NTL_LTER_TR Case Study

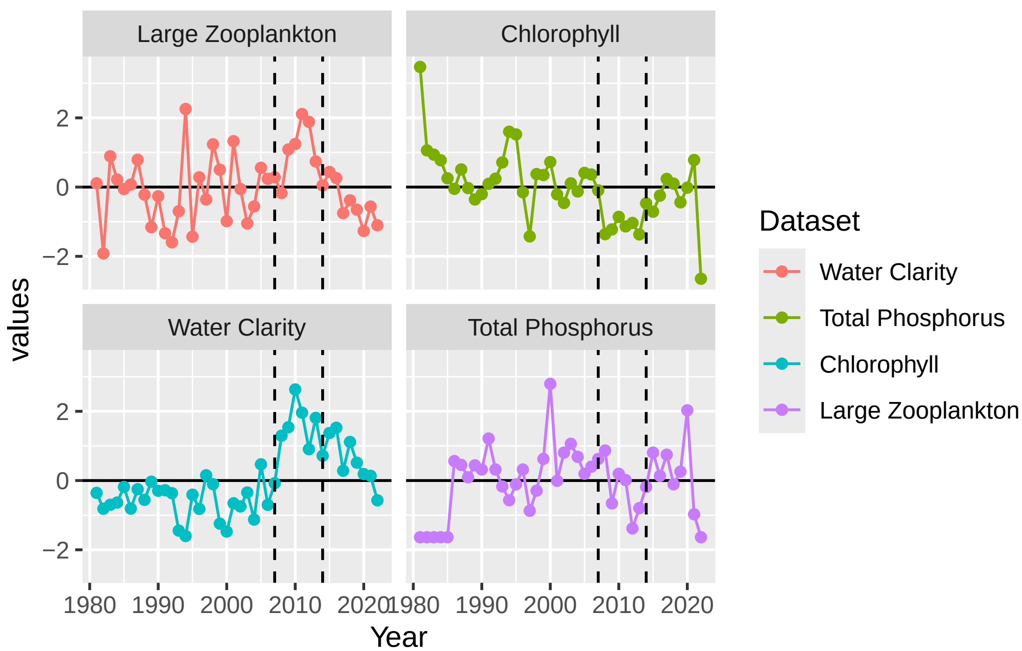
Approach for running LTER-NTL Trout Lake data using temporally structured GAMs and linear regression models with time periods defined a-priori.

First we start with looking at three time periods using linear regression models: 1. The historical regime with low water clarity 2. A clear water regime where the introduction of Lake Trout into the system from stocking in 2006. 3. A novel regime following the introduction of invasive, predatory water flea (*Bythotrephes*) in 2014 which lead to a reversion of water clarity to a less clear state.

To examine this we start by looking at a few key food web conditions using intercept only models through time: 1. Water clarity 2. Phosphorus - which impacts water clarity and is a common bottom-up process that could impact water clarity and we examine as an alternative hypothesis to the top down processes of Lake Trout and invasive speceis. 3. Abundance of large zooplankton *Daphnia* and *Calanoids* 4. Chlorophyll

Variable	No.Period.AIC	Period.AIC	Best.Model
Water Clarity	122.18	83.43	Period
Total Phosphorus	122.18	125.75	No difference
Chlorophyll	122.18	111.00	Period
Large Zooplankton	122.18	115.84	Period

Covariates	Period	Interaction
Chlorophyll	43.99	39.56
Total Phosphorus	44.45	46.31
Large Zooplankton	43.58	35.99



We approach this by fitting a linear model with the a priori time periods as a factor and compare AIC to a single intercept model for wach variable.

We find that water clarity, chlorophyll, and large zooplankton abundance are all better explained by a model that includes apriori defined time periods improves model fit.

We have identified time-varying mean abundance or amount. We see that the relationship between large zooplankton and water clarity, and large zooplankton and chlorophyll, appears to be time-varying. Next we consider whether there are changing relationships between water clarity and these ecosystem dynamics by including a slope parameter and its interaction with time period.

