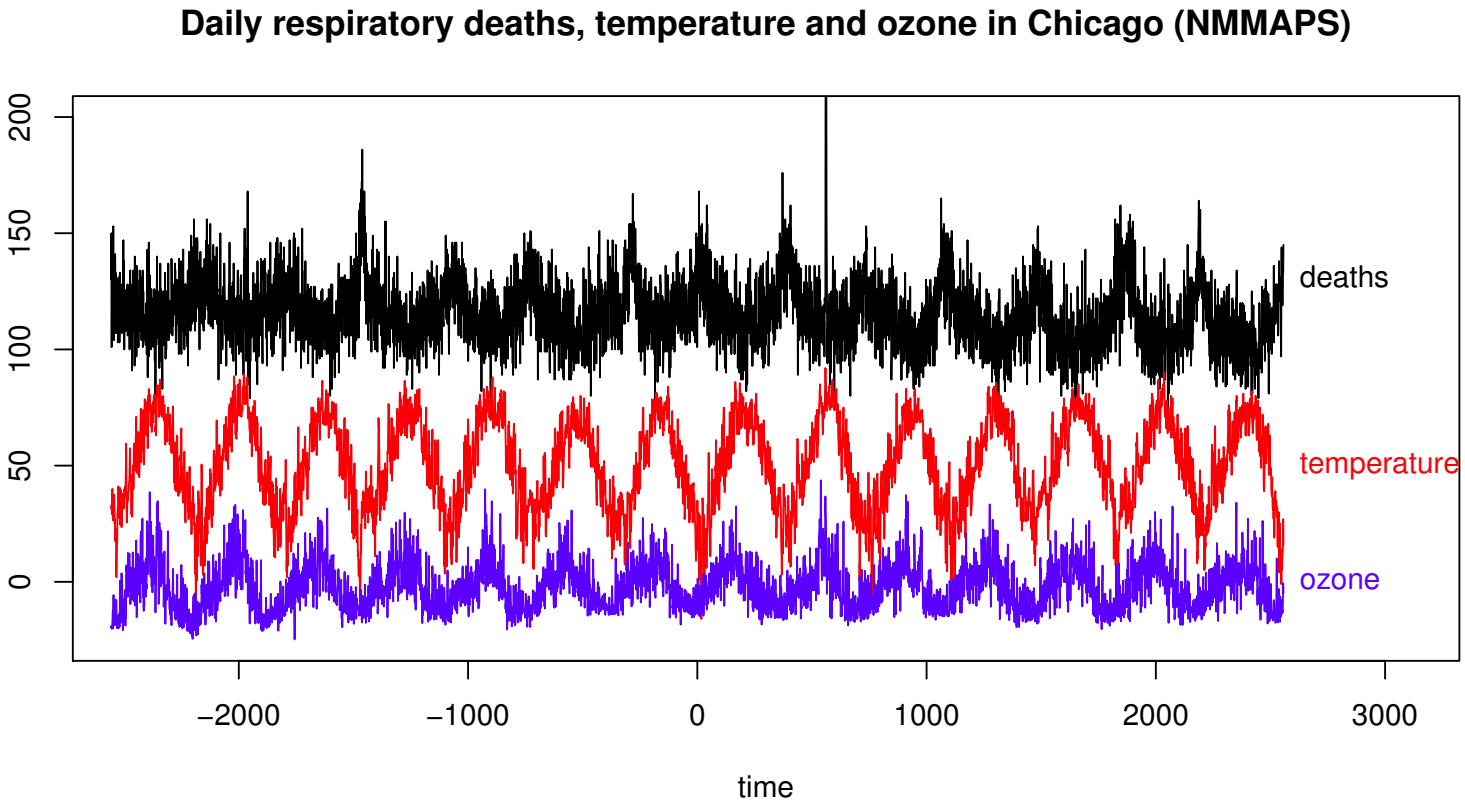
Modelling with smooth functions

Simon Wood University of Bath, EPSRC funded

Some data. . .



- Apparently daily respiratory deaths are ‘significantly’ negatively correlated with ozone and temperature.
- But obviously there is confounding here - we need a model.

A model...

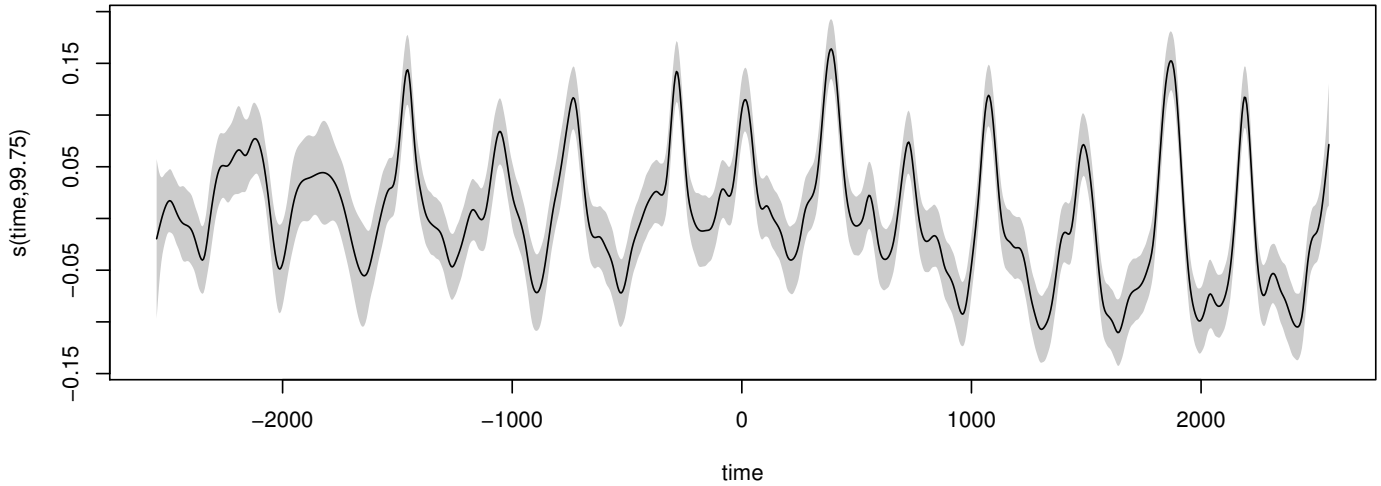
- $\texttt{death}_i \sim \text{Poi}(\mu_i)$

