Generalized Additive Models

Simon Wood and **Matteo Fasiolo**
School of Mathematics, University of Bristol, U.K.

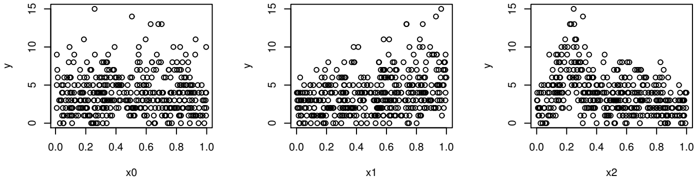
The basic model

- Response, y_i, predictors x_{ji}, model

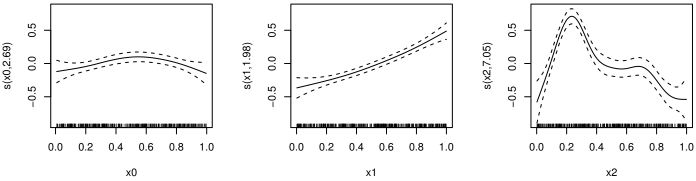
$$y_i \underset{\text{ind.}}{\sim} \pi(y_i|\mu_i, \boldsymbol{\theta}) \text{ where } g(\mu_i) = \mathbf{A}_i\boldsymbol{\gamma} + \sum_j f_j(x_{ji}).$$

  - π is a p(d)f: location parameter μ and other parameters $\boldsymbol{\theta}$.
  - The f_j are *smooth functions* to be estimated.
  - $\mathbf{A}$ is a model matrix: associated parameters $\boldsymbol{\gamma}$ to be estimated.
  - g is a known *link function* (e.g. identity or log).
- If π is an exponential family distribution then this is a GLM with linear predictor dependent on smooth functions of predictors.

Example: Poisson regression

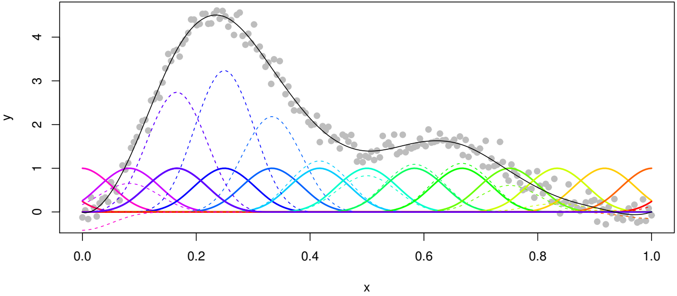


- $y_i \sim \text{Poi}(\mu_i)$ where $\log(\mu_i) = \alpha + f_0(x_{0i}) + f_1(x_{1i}) + f_2(x_{2i})$.
- `gam(y~s(x0)+s(x1)+s(x2),family=poisson())`



Model representation and estimation

- Without $\sum f_j(x_{ji})$ the model is a standard regression model: use maximum likelihood estimation via Newton's method.
- With $\sum f_j(x_{ji})$ there are two problems:
  1. How to represent the f_j for estimation.
  2. How to control and estimate the degree of smoothness for the f_j.
- For 1 use a basis expansion $f_j(x) = \sum_k \beta_{jk} b_{jk}(x)$. $b_{jk}(x)$ is a known *basis function*, β_{jk} a coefficient to estimate.



Model representation with basis

- The basis expansions for the f_j turn the model into

$$y_i \underset{\text{ind.}}{\sim} \pi(y_i|\mu_i, \boldsymbol{\theta}) \text{ where } g(\mu_i) = \mathbf{X}_i\boldsymbol{\beta},$$

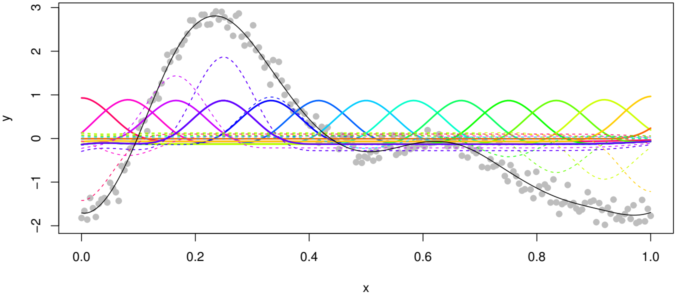
$$\boldsymbol{\beta}^{\mathsf{T}} = (\boldsymbol{\gamma}^{\mathsf{T}}, \boldsymbol{\beta}_1^{\mathsf{T}}, \boldsymbol{\beta}_2^{\mathsf{T}} \ldots) \text{ and}$$