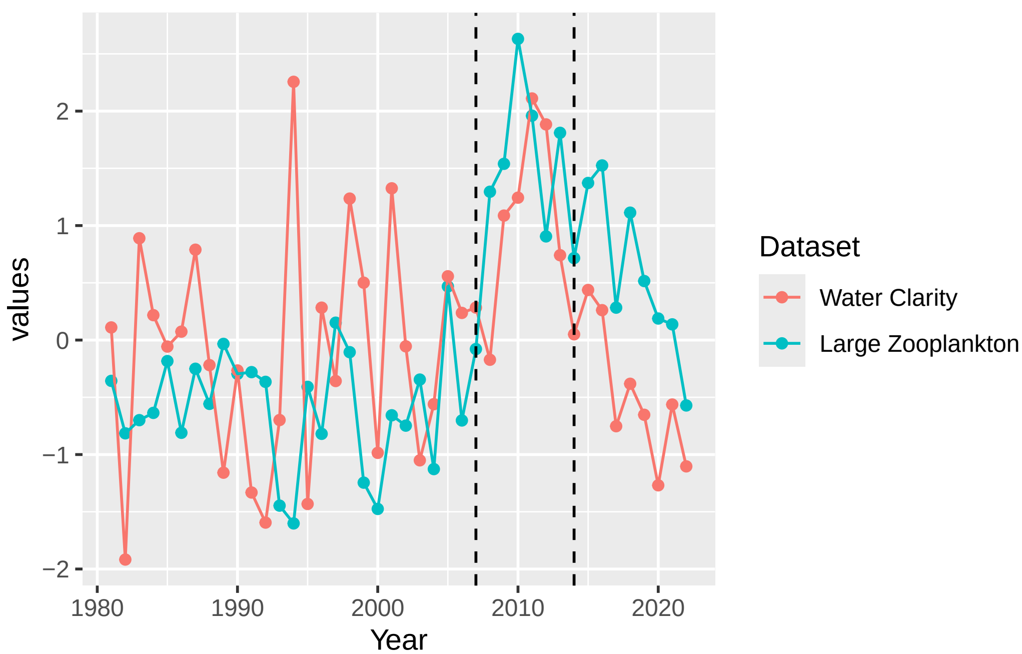
term	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	4.8349816	0.2528329	19.1232307	0.0000000
mean_chl	-0.0782362	0.0788946	-0.9916537	0.3279853
as.factor(period)2	2.2005743	0.5785710	3.8034645	0.0005327
as.factor(period)3	0.1117831	0.4101245	0.2725590	0.7867496
mean_chl:as.factor(period)2	-0.6484809	0.3038817	-2.1339917	0.0397262
mean_chl:as.factor(period)3	0.2348195	0.1514716	1.5502543	0.1298281

term	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	4.6974936	0.1973695	23.8005078	0.0000000
Large	-0.0440188	0.0789582	-0.5574951	0.5806395
as.factor(period)2	0.0552649	0.6211177	0.0889765	0.9295938
as.factor(period)3	-0.8929801	0.5510514	-1.6205024	0.1138523
Large:as.factor(period)2	0.3623181	0.1945706	1.8621421	0.0707613
Large:as.factor(period)3	0.7863286	0.2594449	3.0308115	0.0044983

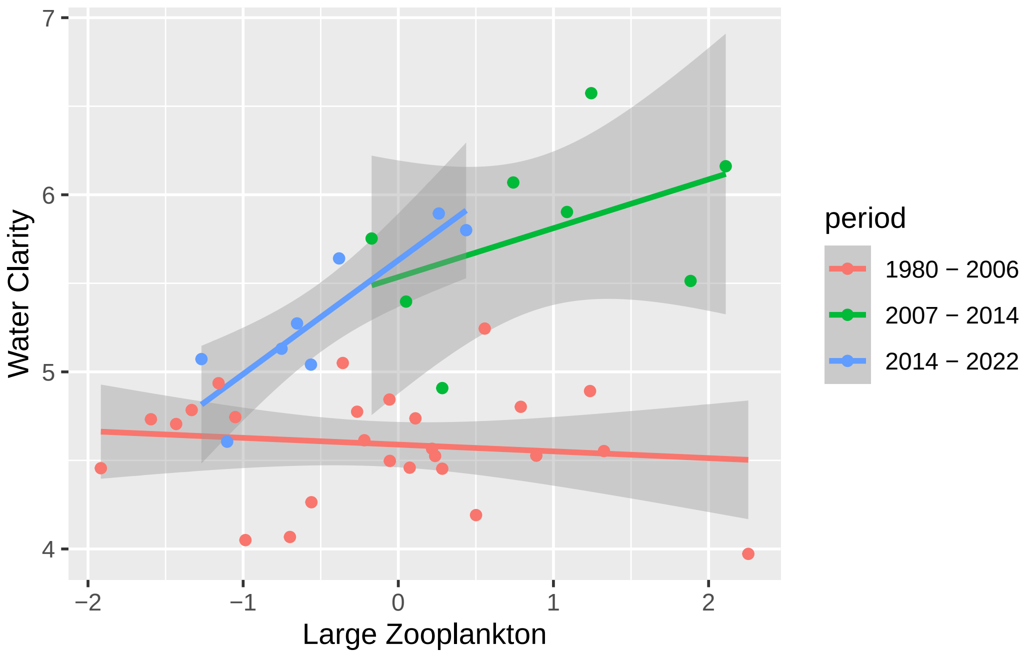
Both Chlorophyll and Large Zooplankton best explain water clarity with a time varying relationship. We examine how these relationships change through time.

