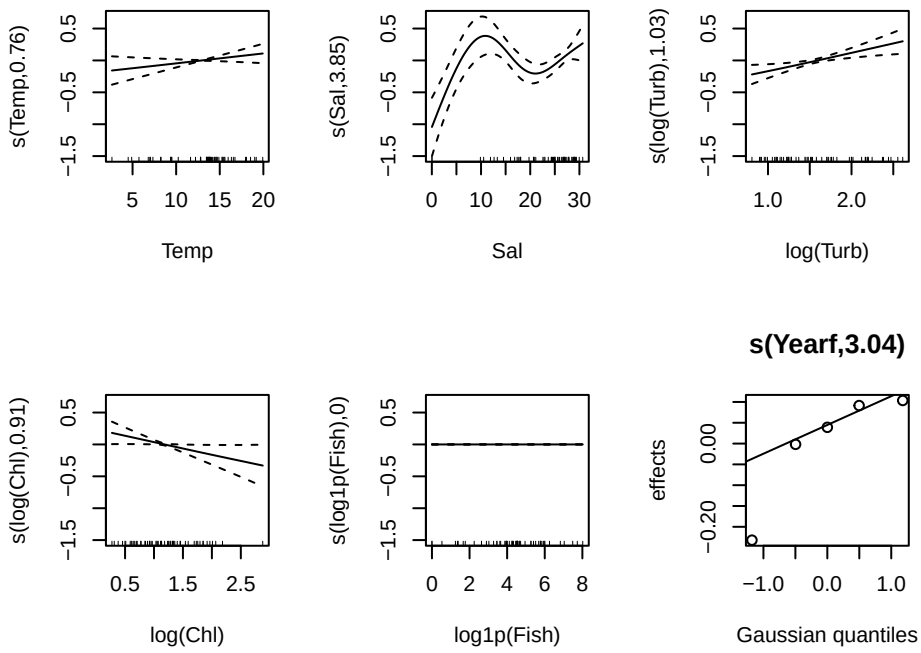

```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 16 iterations.
#> The RMS GCV score gradient at convergence was 4.482916e-08 .
#> The Hessian was positive definite.
#> Model rank =  26 / 26
#>
#> Basis dimension (k) checking results. Low p-value (k-index<1) may
#> indicate that k is too low, especially if edf is close to k'.
#>
#>                      k'      edf k-index p-value
#> s(Temp)        4.00e+00 1.14e+00    1.08    0.70
#> s(Sal)         4.00e+00 1.70e+00    1.08    0.66
#> s(log(Turb))   4.00e+00 2.34e+00    0.98    0.37
#> s(log(Chl))    4.00e+00 3.73e-10    1.14    0.80
#> s(log1p(Fish)) 4.00e+00 1.04e-09    1.07    0.70
#> s(Yearf)       5.00e+00 3.18e+00      NA      NA
par(oldpar)
```

The model is fairly well behaved. No obvious problems here.

Axis 2

```
gam_2_1 <- gam(MDS2 ~
               s(Temp, bs="ts", k = 5) +
               s(Sal, bs="ts", k = 5) +
               s(log(Turb), bs="ts", k = 5) +
               s(log(Chl), bs="ts", k = 5) +
               s(log1p(Fish),bs="ts", k = 5) +
               s(Yearf, bs = 're'),
             data = envNMDS, family = 'gaussian')
summary(gam_2_1)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> MDS2 ~ s(Temp, bs = "ts", k = 5) + s(Sal, bs = "ts", k = 5) +
#>     s(log(Turb), bs = "ts", k = 5) + s(log(Chl), bs = "ts", k = 5) +
#>     s(log1p(Fish), bs = "ts", k = 5) + s(Yearf, bs = "re")
#>
#> Parametric coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) -0.01782    0.08863  -0.201    0.842
#>
#> Approximate significance of smooth terms:
#>                     edf Ref.df      F  p-value
#> s(Temp)        7.574e-01      4  0.733 0.094026 .
#> s(Sal)         3.849e+00      4 10.736 0.000116 ***
#> s(log(Turb))   1.026e+00      4  3.144 0.001845 **
#> s(log(Chl))    9.141e-01      4  2.123 0.025532 *
#> s(log1p(Fish)) 5.095e-10      4  0.000 0.967772
#> s(Yearf)       3.036e+00      4  2.480 0.017158 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.439   Deviance explained = 53.4%
#> GCV = 0.11603  Scale est. = 0.094856  n = 58
```

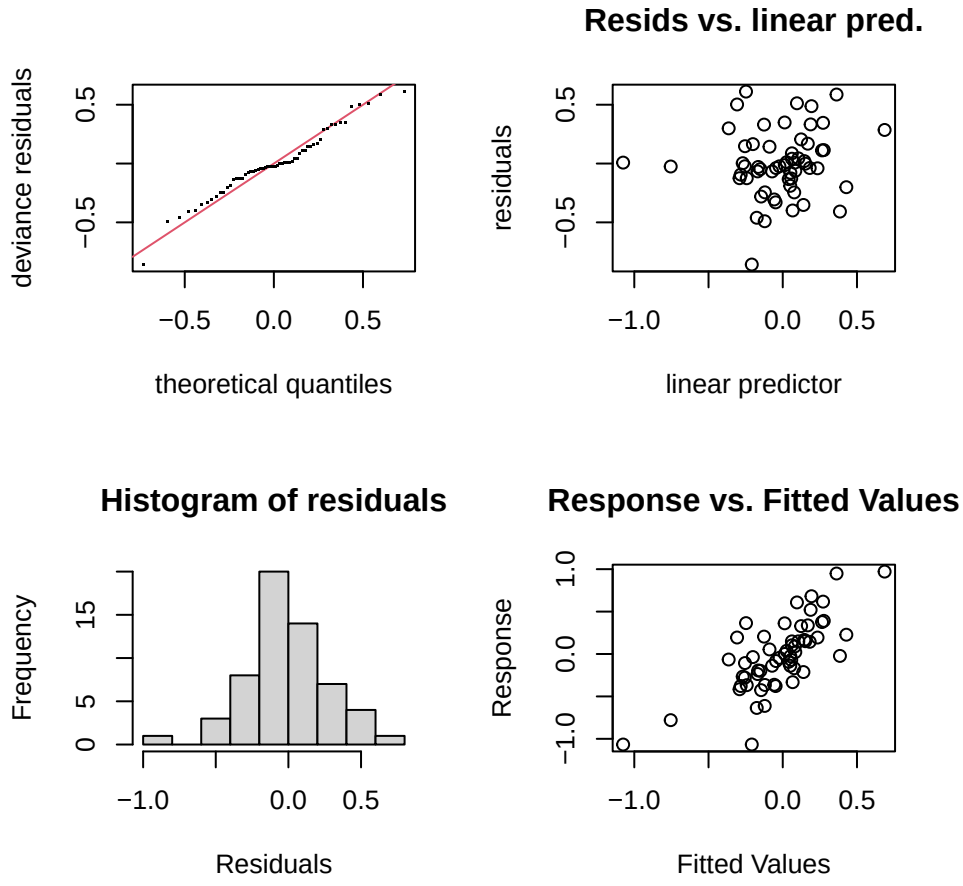
```
oldpar <- par(mfrow = c(2,3))
plot(gam_2_1)
```




```
par(oldpar)
```

Axis 2 is dominated by the salinity response, which is itself driven in part by the low salinity samples.

```
oldpar <- par(mfrow = c(2,2))
gam.check(gam_2_1)
```




```
#>
#> Method: GCV   Optimizer: magic
#> Smoothing parameter selection converged after 21 iterations.
#> The RMS GCV score gradient at convergence was 5.987553e-08 .
#> The Hessian was positive definite.
#> Model rank =  26 / 26
#>
#> 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)        4.00e+00 7.57e-01    1.09    0.71
#> s(Sal)         4.00e+00 3.85e+00    1.07    0.61
#> s(log(Turb))   4.00e+00 1.03e+00    0.96    0.38
#> s(log(Chl))    4.00e+00 9.14e-01    1.07    0.71
#> s(log1p(Fish)) 4.00e+00 5.10e-10    0.97    0.39
#> s(Yearf)       5.00e+00 3.04e+00      NA      NA
par(oldpar)
```

Axis 2 shows the effect of a couple of low salinity samples. Those Samples plot in NMDS space in the lower left, far from most Station 1 samples, so the GAM smoother fits a strong change in axis 2 scores to better predict those low salinity, early spring "washout" samples.

