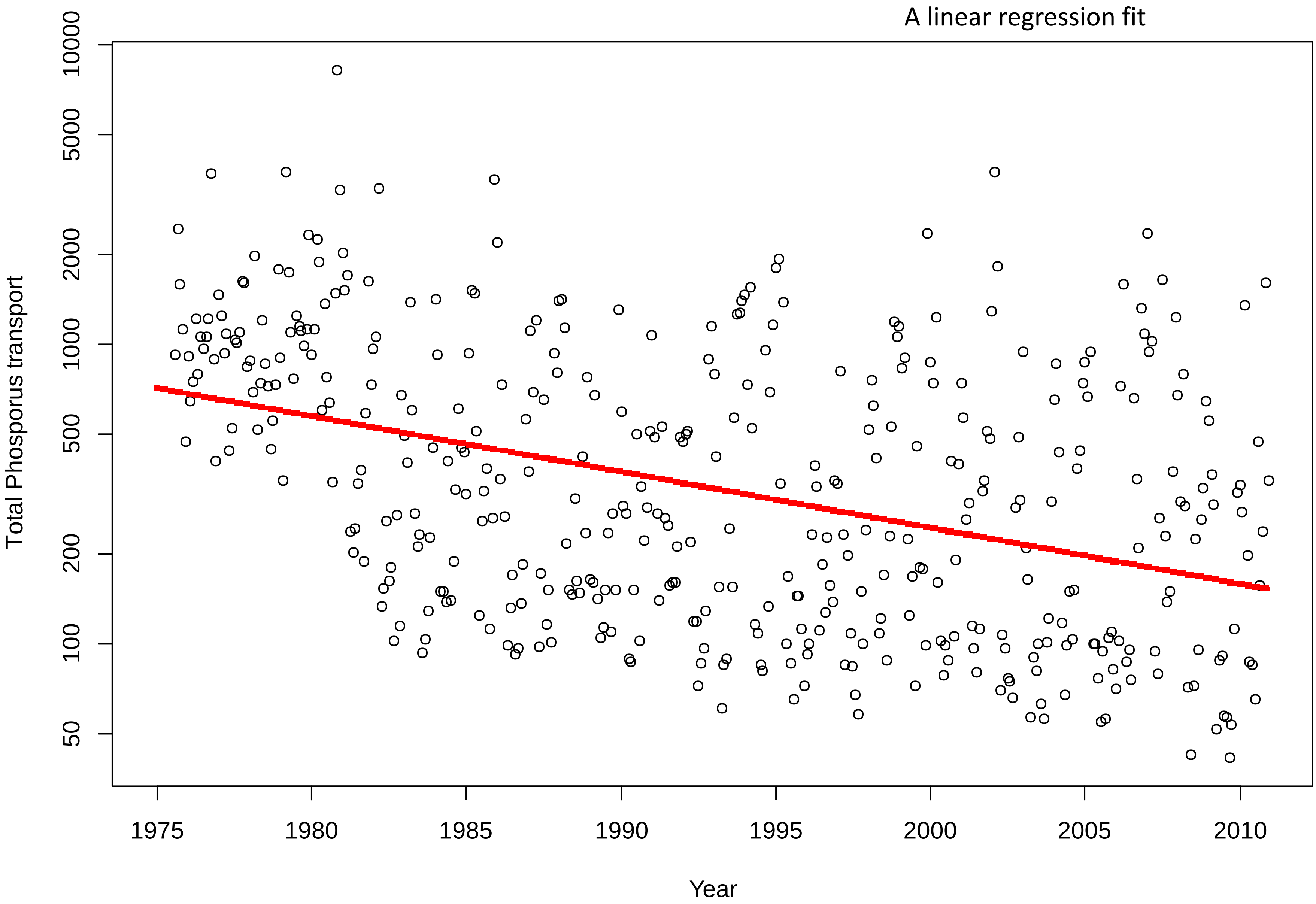
An introduction to General Additive Models

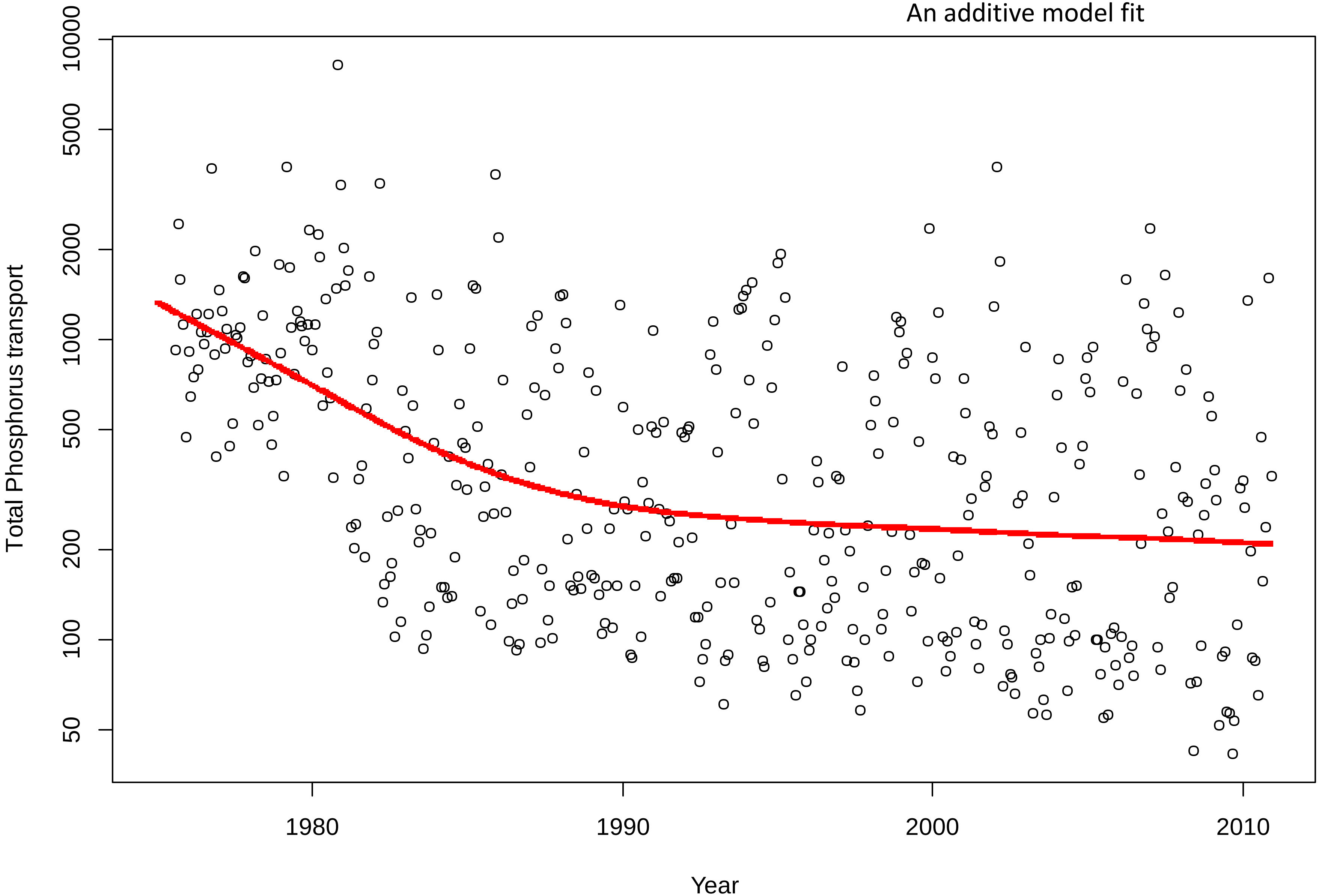
Claudia von Brömssen

Dept. of Energy and Technology

Why use General Additive Models (GAMs)?



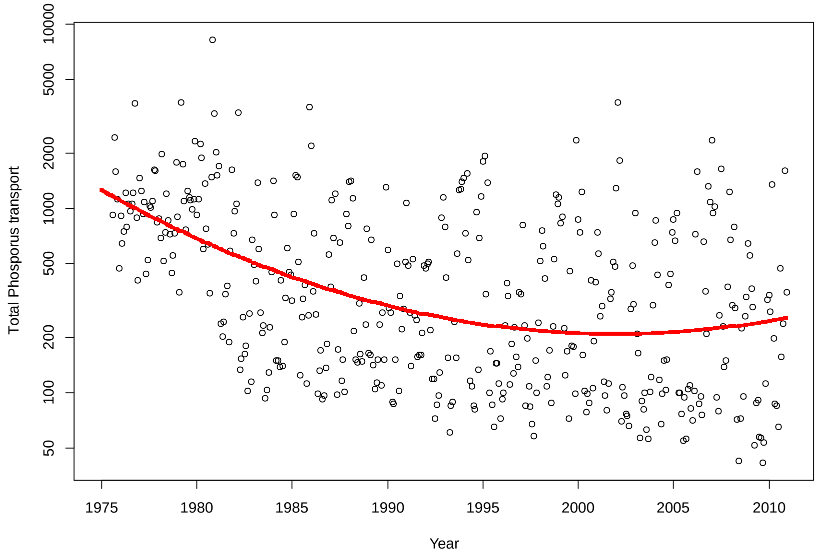
Why use General Additive Models (GAMs)?



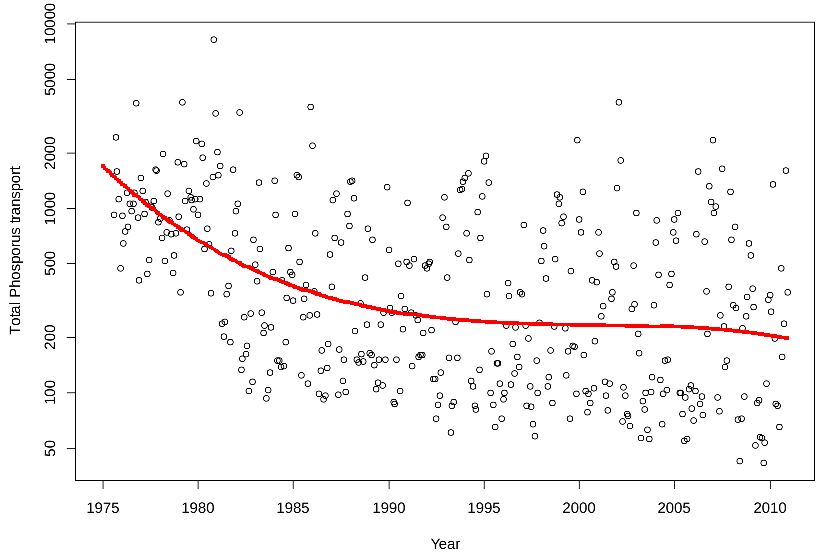
Why use General Additive Models (GAMs)?