$$\mathbf{X} = \begin{bmatrix} A_{11} & A_{12} & \cdots & b_{11}(x_{11}) & b_{12}(x_{11}) & \cdots & b_{21}(x_{21}) & \cdots \\ A_{21} & A_{22} & \cdots & b_{11}(x_{12}) & b_{12}(x_{12}) & \cdots & b_{21}(x_{22}) & \cdots \\ . & . & \ldots & . & . & \ldots & . & \ldots \\ . & . & \ldots & . & . & \ldots & . & \ldots \end{bmatrix}$$

- If π is an exponential family distribution this is just a richly parameterized GLM.

Identifiability

- The f_j in $\sum_j f_j(x_{ji})$ are only identifiable up to an additive constant.
- Impose identifiability constraints $\sum_i f_j(x_{ji}) = 0$, for all j.
- Can absorb into the basis (modifies the $b_{ij}(x)$ and loses one).



Controlling smoothness

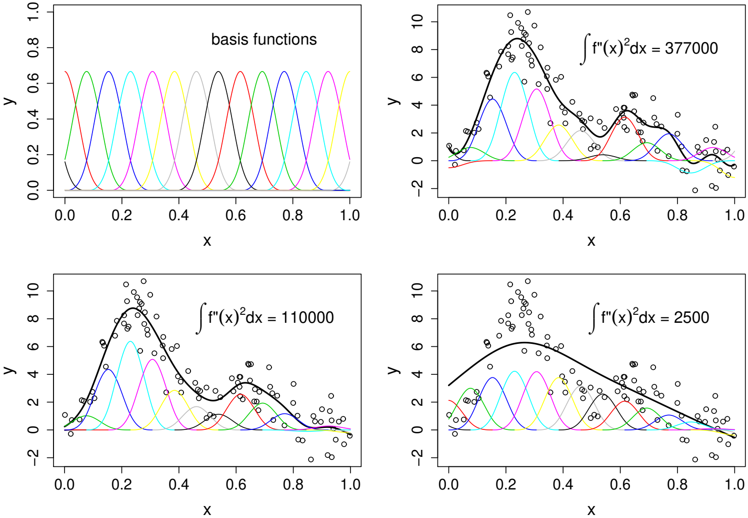
- We could control smoothness via the number of basis functions, but this is computationally awkward to optimize.
- Instead define a smoothing penalty $\lambda_j \boldsymbol{\beta}^{\mathsf{T}} \mathbf{S}_j \boldsymbol{\beta}$ to penalize and control the wiggliness of each f_j in fitting.
  - $\mathbf{S}_j$ fixed and defines what we mean by smooth.
  - *Smoothing parameter* λ_j varied to control smoothness of $\hat{f}_j$.
- E.g. given $f_j(x) = \boldsymbol{\beta}^{\mathsf{T}} \mathbf{b}(x)$ where $\mathbf{b}(x)^{\mathsf{T}} = (b_{j1}(x), b_{j2}(x), \ldots)$ then $f_j(x)'' = \boldsymbol{\beta}^{\mathsf{T}} \mathbf{b}''(x)$ so that, by definition of $\mathbf{S}_j$,

$$\int f_j''(x)^2 dx = \int \boldsymbol{\beta}^{\mathsf{T}} \mathbf{b}''(x) \mathbf{b}''(x)^{\mathsf{T}} \boldsymbol{\beta} dx = \boldsymbol{\beta}^{\mathsf{T}} \mathbf{S}_j \boldsymbol{\beta}.$$

  — a *cubic spline penalty*. Many other possibilities[1].
- So each f_j is represented by a basis and a quadratic penalty.

[1] including allowing the penalization of wiggliness to vary adaptively with x.

