DLMs and GAMs for Lake Washington data

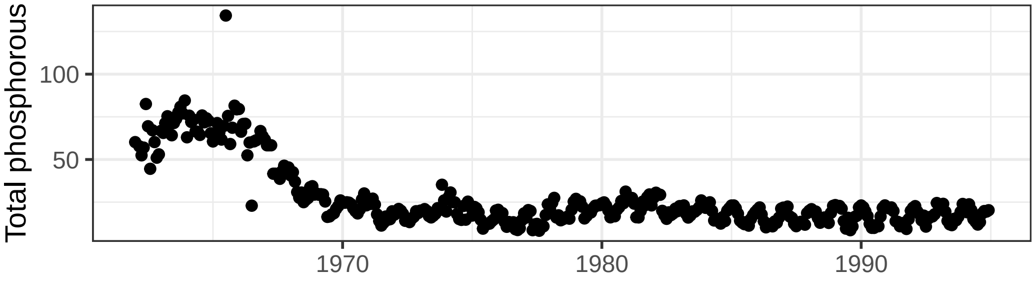
```r
library(MARSS)
library(mgcv)
library(dplyr)
library(forecast)
library(ggplot2)
library(lubridate)
```

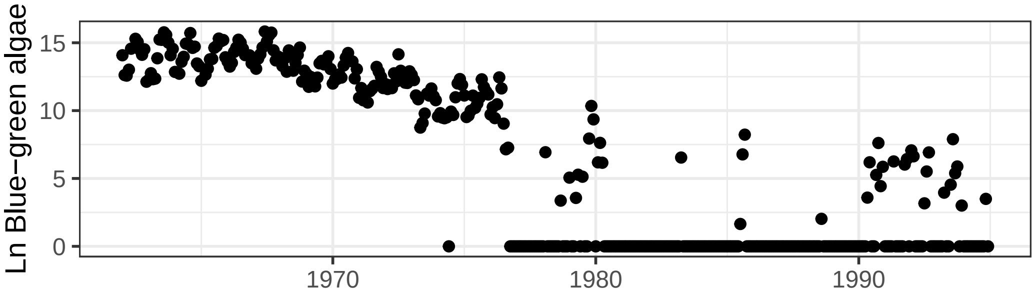
Data

```r
dat <- as.data.frame(MARSS::lakeWAplanktonRaw)
dat$date <- make_date(dat$Year, dat$Month, day = 1)
```

The major change here is related to sewage in the lake – adding wastewater plants in the 1960s reduced phosphorous (TP) and in turn, blue green algae.

```r
p1 <- ggplot(dat, aes(date, TP)) +
  geom_point() + ylab("Total phosphorous") + xlab("") + theme_bw()
p2 <- ggplot(dat, aes(date, log(Bluegreens+1))) +
  geom_point() + ylab("Ln Blue-green algae") + xlab("") + theme_bw()
gridExtra::grid.arrange(p1,p2,nrow=2)
```





We can use the MARSS package to fit a DLM. We'll filter the data to only use counts before 1974, and log transform densities. We also z-score the covariate. This is a good example of imperfect data – there's a couple of missing observations in the covariate, which we can't have in a DLM or GAM – so let's just fill them in with a spline.

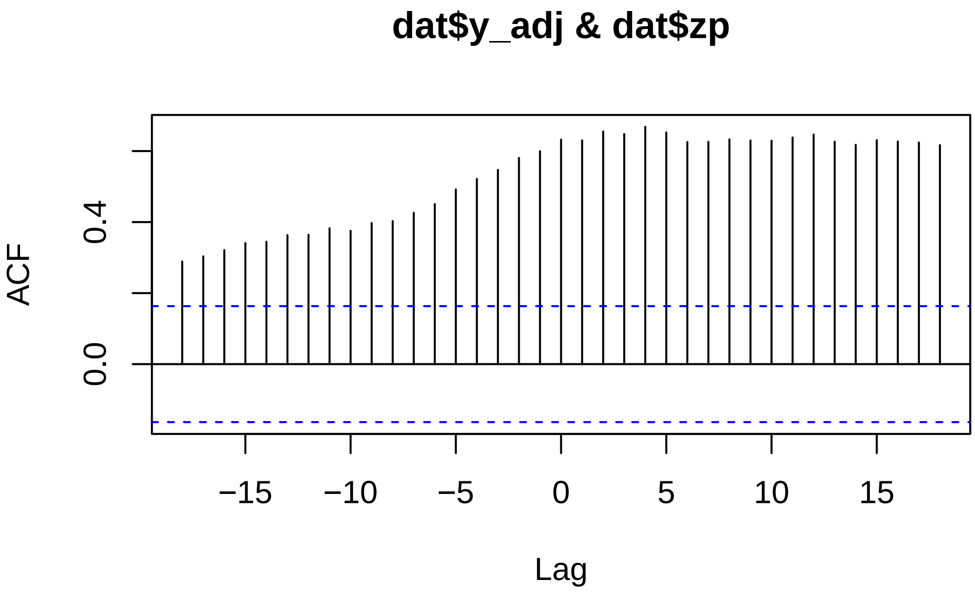
```r
dat <- dplyr::filter(dat, Year < 1974) %>%
  dplyr::mutate(y = log(Bluegreens), zp = as.numeric(scale(TP)))

interpolated <- spline(dat$zp, xout = 1:nrow(dat))$y
dat$zp[which(is.na(dat$zp))] <- interpolated[which(is.na(dat$zp))]
```