Using GAMs we relax the **assumption of linearity** between predictors and response variable.

We could do this also with a general linear model (GLM) using a quadratic or cubic fit or with a nonlinear regression model.



quadratic fit



cubic fit

Why use General Additive Models (GAMs)?

- we do not need to determine the functional form of the relationship in beforehand

- if relationships are best approximated by linear, quadratic or cubic function the result of GAM simplifies to that

- we have most of the possibilities we have with GLMs, GLiMs, and GLMM, e.g. we can

    - include categorical predictors and interactions and
    - use other distributions than normal for the response
    - use mixed approaches to include autocorrelation estimates or hierarchical sampling structures

So, how do I use GAMs anyway?

A couple of statistical softwares have some GAM functions available, but if you want to use all available features you need to choose R.

Package gam:
https://cran.r-project.org/web/packages/gam/index.html

Package mgcv:
https://cran.r-project.org/web/packages/mgcv/mgcv.pdf

Book:
Generalised Additive Models – An introduction with R, Simon Wood

The code I show you will be using mgcv:gamm.

So, how do I use GAMs anyway?

In SAS there is `PROC GAM`, but with much fewer choices.

You can also model smooth relationships between response and predictors within `PROC GLIMMIX`. Use the `effect` statement.

But what is a GAM?

A GAM can be written as:

$$Y = a + f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n) + \varepsilon$$

where a is an intercept and f are smooth functions.

As smooths different types of functions can be used such as local linear regression (loess) or splines.

Generally splines have better mathematical properties and are most often used in GAM fitting.

But what is a GAM?

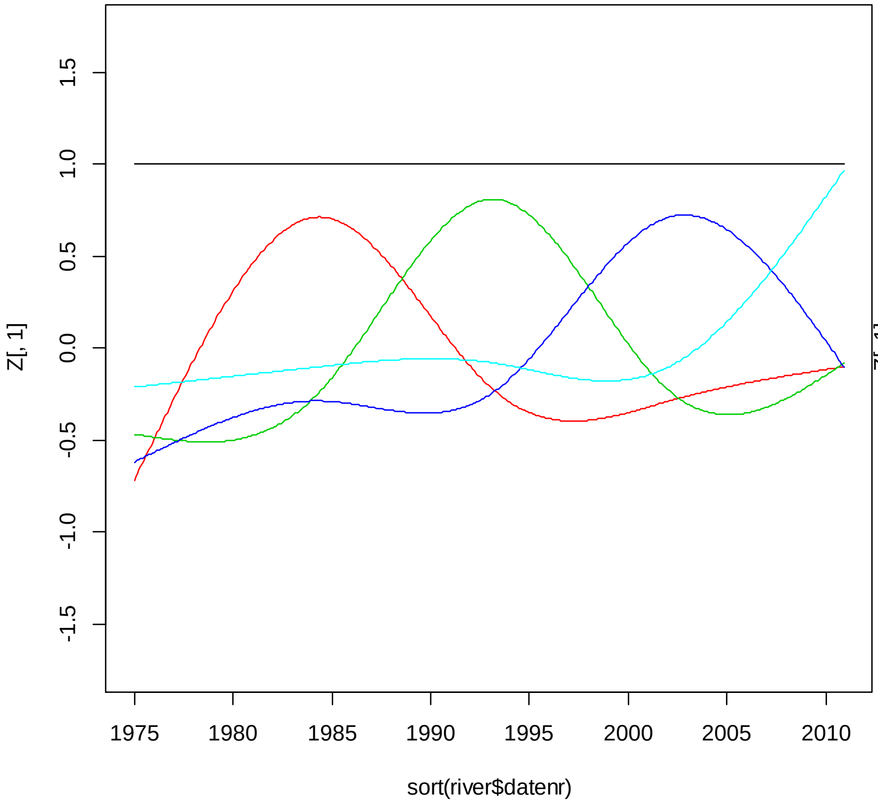
Splines are sums of weighted basis functions.

The flexibility of the fit is determined by the amount of basis functions.

Depending on which type of spline is used the basis function look differently and have different properties.

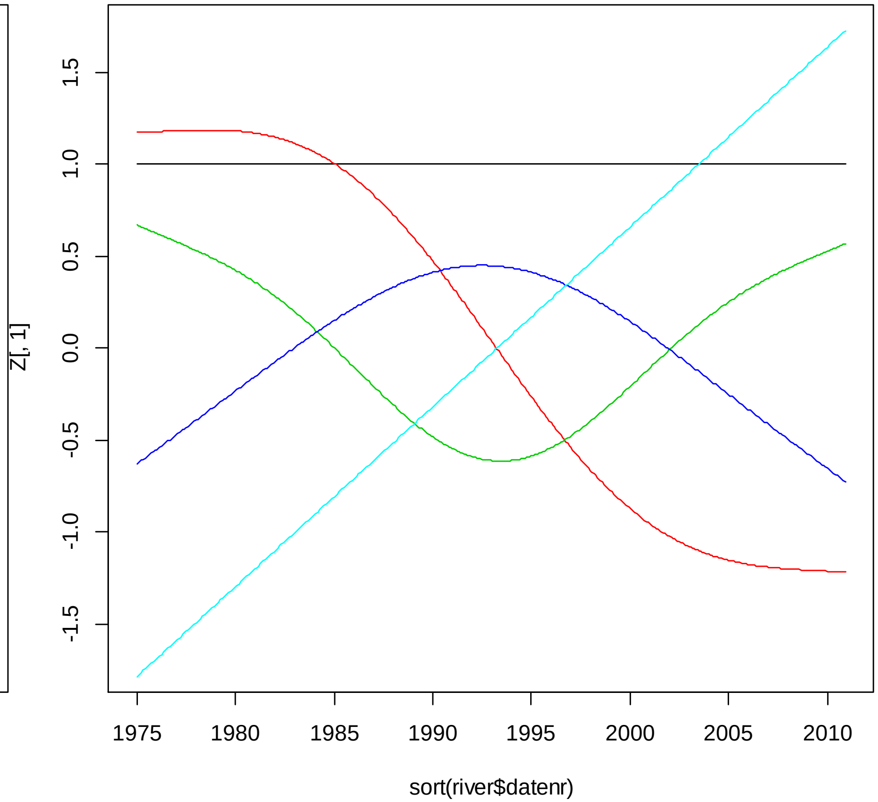
And what are basis functions?

Basis functions of a cubic regression spline



Using knots

Basis functions of a thin plate spline



Not using knots

And what are basis functions?

Combining basis functions to create a smooth.

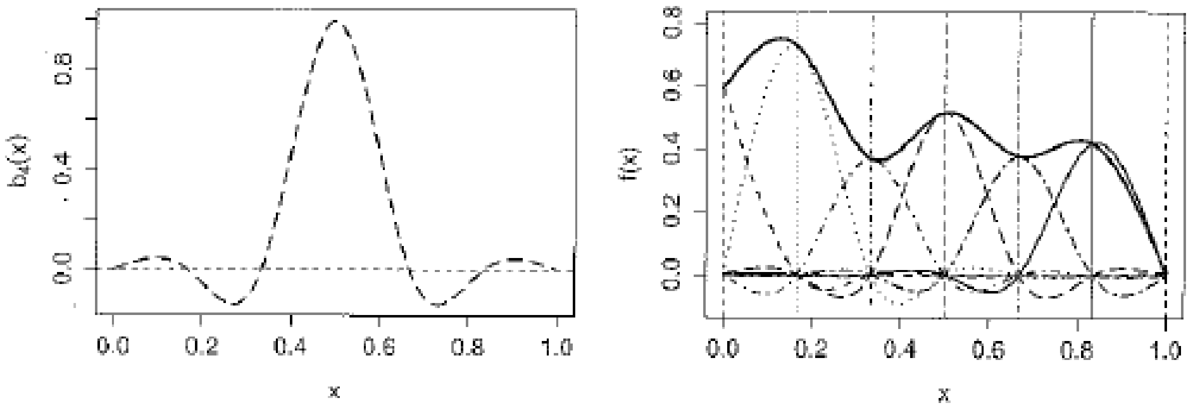


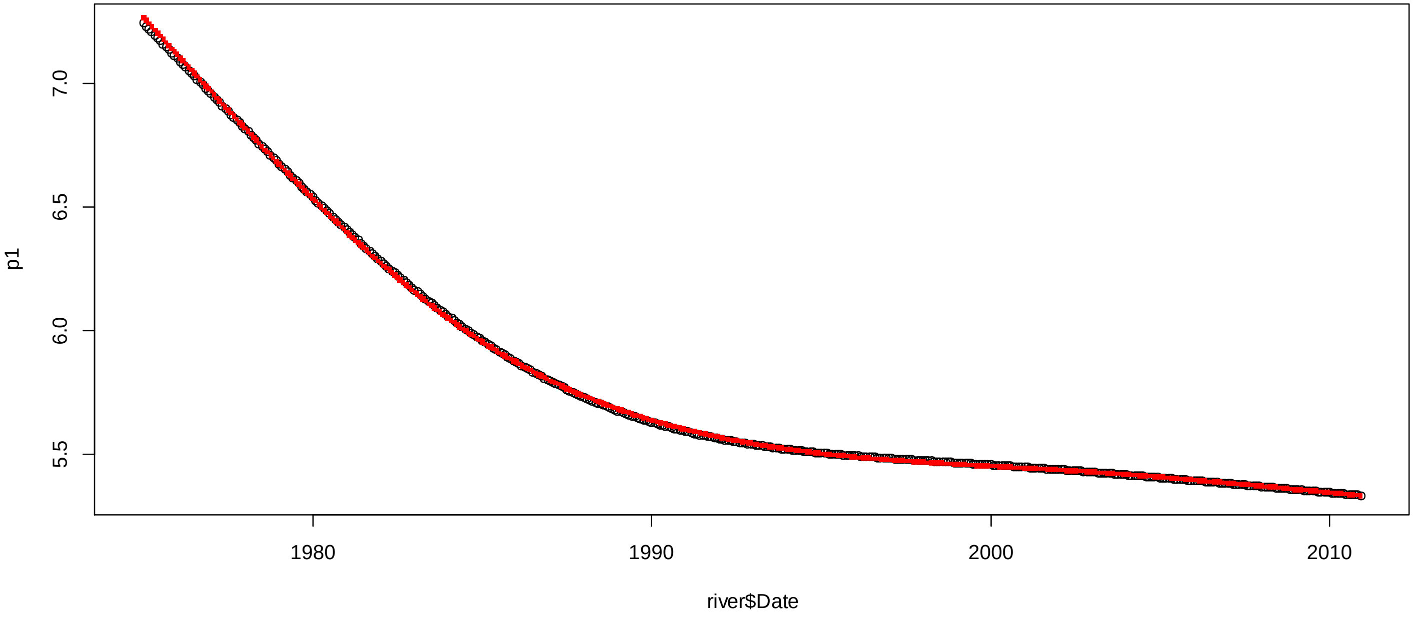
Figure 4.1. in Generalised Additive models – An introduction with R, Simon Wood, Chapman & Hall

But what is a GAM?