Basis-penalty smoothing 1D example



Penalized model fitting

- Incorporating any $\boldsymbol{\theta}$ into $\boldsymbol{\beta}$, the model p(m)f can be written as $\pi(\mathbf{y}|\boldsymbol{\beta})$ and the log likelihood as $l(\boldsymbol{\beta}) = \log \pi(\mathbf{y}|\boldsymbol{\beta})$.
- For given smoothing parameters, λ_j, the maximum penalised likelihood estimates are

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}}\, l(\boldsymbol{\beta}) - \frac{1}{2}\sum \lambda_j \boldsymbol{\beta}^{\mathsf{T}}\mathbf{S}_j\boldsymbol{\beta}$$

- The Bayesian view. For compactness write $\mathbf{S}_\lambda = \sum_j \lambda_j \mathbf{S}_j$ and consider the *smoothing prior* $\boldsymbol{\beta} \sim N(\mathbf{0}, \mathbf{S}_\lambda^-)$. By Bayes theorem

$$\begin{aligned}\log \pi(\boldsymbol{\beta}|\mathbf{y}) &= \log \pi(\mathbf{y}|\boldsymbol{\beta}) + \log \pi(\boldsymbol{\beta}) - \log \pi(\mathbf{y}) \\ &= l(\boldsymbol{\beta}) - \boldsymbol{\beta}^{\mathsf{T}}\mathbf{S}_\lambda\boldsymbol{\beta}/2 + \text{const.}\end{aligned}$$

- i.e. $\hat{\boldsymbol{\beta}}$ is the *posterior mode*, and f_j are equivalent to Gaussian random effects/fields.

Computing $\hat{\boldsymbol\beta}$ and $\pi(\boldsymbol\beta|\mathbf{y})$

- $\hat{\boldsymbol\beta} = \underset{\boldsymbol\beta}{\text{argmax}}\; l(\boldsymbol\beta) - \boldsymbol\beta^{\mathsf{T}}\mathbf{S}_\lambda\boldsymbol\beta/2 \Rightarrow \left.\frac{\partial l}{\partial \boldsymbol\beta}\right|_{\hat{\boldsymbol\beta}} - \mathbf{S}_\lambda\hat{\boldsymbol\beta} = \mathbf{0}$
- In practice use Newton iteration (until $\hat{\boldsymbol\beta}$ stops changing): $\hat{\boldsymbol\beta} \leftarrow \hat{\boldsymbol\beta} + (\hat{\boldsymbol{\mathcal{I}}} + \mathbf{S}_\lambda)^{-1}\left(\left.\frac{\partial l}{\partial \boldsymbol\beta}\right|_{\hat{\boldsymbol\beta}} - \mathbf{S}_\lambda\hat{\boldsymbol\beta}\right)$, where $\hat{\boldsymbol{\mathcal{I}}} = -\frac{\partial^2 l}{\partial\boldsymbol\beta\partial\boldsymbol\beta^{\mathsf{T}}}$.
- Taylor expand for approximate posterior

$$\begin{aligned}\log\pi(\boldsymbol\beta|\mathbf{y}) &= l(\boldsymbol\beta) - \boldsymbol\beta^{\mathsf{T}}\mathbf{S}_\lambda\boldsymbol\beta/2 + c \\ &\simeq l(\hat{\boldsymbol\beta}) - \frac{1}{2}\hat{\boldsymbol\beta}^{\mathsf{T}}\mathbf{S}_\lambda\hat{\boldsymbol\beta} - \frac{1}{2}(\boldsymbol\beta - \hat{\boldsymbol\beta})^{\mathsf{T}}(\hat{\boldsymbol{\mathcal{I}}} + \mathbf{S}_\lambda)(\boldsymbol\beta - \hat{\boldsymbol\beta}) + c\end{aligned}$$

- Hence[2] approximately $\pi_G(\boldsymbol\beta|\mathbf{y}) \propto e^{-\frac{1}{2}(\boldsymbol\beta-\hat{\boldsymbol\beta})^{\mathsf{T}}(\hat{\boldsymbol{\mathcal{I}}}+\mathbf{S}_\lambda)(\boldsymbol\beta-\hat{\boldsymbol\beta})}$, so

$$\boldsymbol\beta|\mathbf{y} \sim \text{N}(\hat{\boldsymbol\beta}, (\hat{\boldsymbol{\mathcal{I}}} + \mathbf{S}_\lambda)^{-1})$$