The response data exhibit a strong seasonal cycle – and it's a good idea to de-season that to focus on the phosphorous effect.

```r
# Fill in a few missing values
dat$y_interp <- dat$y
interpolated <- spline(dat$y, xout = 1:nrow(dat))$y
missing <- which(is.na(dat$y_interp))
dat$y_interp[missing] <- interpolated[missing]

dat <- dplyr::group_by(dat, Month) %>%
  dplyr::mutate(month_mean = mean(y,na.rm=T)) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(y_adj = y_interp - month_mean)
```

Last, we should look at the lags between driver and response relationships. The CCF here indicates a 2-4 month lag, where TP and algae are more strongly associated with one another.

```
# We can also try to do a CCF analysis between phosphorous and algae

ccf(dat$y_adj, dat$zp)
```



We'll use a lag of 4 months.

```
lag_n <- 5
dat$lagged_zp <- c(rep(NA, lag_n), dat$zp[1:(nrow(dat)-lag_n)])
dat <- dat[-c(1:lag_n),]
```

DLMs

Define the model – this block is basically copied from the MARSS book (salmon survival case study)

```
m <- 2
TT <- nrow(dat)
B <- diag(m)  ## 2x2; Identity
U <- matrix(0, nrow = m, ncol = 1)  ## 2x1; both elements = 0
Q <- matrix(list(0), m, m)  ## 2x2; all 0 for now
diag(Q)[1] <- 0.0000001
diag(Q)[2] <- c("q.beta")
```

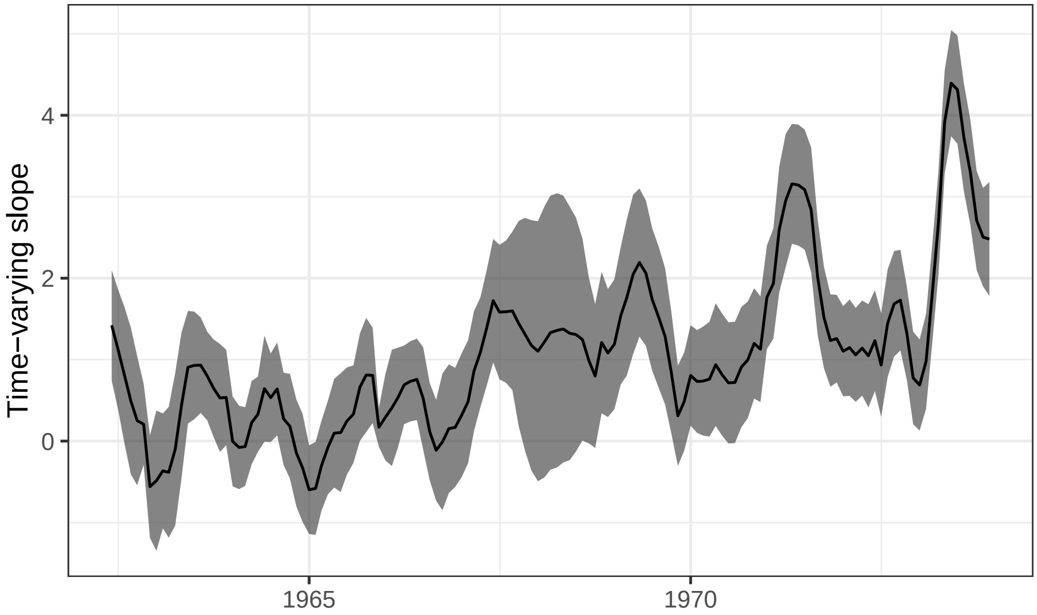
```
#diag(Q) <- c("q.alpha", "q.beta")  ## 2x2; diag = (q1,q2)
Z <- array(NA, c(1, m, TT))  ## NxMxT; empty for now
Z[1, 1, ] <- rep(1, TT)  ## Nx1; 1's for intercept
Z[1, 2, ] <- dat$lagged_zp  ## Nx1; predictor variable
A <- matrix(0)  ## 1x1; scalar = 0
R <- matrix("r")  ## 1x1; scalar = r
## only need starting values for regr parameters
inits_list <- list(x0 = matrix(c(0, 0), nrow = m))

## list of model matrices & vectors
mod_list <- list(B = B, U = U, Q = Q, Z = Z, A = A, R = R)

# convert response to matrix
dat_mat <- matrix(dat$y_adj, nrow = 1)

# fit the model -- crank up the maxit to ensure convergence
dlm_1 <- MARSS(dat_mat, inits = inits_list, model = mod_list,
               control = list(maxit=4000), method="TMB")
```

```
MARSS fit is
Estimation method: TMB
Estimation converged in 18 iterations.
Log-likelihood: -155.082
AIC: 318.1641   AICc: 318.4626

         Estimate
R.r         0.226
Q.q.beta    0.235
x0.X1       0.498
x0.X2       1.422
Initial states (x0) defined at t=0

Standard errors have not been calculated.
Use MARSSparamCIs to compute CIs and bias estimates.
```