Even though basis functions can look very different from each other the final fit is often very similar:

Black curve: thin plate spline

Red curve: cubic regression spline



Some versions of splines in the mgcv package:

Thin plate spline (`tp`):

- does not use knots
- can be used for multiple covariates (modelling interactions)
- computationally expensive

Cubic regression splines (`cr`):

- uses knots
- can only be used for single covariates
- computationally less expensive

Cyclic cubic regression splines (`cc`):

- as `cr`, but has the same start and end point, e.g. for modelling seasonality

Some versions of splines in the mgcv package:

Thin plate spline with shrinkage (`ts`):

- as `tp`, but allows the complete removal of covariates if they are not needed (variable selections)

Cubic regression splines with shrinkage (`cs`):

- as `cr`, but allows complete removal of covariates

Tensor products (`te`):

- another alternative if you have multiple covariates (= interactions), see later

GAM using mgcv:

Fit a model using time (datenr) as covariate to describe a temporal trend.

```
model_1c <- gamm(logTot.P~s(datenr, bs='tp'), data=river)
```

Default, a thin plate spline

```
model_1d <- gamm(logTot.P~s(datenr, bs='cr'), data=river)
```

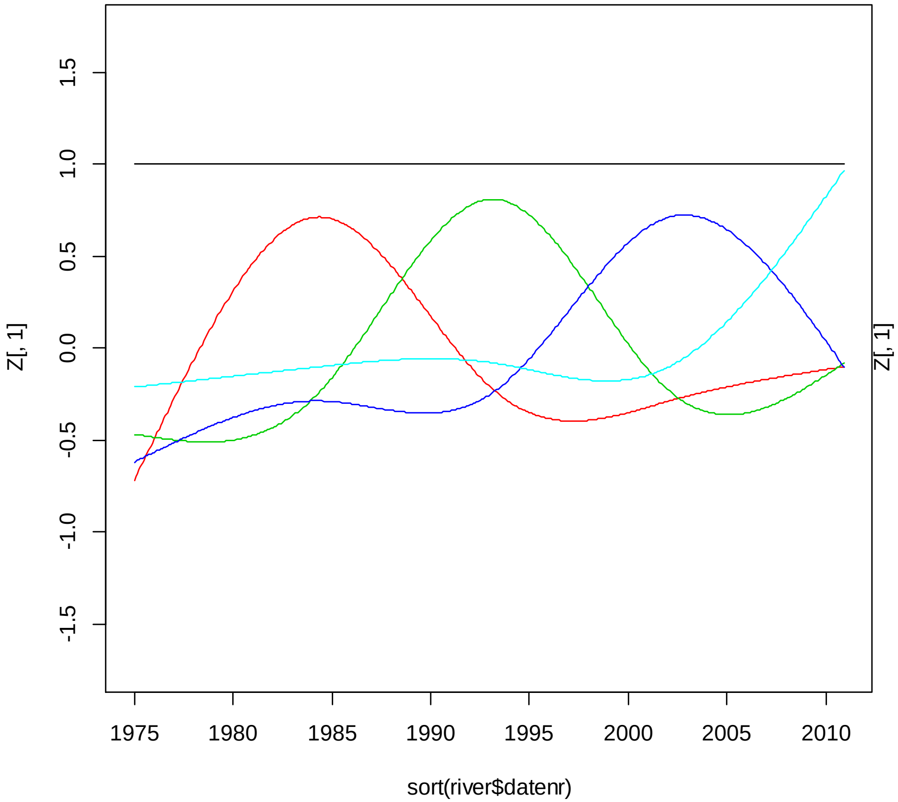
A cubic regression spline

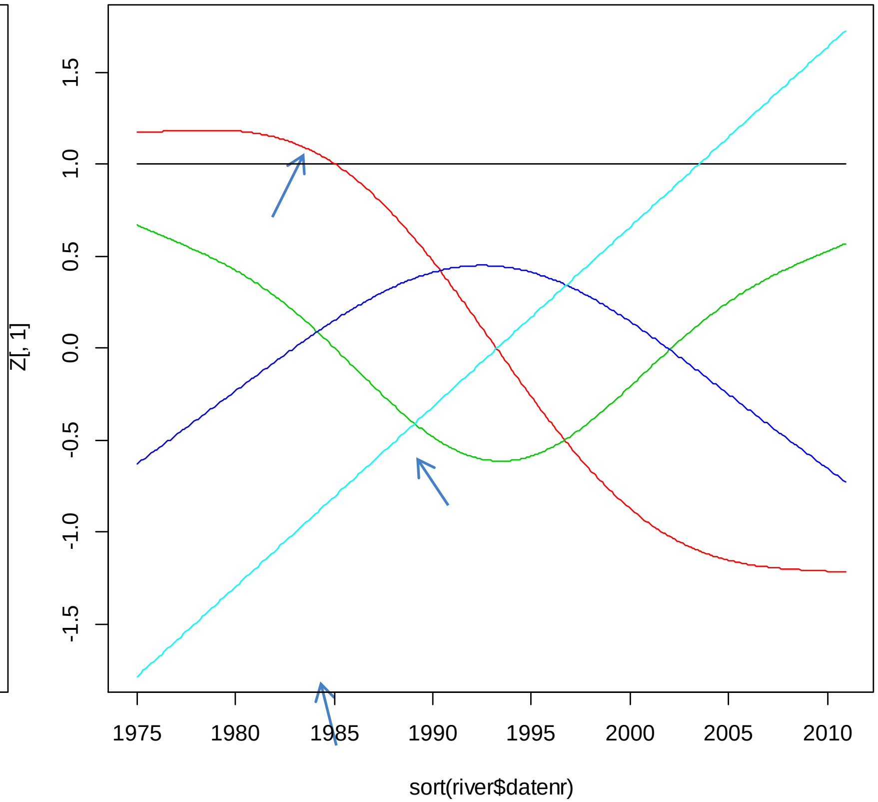
```
model_1e <- gamm(logTot.P~s(datenr, k=20), data=river)
```

Steer the complexity of the initial fit

GAM using mgcv:

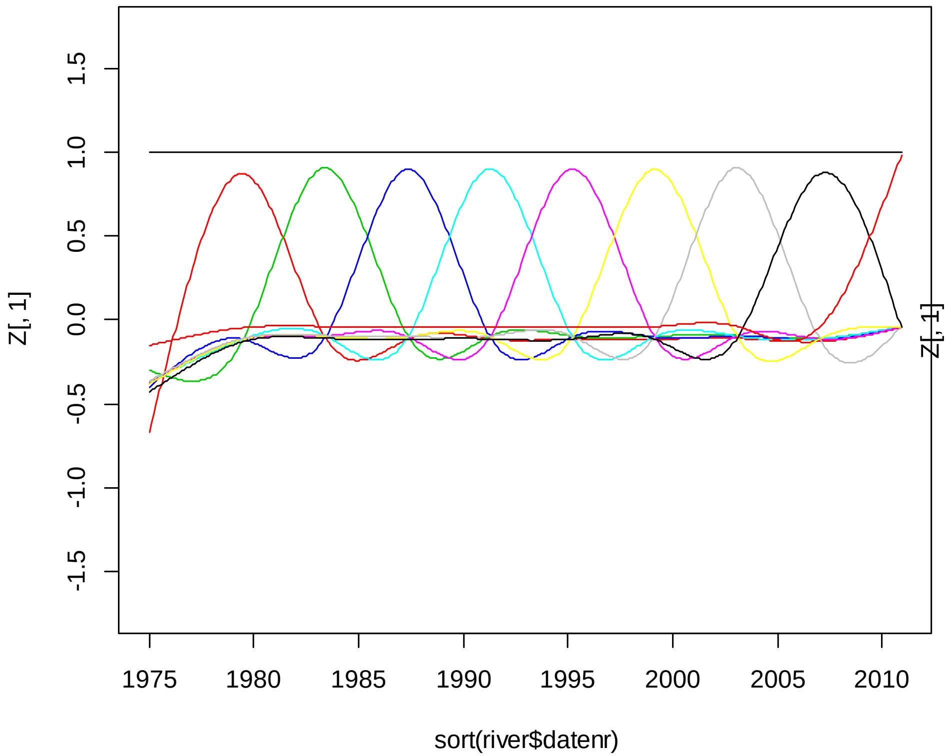
Here k=5:

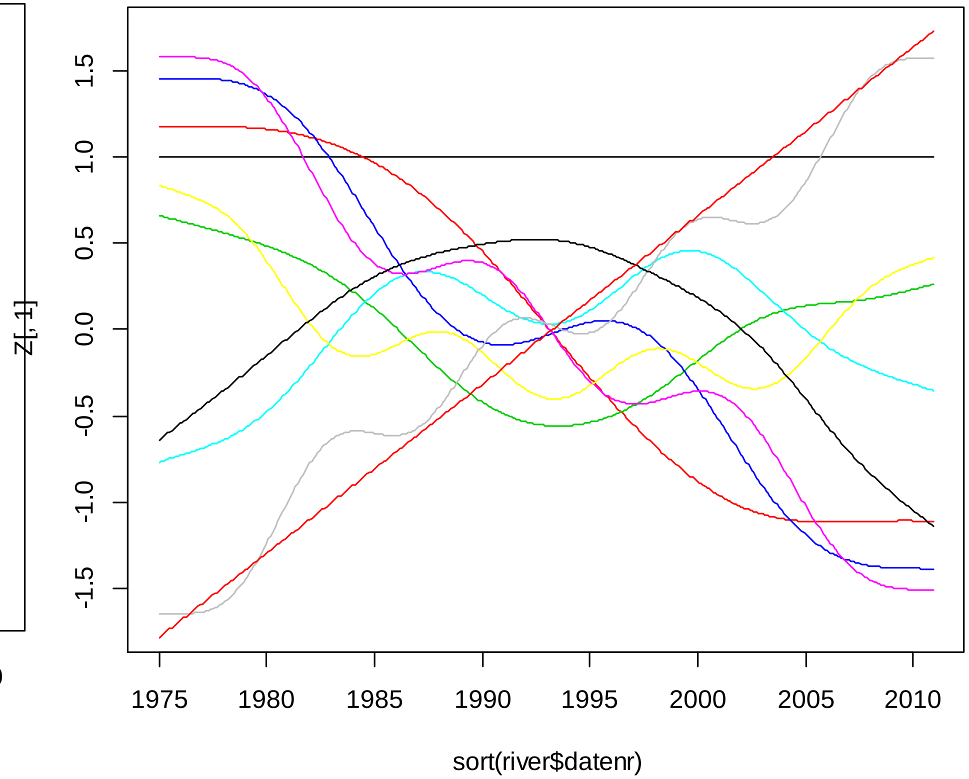




GAM using mgcv:

Here k=10:





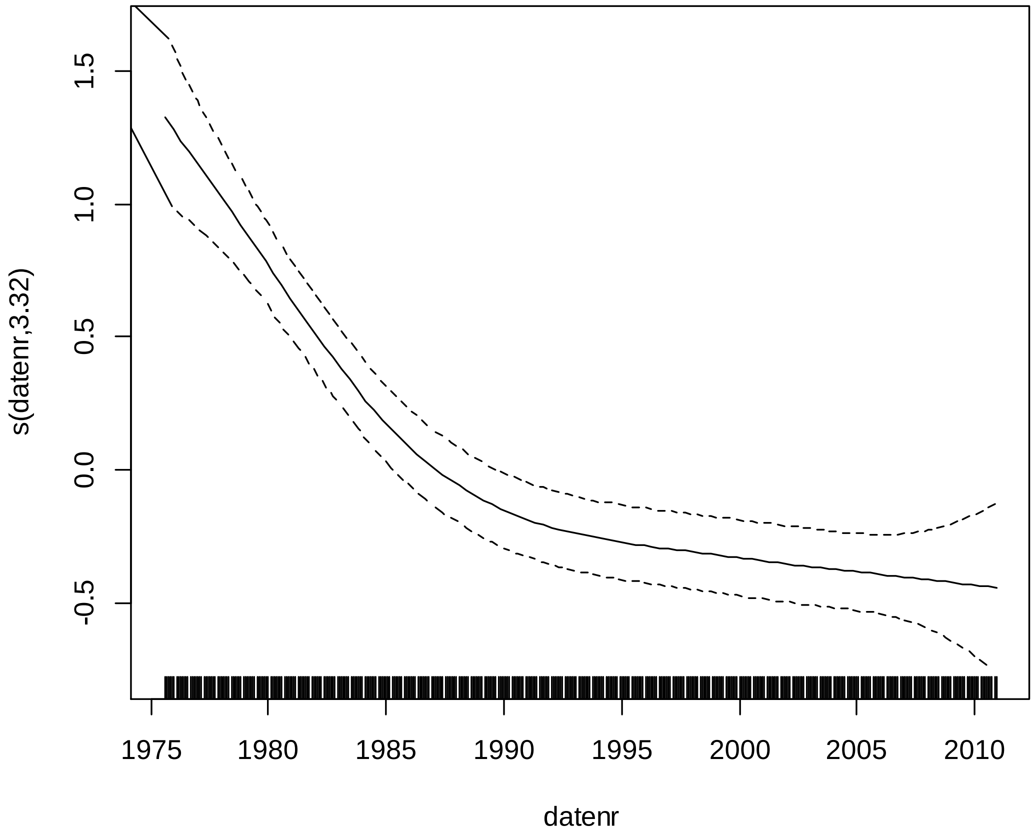
Output and interpretation of GAMs:

```
model_1c <- gamm(logTot.P~s(datenr, bs='tp'), data=river)

plot(model1c$gam)
```

Gives the fit of the smooth.

Smooth functions are usually centered to mean zero taken over the set of covariate values



Output and interpretation of GAMs:

summary(model1c$gam)

```
Family: gaussian
Link function: identity

Formula:
logTot.P ~ s(datenr, bs = "tp")

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.78712    0.04591   126.1   <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(datenr) 3.316  3.316 35.73  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.219
  Scale est. = 0.89367   n = 425
```

Output and interpretation of GAMs:

`summary(model1c$gam)`

```
Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.78712    0.04591   126.1   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Parametric estimate for intercept. Interpretation as usual, but remember that the spline is centered at zero, i.e. the intercept measures in this case the overall mean:

```
> mean(river$logTot.P, na.rm=T)
[1] 5.787118
```

Output and interpretation of GAMs:

`summary(model1c$gam)`

```
R-sq.(adj) =  0.219
  Scale est. = 0.89367   n = 425
```

R^2-value is as usual the proportion of variance explained by the model.

Scale estimate is the variance of the residual.

Output and interpretation of GAMs:

summary(model1c$gam)

```
Approximate significance of smooth terms:
            edf Ref.df     F p-value    
s(datenr) 3.316  3.316 35.73  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see that there is a significant effect of time, but the p-values are only approximate and should be handled with care.

The effective degrees of freedom are just above 3 indicating that the fit is similar to a GLM with a cubic function.

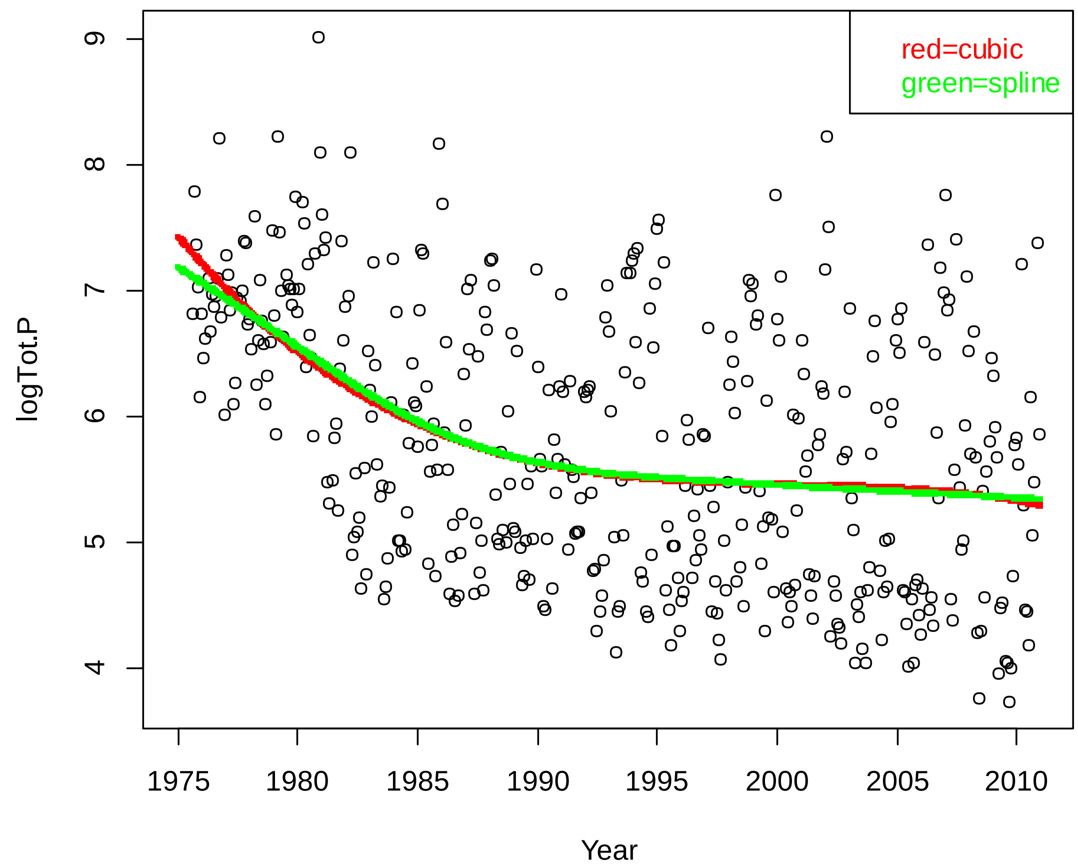
$$Y = a + b_1 x + b_2 x^2 + b_3 x^3$$

Output and interpretation of GAMs:

Using the model above we get the green line.

A cubic regression model would give similar results = the red line.

The complexities of the models are similar.