[2] generally requires $\dim(\boldsymbol\beta) = o(n^{1/3})$

Degrees of freedom

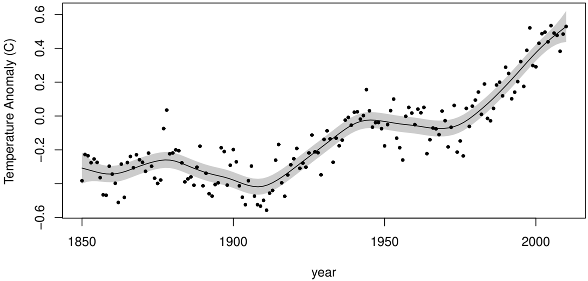
- Penalties restrict the variability of the parameters, $\boldsymbol{\beta}$. We need an alternative definition of degrees of freedom to $p = \dim(\boldsymbol{\beta})$.
- Consider a scaled measure of the variability in $\boldsymbol{\beta}$, which is just p without penalization:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\beta}|\mathbf{y}}\{(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})^{\mathsf{T}}\hat{\boldsymbol{\mathcal{I}}}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})\} &= \mathrm{tr}\{\mathbb{E}_{\boldsymbol{\beta}|\mathbf{y}}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})^{\mathsf{T}}\hat{\boldsymbol{\mathcal{I}}}\} \\ &= \mathrm{tr}\{(\hat{\boldsymbol{\mathcal{I}}}+\mathbf{S}_\lambda)^{-1}\hat{\boldsymbol{\mathcal{I}}}\}\end{aligned}$$

  — this *effective degrees of freedom* (EDF) is p under no penalization and $< p$ otherwise.
- Sum the appropriate subset of $\mathrm{diag}\{(\hat{\boldsymbol{\mathcal{I}}}+\mathbf{S}_\lambda)^{-1}\hat{\boldsymbol{\mathcal{I}}}\}$ to get the effective degrees of freedom of the coefficients of a single $\hat{f}_j$.

Bayesian Credible Interval example

- The approximate posterior $\boldsymbol{\beta}|\mathbf{y} \sim \mathrm{N}(\hat{\boldsymbol{\beta}}, (\hat{\boldsymbol{\mathcal{I}}} + \mathbf{S}_\lambda)^{-1})$ makes it easy to produce credible intervals. For example (EDF 11.7). . .



- Nychka (1988, JASA) shows such CIs have good properties: close to nominal coverage *on average across the function*.

Smoothing parameter selection

1. *Prediction error optimization*. Which $\boldsymbol{\lambda}$ would be best for predicting data not fitted? Optimize GCV/AIC like criteria, e.g.

$$-2l(\hat{\boldsymbol{\beta}}) + 2\text{EDF}.$$

2. *Marginal likelihood maximisation*. Choose $\boldsymbol{\lambda}$ to maximize the average likelihood of random draws from the prior. i.e. maximize

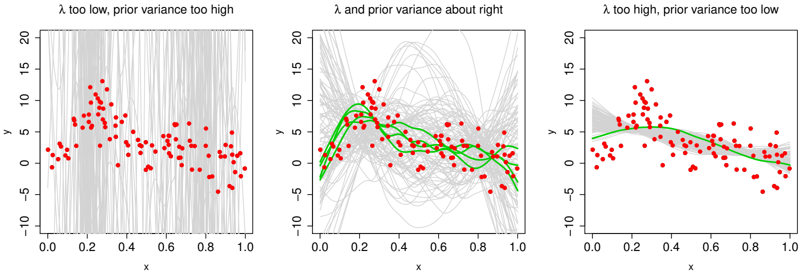
$$\text{REML} = \int \pi(\mathbf{y}|\boldsymbol{\beta})\pi(\boldsymbol{\beta}|\boldsymbol{\lambda})d\boldsymbol{\beta}$$

— intractable, but re-using Gaussian approximate posterior, π_G

$$\text{REML} = \pi(\mathbf{y}) = \frac{\pi(\mathbf{y}|\hat{\boldsymbol{\beta}})\pi(\hat{\boldsymbol{\beta}}|\boldsymbol{\lambda})}{\pi(\hat{\boldsymbol{\beta}}|\mathbf{y})} \simeq \frac{\pi(\mathbf{y}|\hat{\boldsymbol{\beta}})\pi(\hat{\boldsymbol{\beta}})|\boldsymbol{\lambda})}{\pi_G(\hat{\boldsymbol{\beta}}|\mathbf{y})}$$