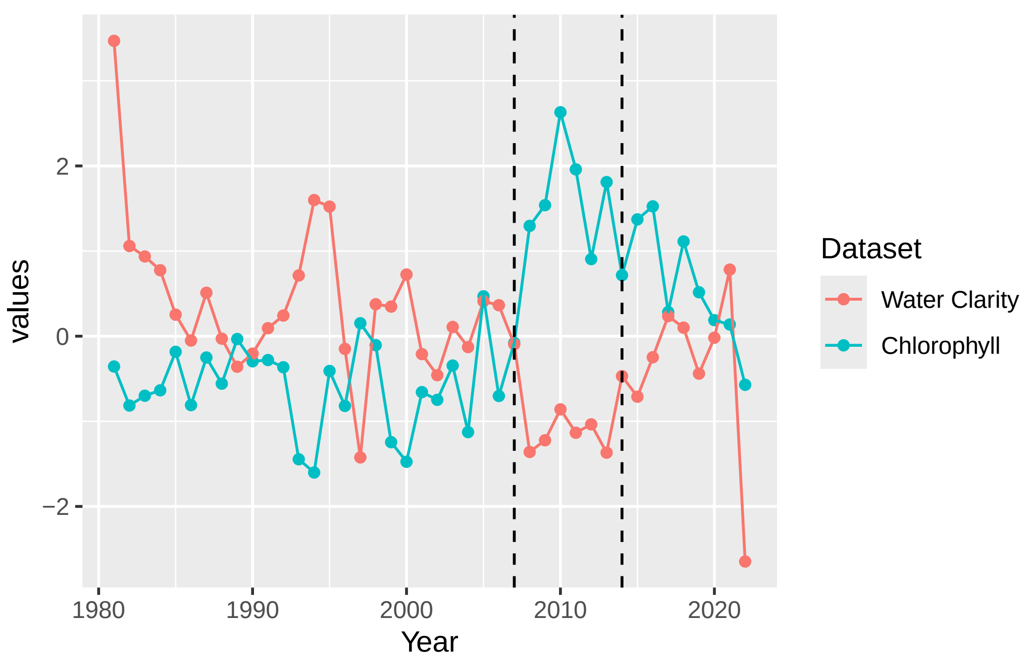
```
[1] "Chlorophyll"
```

```
[1] "Zooplankton"
```

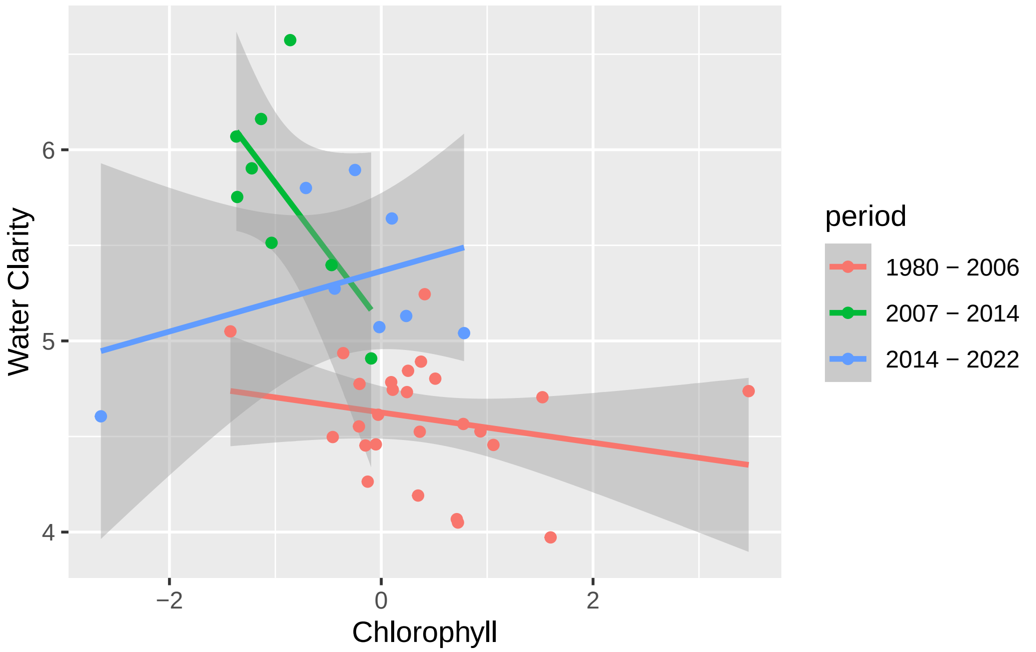


```
`geom_smooth()` using formula = 'y ~ x'
```