Output and interpretation of GAMs:

summary(model1c$gam)

```
Approximate significance of smooth terms:
            edf Ref.df     F p-value    
s(datenr) 3.316  3.316 35.73  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Unlike 'traditional' regression we can not interpret any coefficients or express the estimated curve by a formula.

Instead we visualise the fit by plotting.

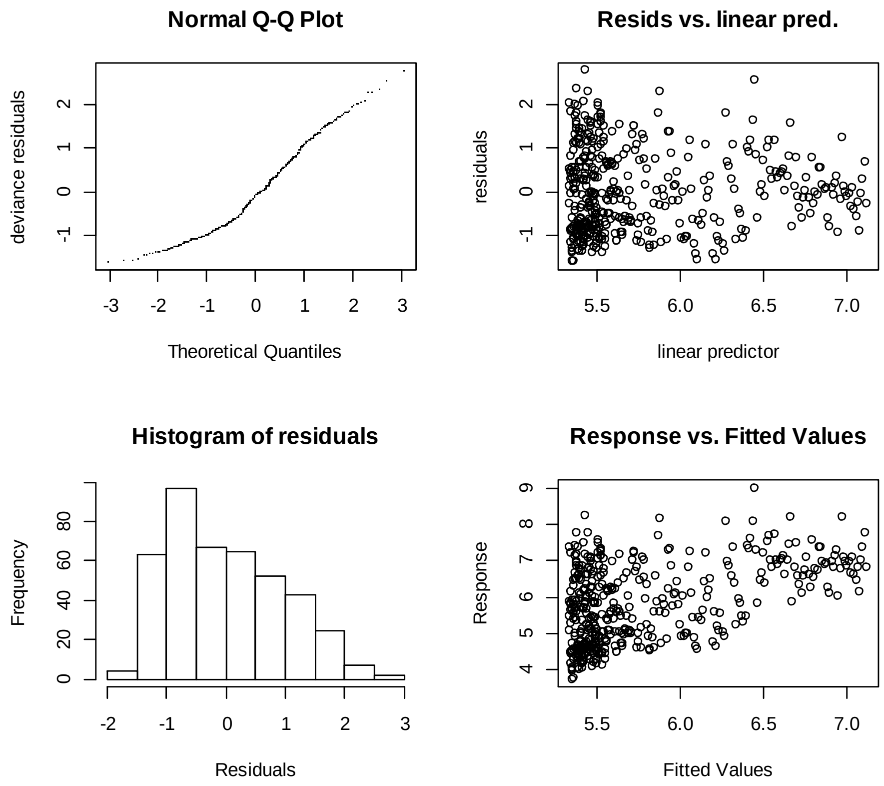
Output and interpretation of GAMs:

```
gam.check(model1c$gam)
```

Residual checking can be done in much the same way as for traditional GLMs.

Basic assumptions:

- normality of residuals
- equality of variances



Output and interpretation of GAMs:

As usual, p-values and confidence intervals rely on the assumption of independence of observations.

Independence needs to be ensured when collecting data.

Here we have a time series of observations → the observations are certainly not independent from each other.

If data is collected as a time series, in space or by a hierarchical/clustered/nested scheme, dependencies can be estimated within the model. See GAMM tomorrow.

Output and interpretation of GAMs:

For a subset of the dataset (December values) we get the following output:

```
Family: gaussian
Link function: identity

Formula:
logTot.P ~ s(datenr, bs = "tp")

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.4865     0.1361   47.67   <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(datenr)   1      1 4.285   0.046 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.0829
  Scale est. = 0.64795   n = 36
```

Output and interpretation of GAMs:

The output above indicates a linear fit, i.e. a traditional linear regression equation.

Since the smooth is centered around 0 leading to the intercept being the mean of the response variable we estimate the regression line corresponding to:

$$y = \beta_0 + \beta_1 \cdot \left(\frac{x - \bar{x}}{s_x}\right)$$

A regression line for standardized values of the explanatory variable.

Output and interpretation of GAMs:

Create a standardised explanatory variable

```
river_subset$date_std<-(river_subset$datenr-
mean(river_subset$datenr, na.rm=T))/sd(river_subset$datenr)
```

Fit a linear regression model to the data

```
model1g<-lm(logTot.P~date_std, data=river_subset)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.4865     0.1380   46.99   <2e-16 ***
date_cent    -0.2857     0.1400   -2.04   0.0491 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8283 on 34 degrees of freedom
Multiple R-squared:  0.1091,    Adjusted R-squared:  0.08288
F-statistic: 4.163 on 1 and 34 DF,  p-value: 0.04914
```

Output and interpretation of GAMs:

From the GAM output:

```
Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.4865     0.1361   47.67   <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(datenr)   1      1 4.285   0.046 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.0829
  Scale est. = 0.64795   n = 36
```

Residual variance is not exactly the same: $0.8283^2=0.686$

Output and interpretation of GAMs:

The slope estimate can be found in the lme-output of the GAM fit:

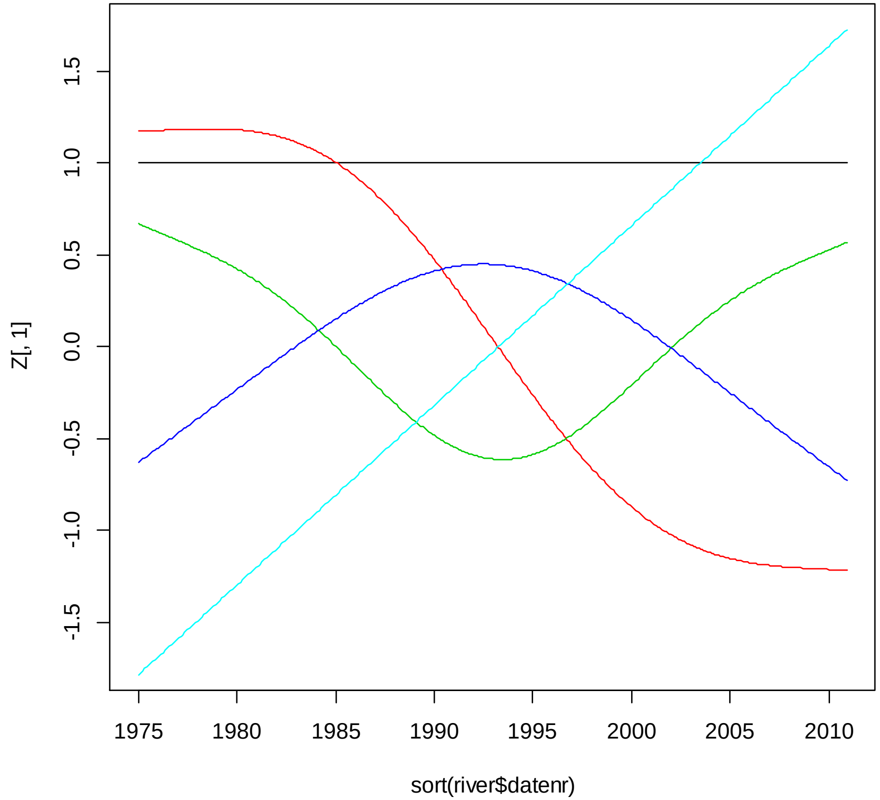
```
summary(model1f$lme)

Random effects:
 Formula: ~Xr - 1 | g
 Structure: pdIdnot
                 Xr1          Xr2          Xr3          Xr4          Xr5
StdDev: 3.795559e-05 3.795559e-05 3.795559e-05 3.795559e-05 3.795559e-05
                 Xr6          Xr7          Xr8  Residual
StdDev: 3.795559e-05 3.795559e-05 3.795559e-05 0.8049558

Fixed effects: y ~ X - 1
                    Value Std.Error DF  t-value p-value
X(Intercept)     6.486455 0.1380488 34 46.98668  0.0000
Xs(datenr)Fx1   -0.281668 0.1380488 34 -2.04035  0.0491
```

The regression coefficient is slightly different since variance estimates are different = standardization results are not exactly the same. This only works for tp splines.

Output and interpretation of GAMs:



The `Xs(datenr)Fx1` coefficient is connected to the light blue basis function.

Specifying GAM models:

As with traditional GLM models we could be interested in specifying models with

- several explanatory variables, categorical or continuous
- interactions between explanatory variables

We could also be interested to let some of the explanatory variables have a parametric form (e.g. linear), whereas others are smooth.

Modelling in GAM: several explanatory variables

Several numerical or categorical explanatory variables can be included in a GAM.

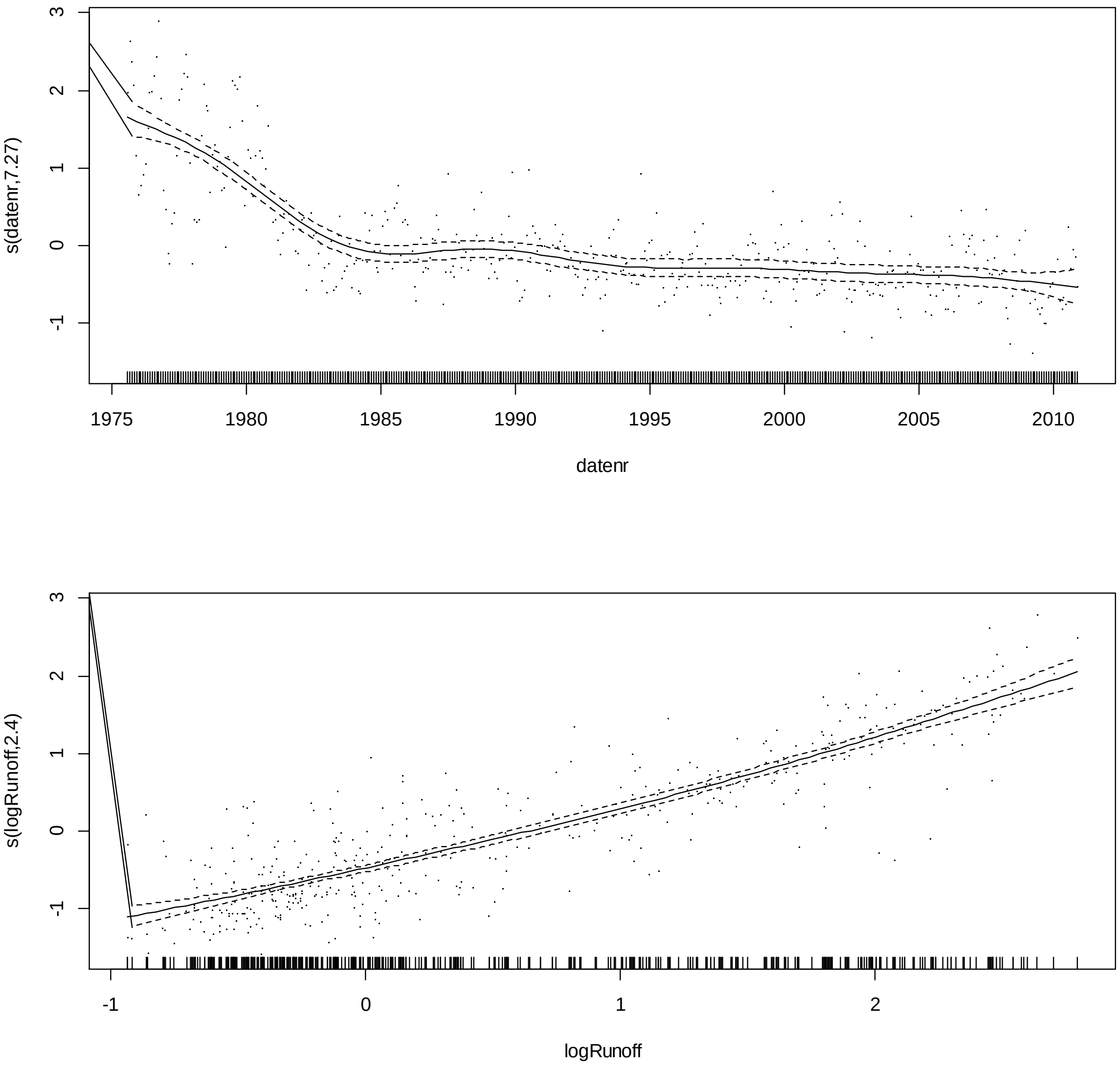
Usually an additive structure is used, but interactions can be specified.