Let's put the state estimates back into the original dataframe and do some plotting. The slope is generally positive, which is what we expect – and exhibits clear variation through time

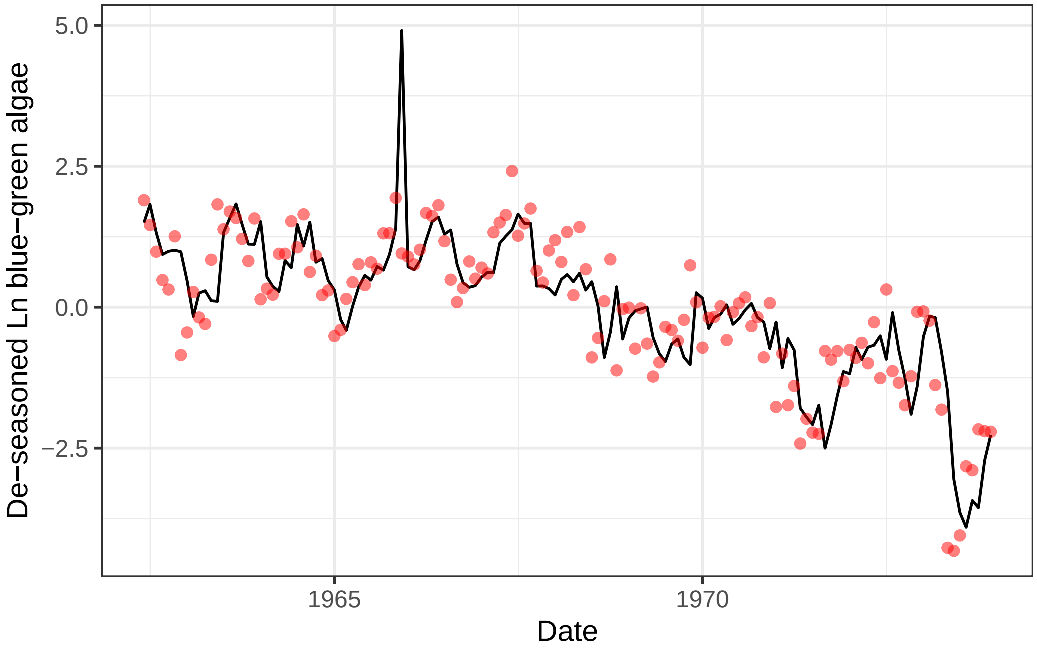
```
dat$int_est <- dlm_1$states[1,]
dat$int_se <- dlm_1$states.se[1,]
```

```
dat$slope_est <- dlm_1$states[2,]
dat$slope_se <- dlm_1$states.se[2,]

ggplot(dat, aes(date, slope_est)) +
  geom_ribbon(aes(ymin=slope_est-2*slope_se, ymax = slope_est+2*slope_se), alpha=0.6) +
  geom_line() + ylab("Time-varying slope") + xlab("") + theme_bw()
```