Season and Station GAM Analysis – Full Data

Recall that the alternative way of constructing a GAM drops quantitative predictors Temperature and Salinity and retains Station and Season as factors. I Include these models for completeness. I did double check concurvity and standard GAM diagnostics, but saw no problems, so do not present them here.

Axis 1

```
ss_gam_1_1 <- gam(MDS1 ~
                    Station +
                    Season +
                  s(log(Turb), bs="ts", k = 5) +
                  s(log(Chl), bs="ts", k = 5) +
                  s(log1p(Fish),bs="ts", k = 5) +
                  s(Yearf, bs = 're'),
                data = envNMDS, family = 'gaussian')
anova(ss_gam_1_1)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> MDS1 ~ Station + Season + s(log(Turb), bs = "ts", k = 5) + s(log(Chl),
#>     bs = "ts", k = 5) + s(log1p(Fish), bs = "ts", k = 5) + s(Yearf,
#>     bs = "re")
#>
#> Parametric Terms:
#>         df      F  p-value
#> Station  3  2.698   0.0566
#> Season   2 81.112 7.71e-16
#>
#> Approximate significance of smooth terms:
#>                       edf    Ref.df     F p-value
#> s(log(Turb))   1.916e+00 4.000e+00 0.815 0.25808
#> s(log(Chl))    6.346e-10 4.000e+00 0.000 0.77036
#> s(log1p(Fish)) 6.284e-01 4.000e+00 0.379 0.13997
#> s(Yearf)       3.266e+00 4.000e+00 4.216 0.00152
```

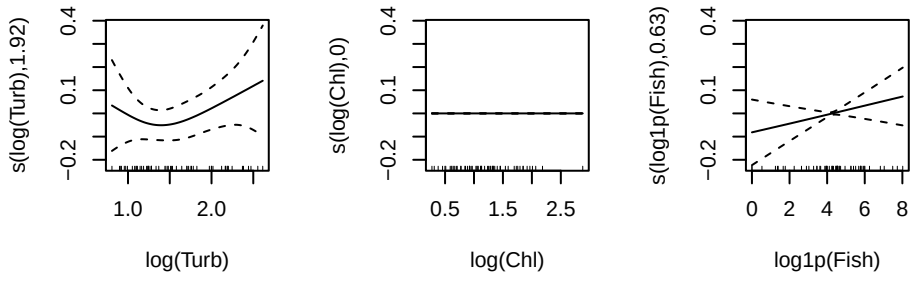
```
summary(ss_gam_1_1)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> MDS1 ~ Station + Season + s(log(Turb), bs = "ts", k = 5) + s(log(Chl),
#>     bs = "ts", k = 5) + s(log1p(Fish), bs = "ts", k = 5) + s(Yearf,
#>     bs = "re")
#>
#> Parametric coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  -0.54442    0.12045  -4.520 4.29e-05 ***
#> Station2     -0.26320    0.09687  -2.717  0.00924 **
```

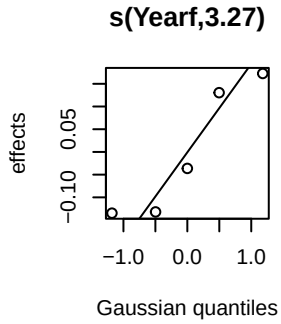
```
#> Station3     -0.18458    0.10197  -1.810  0.07678 .
#> Station4     -0.10332    0.10867  -0.951  0.34666
#> SeasonSummer  0.94100    0.08851  10.632 5.32e-14 ***
#> SeasonFall    1.05463    0.08960  11.770 1.68e-15 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>                       edf Ref.df     F p-value
#> s(log(Turb))   1.916e+00      4 0.815 0.25808
#> s(log(Chl))    6.346e-10      4 0.000 0.77036
#> s(log1p(Fish)) 6.284e-01      4 0.379 0.13997
#> s(Yearf)       3.266e+00      4 4.216 0.00152 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.775   Deviance explained = 81.7%
#> GCV = 0.084115  Scale est. = 0.066987  n = 58
```

The prior quantitative Axis 1 model provides a much better model by AIC.

```
AIC(gam_1_1, ss_gam_1_1)
#>                  df      AIC
#> gam_1_1    10.36354  4.06814
#> ss_gam_1_1 12.81057 20.22277
```

```
oldpar <- par(mfrow = c(2,3))
plot(ss_gam_1_1)
par(oldpar)
```

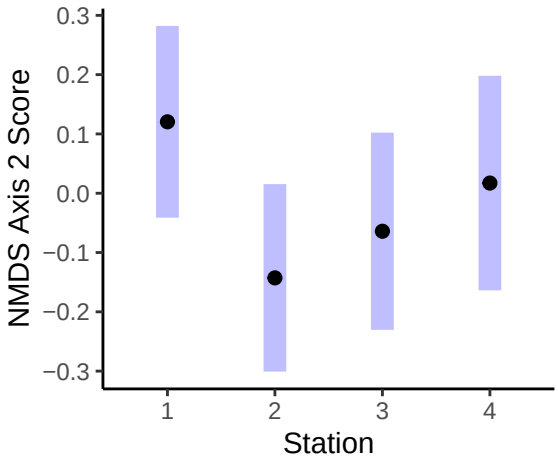




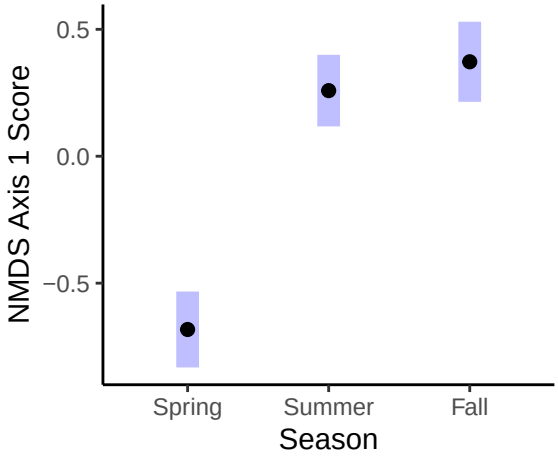
Axis 1 can be interpreted as highly correlated with season axis, as shown here. The spring is quite different from summer or fall in terms of the composition of the plankton community.

The Station pattern is not significant by ANOVA, and parameters are marginally significant by Wald test, so any observed pattern should be interpreted with care. But I present the pairwise comparisons for completeness.

```
emms <- emmeans(ss_gam_1_1, ~Station)
pairs(emms)
#>  contrast            estimate     SE   df t.ratio p.value
#>  Station1 - Station2   0.2632 0.0969 46.2   2.717  0.0442
#>  Station1 - Station3   0.1846 0.1020 46.2   1.810  0.2817
#>  Station1 - Station4   0.1033 0.1087 46.2   0.951  0.7777
#>  Station2 - Station3  -0.0786 0.0995 46.2  -0.790  0.8587
#>  Station2 - Station4  -0.1599 0.1065 46.2  -1.501  0.4454
#>  Station3 - Station4  -0.0813 0.0975 46.2  -0.833  0.8383
#>
#> Results are averaged over the levels of: Season, Yearf
#> P value adjustment: tukey method for comparing a family of 4 estimates

plot(emms) +
  coord_flip() +
  xlab('NMDS Axis 2 Score')
```




```
emms <- emmeans(ss_gam_1_1, ~Season)
pairs(emms)
#>  contrast        estimate     SE   df t.ratio p.value
#>  Spring - Summer   -0.941 0.0885 46.2 -10.632  <.0001
#>  Spring - Fall     -1.055 0.0896 46.2 -11.770  <.0001
#>  Summer - Fall     -0.114 0.0835 46.2  -1.361  0.3693
#>
#> Results are averaged over the levels of: Station, Yearf
#> P value adjustment: tukey method for comparing a family of 3 estimates

plot(emms) +
  coord_flip() +
  xlab('NMDS Axis 1 Score')
```