```
model2 <-gamm(logTot.P~s(datenr)+s(logRunoff), data=river)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.78712    0.02192     264   <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(datenr)     7.27   7.27  87.39  <2e-16 ***
s(logRunoff) 2.40   2.40 585.13  <2e-16 ***
```

Modelling in GAM: several explanatory variables



Modelling in GAM: parametric terms

Not all terms in the model need to be estimated with splines. parametric estimates can be made as well.

```
river$Month1<-as.factor(river$Month)
```

```
model3 <- gamm(logTot.P~s(datenr)+s(logRunoff)+Month1,
data=river)
```

Month1 is a categorical/factor variable indicating months 1-12.

Using the formula above monthly means are estimated in the model.

Modelling in GAM: parametric terms

```
Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.54591    0.07325  75.710  < 2e-16 ***
Month12      0.02043    0.09440   0.216 0.828781
Month13     -0.04433    0.09421  -0.471 0.638197
Month14     -0.27011    0.09628  -2.806 0.005265 **
Month15      0.01639    0.10410   0.157 0.875007
Month16      0.40016    0.10867   3.682 0.000263 ***
Month17      0.53832    0.10882   4.947 1.11e-06 ***
Month18      0.56200    0.10918   5.148 4.13e-07 ***
Month19      0.60374    0.10798   5.591 4.16e-08 ***
Month110     0.57104    0.10366   5.509 6.45e-08 ***
Month111     0.39237    0.09754   4.022 6.87e-05 ***
Month112     0.07596    0.09370   0.811 0.418035
---

Approximate significance of smooth terms:
               edf Ref.df      F p-value
s(datenr)    7.994  7.994  110.9  <2e-16 ***
s(logRunoff) 1.000  1.000 1126.3  <2e-16 ***
---
```

Modelling in GAM: cyclic terms

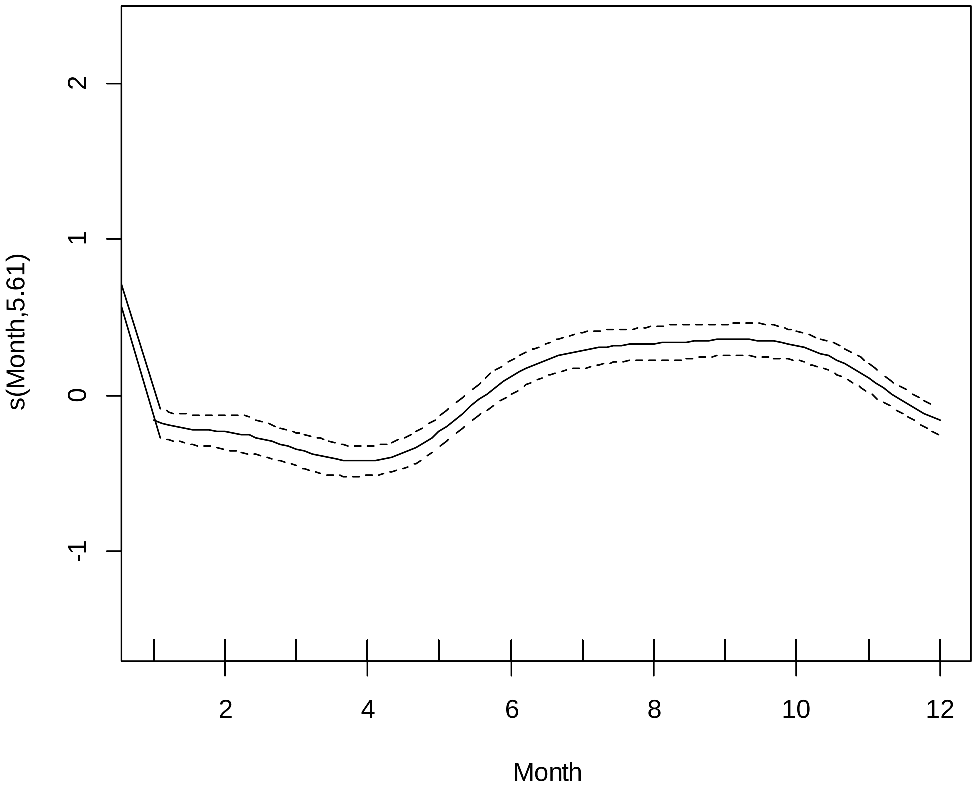
An alternative is to estimate a annual cycle to describe the seasonal variation. For this we can use a cyclic cubic regression spline.

```
model4 <- gamm(logTot.P~s(datenr)+s(logRunoff)+
       s(Month, bs='cc'), data=river)
```

Month is a numerical variable with values 1-12. The choice bs=’cc’ forces the spline to connect the estimate at 12 with the estimate at 1.

Modelling in GAM: cyclic terms

The cyclical spline for seasonal variation. The line connects at 12 and 1.



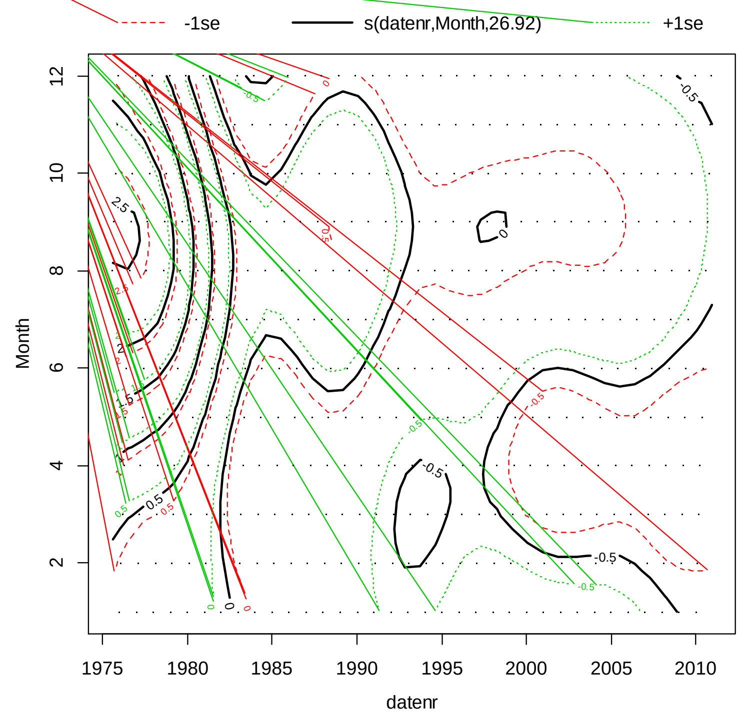
Modelling in GAM: interactions

If interactions between two variables are expected they can be included in model using a two-dimensional spline.

```
model5 <- gamm(logTot.P ~ s(datenr, Month) + s(logRunoff), data=river)
```

Results are difficult to see: highest values in autumn in the 1970s

In the 2000 generally lower values, but still higher in autumn.

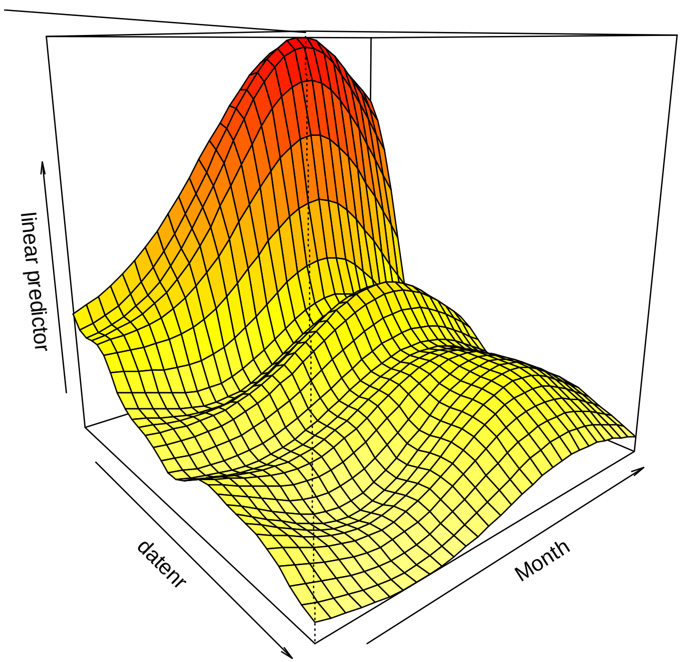


Modelling in GAM: interactions

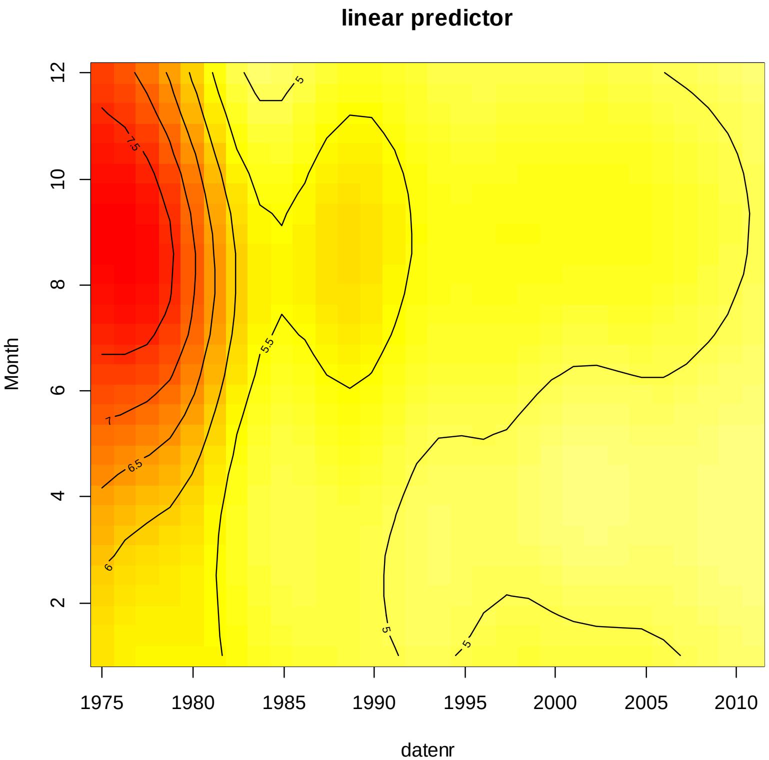
Nicer plots can be obtained by `vis.gam()`