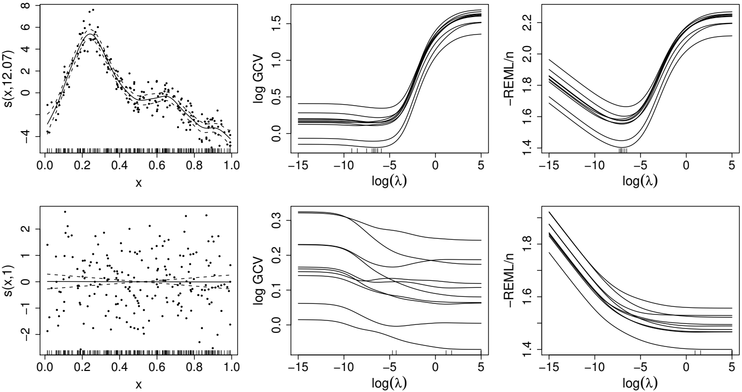
is tractable: *Laplace Approximation*.

Marginal likelihood smoothness selection idea



1. Choose λ to maximize the average likelihood of random draws from the prior implied by λ.
2. If λ too low, then almost all draws are too variable to have high likelihood. If λ too high, then draws all underfit and have low likelihood. The right λ maximizes the proportion of draws close enough to data to give high likelihood.

Prediction error vs. likelihood $\boldsymbol{\lambda}$ estimation



1. Pictures show GCV and REML scores for different replicates from same truth.
2. Compared to REML, GCV penalizes overfit only weakly, and so is more likely to occasionally undersmooth.

Computing the $\boldsymbol{\lambda}$ estimates

- Optimize the LAML[3] (or other criterion) by Newton or Quasi-Newton method w.r.t. $\boldsymbol{\rho} = \log \boldsymbol{\lambda}$. i.e. maximize successive quadratic approximations to REML, based on derivatives of REML w.r.t. $\boldsymbol{\rho}$.
- Each trial $\boldsymbol{\rho}$ requires
  1. an inner Newton iteration to find $\hat{\boldsymbol{\beta}}$ for this $\boldsymbol{\rho}$, and hence evaluate the LAML.
  2. an implicit differentiation step to find derivatives of $\hat{\boldsymbol{\beta}}$ w.r.t. $\boldsymbol{\rho}$ and hence the derivatives of the LAML.
- A less involved approach approximately maximizes the LAML by alternating updates of $\hat{\boldsymbol{\beta}}$ given $\boldsymbol{\lambda}$ with simple *Fellner-Schall*[4] updates of $\boldsymbol{\lambda}$ given $\hat{\boldsymbol{\beta}}$.

[3] Laplace Approximate Marginal Likelihood, Wood, 2011, JRSSB

[4] Wood and Fasiolo, 2017, Biometrics

Model selection

- We need means for comparing models/deciding what terms to include…
  1. Null space penalization: add an extra penalty and smoothing parameter for each f_j which allows it to be penalized to zero during smoothing parameter estimation.
  2. P-values: 'invert' the Bayesian CI for f_j to compute a p-value for $H_0 : f_j = 0$ (different for pure random effects terms).
  3. Akaike's Information Criterion becomes

     $$-2l(\hat{\boldsymbol{\beta}}) + 2EDF$$

     but to use for model comparison, rather than $\boldsymbol{\lambda}$ estimation, we must correct for $\boldsymbol{\lambda}$ estimation uncertainty[5].
- In `mgcv`: 1. `gam(...,select=TRUE)` 2. `summary` or `anova` 3. `AIC`.