Axis 2

```
ss_gam_2_1 <- gam(MDS2 ~
                  Station +
                  Season +
```

```
              s(log(Turb), bs="ts", k = 5) +
              s(log(Chl), bs="ts", k = 5) +
              s(log1p(Fish),bs="ts", k = 5) +
              s(Yearf, bs = 're'),
            data = envNMDS, family = 'gaussian')
anova(ss_gam_2_1)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> MDS2 ~ Station + Season + s(log(Turb), bs = "ts", k = 5) + s(log(Chl),
#>     bs = "ts", k = 5) + s(log1p(Fish), bs = "ts", k = 5) + s(Yearf,
#>     bs = "re")
#>
#> Parametric Terms:
#>         df     F  p-value
#> Station  3 6.675 0.000743
#> Season   2 1.710 0.191816
#>
#> Approximate significance of smooth terms:
#>                       edf    Ref.df     F p-value
#> s(log(Turb))   2.621e-10 4.000e+00 0.000   0.812
#> s(log(Chl))    4.420e-01 4.000e+00 0.215   0.184
#> s(log1p(Fish)) 2.455e+00 4.000e+00 1.411   0.092
#> s(Yearf)       1.314e+00 4.000e+00 0.484   0.213
```

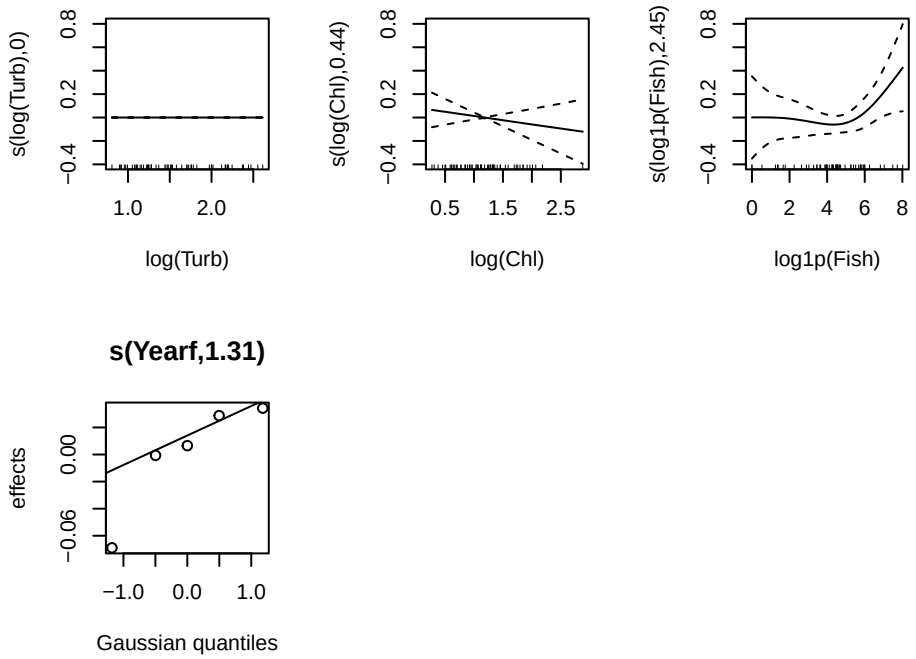
```
summary(ss_gam_2_1)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> MDS2 ~ Station + Season + s(log(Turb), bs = "ts", k = 5) + s(log(Chl),
#>     bs = "ts", k = 5) + s(log1p(Fish), bs = "ts", k = 5) + s(Yearf,
#>     bs = "re")
#>
#> Parametric coefficients:
#>               Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  -0.008016   0.137947  -0.058  0.95390
#> Station2      0.125263   0.133148   0.941  0.35155
#> Station3     -0.370048   0.132005  -2.803  0.00729 **
#> Station4     -0.305332   0.138485  -2.205  0.03231 *
#> SeasonSummer  0.130083   0.129846   1.002  0.32147
#> SeasonFall    0.233372   0.126729   1.842  0.07176 .
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>                       edf Ref.df     F p-value
#> s(log(Turb))   2.621e-10      4 0.000   0.812
#> s(log(Chl))    4.420e-01      4 0.215   0.184
#> s(log1p(Fish)) 2.455e+00      4 1.411   0.092 .
```

```
#> s(Yearf)       1.314e+00      4 0.484   0.213
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.303   Deviance explained = 41.6%
#> GCV = 0.14301  Scale est. = 0.11784   n = 58
```

Again, the quantitative predictors model provides a significantly better model by AIC.

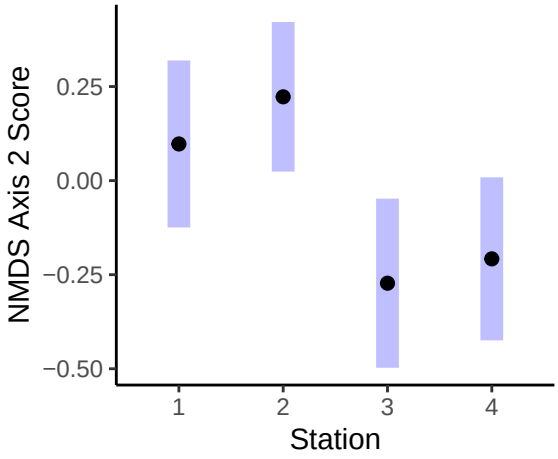
```
AIC(gam_2_1, ss_gam_2_1)
#>                  df      AIC
#> gam_2_1    11.58236 39.46494
#> ss_gam_2_1 11.21009 51.75576
```

```
oldpar <- par(mfrow = c(2,3))
plot(ss_gam_2_1)
par(oldpar)
```



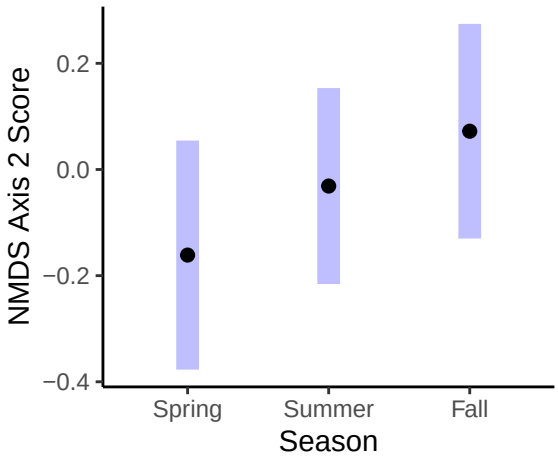

```
emms <- emmeans(ss_gam_2_1, ~Station)
pairs(emms)
#>  contrast           estimate    SE   df t.ratio p.value
#>  Station1 - Station2  -0.1253 0.133 47.8  -0.941  0.7831
#>  Station1 - Station3   0.3700 0.132 47.8   2.803  0.0355
#>  Station1 - Station4   0.3053 0.138 47.8   2.205  0.1366
#>  Station2 - Station3   0.4953 0.130 47.8   3.824  0.0021
#>  Station2 - Station4   0.4306 0.129 47.8   3.336  0.0087
#>  Station3 - Station4  -0.0647 0.131 47.8  -0.494  0.9599
```

```
#>
#> Results are averaged over the levels of: Season, Yearf
#> P value adjustment: tukey method for comparing a family of 4 estimates

plot(emms) +
  coord_flip() +
  xlab('NMDS Axis 2 Score')
```



Axis 2 scores show no significant patterns with Season I show pairwise comparisons only for completeness.

```
emms <- emmeans(ss_gam_2_1, ~Season)
pairs(emms)
#>  contrast        estimate    SE   df t.ratio p.value
#>  Spring - Summer   -0.130 0.130 47.8  -1.002  0.5793
#>  Spring - Fall     -0.233 0.127 47.8  -1.842  0.1672
#>  Summer - Fall     -0.103 0.112 47.8  -0.918  0.6316
#>
#> Results are averaged over the levels of: Station, Yearf
#> P value adjustment: tukey method for comparing a family of 3 estimates

plot(emms) +
  coord_flip() +
  xlab('NMDS Axis 2 Score')
```



Gam Analysis – Reduced Data

Preparation

We need to rerun the NMDS analysis, omitting the low salinity samples. While that should not alter the NMDS much, it at produces NMDS scores based on the same sample as we will use to fit the GAM.