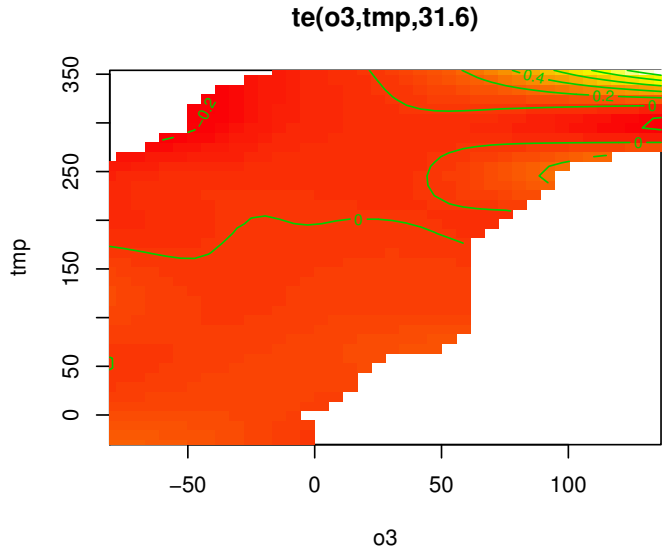
$$\log(\mu_i) = \alpha + f_1(t_i) + f_2(\texttt{o3}_i, \texttt{tmp}_i) + f_3(\texttt{pm10}_i)$$

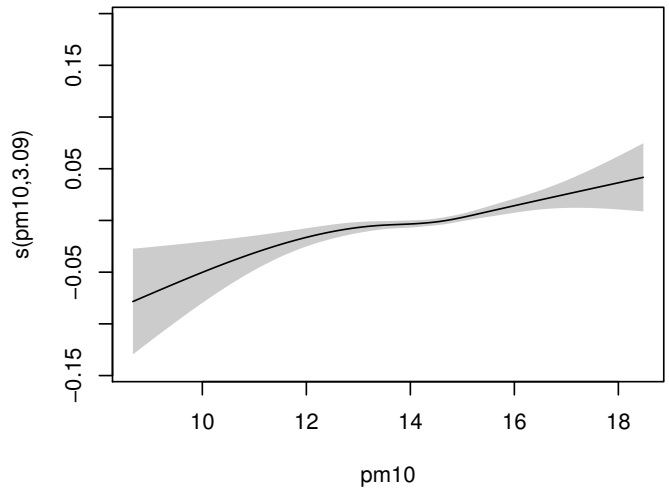
- The f_j are smooth functions to estimate.
  1. $\alpha + f_1$ is the (log) background respiratory mortality rate.
  2. f_2 is modification by ozone and temperature (interacting).
  3. f_3 is the modification from particulates.
- Actually the predictors are aggregated over the 3 days preceding death.
- Fit in R (`mgcv` package)

```
gam(death~s(time,k=200)+te(o3,tmp)+s(pm10),
    family=poisson)
```

Air pollution model estimates. . .

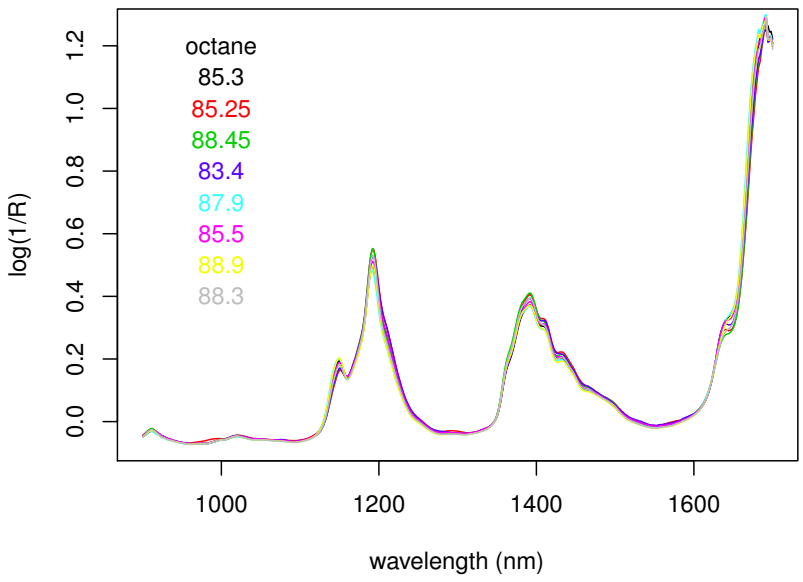






- High ozone and temp associated with increased risk.

Some more data...



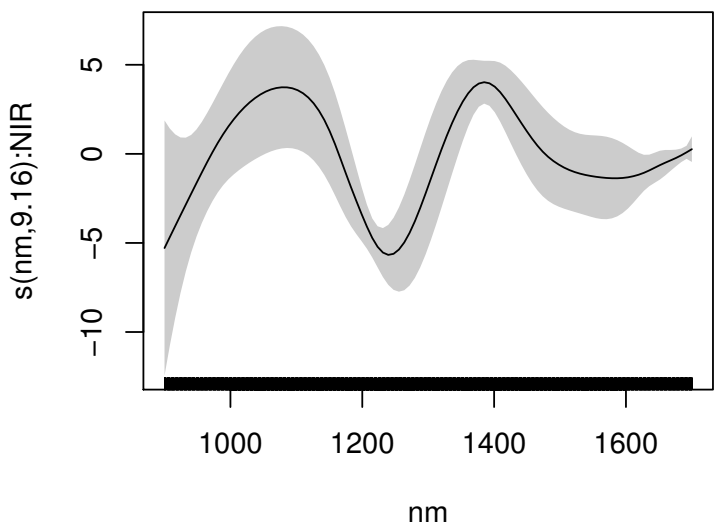
- Can we predict octane rating of fuel from near infra-red spectrum?
- 60 octane/spectrum pairs available (from `pls` package).

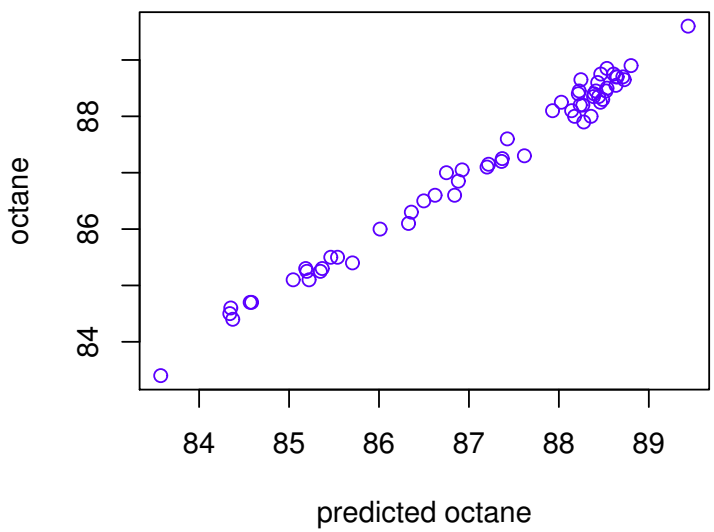
- Try a 'signal regression' model. k_i is i^{th} spectrum, f is a smooth function...