[5] Problem: Greven & Kneib 2010 Biometrika. Solution: Wood et al. 2016 JASA

Model extensions

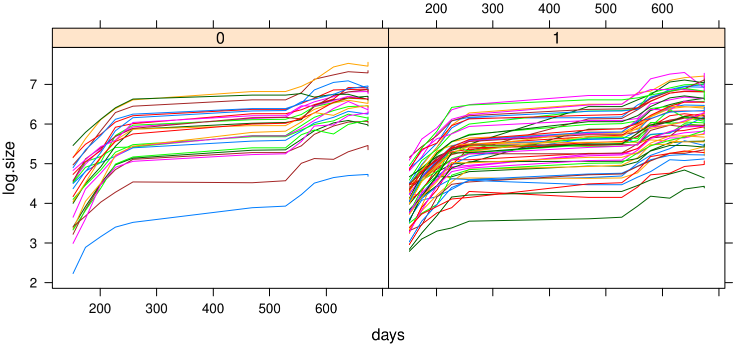
- The preceding framework applies more generally than to the models we started with!
- For example: $y_i \underset{\text{ind.}}{\sim} \pi(y_i|\theta_{1i}, \theta_{2i}, \ldots)$ where $g_k(\theta_{ki}) = \sum f_j(x_{ji})$. A *distributional regression* or GAMLSS model, with a smooth additive predictor for each of several distribution parameters (e.g. mean, variance, …).
- As hinted at earlier, any random effect that makes a contribution $\mathbf{Zb}$ to a linear predictor, where $\mathbf{Z}$ is a model matrix for the term and $\mathbf{b} \sim \text{N}(\mathbf{0}, \mathbf{I}\lambda^{-1})$, can be treated just like any smooth function in the model (only p-value computation differs).
- We can include any bounded linear functional of a smooth $L_{ij} f_j$, in place of simple evaluation of f_j. e.g. $L_{ij} f_j = \int k_i(x) f_j(x) dx$ where $k_i(x)$ is an observed function (*signal regression*). See `?linear.functional.terms` in `mgcv`.

GAMs with `mgcv::gam` in R

- Basically like any other regression model function in R.
- Modelling function `gam` has several key arguments:
    - a model formula: response on l.h.s and linear predictor on r.h.s.
        - the linear predictor can include smooth functions of predictors: e.g. `s(x,k=15,bs="cr")` is a rank 15 cubic spline.
        - if there are several linear predictors a list of formulae is supplied.
    - A `family`, specifying the distribution and any link functions.
    - A data frame containing the variables referred to in the formula.
- `gam` returns a fitted model object of class `gam`. Various methods functions are used to extract its components and summarize it...
    - `plot`, `gam.check`, `vis.gam`, `qq.gam`, `fitted`, `residuals` etc. are for visualization and checking.
    - `summary`, `anova`, `AIC`, `predict`, `vcov`, `gam.vcomp` etc. are for further inference and prediction.

Example: Sitka spruce growth data

```
require(gamair); require(lattice); require(mgcv)
data(sitka); sitka$id.num <- factor(sitka$id.num)
xyplot(log.size~days|as.factor(ozone),data=sitka,
       type="l",groups=id.num)
```



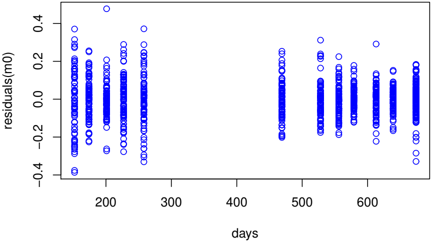
Example: Sitka spruce growth model

- $\texttt{log.size}_i = f(\texttt{days}_i) + \gamma\texttt{ozone}_i + a_{\texttt{id}(i)} + b_{\texttt{id}(i)}\texttt{days}_i + \epsilon_i$
  $a_j \sim N(0, \sigma_a^2)$, $b_j \sim N(0, \sigma_b^2)$ and $\epsilon_i \sim N(0, \sigma^2)$.
- Fit with `mgcv` (family `gaussian` is default)

```
m0 <- gam(log.size ~ s(days) + (id.num,bs="re")
          + s(id.num,days,bs="re")+ozone,data=sitka,method="REML")
```

- Basic checking with `gam.check(m0)` and `plot(m0)` and residual checks like. . .

```
plot(sitka$days,residuals(m0),xlab="days")
```



. . . variance not constant? Constant additive `ozone` effect?

Example: Sitka spruce growth model 2

- $\texttt{log.size}_i = f(\texttt{days}_i) + \texttt{ozone}_i f_1(\texttt{days}_i) + a_{\texttt{id}(i)} + b_{\texttt{id}(i)}\texttt{days}_i + \epsilon_i$, $a_j \sim N(0, \sigma_a^2)$, $b_j \sim N(0, \sigma_b^2)$, $\epsilon_i \sim N(0, \sigma_i^2)$, $\log \sigma_i = f_2(\texttt{days}_i)$.
- In `mgcv`

```
m1 <- gam(list(log.size ~ s(days) + s(days,by=ozone) +
      s(id.num,bs="re") + s(id.num,days,bs="re"), ~ s(days)),
      family=gaulss,data=sitka, method="REML")
```