We can also plot the fitted values from our model, and some diagnostics

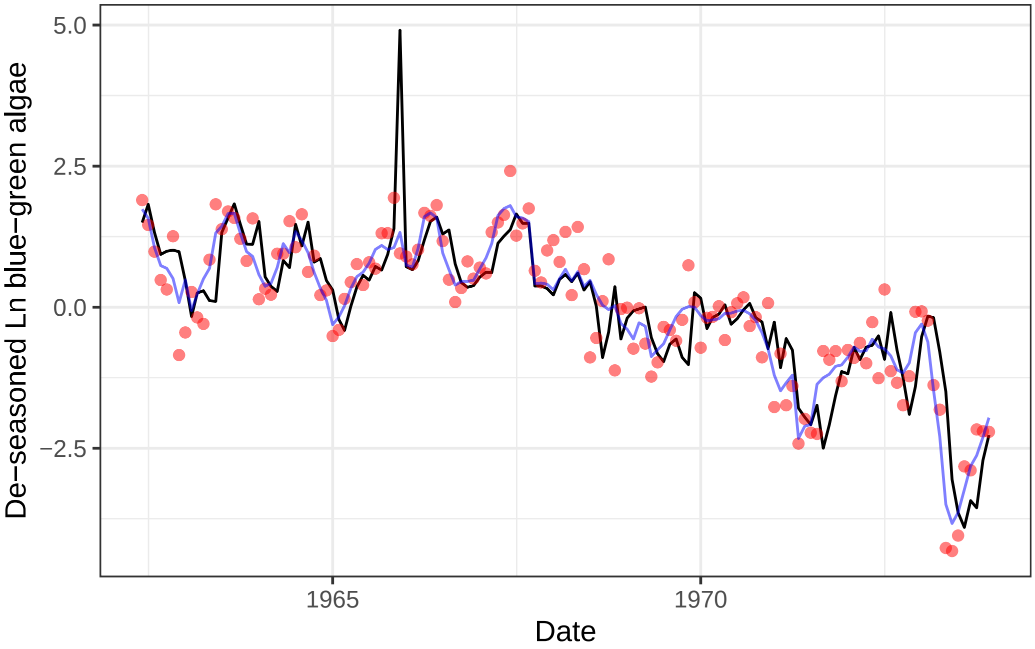
```
pred <- fitted(dlm_1)

dat$pred <- pred$.fitted
ggplot(dat, aes(date, pred)) +
  geom_line() +
  ylab("De-seasoned Ln blue-green algae") +
  xlab("Date") +
  theme_bw() +
  geom_point(aes(date,y_adj), col="red",alpha=0.5)
```