$$\texttt{octane}_i = \int f(\nu) k_i(\nu) d\nu + \epsilon_i$$

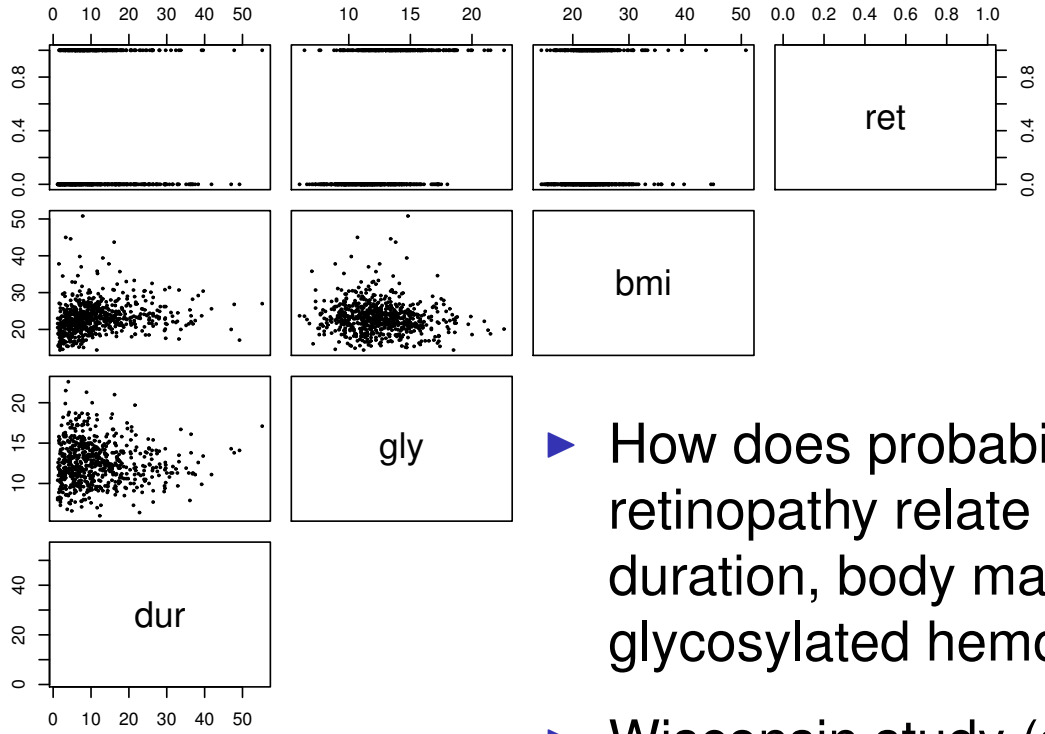
Signal regression model fit. . .

- If each row of matrix `NIR` contains a spectral intensity, and each row of `nm` the corresponding wavelengths, then `gam(octane~s(nm,by=NIR))` estimates the model.
- Plots of estimated $f(\nu)$ and fitted vs. data:





And one more dataset. . .

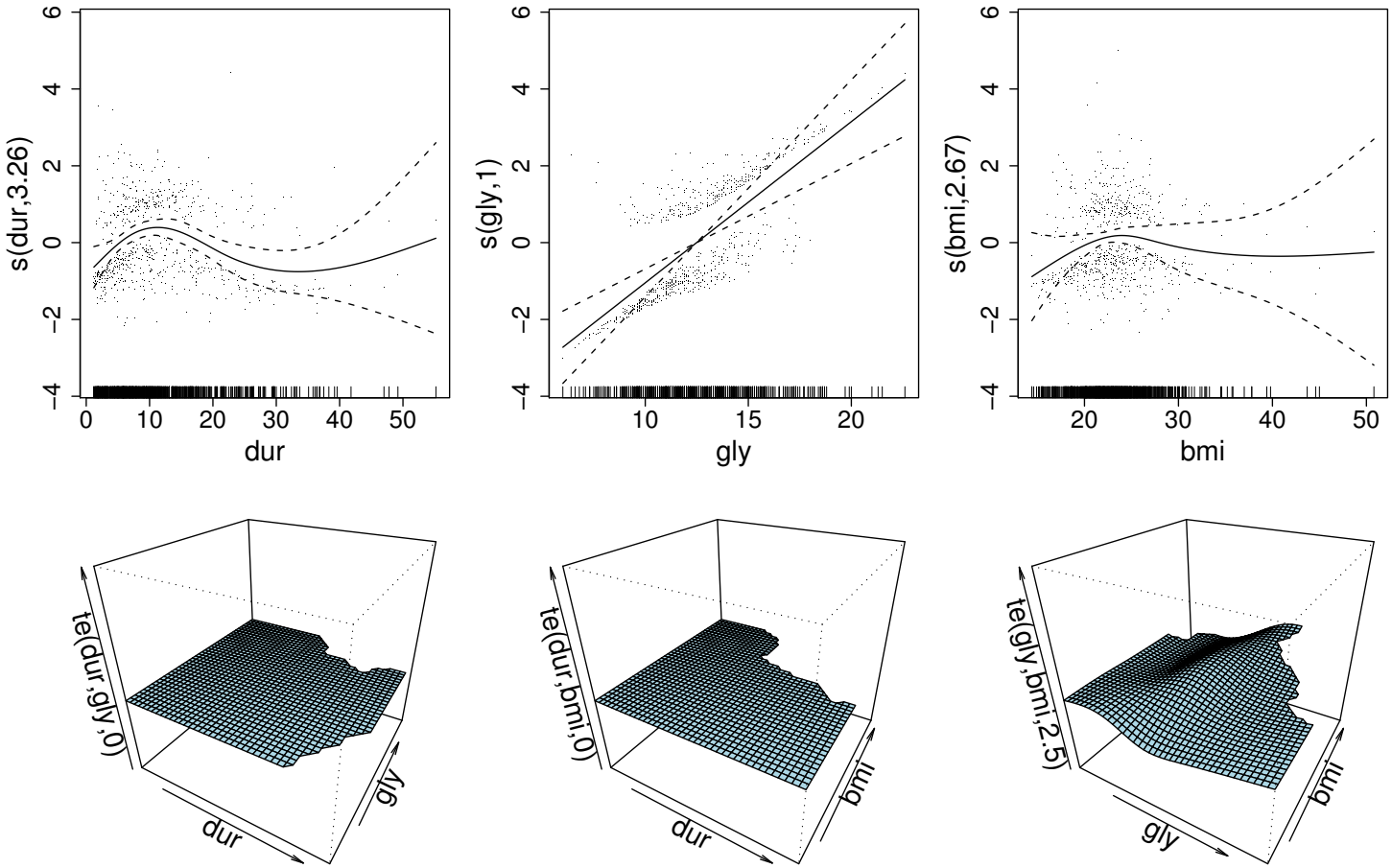


- How does probability of diabetic retinopathy relate to previous disease duration, body mass index and % glycosylated hemoglobin?
- Wisconsin study (see `gss` package).
- Model: $\texttt{ret}_i \sim$ Bernouilli,

$$\text{logit}\{\mathbb{E}(\texttt{ret})\} = f_1(\texttt{dur}) + f_2(\texttt{bmi}) + f_3(\texttt{gly}) + f_4(\texttt{dur},\texttt{bmi}) + f_5(\texttt{dur},\texttt{gly}) + f_6(\texttt{gly},\texttt{bmi})$$