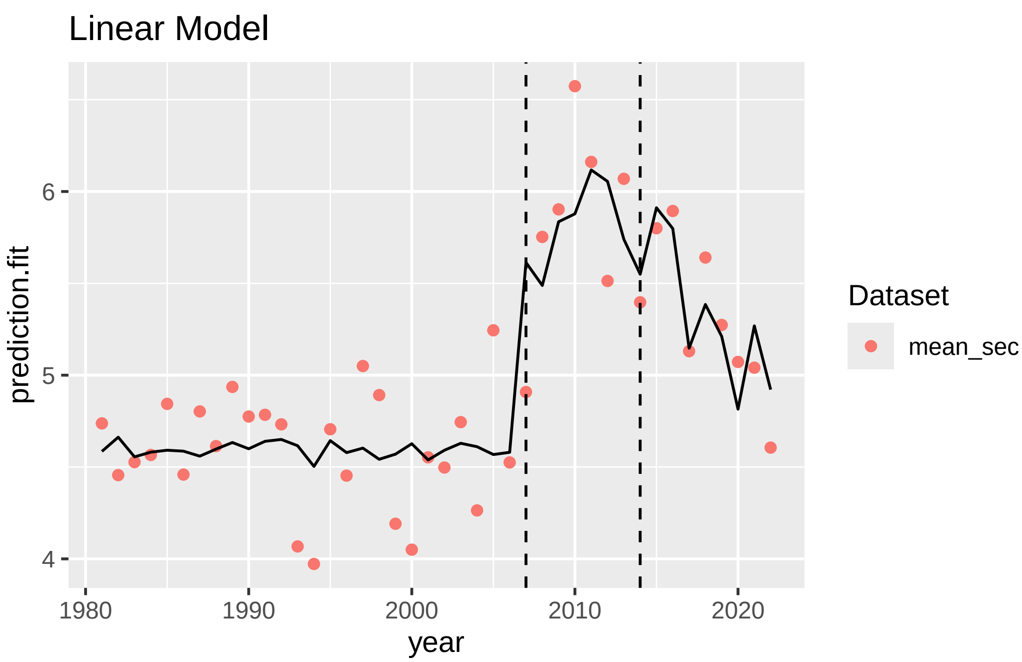



```
`geom_smooth()` using formula = 'y ~ x'
```



If we take the Zooplankton-water quality relationship and fit a temporally structured gam and compare it to the linear model, we find similar results. Both predictions also acccurately identify 2006 as a breakpoint and a second break point in 2016.

```
#| echo: false

prediction=data.frame(prediction=predict(zoop_slope_int,interval='confidence'),year=unique(ar
ggplot(data=prediction,aes(y=prediction.fit, x=year)) +
  #facet_wrap(~area1,scales="free_y") +
 # geom_point()+
  geom_point(data=analysis_long%>%
         filter(Dataset=="mean_sec"), aes(y=values, x=year4, group=Dataset, col=Dataset))+
  geom_line()+
  ggtitle('Linear Model')+
  geom_vline(xintercept=2007, lty=2)+
  geom_vline(xintercept=2014, lty=2)+
  guides(shape = FALSE)
```




```
y.ts <- ts(data=prediction, frequency=1)
# fit breakpoint model
bp.y <- breakpoints(y.ts ~ 1)
summary(bp.y)
```

```
	 Optimal (m+1)-segment partition:

Call:
breakpoints.formula(formula = y.ts ~ 1)

Breakpoints at observation number:

m = 1            26
m = 2            26 36
m = 3      13    26 36
m = 4    7 13    26 36
m = 5    7 13 19 26 36

Corresponding to breakdates:

m = 1            26
```

```
m = 2            26 36
m = 3      13    26 36
m = 4    7 13    26 36
m = 5    7 13 19 26 36

Fit:

m   0         1         2         3         4         5
RSS  11.3356   2.3579   0.6534   0.6505   0.6468   0.6457
BIC  71.6581  13.1864 -33.2407 -25.9477 -18.7148 -11.3127
```