Omit Low Salinity Samples

We have to be careful to do this for **both** Environmental Data and community data.

```
# First confirm alignment of samples
zoopl2 <- zoopl2 %>%
  arrange(DATE, STATION)
station_data <- station_data %>%
  arrange(Date, Station)

# Then calculate which samples to keep.
dont_drop <- which(station_data$Sal >= 10)
zoopl3<-zoopl2[dont_drop, ]
station_data_3 <- station_data[dont_drop,]

# Assemble the species only data set for NMDS analysis.
CDATA3 <- zoopl3[,-c(1:4)]
```

NMDS Analyses on Reduced Data

```
NMDSE3 <- metaMDS(CDATA3, autotransform = FALSE, k = 2, trymax = 75)
#> Run 0 stress 0.138909
#> Run 1 stress 0.138909
#> ... Procrustes: rmse 2.12258e-05  max resid 9.364902e-05
#> ... Similar to previous best
#> Run 2 stress 0.138909
```

```
#> ... New best solution
#> ... Procrustes: rmse 2.971204e-06  max resid 1.089249e-05
#> ... Similar to previous best
#> Run 3 stress 0.138909
#> ... New best solution
#> ... Procrustes: rmse 4.884639e-06  max resid 2.205477e-05
#> ... Similar to previous best
#> Run 4 stress 0.138909
#> ... Procrustes: rmse 6.688158e-06  max resid 2.753458e-05
#> ... Similar to previous best
#> Run 5 stress 0.1852538
#> Run 6 stress 0.1518512
#> Run 7 stress 0.1517944
#> Run 8 stress 0.138909
#> ... Procrustes: rmse 2.328478e-06  max resid 1.056825e-05
#> ... Similar to previous best
#> Run 9 stress 0.1839309
#> Run 10 stress 0.138909
#> ... Procrustes: rmse 8.537858e-06  max resid 3.581578e-05
#> ... Similar to previous best
#> Run 11 stress 0.138909
#> ... Procrustes: rmse 1.524268e-05  max resid 7.034005e-05
#> ... Similar to previous best
#> Run 12 stress 0.1579655
#> Run 13 stress 0.138909
#> ... Procrustes: rmse 8.16158e-06  max resid 3.650435e-05
#> ... Similar to previous best
#> Run 14 stress 0.1952362
#> Run 15 stress 0.1642028
#> Run 16 stress 0.1696707
#> Run 17 stress 0.1839308
#> Run 18 stress 0.1517944
#> Run 19 stress 0.157966
#> Run 20 stress 0.1642027
#> *** Solution reached
NMDSE3
#>
#> Call:
#> metaMDS(comm = CDATA3, k = 2, trymax = 75, autotransform = FALSE)
#>
#> global Multidimensional Scaling using monoMDS
#>
#> Data:     CDATA3
#> Distance: bray
#>
#> Dimensions: 2
#> Stress:     0.138909
#> Stress type 1, weak ties
#> Two convergent solutions found after 20 tries
#> Scaling: centring, PC rotation, halfchange scaling
#> Species: expanded scores based on 'CDATA3'
```

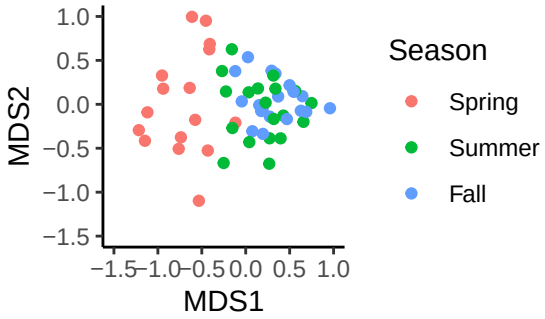
Combining the NMDS Results with Environmental Data

As before, I capitalize variable names here, so they will appear capitalized in graphics. I also create transformed versions of the predictor variables.

```
envNMDS3 <- station_data_3 %>%
  select(-Date, -Month, -DOY, -riv_km, -AvgTurb) %>%
  mutate(Turb2 = log(Turb),
         Chl2 = log(Chl),
         RH2 = log1p(RH),
         Fish2 = log1p(Fish)) %>%
  mutate(sample_seq = as.numeric(Season) + (Year-2013)*3,
         sample_event = factor(sample_seq)) %>%
  cbind(as_tibble(NMDSE3$points))
```

```
ggplot(envNMDS3, aes(MDS1, MDS2)) +
  geom_point(aes(color=Season)) +
  xlim(c(-1.5,1)) +
  ylim(c(-1.5,1)) +
  theme(aspect.ratio=1)
#> Warning: Removed 1 rows containing missing values (geom_point).
```

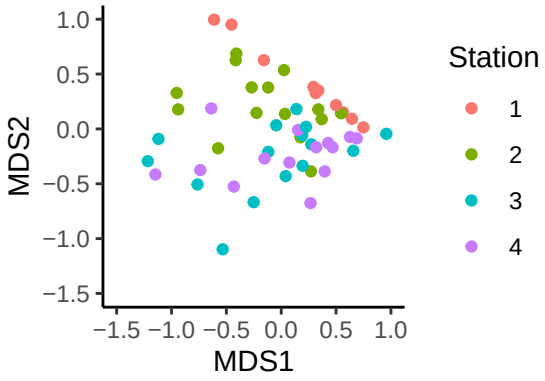
Plot NMDS By Season



The overall shape of the NMDS plot is nearly unchanged, and Axis 1 still shows **strong** association with season.

```
ggplot(envNMDS3, aes(MDS1, MDS2)) +
    geom_point(aes(color=Station)) +
  xlim(c(-1.5,1)) +
  ylim(c(-1.5,1)) +
  theme(aspect.ratio=1)
#> Warning: Removed 1 rows containing missing values (geom_point).
```

Plot NMDS By Station



The two "weird" station 1 samples that plotted to the lower left are now gone. This makes the association of Axis 2 with Station a little more obvious.

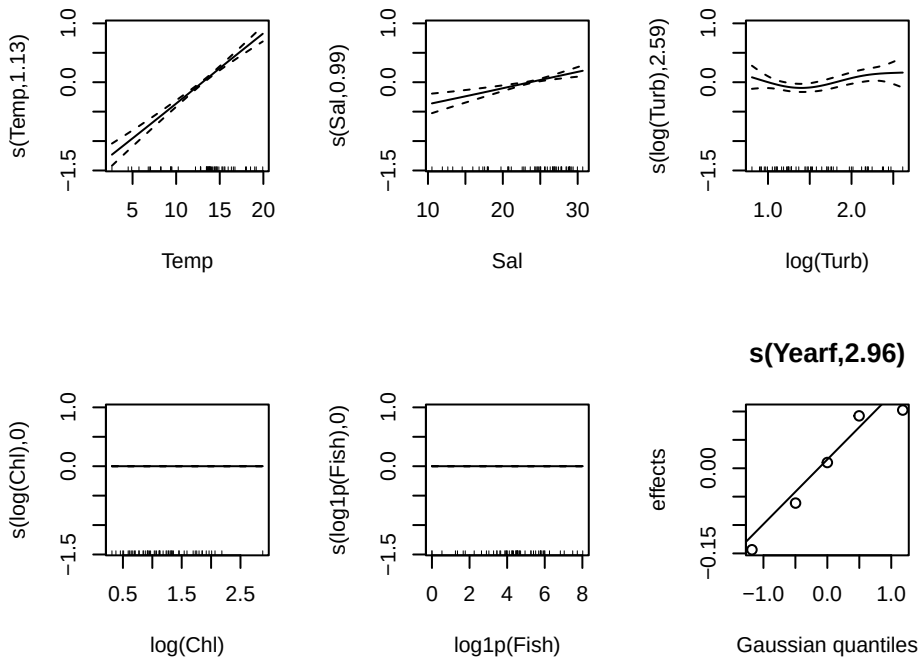
Environmental Drivers

Axis 1

```
gam_1_1_no_low <- gam(MDS1 ~
                s(Temp, bs="ts", k = 5) +
                s(Sal, bs="ts", k = 5) +
                s(log(Turb), bs="ts", k = 5) +
                s(log(Chl), bs="ts", k = 5) +
                s(log1p(Fish),bs="ts", k = 5) +
                s(Yearf, bs = 're'),
              data = envNMDS3, family = 'gaussian')
summary(gam_1_1_no_low)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> MDS1 ~ s(Temp, bs = "ts", k = 5) + s(Sal, bs = "ts", k = 5) +
#>     s(log(Turb), bs = "ts", k = 5) + s(log(Chl), bs = "ts", k = 5) +
#>     s(log1p(Fish), bs = "ts", k = 5) + s(Yearf, bs = "re")
#>
#> Parametric coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  0.00168    0.06102   0.028    0.978
#>
#> Approximate significance of smooth terms:
#>                     edf Ref.df      F  p-value
#> s(Temp)        1.126e+00      4 64.095  < 2e-16 ***
#> s(Sal)         9.949e-01      4  5.270 2.24e-05 ***
```

```
#> s(log(Turb))   2.591e+00      4  2.758  0.02025 *
#> s(log(Chl))    7.858e-10      4  0.000  0.92191
#> s(log1p(Fish)) 8.974e-10      4  0.000  0.51395
#> s(Yearf)       2.963e+00      4  3.008  0.00589 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.832   Deviance explained = 85.6%
#> GCV = 0.05937  Scale est. = 0.050005  n = 55
```

```
oldpar <- par(mfrow = c(2,3))
plot(gam_1_1_no_low)
```