Retinopathy model estimates

```
gam(ret ~ s(dur) + s(gly) + s(bmi) + ti(dur,gly) +
          ti(dur,bmi) + ti(gly,bmi), family=binomial)
```



Additive smooth models: structured flexibility

- The preceding are examples of *generalized additive models* (GAM).
- A GAM is a GLM, in which the linear predictor depends linearly on unknown smooth functions of covariates...

$$y_i \sim EF(\mu_i, \phi), \quad g(\mu_i) = f_1(x_{1i}) + f_2(x_{2i}) + \cdots$$

- Hastie and Tibshirani (1990) and Wahba (1990) laid the foundations for these models as discussed here.
- If we can work with GAMs at all we can easily generalize:

$$g(\mu_i) = \mathbf{A}_i\theta + \sum_j L_{ij} f_j(x_j) + \mathbf{Z}_i\mathbf{b}$$

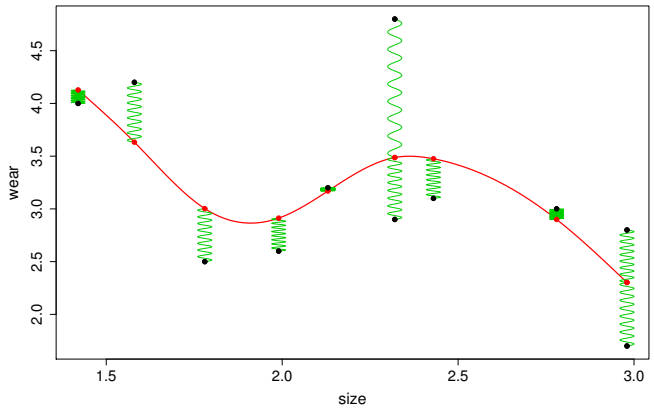
— L_{ij} are linear functionals; $\mathbf{A}$ & $\mathbf{Z}$ are model matrices; θ and $\mathbf{b}$ are parameters and Gaussian random effects.

Additive smooth models & practical computation

- Making GAMs work in practice requires at least 3 things...
  1. A way of representing the smooth functions f_j.
  2. A way of estimating the f_j.
  3. Some means of deciding *how smooth* the f_j should be.
- 3 is the awkward part of the enterprise, and strongly influences 1 and 2.
- As well as point estimates we also need further inferential tools, such as interval estimates, model selection methods, AIC, p-values etc.
- We'd also like to go beyond univariate exponential family.
- This talk will cover these things in order.

Representing smooth functions: splines

- To motivate how to represent several smooth terms in a model, first consider a simpler smoothing problem.
- Consider estimating the smooth function f in the model $y_i = f(x_i) + \epsilon_i$ from x_i, y_i data using *smoothing splines*...

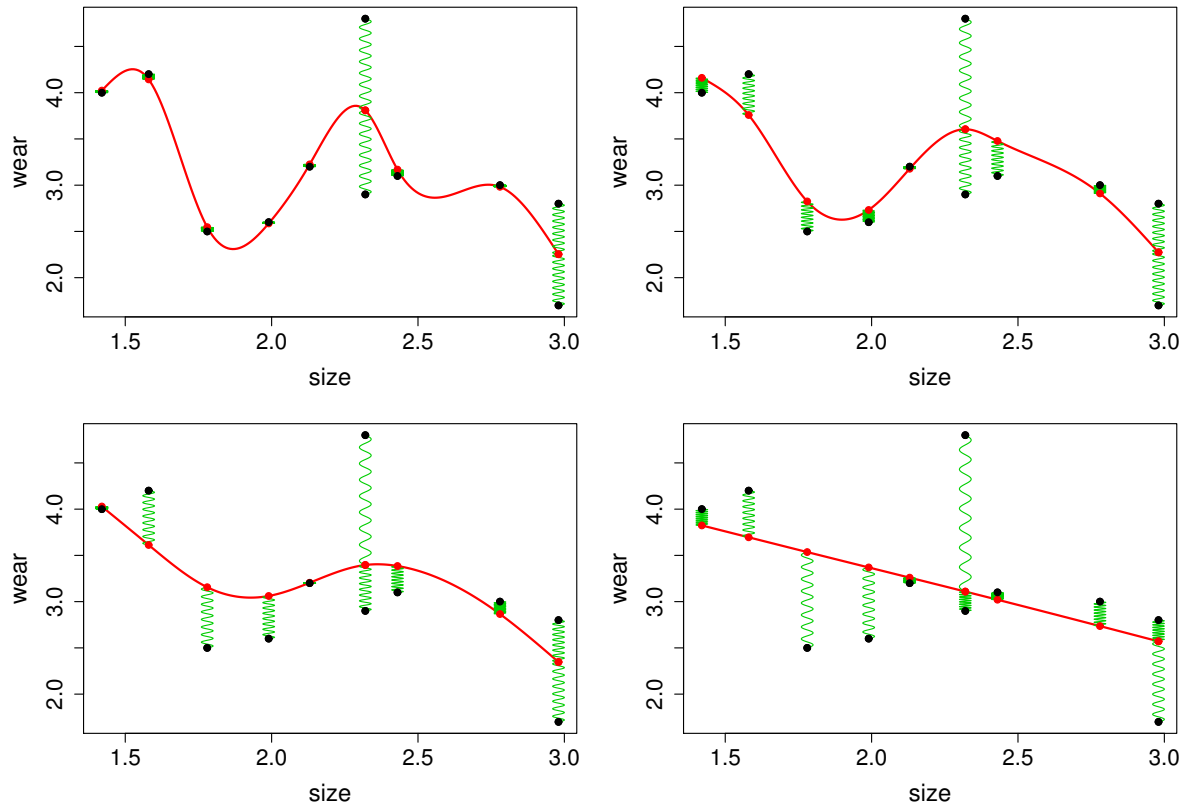


- The red curve is the *function* minimizing

$$\sum_i (y_i - f(x_i))^2 + \lambda \int f''(x)^2 dx.$$

Splines and the smoothing parameter

- Smoothing parameter λ controls the stiffness of the spline.