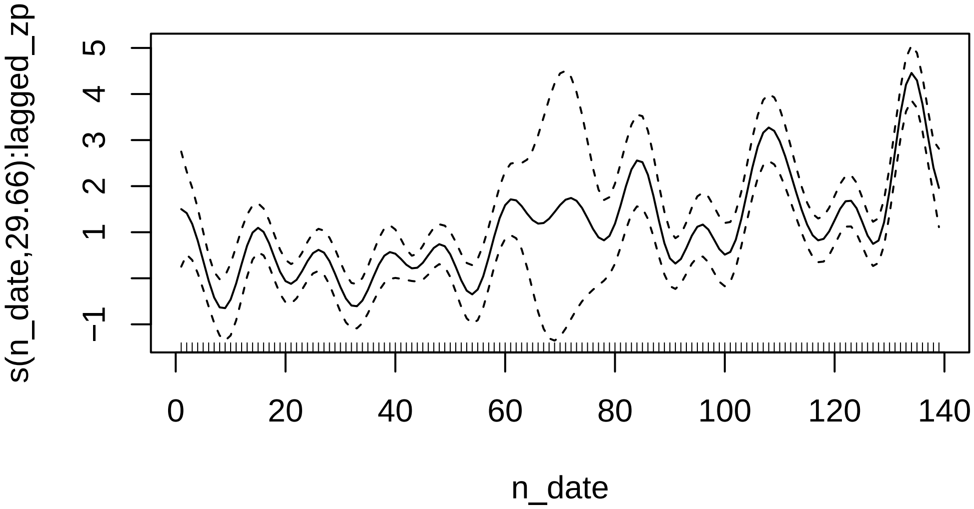
GAMs

Let's fit the same model with a GAM

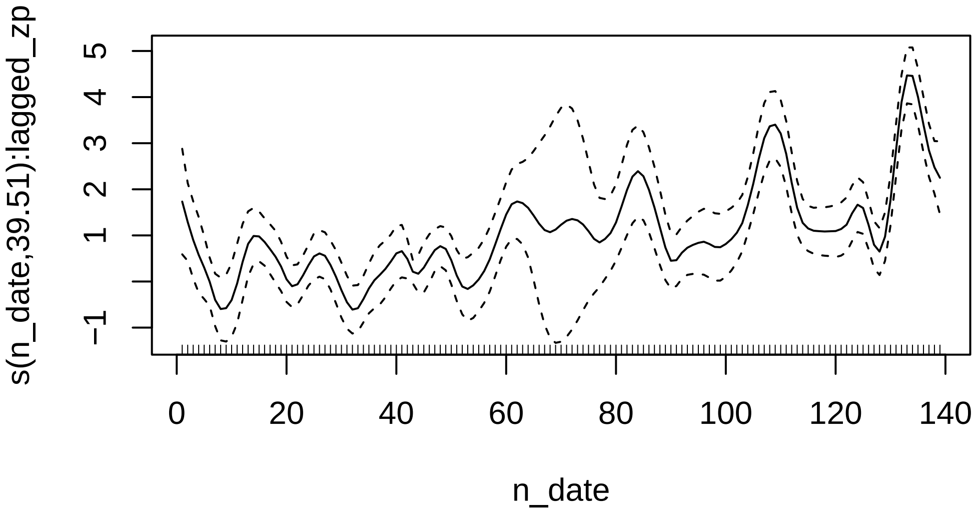
```
# Maybe some discussion in the paper about why s(n_date,lagged_zp) and not
# the other way around?

# The 'sp' parameter also controls the penalty on smoothing (below)
dat$n_date <- as.numeric(as.factor(dat$date))
gam_cr <- gam(y_adj ~ s(n_date, by = lagged_zp, bs = "cr",k=nrow(dat)),
              data = dat)
gam_tp <- gam(y_adj ~ s(n_date, by = lagged_zp,k=nrow(dat)),
              data = dat) # k is ignored
gam_gp <- gam_cr <- gam(y_adj ~ s(n_date, by = lagged_zp, bs = "gp",k=nrow(dat)),
                        data = dat)

dat$pred_gam <- predict(gam_tp)

ggplot(dat, aes(date, pred)) +
  geom_line() + ylab("De-seasoned Ln blue-green algae") +
  xlab("Date") + theme_bw() +
  geom_point(aes(date,y_adj), col="red",alpha=0.5) +
  geom_line(aes(date,pred_gam), col="blue",alpha=0.5)
```



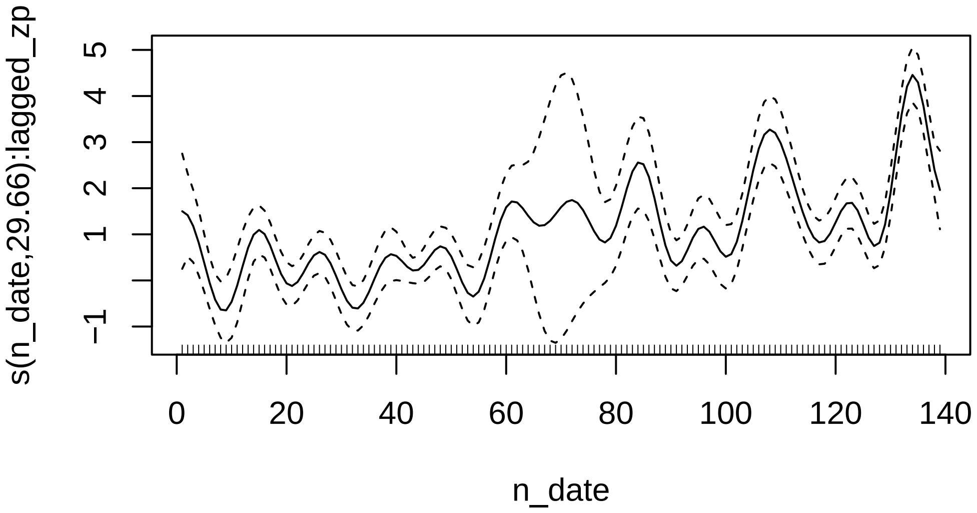
```r
# extract the smooth estimates with SEs to compare to DLM
plot_cr <- plot(gam_cr, seWithMean = TRUE, n = nrow(dat))
```



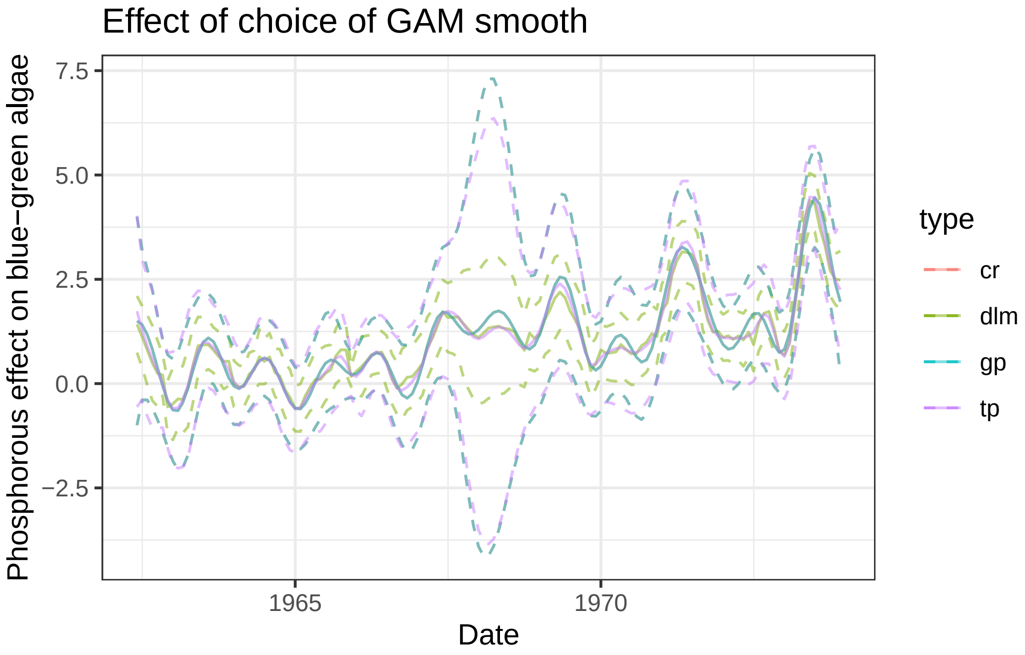
```r
plot_tp <- plot(gam_tp, seWithMean = TRUE, n = nrow(dat))
```