```
vis.gam(model5$gam,view=c("datenr","Month"), theta=50)
```

Perspective plot



Contour plot: `plot.type='contour'`



Modelling in GAM: interactions

Using the usual `s()` function for the smooth for interactions uses thin plate splines.

In this option isotropy is assumed, i.e. the same amount of smoothing is used in both directions (time and month).

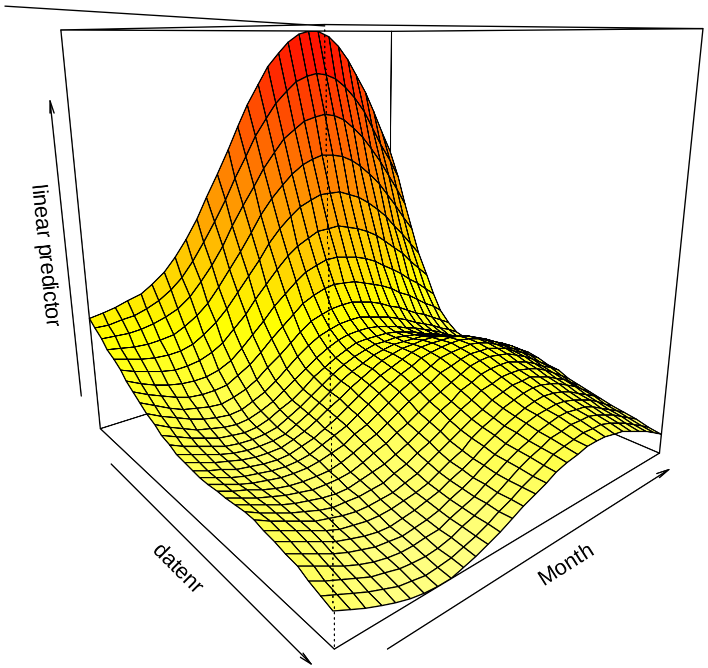
This could be reasonable for spatial fitting, or for interactions where both variables are in the same unit, but not in our case.

For interactions between variables that should not be smoothed with the same amount, we can use tensor products (`te`)

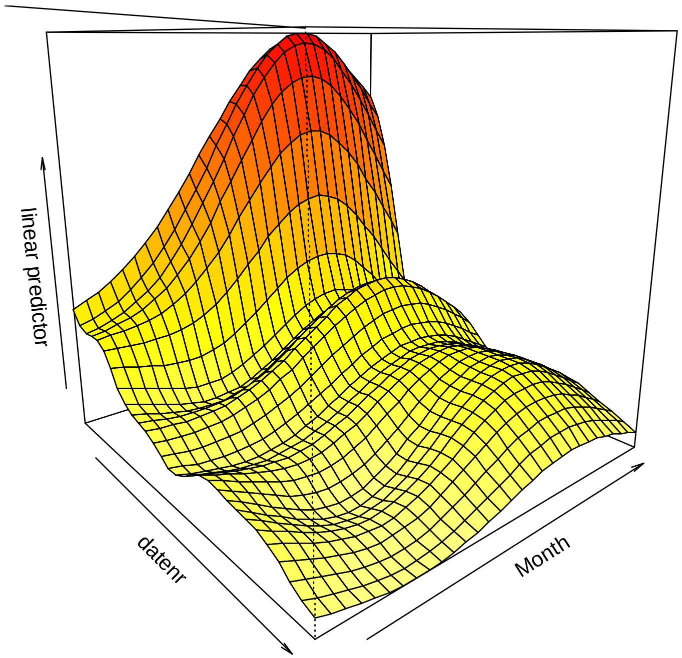
Modelling in GAM: interactions

```
model6 <- gamm(logTot.P~s(logRunoff)+te(datenr, Month),
data=river)
```

Using tensor product



Using thin plate spline



Modelling in GAM: main effects and interactions

```
model7 <- gamm(logTot.P~s(logRunoff)+ti(datenr)+ti(Month)+
ti(datenr, Month) , data=river)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.78607    0.01733   333.9   <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(logRunoff)      1.000  1.000 1449.71  <2e-16 ***
ti(datenr)        3.932  3.932  259.71  <2e-16 ***
ti(Month)         3.880  3.880   40.41  <2e-16 ***
ti(datenr,Month) 11.510 11.510   12.16  <2e-16 ***
```

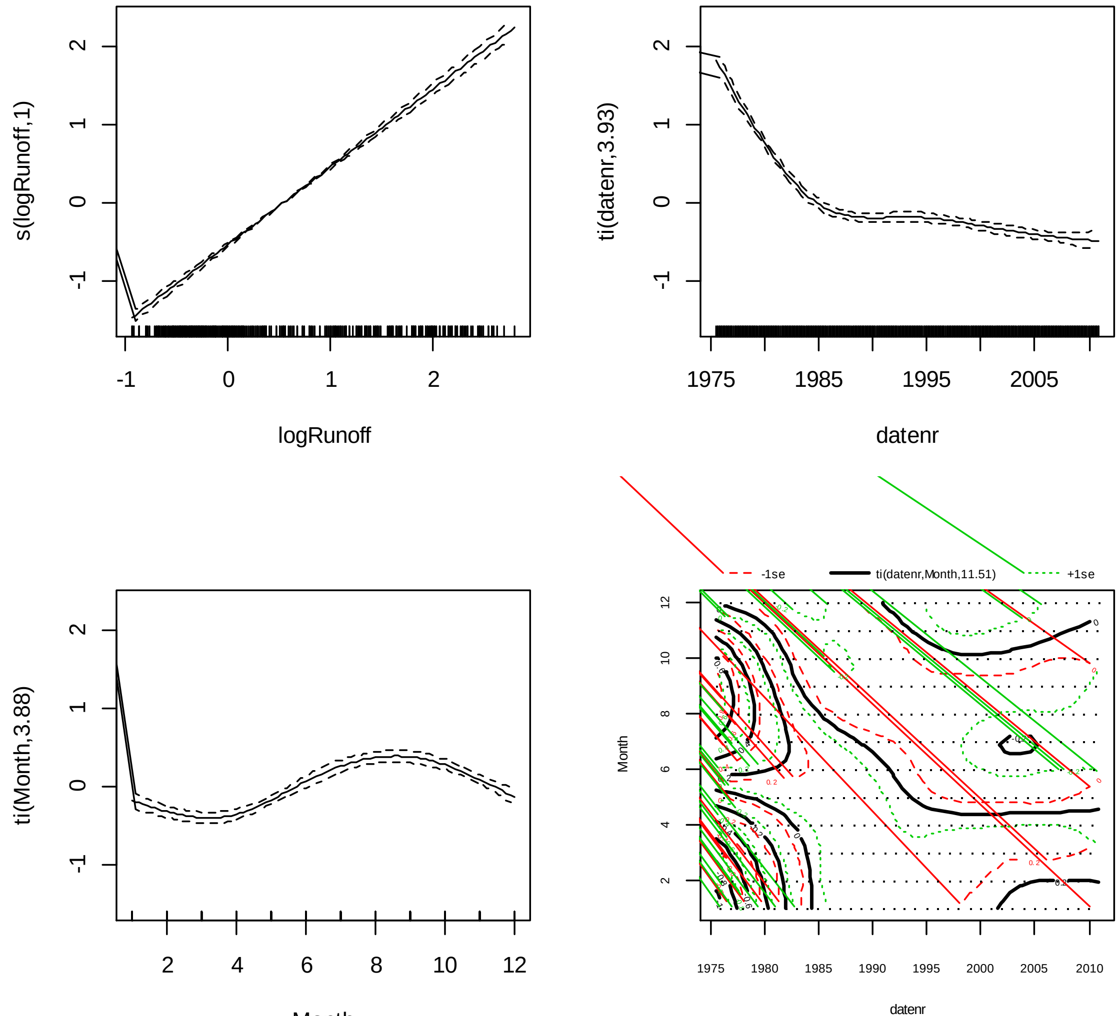
Modelling in GAM: interactions

- smooth for Runoff

- smooth for the main effect date/time

- smooth for the main effect month

- interaction between date and month,



Modelling in GAM: interaction with factor variables

Instead of only modelling seasonality as change in mean values we could be interested to model seasonal trends, i.e. estimating a trend function for each month.