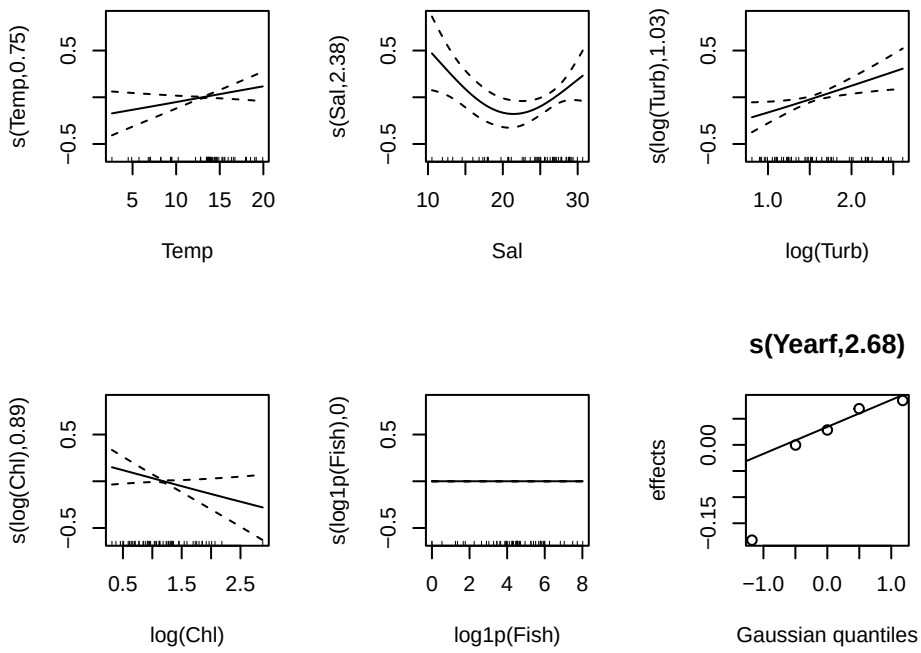

```
par(oldpar)
```

Axis 2

```
gam_2_1_no_low <- gam(MDS2 ~
                s(Temp, bs="ts", k = 5) +
                s(Sal, bs="ts", k = 5) +
                s(log(Turb), bs="ts", k = 5) +
                s(log(Chl), bs="ts", k = 5) +
                s(log1p(Fish),bs="ts", k = 5) +
                s(Yearf, bs = 're'),
              data = envNMDS3, family = 'gaussian')
```

```
summary(gam_2_1_no_low)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> MDS2 ~ s(Temp, bs = "ts", k = 5) + s(Sal, bs = "ts", k = 5) +
#>     s(log(Turb), bs = "ts", k = 5) + s(log(Chl), bs = "ts", k = 5) +
#>     s(log1p(Fish), bs = "ts", k = 5) + s(Yearf, bs = "re")
#>
#> Parametric coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept) -0.003309   0.081641  -0.041    0.968
#>
#> Approximate significance of smooth terms:
#>                      edf Ref.df     F p-value
#> s(Temp)        7.485e-01      4 0.723 0.08149 .
#> s(Sal)         2.382e+00      4 3.948 0.00397 **
#> s(log(Turb))   1.026e+00      4 2.508 0.00393 **
#> s(log(Chl))    8.879e-01      4 1.268 0.07245 .
#> s(log1p(Fish)) 1.895e-09      4 0.000 0.86869
#> s(Yearf)       2.685e+00      4 1.424 0.08254 .
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =   0.32   Deviance explained = 41.7%
#> GCV = 0.12775  Scale est. = 0.10748   n = 55
```

```
oldpar <- par(mfrow = c(2,3))
plot(gam_2_1_no_low)
```




```
par(oldpar)
```

Season and Station GAM Analysis – Full Data

Recall that the alternative way of constructing a GAM drops quantitative predictors Temperature and Salinity and retains Station and Season as factors. I Include these models for completeness.

Axis 1

In a Season and Station model, axis 1 shows strong relationship to Season, and a possible relationship to Station as well.

```
ss_gam_1_1_no_low <- gam(MDS1 ~
                   Station +
                   Season +
                #s(Temp, bs="ts", k = 5) +
                #s(Sal, bs="ts", k = 5) +
                s(log(Turb), bs="ts", k = 5) +
                s(log(Chl), bs="ts", k = 5) +
                s(log1p(Fish),bs="ts", k = 5) +
                s(Yearf, bs = 're'),
              data = envNMDS, family = 'gaussian')
anova(ss_gam_1_1_no_low)
#>
#> Family: gaussian
#> Link function: identity
#>
```

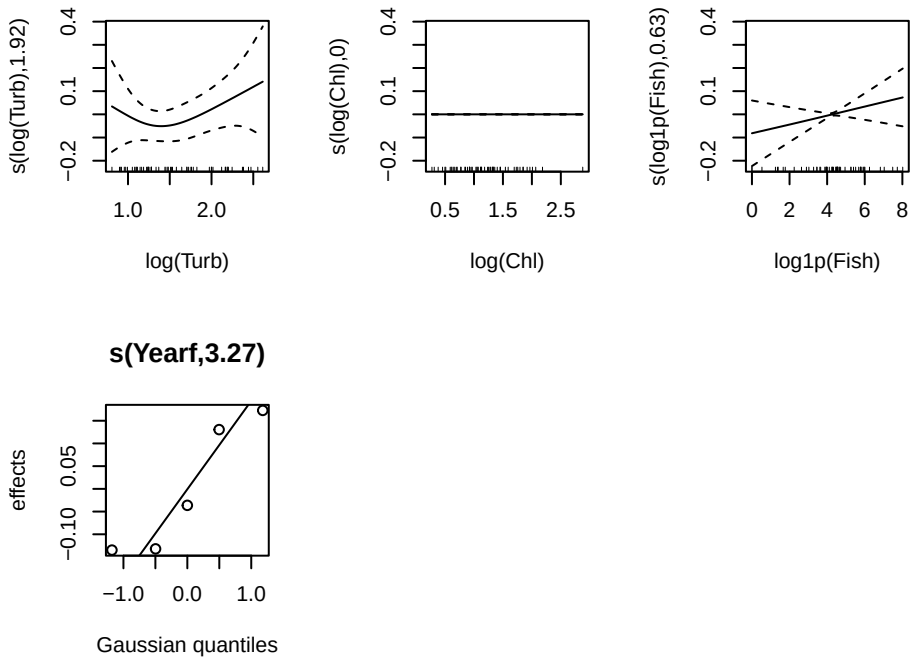
```
#> Formula:
#> MDS1 ~ Station + Season + s(log(Turb), bs = "ts", k = 5) + s(log(Chl),
#>     bs = "ts", k = 5) + s(log1p(Fish), bs = "ts", k = 5) + s(Yearf,
#>     bs = "re")
#>
#> Parametric Terms:
#>         df      F  p-value
#> Station  3  2.698   0.0566
#> Season   2 81.112 7.71e-16
#>
#> Approximate significance of smooth terms:
#>                       edf    Ref.df     F p-value
#> s(log(Turb))   1.916e+00 4.000e+00 0.815 0.25808
#> s(log(Chl))    6.346e-10 4.000e+00 0.000 0.77036
#> s(log1p(Fish)) 6.284e-01 4.000e+00 0.379 0.13997
#> s(Yearf)       3.266e+00 4.000e+00 4.216 0.00152
```

```
summary(ss_gam_1_1_no_low)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> MDS1 ~ Station + Season + s(log(Turb), bs = "ts", k = 5) + s(log(Chl),
#>     bs = "ts", k = 5) + s(log1p(Fish), bs = "ts", k = 5) + s(Yearf,
#>     bs = "re")
#>
#> Parametric coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  -0.54442    0.12045  -4.520 4.29e-05 ***
#> Station2     -0.26320    0.09687  -2.717  0.00924 **
#> Station3     -0.18458    0.10197  -1.810  0.07678 .
#> Station4     -0.10332    0.10867  -0.951  0.34666
#> SeasonSummer  0.94100    0.08851  10.632 5.32e-14 ***
#> SeasonFall    1.05463    0.08960  11.770 1.68e-15 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>                       edf Ref.df     F p-value
#> s(log(Turb))   1.916e+00      4 0.815 0.25808
#> s(log(Chl))    6.346e-10      4 0.000 0.77036
#> s(log1p(Fish)) 6.284e-01      4 0.379 0.13997
#> s(Yearf)       3.266e+00      4 4.216 0.00152 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.775   Deviance explained = 81.7%
#> GCV = 0.084115  Scale est. = 0.066987  n = 58
```