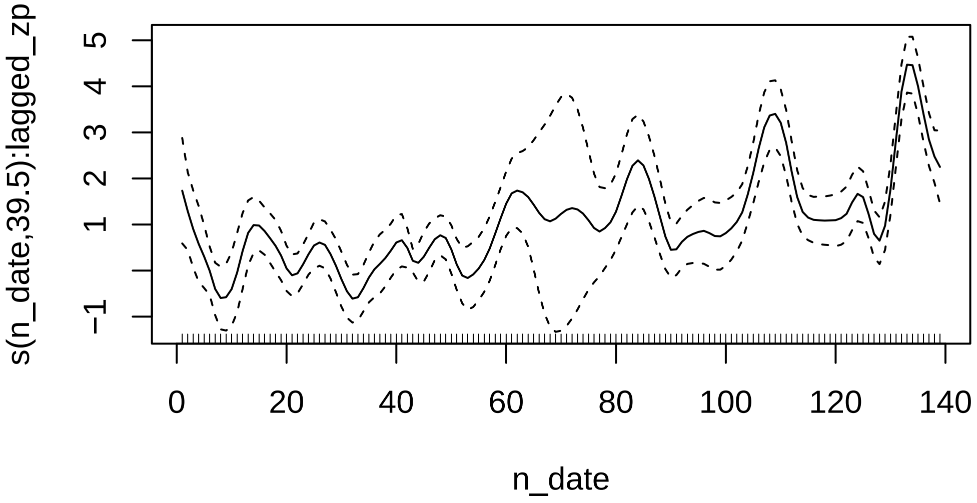
```
plot_gp <- plot(gam_gp, seWithMean = TRUE, n = nrow(dat))
```



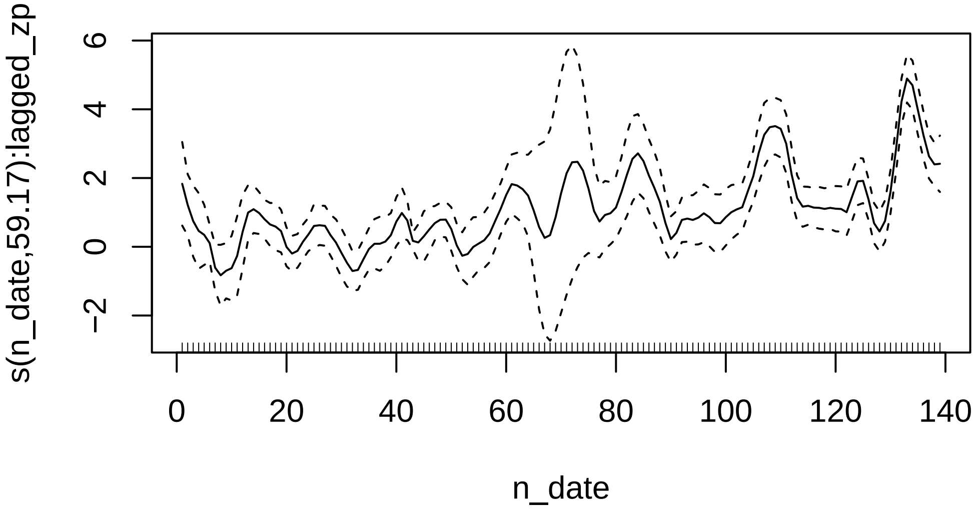
```
df_cr <- data.frame(date = dat$date,
                    est = plot_cr[[1]]$fit,
                    se = plot_cr[[1]]$se, type="cr")
df_tp <- data.frame(date = dat$date,
                    est = plot_tp[[1]]$fit,
                    se = plot_tp[[1]]$se, type="tp")
df_gp <- data.frame(date = dat$date,
                    est = plot_gp[[1]]$fit,
                    se = plot_gp[[1]]$se, type="gp")
df_dlm <- data.frame(date = dat$date,
                      est = dat$slope_est,
                      se = dat$slope_se, type="dlm")
df_smooth <- rbind(df_cr, df_tp, df_gp, df_dlm)

ggplot(df_smooth, aes(date, est, col=type, fill=type)) +
  geom_line(aes(date,est-2*se), alpha=0.5, linetype=2) +
  geom_line(aes(date,est+2*se), alpha=0.5, linetype=2) +
  geom_line(aes(date,est), alpha=0.5) +
  theme_bw() +
  xlab("Date") +
  ylab("Phosphorous effect on blue-green algae") +
  ggtitle("Effect of choice of GAM smooth")
```