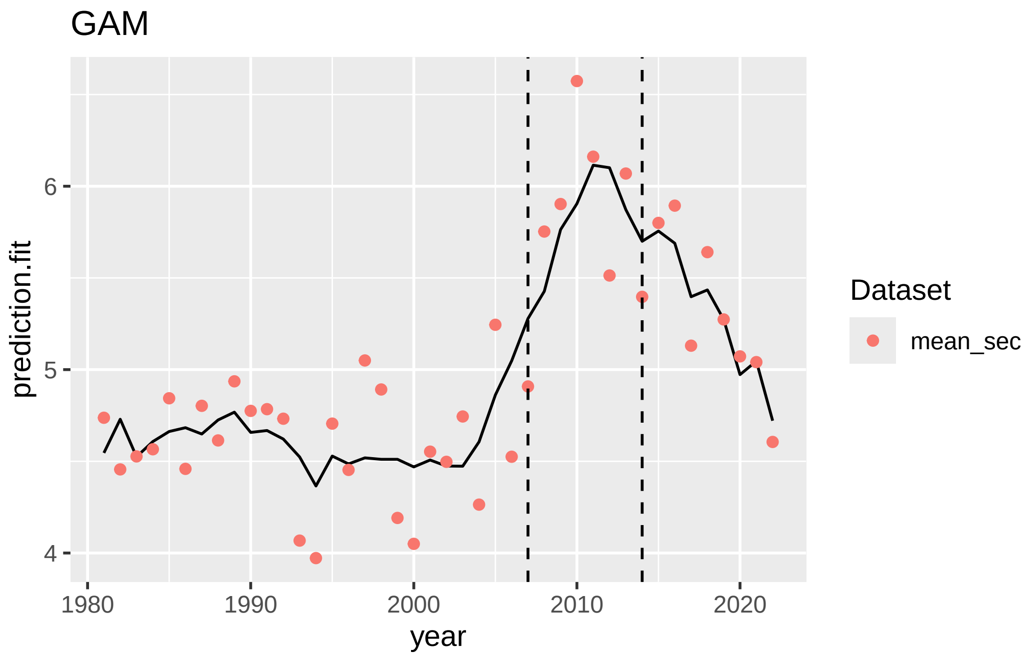
```
prediction[26,]
```

```
   prediction.fit prediction.lwr prediction.upr year
26       4.580172       4.436148       4.724196 2006
```

```
prediction[36,]
```

```
   prediction.fit prediction.lwr prediction.upr year
36       5.798336       5.386912       6.209761 2016
```

```
gam1<-gam(mean_sec~s(year4, by=Large,k=10)+s(year4,k=10), data=full_data)
#plot(gam1)
#gam1<-gam(mean_sec~s(year4, by=Large,k=10), data=full_data)

gam2<-gam(mean_sec~s(year4,k=10), data=full_data)
gam3<-gam(mean_sec~s(Large,k=10), data=full_data)


prediction2=data.frame(prediction=predict(gam1, se.fit=TRUE),year=unique(analysis_long$year4)
ggplot(data=prediction2,aes(y=prediction.fit, x=year)) +
  #facet_wrap(~area1,scales="free_y") +
  #geom_point()+
  geom_line()+
    geom_point(data=analysis_long%>%
         filter(Dataset=="mean_sec"), aes(y=values, x=year4, group=Dataset, col=Dataset))+
  ggtitle('GAM')+
  geom_vline(xintercept=2007, lty=2)+
  geom_vline(xintercept=2014, lty=2)+
  guides(shape = FALSE)
```




```
y.ts <- ts(data=prediction2, frequency=1)
# fit breakpoint model
bp.y <- breakpoints(y.ts ~ 1)
summary(bp.y)
```

```
     Optimal (m+1)-segment partition:

Call:
breakpoints.formula(formula = y.ts ~ 1)

Breakpoints at observation number:

m = 1            26
m = 2            26 36
m = 3      12    26 36
m = 4      12 22 28 36
m = 5    6 12 22 28 36

Corresponding to breakdates:

m = 1            26
```

```
m = 2           26 36
m = 3     12    26 36
m = 4     12 22 28 36
m = 5   6 12 22 28 36

Fit:

m   0      1      2      3      4      5
RSS 11.413  2.969  1.528  1.476  1.358  1.349
BIC 71.943 22.858  2.454  8.451 12.448 19.633
```