The prior quantitative Axis 1 model provides a much better model by AIC.

```
AIC(gam_1_1_no_low, ss_gam_1_1_no_low)
#>                             df       AIC
#> gam_1_1_no_low      9.675292  1.233298
#> ss_gam_1_1_no_low 12.810574 20.222766
```

```
oldpar <- par(mfrow = c(2,3))
plot(ss_gam_1_1_no_low)
par(oldpar)
```

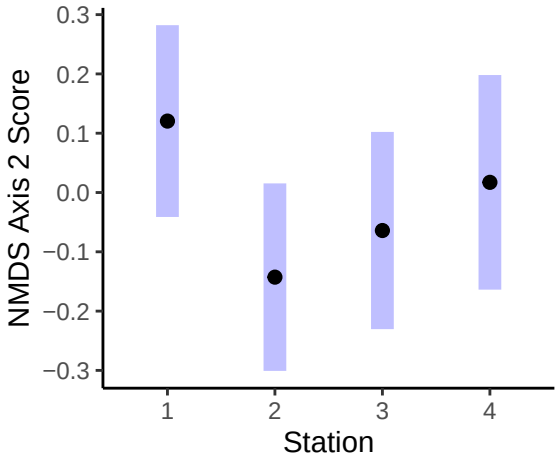


Axis 1 is highly correlated with season, as shown here.

The Station pattern is not significant by ANOVA, so any observed pattern should be interpreted with care. But I present the pairwise comparisons for completeness.

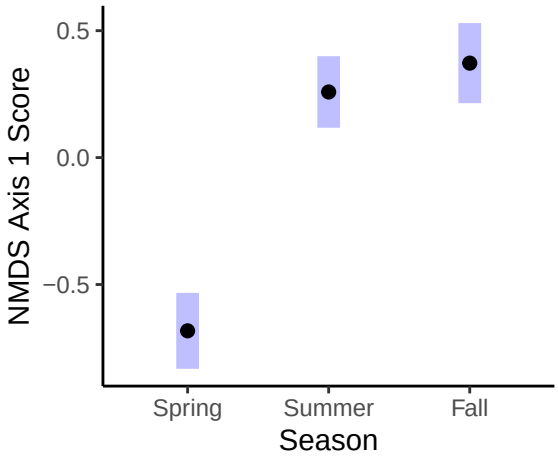
```
emms <- emmeans(ss_gam_1_1_no_low, ~Station)
pairs(emms)
#>  contrast            estimate     SE   df t.ratio p.value
#>  Station1 - Station2   0.2632 0.0969 46.2   2.717  0.0442
#>  Station1 - Station3   0.1846 0.1020 46.2   1.810  0.2817
#>  Station1 - Station4   0.1033 0.1087 46.2   0.951  0.7777
#>  Station2 - Station3  -0.0786 0.0995 46.2  -0.790  0.8587
#>  Station2 - Station4  -0.1599 0.1065 46.2  -1.501  0.4454
#>  Station3 - Station4  -0.0813 0.0975 46.2  -0.833  0.8383
#>
#> Results are averaged over the levels of: Season, Yearf
#> P value adjustment: tukey method for comparing a family of 4 estimates

plot(emms) +
```

```
  coord_flip() +
  xlab('NMDS Axis 2 Score')
```



The spring is quite different from summer or fall in terms of the composition of the plankton community.

```
emms <- emmeans(ss_gam_1_1_no_low, ~Season)
pairs(emms)
#>  contrast        estimate     SE   df t.ratio p.value
#>  Spring - Summer   -0.941 0.0885 46.2 -10.632  <.0001
#>  Spring - Fall     -1.055 0.0896 46.2 -11.770  <.0001
#>  Summer - Fall     -0.114 0.0835 46.2  -1.361  0.3693
#>
#> Results are averaged over the levels of: Station, Yearf
#> P value adjustment: tukey method for comparing a family of 3 estimates

plot(emms) +
  coord_flip() +
  xlab('NMDS Axis 1 Score')
```



Axis 2

```
ss_gam_2_1_no_low <- gam(MDS2 ~
                   Station +
                   Season +
                #s(Temp, bs="ts", k = 5) +
                #s(Sal, bs="ts", k = 5) +
                s(log(Turb), bs="ts", k = 5) +
                s(log(Chl), bs="ts", k = 5) +
                s(log1p(Fish),bs="ts", k = 5) +
                s(Yearf, bs = 're'),
              data = envNMDS, family = 'gaussian')
anova(ss_gam_2_1_no_low)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> MDS2 ~ Station + Season + s(log(Turb), bs = "ts", k = 5) + s(log(Chl),
#>     bs = "ts", k = 5) + s(log1p(Fish), bs = "ts", k = 5) + s(Yearf,
#>     bs = "re")
#>
#> Parametric Terms:
#>         df     F  p-value
#> Station  3 6.675 0.000743
#> Season   2 1.710 0.191816
#>
#> Approximate significance of smooth terms:
#>                       edf    Ref.df     F p-value
#> s(log(Turb))   2.621e-10 4.000e+00 0.000   0.812
#> s(log(Chl))    4.420e-01 4.000e+00 0.215   0.184
#> s(log1p(Fish)) 2.455e+00 4.000e+00 1.411   0.092
#> s(Yearf)       1.314e+00 4.000e+00 0.484   0.213
```