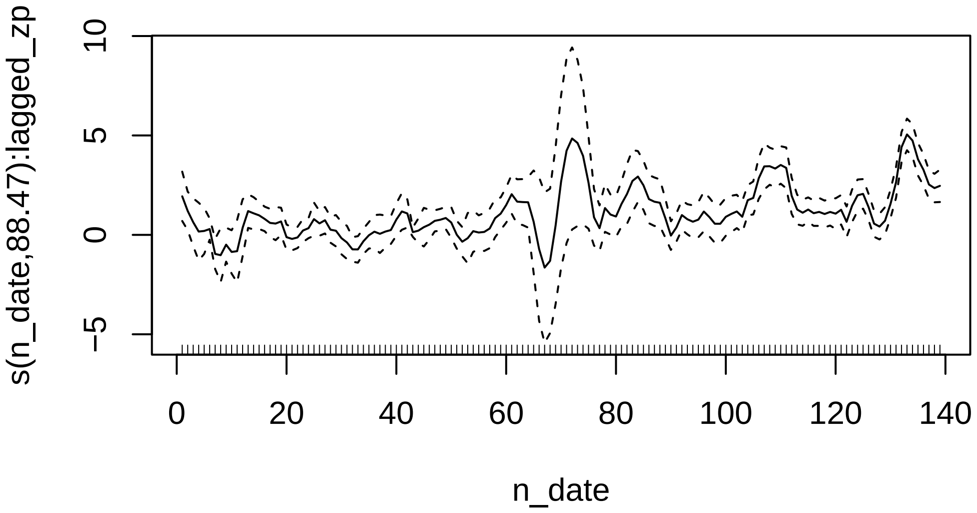
```
gam_cr30 <- gam(y_adj ~ s(n_date, by = lagged_zp, bs = "cr",
                          k=nrow(dat)), data = dat)
gam_cr60 <- gam(y_adj ~ s(n_date, by = lagged_zp, bs = "cr",
                          k=nrow(dat),sp=16), data = dat)
gam_cr90 <- gam(y_adj ~ s(n_date, by = lagged_zp, bs = "cr",
                          k=nrow(dat),sp=2.5), data = dat)
gam_cr120 <- gam(y_adj ~ s(n_date, by = lagged_zp, bs = "cr",
                           k=nrow(dat),sp=0.3), data = dat)

plot_cr30 <- plot(gam_cr30, seWithMean = TRUE, n = nrow(dat))
```



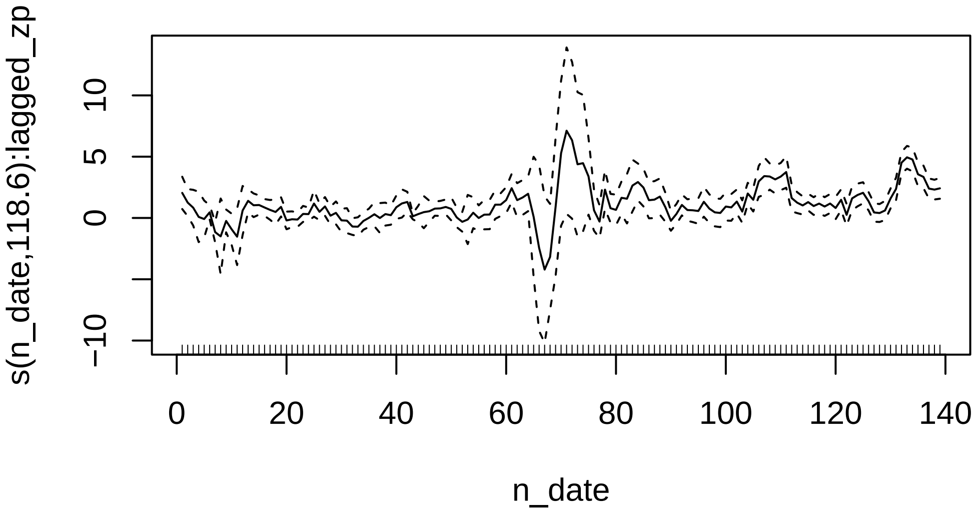
```
plot_cr60 <- plot(gam_cr60, seWithMean = TRUE, n = nrow(dat))
```