- But the spline can be written $\hat{f}(x) = \sum_i \beta_i b_i(x)$, where the basis functions $b_i(x)$ do not depend on λ.

General spline theory background

- Consider a Hilbert space of real valued functions, f, on some domain τ (e.g. $[0, 1]$).
- It is a *reproducing kernel Hilbert space*, $\mathcal{H}$, if evaluation is bounded. i.e. $\exists M$ s.t. $|f(t)| \leq M\|f\|$.
- Then the Riesz representation thm says that there is a function $R_t \in \mathcal{H}$ s.t. $f(t) = \langle R_t, f \rangle$.
- Now consider $R_t(u)$ as a function of t: $R(t, u)$

$$\langle R_t, R_s \rangle = R(t, s)$$

  — so $R(t, s)$ is known as *reproducing kernel* of $\mathcal{H}$.
- Actually, to every positive definite function $R(t, s)$ corresponds a unique r.k.h.s.

Spline smoothing problem

- RKHS are quite useful for constructing smooth models, to see why consider finding $\hat{f}$ to minimize

$$\sum_i \{y_i - f(t_i)\}^2 + \lambda \int f''(t)^2 dt.$$

- Let $\mathcal{H}$ have $\langle f, g \rangle = \int g''(t) f''(t) dt$.
- Let $\mathcal{H}_0$ denote the RKHS of functions for which $\int f''(t)^2 dt = 0$, with basis $\phi_1(t) = 1,\ \phi_2(t) = t$, say.
- Spline problem seeks $\hat{f} \in \mathcal{H}_0 \oplus \mathcal{H}$ to minimize

$$\sum_i \{y_i - f(t_i)\}^2 + \lambda \|Pf\|^2.$$

Spline smoothing solution

- $\hat{f}(t) = \sum_{i=1}^{n} c_i R_{t_i}(t) + \sum_{i=1}^{M} d_i \phi_i(t)$ is the basis representation of $\hat{f}$. Why?
- Suppose minimizer were $\tilde{f} = \hat{f} + \eta$ where $\eta \in \mathcal{H}$ and $\eta \perp \hat{f}$:
  1. $\eta(t_i) = \langle R_{t_i}, \eta \rangle = 0$.
  2. $\|P\tilde{f}\|^2 = \|P\hat{f}\|^2 + \|\eta\|^2$ which is minimized when $\eta = 0$.
- . . . obviously this argument is rather general.
- So if $E_{ij} = \langle R_{t_i}, R_{t_j} \rangle$ and $T_{ij} = \phi_j(t_i)$ then we seek $\hat{c}$ and $\hat{d}$ to minimize

$$\|y - Td - Ec\|_2^2 + \lambda c^{\mathrm{T}} Ec.$$

. . . straightforward to compute (but at $O(n^3)$ cost).

Reduced rank smoothers

- Can obtain efficient reduced rank basis (and penalty) by
  1. using spline basis for a 'representative' subset of data, or
  2. using Lanczos methods to find a low order 'eigenbasis'.
- In either case we end up representing the smoother as

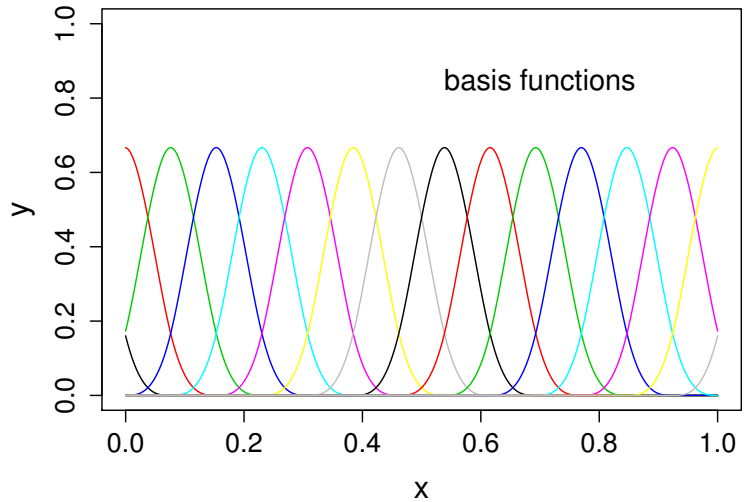
$$f(x) = \sum_j \beta_j b_j(x)$$

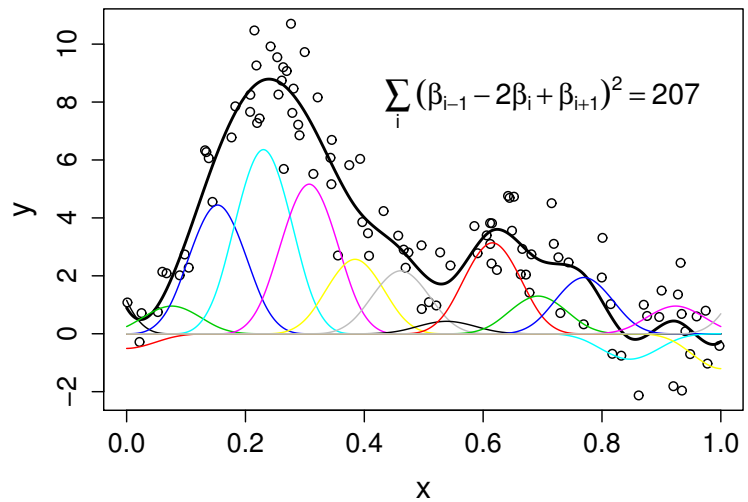
  — basis functions, $b_j(x)$ known; coefficients β not.
- Corresponding *smoothing penalty* is $\beta^{\mathrm{T}}\mathbf{S}\beta$. $\mathbf{S}$ is known.
- So $y_i = f(x_i) + \epsilon_i$ becomes $\mathbf{y} = \mathbf{X}\beta + \epsilon$ where $X_{ij} = b_j(x_i)$ and $\hat{\beta} = \mathrm{argmin}_\beta \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda\beta^{\mathrm{T}}\mathbf{S}\beta$.
- Examples follow asymptotic justification of rank reduction.

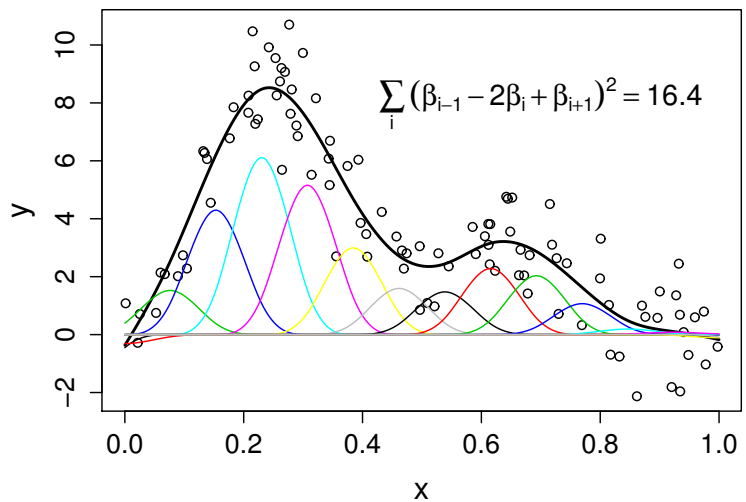
What rank?