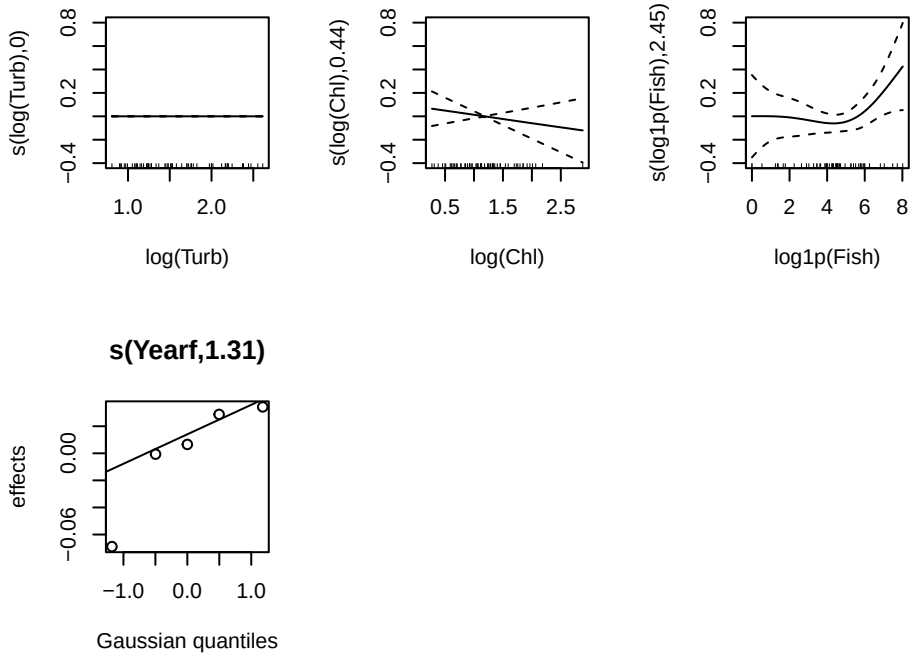
```
summary(ss_gam_2_1_no_low)
#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> MDS2 ~ Station + Season + s(log(Turb), bs = "ts", k = 5) + s(log(Chl),
#>     bs = "ts", k = 5) + s(log1p(Fish), bs = "ts", k = 5) + s(Yearf,
#>     bs = "re")
#>
#> Parametric coefficients:
#>               Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  -0.008016   0.137947  -0.058  0.95390
#> Station2      0.125263   0.133148   0.941  0.35155
#> Station3     -0.370048   0.132005  -2.803  0.00729 **
#> Station4     -0.305332   0.138485  -2.205  0.03231 *
#> SeasonSummer  0.130083   0.129846   1.002  0.32147
#> SeasonFall    0.233372   0.126729   1.842  0.07176 .
#> ---
```

```
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Approximate significance of smooth terms:
#>                       edf Ref.df     F p-value
#> s(log(Turb))   2.621e-10      4 0.000   0.812
#> s(log(Chl))    4.420e-01      4 0.215   0.184
#> s(log1p(Fish)) 2.455e+00      4 1.411   0.092 .
#> s(Yearf)       1.314e+00      4 0.484   0.213
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.303   Deviance explained = 41.6%
#> GCV = 0.14301  Scale est. = 0.11784   n = 58
```

Again, the quantitative predictors model provides a significantly better model by AIC.

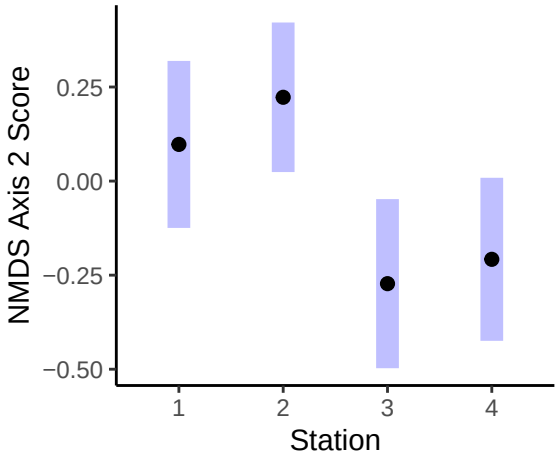
```
AIC(gam_2_1_no_low, ss_gam_2_1_no_low)
#>                           df      AIC
#> gam_2_1_no_low     9.728945 43.35952
#> ss_gam_2_1_no_low 11.210093 51.75576
```

```
oldpar <- par(mfrow = c(2,3))
plot(ss_gam_2_1_no_low)
par(oldpar)
```



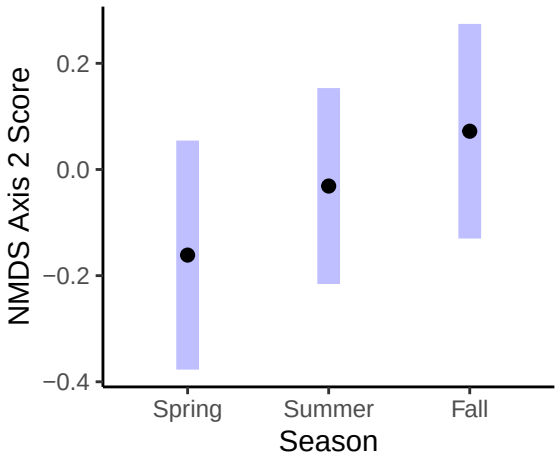
Stations differ.

```
emms <- emmeans(ss_gam_2_1_no_low, ~Station)
pairs(emms)
#>  contrast            estimate    SE   df t.ratio p.value
#>  Station1 - Station2  -0.1253 0.133 47.8  -0.941  0.7831
#>  Station1 - Station3   0.3700 0.132 47.8   2.803  0.0355
#>  Station1 - Station4   0.3053 0.138 47.8   2.205  0.1366
#>  Station2 - Station3   0.4953 0.130 47.8   3.824  0.0021
#>  Station2 - Station4   0.4306 0.129 47.8   3.336  0.0087
#>  Station3 - Station4  -0.0647 0.131 47.8  -0.494  0.9599
#>
#> Results are averaged over the levels of: Season, Yearf
#> P value adjustment: tukey method for comparing a family of 4 estimates

plot(emms) +
  coord_flip() +
  xlab('NMDS Axis 2 Score')
```



Seasons don't differ with respect to axis 2.

```
emms <- emmeans(ss_gam_2_1, ~Season)
pairs(emms)
#>  contrast        estimate    SE   df t.ratio p.value
#>  Spring - Summer   -0.130 0.130 47.8  -1.002  0.5793
#>  Spring - Fall     -0.233 0.127 47.8  -1.842  0.1672
#>  Summer - Fall     -0.103 0.112 47.8  -0.918  0.6316
#>
#> Results are averaged over the levels of: Station, Yearf
#> P value adjustment: tukey method for comparing a family of 3 estimates

plot(emms) +
  coord_flip() +
  xlab('NMDS Axis 2 Score')
```



Possible Publication Graphics – Reduced Data Set

Run `envfit`

```
ef_3 <- envfit(NMDSE3, envNMDS3[,c(1, 3:9, 11:17)], permu = 9999, na.rm = TRUE)
ef_3
#>
#> ***VECTORS
#>
#>             NMDS1    NMDS2     r2 Pr(>r)
#> disch_wk -0.97675 -0.21437 0.5667 0.0001 ***
#> Temp      0.96608  0.25824 0.7385 0.0001 ***
#> Sal       0.62573 -0.78004 0.1218 0.0347 *
#> Turb     -0.19758  0.98029 0.1020 0.0626 .
#> Chl       0.66456 -0.74723 0.0617 0.1916
#> RH       -0.26698  0.96370 0.1342 0.0241 *
#> Fish     -0.46297  0.88637 0.0794 0.1112
#> Turb2    -0.18123  0.98344 0.0992 0.0690 .
#> Chl2      0.78352 -0.62136 0.1544 0.0143 *
#> RH2       0.28722  0.95786 0.1135 0.0414 *
#> Fish2    -0.74267  0.66966 0.0231 0.5514
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#> Permutation: free
#> Number of permutations: 9999
#>
#> ***FACTORS:
#>
#> Centroids:
#>                NMDS1   NMDS2
#> Station1      0.2896  0.3711
#> Station2     -0.1430  0.2109
#> Station3     -0.0907 -0.2563
#> Station4      0.0228 -0.2429
#> Yearf2013     0.1187 -0.1599
#> Yearf2014    -0.0015  0.0114
```

```
#> Yearf2015     -0.1371  0.1019
#> Yearf2016      0.1756  0.0491
#> Yearf2017     -0.1609 -0.0216
#> SeasonSpring  -0.6910  0.0163
#> SeasonSummer   0.2347 -0.0591
#> SeasonFall     0.3349  0.0484
#> is_sp_upFALSE  0.0201 -0.0367
#> is_sp_upTRUE  -0.5324  0.9736
#>
#> Goodness of fit:
#>              r2 Pr(>r)
#> Station  0.2176 0.0004 ***
#> Yearf    0.0572 0.6336
#> Season   0.4469 0.0001 ***
#> is_sp_up 0.1039 0.0040 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#> Permutation: free
#> Number of permutations: 9999
```

Extract Vector Information

```
arrows <- ef_3$vectors$arrows
rsq    <- ef_3$vectors$r
scaled_arrows <- as_tibble(arrows*sqrt(rsq)) %>%
  mutate(parameter = rownames(arrows))
```

```
scale_factor = .15  # Scale position of text annotations this far off
                    # end or arrow

scaled_arrows <- scaled_arrows %>%
  mutate(ann_xpos = NMDS1 + arrows[,1] * scale_factor,
         ann_ypos = NMDS2 + arrows[,2] * scale_factor)
```