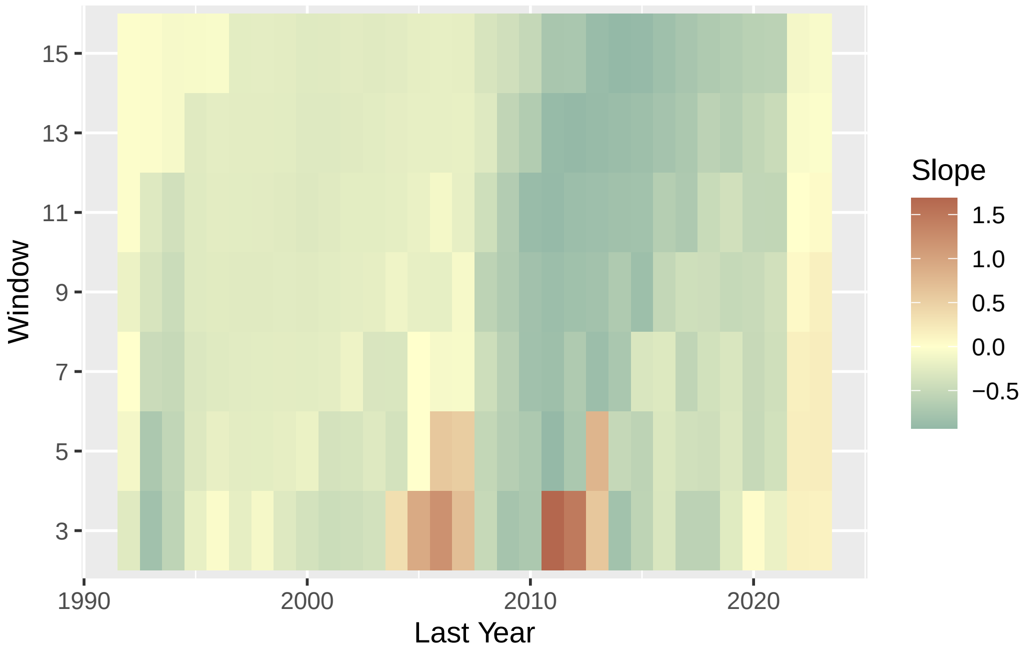
```r
prediction2[26,]
```

```
   prediction.fit prediction.se.fit year
26       5.047866         0.1336191 2006
```

```r
prediction2[36,]
```

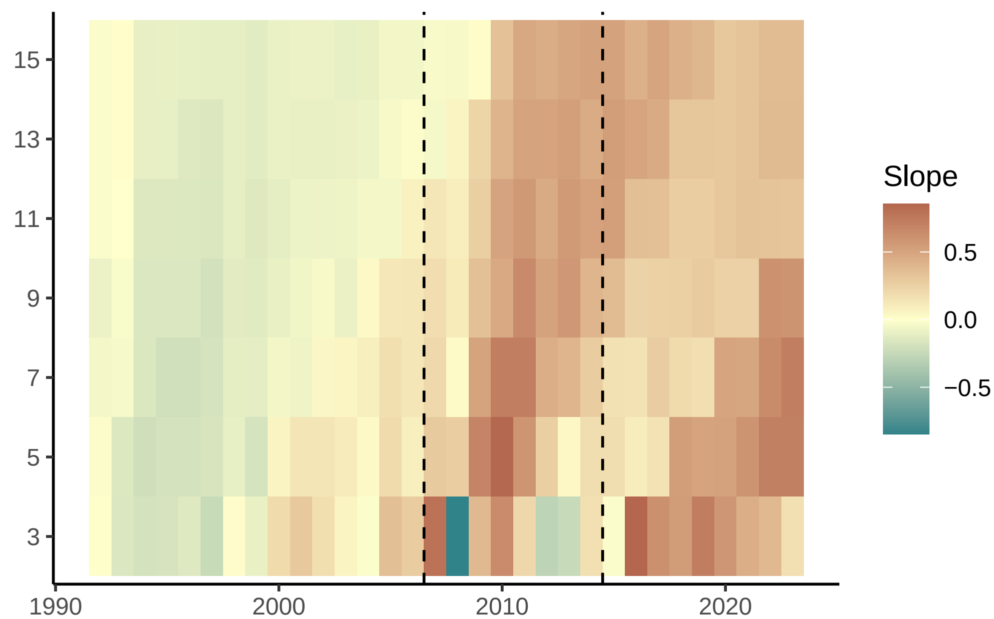
```
   prediction.fit prediction.se.fit year
36       5.688957          0.143298 2016
```

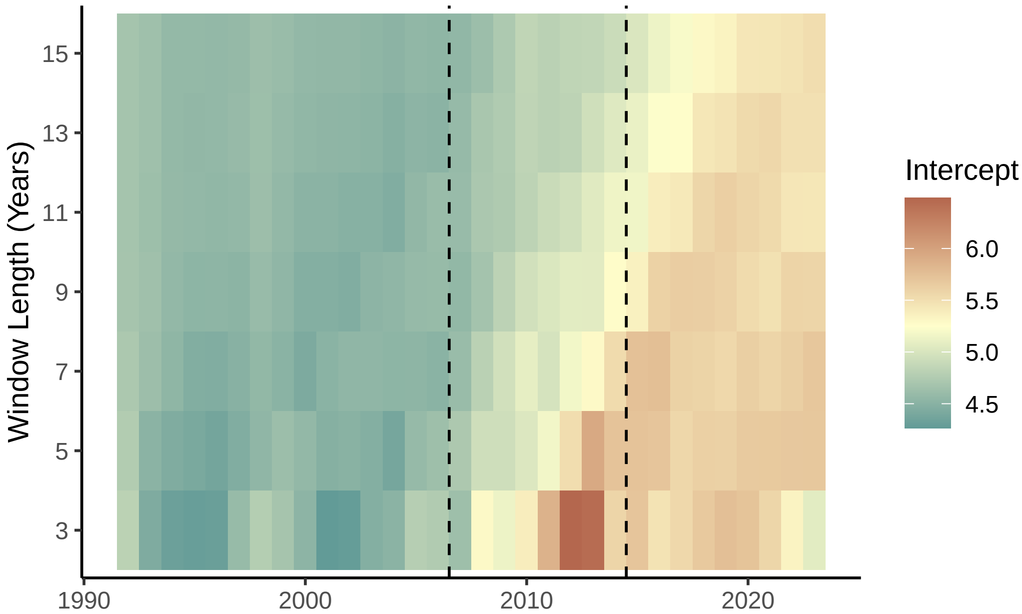
Looking at a 10 year rolling window where the first and last year of the time series changes and see what the slop estimate is as a function of those? Maybe also do a 5-year and a 15-year to also show a function of time series length?