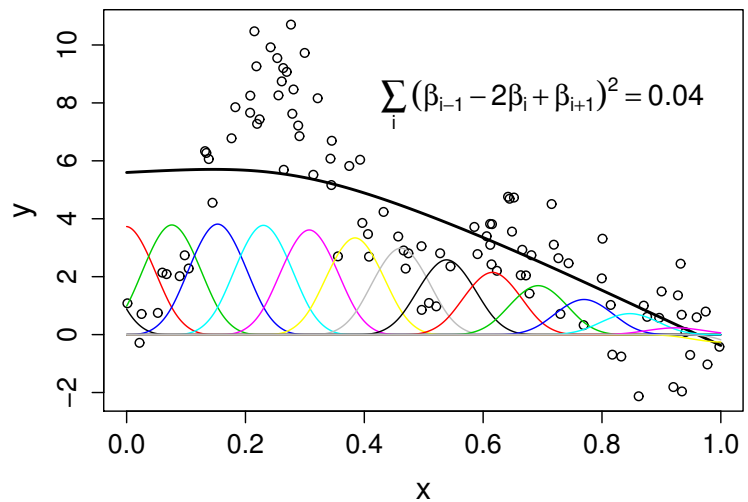
- Consider, f, a rank k cubic regression spline (i.e. $\lambda = 0$), parameterized in terms of function values at evenly spaced 'knots'.
- From basic regression theory, average sampling standard error of $\hat{f}$ is $O(\sqrt{k/n})$.
- From approximation theory for cubic splines, asymptotic bias of $\hat{f}$ is bounded by $O(k^{-4})$.
- So if we let $k = O(n^{1/9})$ we minimize the asymptotic MSE at $O(n^{-8/9})$.
- Actually in practice we would want to penalize and then $k = O(n^{1/5})$ is appropriate.
- Point is that asymptotically $k \ll n$ is appropriate.

P-splines: B-spline basis & approx penalty







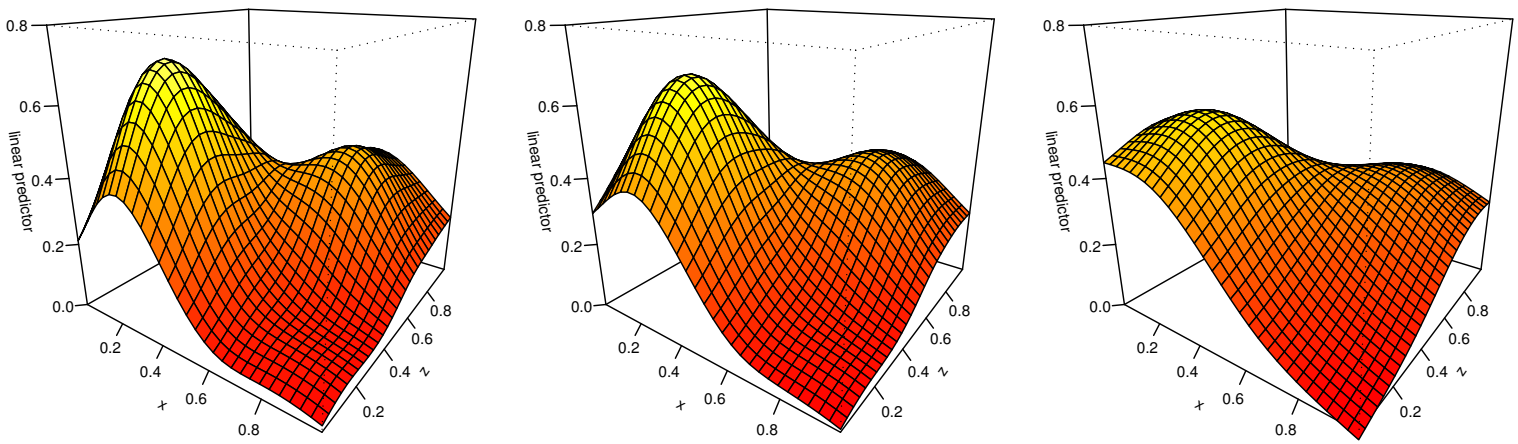


Example: Thin plate splines

- One way of generalizing splines from 1D to several D is to turn the flexible strip into a flexible sheet $\hat{f}$ minimizing e.g.

$$\sum_i \{y_i - f(x_i, z_i)\}^2 + \lambda \int f_{xx}^2 + 2f_{xz}^2 + f_{zz}^2 dxdz$$

- This results in a *thin plate spline*. It is an *isotropic* smooth.
- Isotropy may be appropriate when different covariates are naturally on the same scale.

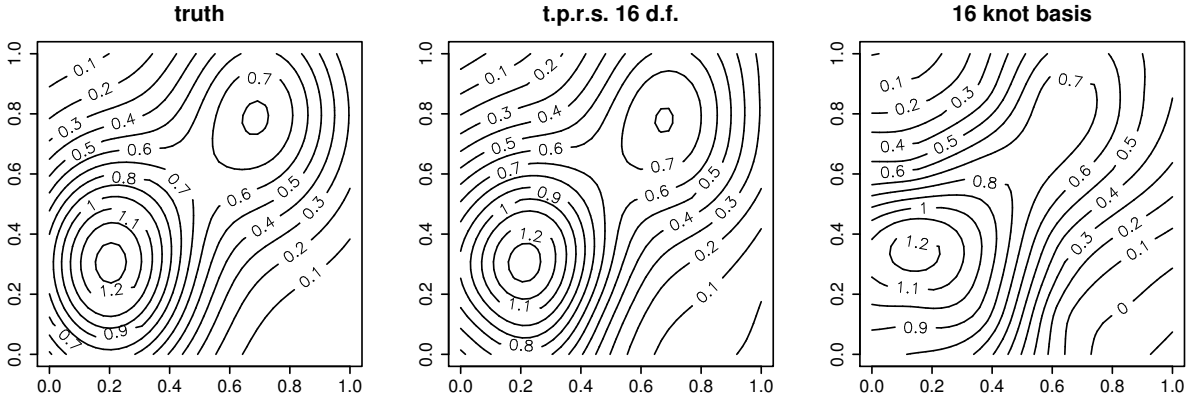


Cheaper thin plate splines

- RKHS or similar theory gives explicit basis (for any dimension), with coefficients ($\mathbf{c}$ and $\mathbf{d}$) minimizing