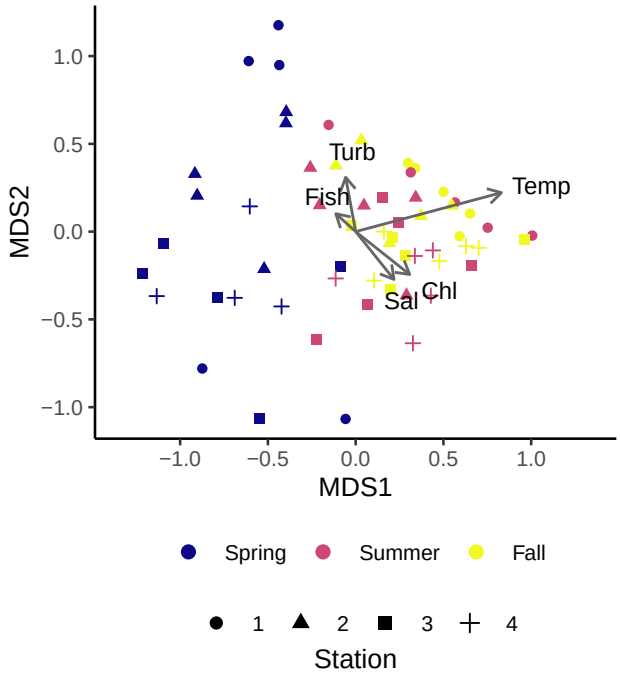
While we are creating vectors, we also want to create points for placing the annotations identifying each vector. We want to space the labels so they are a fixed distance beyond the end of each vector. We do that with a little vector addition.

Environmental Variables Used in Models

```
tmp_arrows <- scaled_arrows %>%
  filter(parameter %in% c('Temp', 'Sal', 'Turb2', 'Chl2', 'Fish2')) %>%
  mutate(parameter = factor(parameter,
                            levels = c('Temp', 'Sal',
                                       'Turb2', 'Chl2', 'Fish2'),
                            labels = c('Temp', 'Sal',
                                       'Turb', 'Chl', 'Fish')))  %>%
   mutate(ann_xpos = case_when(
     parameter == 'Sal' ~ ann_xpos - 0.1,
```

```
    TRUE ~ ann_xpos))

plt <- ggplot(data = envNMDS, aes(MDS1, MDS2)) +
  geom_point(aes(color = Season, shape = Station), size = 1.5) +
  geom_segment(data=tmp_arrows,
               mapping = aes(x=0,xend=NMDS1,y=0,yend=NMDS2),
               arrow = arrow(length = unit(0.2, "cm")) ,colour="grey40") +
  geom_text(data=tmp_arrows,
            mapping = aes(x=ann_xpos,
                          y=ann_ypos,label=parameter),
            size=3, nudge_x =0, nudge_y = 0, hjust = .25)+

  scale_color_viridis_d(option = 'C', name = '') +
  scale_shape(name = 'Station') +

  coord_fixed(xlim =c(-1.35, 1.35)) +

  guides(color = guide_legend(title.position = "bottom",
                              title.hjust = 0.5,
                              override.aes = list(size = 2),
                              order = 1),
         shape = guide_legend(title.position = "bottom",
                              title.hjust = 0.5,
                              override.aes = list(size = 2),
                              order = 2))
```

```
plt +
  theme_classic(base_size = 10) +
  theme(
        legend.position = 'bottom',
        legend.box = 'Vertical',
        legend.spacing.y = unit(0, 'cm'),
        legend.margin = margin(0,0,0,0))
```




```
ggsave('figures/nmds_env_selected_no_low.png', type='cairo',
         width = 3.3, height = 3.4)
ggsave('figures/nmds_env_selected_no_low.pdf', device = cairo_pdf,
       width = 3.3, height = 3.4)
```

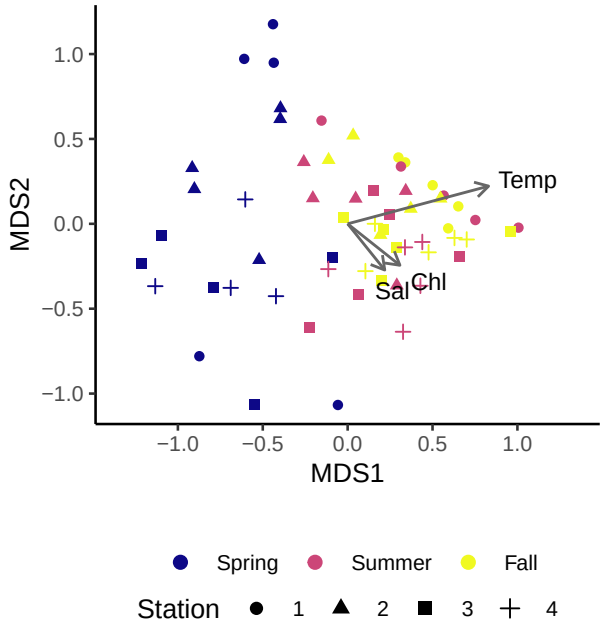
```
ef_3$vectors
#>              NMDS1    NMDS2     r2 Pr(>r)
#> disch_wk -0.97675 -0.21437 0.5667 0.0001 ***
#> Temp      0.96608  0.25824 0.7385 0.0001 ***
#> Sal       0.62573 -0.78004 0.1218 0.0347 *
#> Turb     -0.19758  0.98029 0.1020 0.0626 .
#> Chl       0.66456 -0.74723 0.0617 0.1916
#> RH       -0.26698  0.96370 0.1342 0.0241 *
#> Fish     -0.46297  0.88637 0.0794 0.1112
#> Turb2    -0.18123  0.98344 0.0992 0.0690 .
#> Chl2      0.78352 -0.62136 0.1544 0.0143 *
#> RH2       0.28722  0.95786 0.1135 0.0414 *
#> Fish2    -0.74267  0.66966 0.0231 0.5514
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#> Permutation: free
#> Number of permutations: 9999
```

Significant Environmental Variables Used on Models

```
tmp_arrows <- scaled_arrows %>%
  filter(parameter %in% c('Temp', 'Sal', 'Chl2')) %>%
  mutate(parameter = factor(parameter,
                            levels = c('Temp', 'Sal', 'Chl2'),
                            labels = c('Temp', 'Sal', 'Chl')))  %>%
   mutate(ann_xpos = case_when(
     parameter == 'Sal' ~ ann_xpos - 0.1,
     TRUE ~ ann_xpos))

plt <- ggplot(data = envNMDS, aes(MDS1, MDS2)) +
  geom_point(aes(color = Season, shape = Station), size = 1.5) +
  geom_segment(data=tmp_arrows,
               mapping = aes(x=0,xend=NMDS1,y=0,yend=NMDS2),
               arrow = arrow(length = unit(0.2, "cm")) ,colour="grey40") +
  geom_text(data=tmp_arrows,
            mapping = aes(x= ann_xpos,
                          y= ann_ypos,
                          label=parameter),
            size=3, nudge_x =0, nudge_y = 0, hjust = 0.25)+

  scale_color_viridis_d(option = 'C', name = '') +
  scale_shape(name = 'Station') +

  coord_fixed(xlim =c(-1.35, 1.35)) +

  guides(color = guide_legend(title.position = "top",
                              title.hjust = 0.5,
                              override.aes = list(size = 2),
                              order = 1),
         shape = guide_legend(title.position = "left",
                              title.hjust = 0.5,
                              override.aes = list(size = 2),
                              order = 2))
```

We can't change the figure width, but we can alter the figure height. With that in mind, let's reorient the legends and juggle dimensions

```
plt +
  theme_classic(base_size = 10) +
  theme(
        legend.position = 'bottom',
        legend.box = 'Vertical',
        legend.spacing.y = unit(0, 'cm'),
        legend.box.just = 'top',
        legend.margin = margin(0,0,0,0),
        legend.box.margin = margin(0,0,0,0)
        )
```




```
ggsave('figures/nmds_env_significant_no_low.png', type='cairo',
         width = 3.3, height = 3.4)
ggsave('figures/nmds_env_significant_no_low.pdf', device = cairo_pdf,
       width = 3.3, height = 3.4)
```