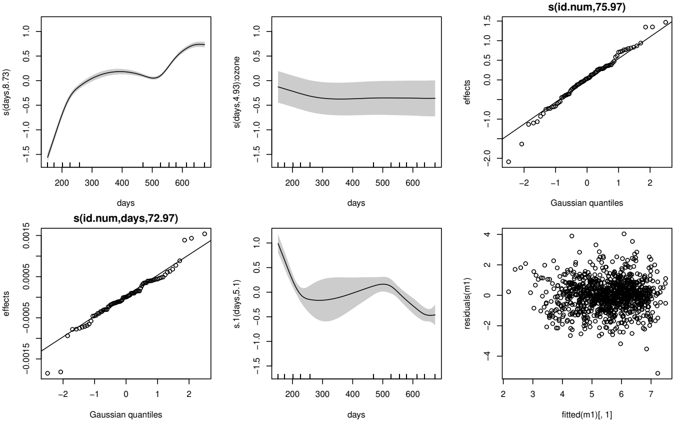
- AIC improves by about 180. Residual plots better.

```
> anova(m1)
...
Approximate significance of smooth terms:
                   edf Ref.df    Chi.sq  p-value
s(days)          8.733  8.955 2.239e+03  < 2e-16
s(days):ozone    4.933  5.752 2.751e+01 0.000106
s(id.num)       75.969 77.000 6.649e+06  < 2e-16
s(id.num,days)  72.971 77.000 1.675e+06  < 2e-16
s.1(days)        5.096  5.927 2.056e+02  < 2e-16
```

- Ozone effect significant (unlike if it’s a constant). Also, dropping it increases AIC by 17.

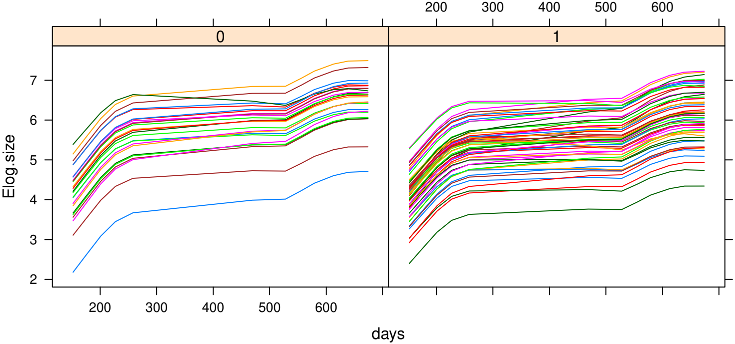
Example: Sitka model 2 effects

```
par(mfrow=c(2,3),mar=c(4,4,2,2)); plot(m1,scheme=1)
plot(fitted(m1)[,1],residuals(m1))
```



Example: Sitka model 2 predictions

```
sitka$Elog.size <- predict(m1)[,1]
xyplot(Elog.size~days|as.factor(ozone),data=sitka,type="l",
       groups=id.num)
```



Model checking introduction

- As for any regression, examine standardised residuals to check for mean-variance and independence assumption violations.
- Details of the distribution beyond these properties are often less important (consider quasi-likelihood theory), but problems may have some influence on smoothness selection. See `qq.gam`.
- Careful residual plotting can indicate what is missing in a model.
- Are the smooth basis dimensions overly restrictive? Must check!
  - EDF close to its upper limit (k', say) is *suspicious*.
  - Randomization test for residual pattern w.r.t. x_j: compare mean square difference between residuals for neighbouring x_j values to mean square difference between randomly selected residual pairs. Pattern *may* indicate oversmoothing because basis too small.
  - `gam.check` provides such checks, amongst others. e.g...

```
       k' edf k-index p-value
s(x0) 9.0 2.5    1.04    0.77
```

- See `gam.check`, `residuals`, `fitted` etc. for more.

Summary

- GAMs allow a response to depend on smooth functions of predictor variables.
- The smooth functions are represented using a basis expansion and quadratic smoothing penalty.
- Basis coefficients are estimated by penalized MLE.
- The quadratic penalties are equivalent to Gaussian priors on the coefficients, providing a Bayesian interpretation.
- Penalization implies a notion of effective degrees of freedom.
- The Bayesian approach provides useful confidence intervals, and an approach to smoothness estimation via marginal likelihood maximization.
- Model selection and checking are similar to any regression model (but check the basis dimensions).