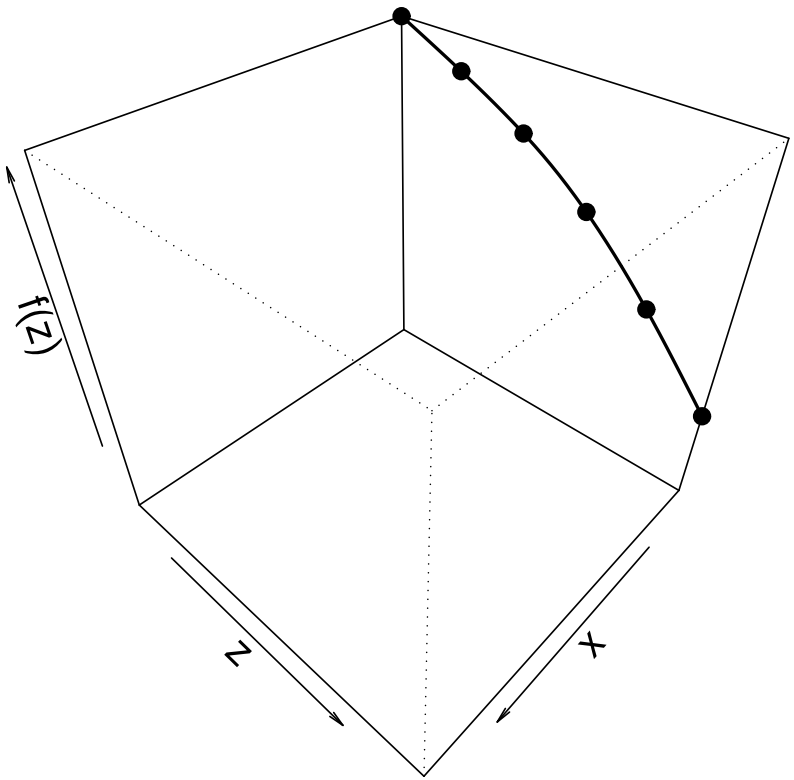
$$\|y - \mathbf{Td} - \mathbf{Ec}\|^2 + \lambda \mathbf{c}^{\mathrm{T}}\mathbf{Ec} \quad s.t. \quad \mathbf{T}^{\mathrm{T}}\mathbf{c} = \mathbf{0}.$$

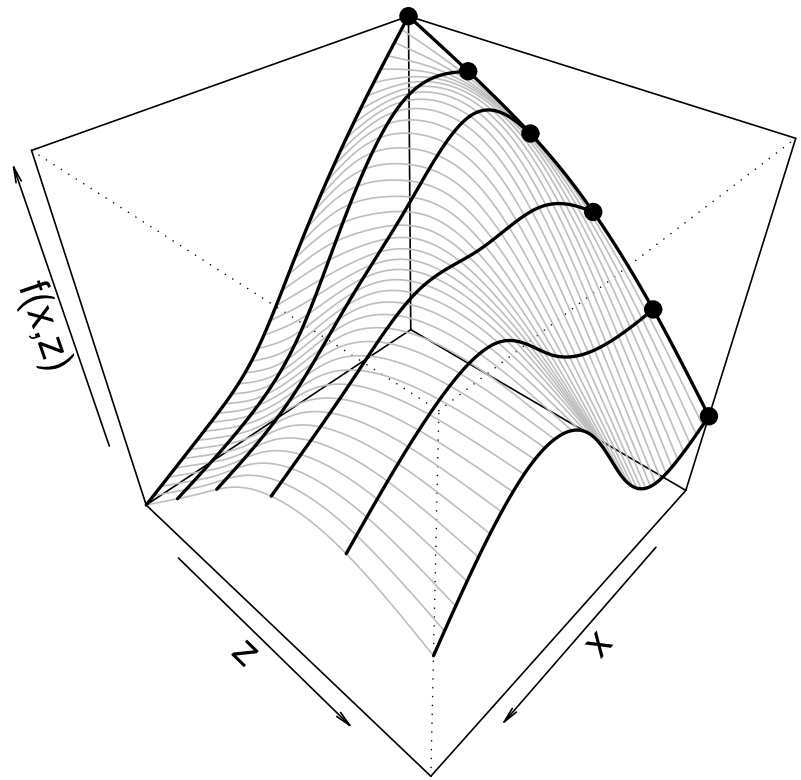
- Can reduce $O(n^3)$ cost to $O(k^3)$ by replacing $\mathbf{E}$ by its rank k truncated eigen-decomposition (computed by Lanczos methods).
- Cheap and somewhat optimal...



Non-isotropic tensor product smooths

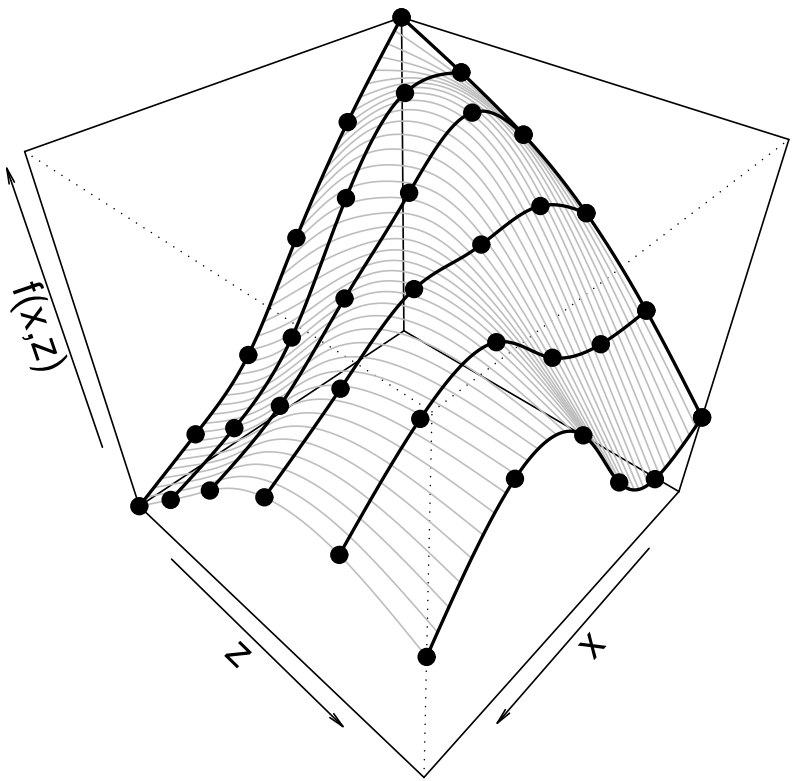
- A different construction is needed if covariates have different scales.
- Start with marginal spline f_z and let its coefficients vary smoothly with x