Looking at GAMs for Chlorophyll

Same process with Zooplankton

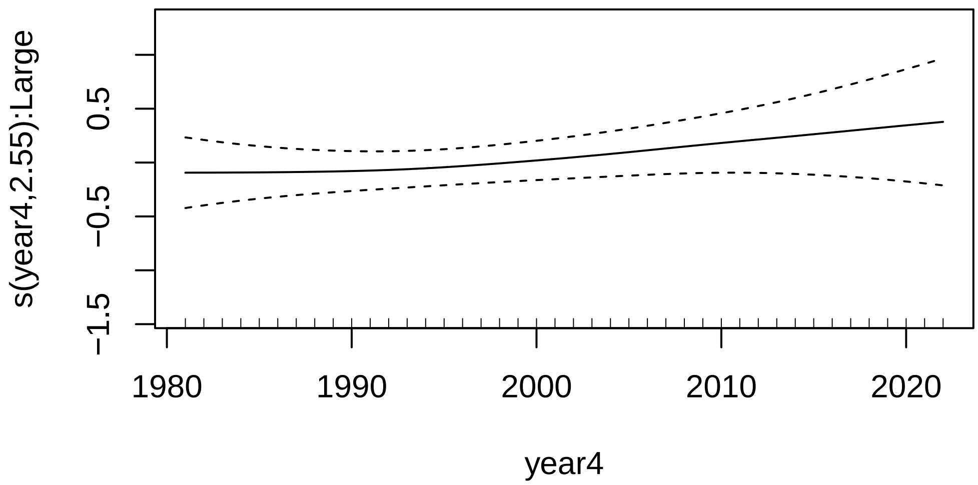


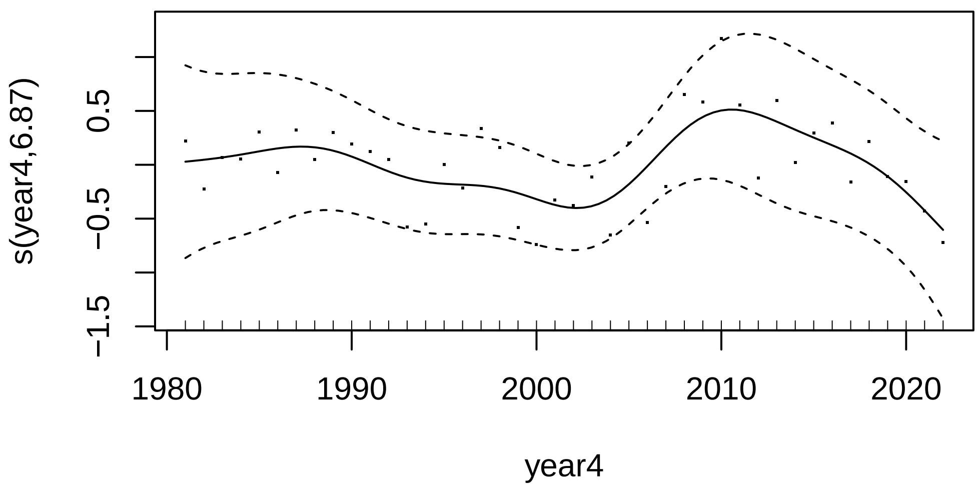


Now that we have looked at both the LM and GAM, lets generate a plot of them together:

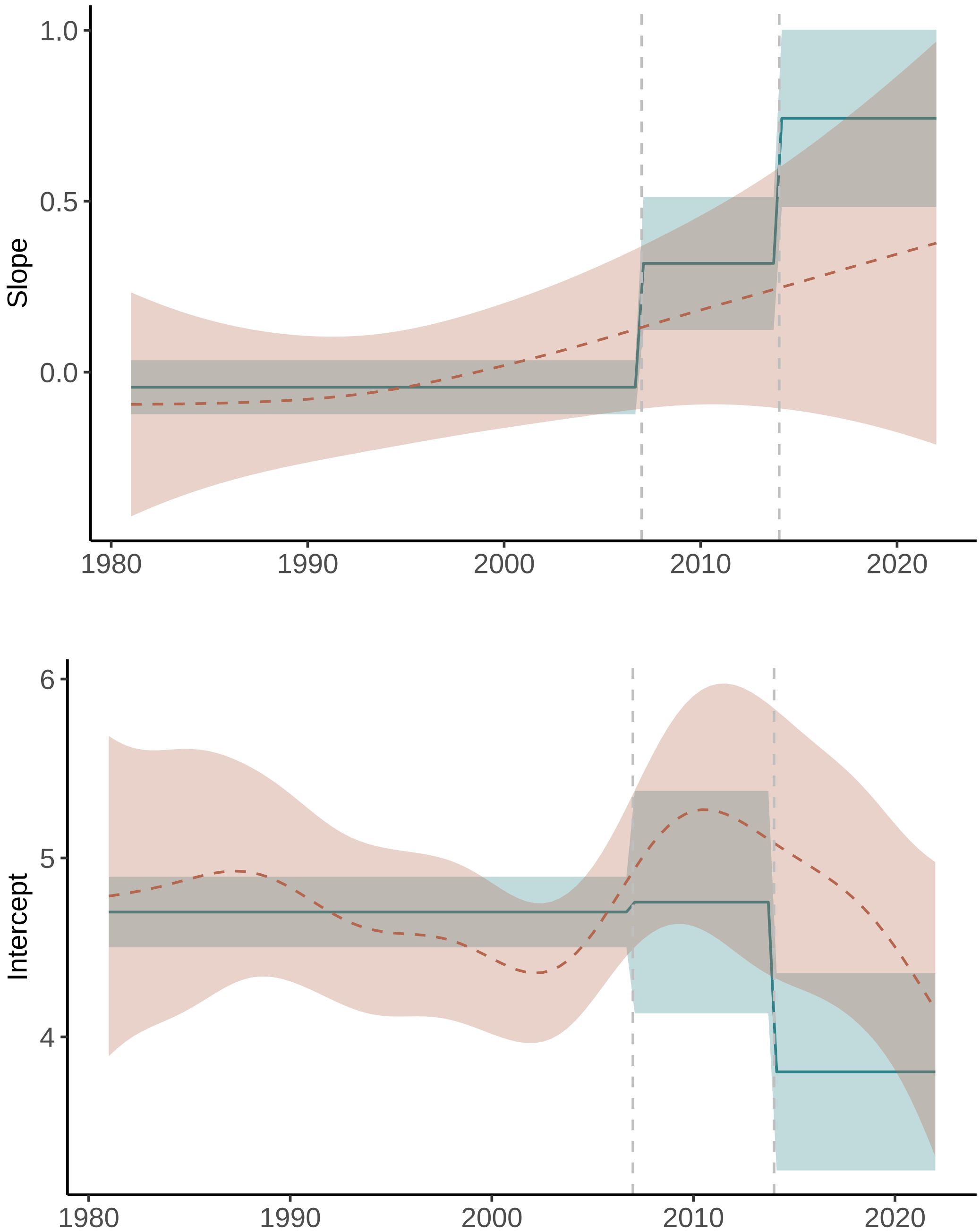
```
Call:
lm(formula = mean_sec ~ Large * as.factor(period), data = full_data)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.70551 -0.14640  0.03001  0.22655  0.69503 

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)               4.69749    0.19737  23.801   <2e-16 ***
Large                    -0.04402    0.07896  -0.557   0.5806    
as.factor(period)2        0.05526    0.62112   0.089   0.9296    
as.factor(period)3       -0.89298    0.55105  -1.621   0.1139    
Large:as.factor(period)2  0.36232    0.19457   1.862   0.0708 .  
Large:as.factor(period)3  0.78633    0.25944   3.031   0.0045 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3396 on 36 degrees of freedom
Multiple R-squared:  0.7319,    Adjusted R-squared:  0.6947
```

```
F-statistic: 19.66 on 5 and 36 DF,  p-value: 2.091e-09


Family: gaussian
Link function: identity

Formula:
mean_sec ~ s(year4, by = Large, k = 10) + s(year4, k = 10)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    4.758      0.228   20.86   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                 edf Ref.df     F p-value
s(year4):Large 2.553  2.879 0.781   0.498
s(year4)       6.869  7.934 1.955   0.086 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.716   Deviance explained = 78.2%
GCV = 0.14249  Scale est. = 0.10713   n = 42
```






```
Joining with `by = join_by(x)`
```


1.0
0.5
0.0
Slope
1980
1990
2000
2010
2020
6
5
4
Intercept
1980
1990
2000
2010
2020