```
model8<- gamm(logTot.P ~ Month1 + s(datenr, by= Month1) +
s(logRunoff), data=river)
```

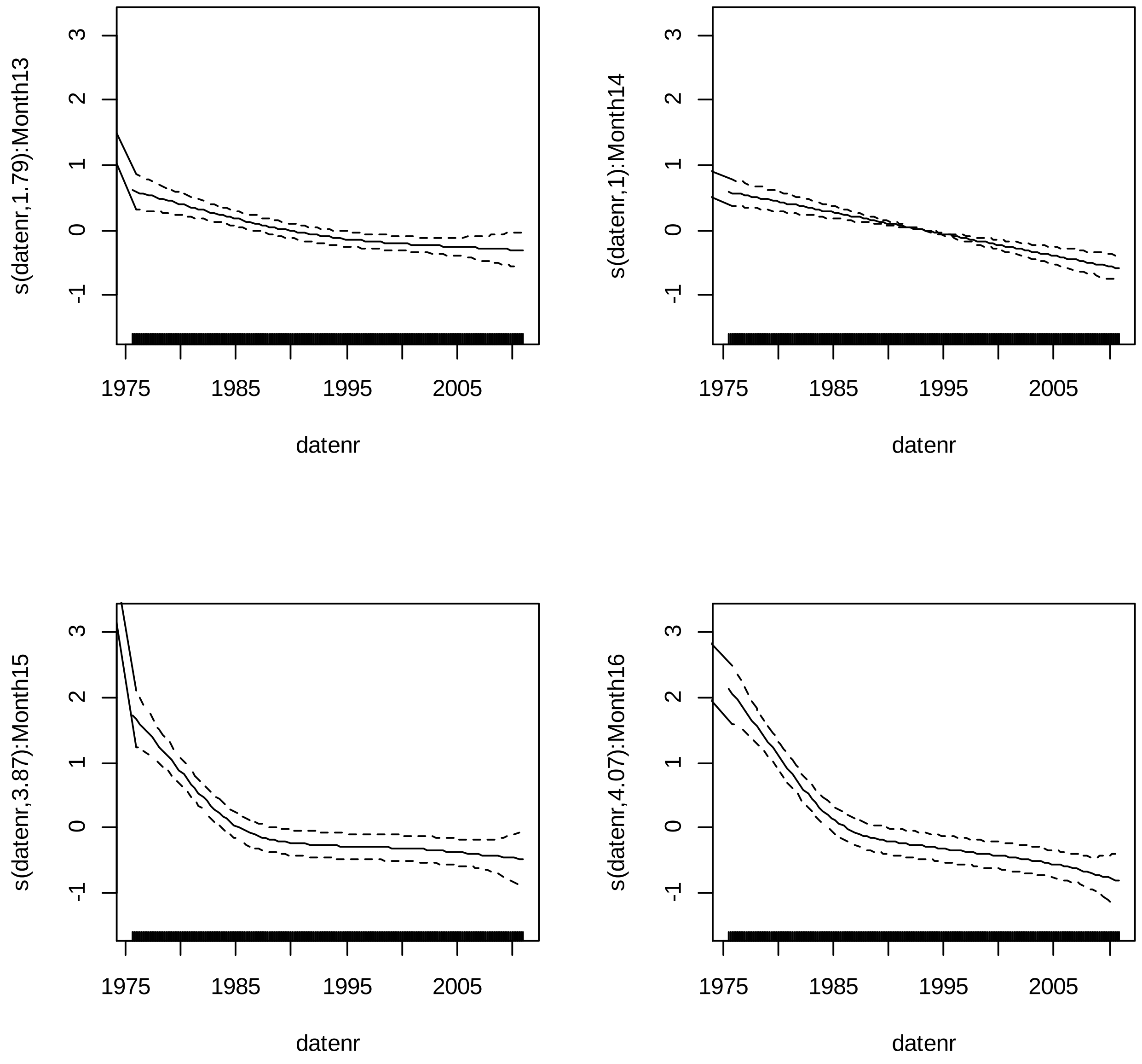
The basis functions for date are multiplied by new coefficients for each month allowing separate trend lines.

Modelling in GAM: interaction with factor variables

```
 Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.54419    0.06349  87.330  < 2e-16 ***
Month12      0.01643    0.08136   0.202 0.840025
Month13     -0.05111    0.08120  -0.629 0.529471
Month14     -0.27725    0.08306  -3.338 0.000931 ***
Month15      0.02210    0.09024   0.245 0.806673
Month16      0.41154    0.09441   4.359 1.70e-05 ***
Month17      0.55346    0.09458   5.852 1.08e-08 ***
Month18      0.56096    0.09503   5.903 8.11e-09 ***
Month19      0.60296    0.09390   6.421 4.17e-10 ***
…
Approximate significance of smooth terms:
                    edf Ref.df       F  p-value
s(datenr):Month11  1.000  1.000   10.99  0.00101 **
s(datenr):Month12  1.000  1.000   23.73 1.62e-06 ***
s(datenr):Month13  1.785  1.785   15.12 8.69e-05 ***
s(datenr):Month14  1.000  1.000   34.19 1.06e-08 ***
s(datenr):Month15  3.867  3.867   24.36  < 2e-16 ***
s(datenr):Month16  4.074  4.074   38.89  < 2e-16 ***
s(datenr):Month17  5.106  5.106   35.63  < 2e-16 ***
s(datenr):Month18  4.546  4.546   41.31  < 2e-16 ***
s(datenr):Month19  4.945  4.945   38.09  < 2e-16 ***
…
```

Modelling in GAM: interaction with factor variables

Trend lines for
March until June.



Modelling in GAM: variable selection

There are no procedures for automatic variable selection in mgcv::gamm (but there is in the gam package).

Instead there are some specialised basis functions that allow shrinkage: `’ts’` and `’cs’`

```
model9 <- gamm(logTot.P ~ s(datenr, bs='ts') + s(Month,
bs='ts') + s(logRunoff, bs=’ts’) + s(Abs._F, bs='ts'),
data=river)
```

If we add a variable that is not improving the model fit the smooth is put to a straight line (0 estimated degrees of freedom).

Modelling in GAM: variable selection

with shrinkage

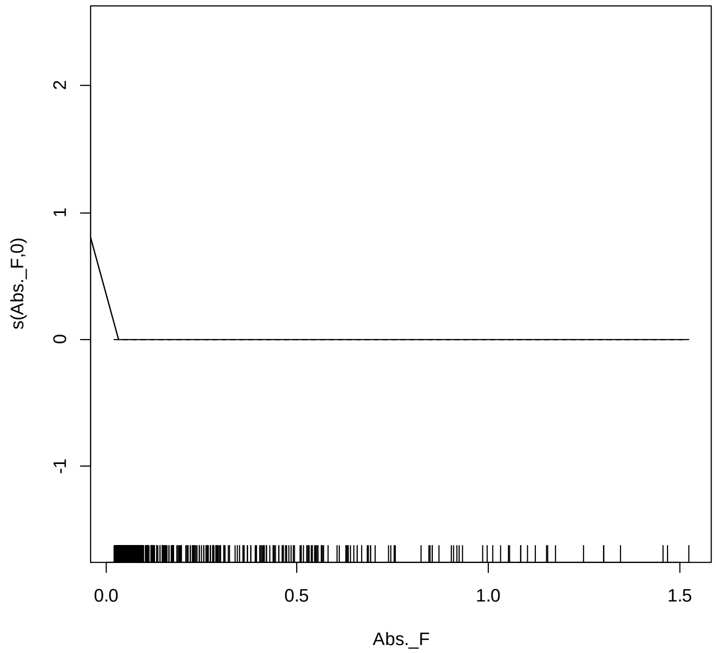
```
Approximate significance of smooth terms:
                   edf Ref.df      F p-value    
s(datenr)    7.816e+00      9  97.10  <2e-16 ***
s(Month)     6.282e+00      9  14.27  <2e-16 ***
s(logRunoff) 2.390e+00      9 128.32  <2e-16 ***
s(Abs._F)    2.727e-07      9   0.00   0.931    
```

without shrinkage:

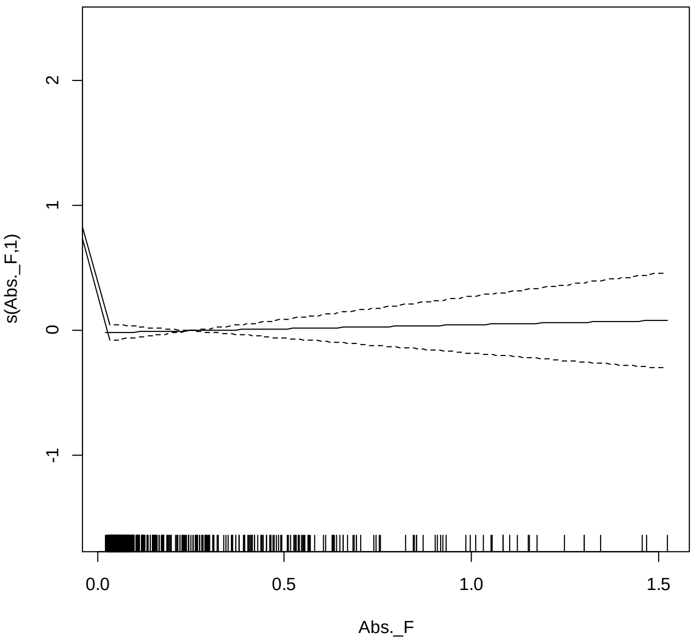
```
Approximate significance of smooth terms:
               edf Ref.df       F p-value    
s(datenr)    7.953  7.953 109.068  <2e-16 ***
s(Month)     6.482  6.482  19.042  <2e-16 ***
s(logRunoff) 1.000  1.000 305.858  <2e-16 ***
s(Abs._F)    1.000  1.000   0.183   0.669    
```

Modelling in GAM: variable selection

with shrinkage



without shrinkage



Modelling in GAM: inference and uncertainty

For all models the smoothing paramters are determined by optimising the ability to predict new data.

For the function

- `gam` usually generalised cross validation (GCV) is used,
- `gamm` (restricted) maximum likelihood estimation for the smoothing parameter is used.

Unless specifically stated a smoothness selection is therefore conducted in the model.

Modelling in GAM: inference and uncertainty

p-values and confidence intervals do not take inte account the uncertainty from the smoothing parameter estimate.

They are therefore only approximate for smooth terms and are often underestimated.

Use p-values and intervals with care.

Tomorrow: generalized and mixed models

The GAM models can also be used for other distributions than normal, e.g. Poisson for count data or Binomial for 0/1 data.

Correlation between residuals can be estimated in the models to account for temporal or spatial autocorrelations.

Data collected in hierarchical sampling designs can be analyzed with GAMM as well by including random factors that describe the sampling design.