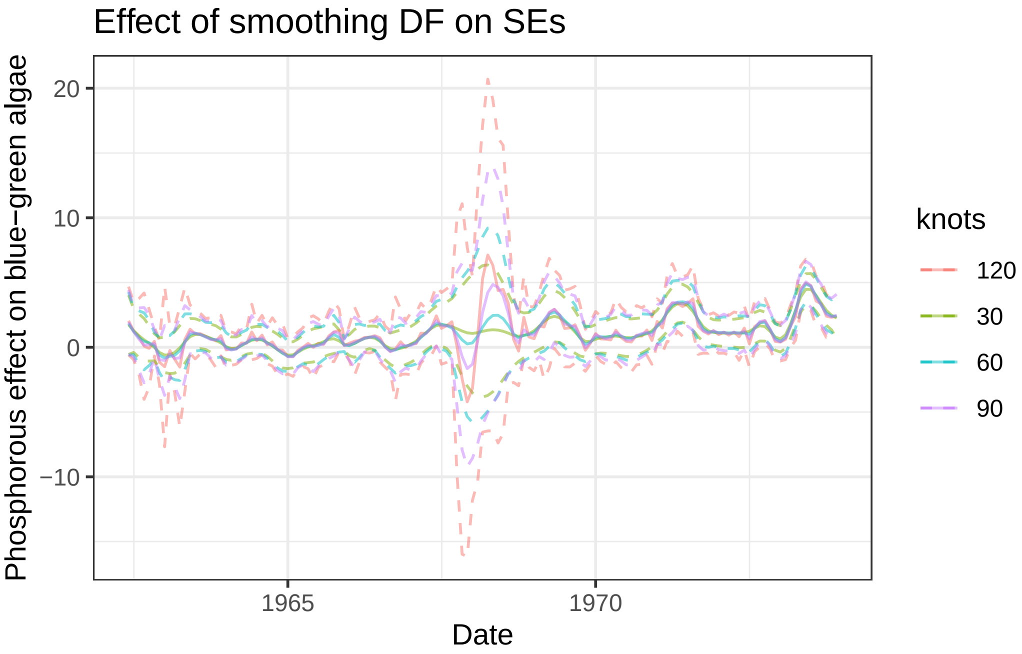
```
plot_cr90 <- plot(gam_cr90, seWithMean = TRUE, n = nrow(dat))
```



```
plot_cr120 <- plot(gam_cr120, seWithMean = TRUE, n = nrow(dat))
```



```
df_30 <- data.frame(date = dat$date,
                    est = plot_cr30[[1]]$fit,
                    se = plot_cr30[[1]]$se, knots="30")
df_60 <- data.frame(date = dat$date,
                    est = plot_cr60[[1]]$fit,
                    se = plot_cr60[[1]]$se, knots="60")
df_90 <- data.frame(date = dat$date,
                    est = plot_cr90[[1]]$fit,
                    se = plot_cr90[[1]]$se, knots="90")
df_120 <- data.frame(date = dat$date,
                     est = plot_cr120[[1]]$fit,
                     se = plot_cr120[[1]]$se, knots="120")
df_smooth <- rbind(df_30, df_60, df_90, df_120)

ggplot(df_smooth, aes(date, est, col=knots, fill=knots)) +
  geom_line(aes(date,est-2*se), alpha=0.5, linetype=2) +
  geom_line(aes(date,est+2*se), alpha=0.5, linetype=2) +
  geom_line(aes(date,est), alpha=0.5) +
  theme_bw() + xlab("Date") + ylab("Phosphorous effect on blue-green algae") +
  ggtitle("Effect of smoothing DF on SEs")
```



Interpretation:

- Sewage into Lake WA was a driver of phosphorous, and thought to cause major increases in blue-green algae into the lake
- This example includes a non-stationary time series of phosphorous into Lake WA, as well as non-stationary driver-response relationship
- Effects of phosphorous on blue-green algae also appear non-stationary – increasing into the 1970s and becoming stronger (more positive). Why? Biologically this may because the Daphnia population really expanded in the late 1970s (Daphnia sometimes eat blue green algae, but it’s a poor food source) – big picture, the lake community changed, and

the effect of phosphorous would have to increase to create the same response ("Grazers are known to inhibit cyanobacteria" paragraph in their Discussion)

- Doesn't account for external effects. Community changed slightly across decades (Francis et al.), and lake has warmed.
- GAMs and DLMs can give very similar answers. Implicit mgcv implementation involves penalty to prevent overfitting – so is less flexible than DLM (AND has wider SEs)
- Choice of smooth type has little effect on inference about time varying parameters
- Adjusting the smoothness DF in mgcv impacts the SEs, and can get the DF much closer to a DLM. If inference is focused on just the mean, this is unnecessary, these give similar answers. But increasing knots increases the SEs