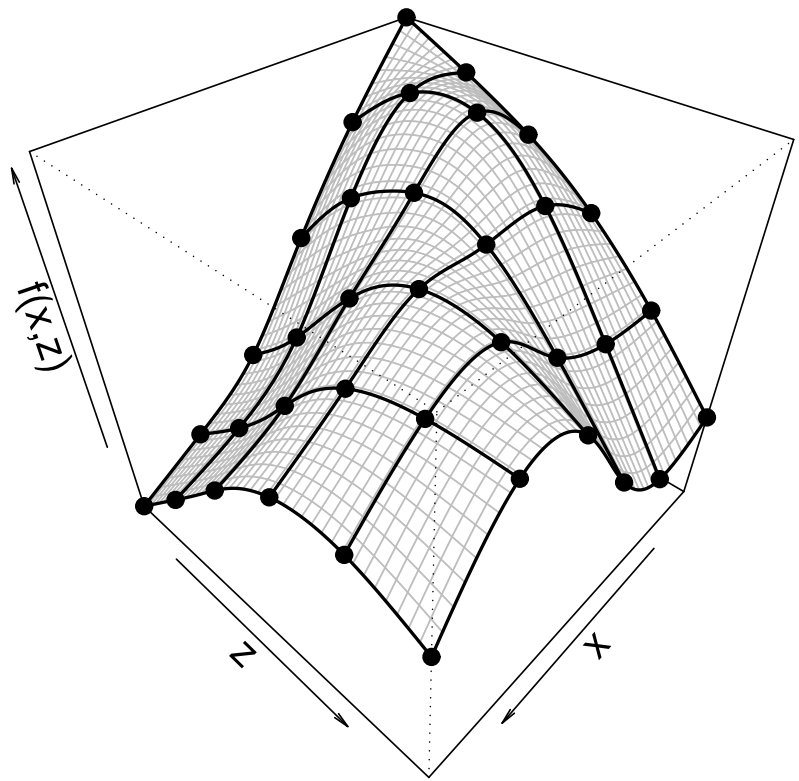




The complete tensor product smooth

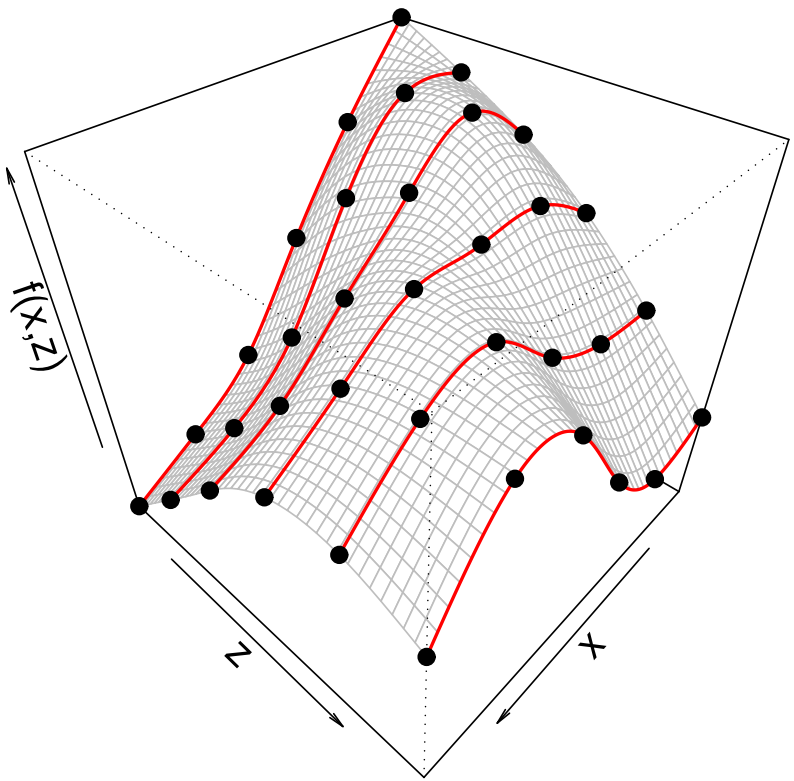
- Let each coefficient of f_z be a spline of x.
- Construct is symmetric (see right).

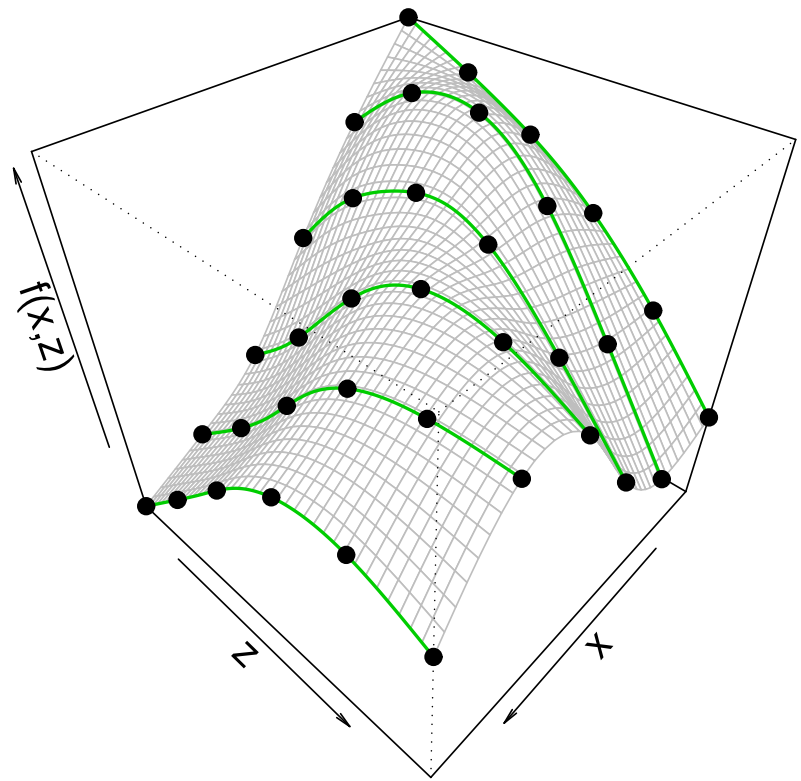




Tensor product penalties - one per dimension

- x-wiggliness: sum marginal x penalties over red curves.
- z-wiggliness: sum marginal z penalties over green curves.





Tensor product expressions

- Suppose the basis expansions for smoothing w.r.t. x and z *marginally* are $\sum_j \mathcal{B}_j b_j(z)$ and $\sum_i \alpha_i a_i(x)$.
- . . . and the marginal penalties are $\boldsymbol{\mathcal{B}}^{\mathrm{T}}\mathbf{S}_z\boldsymbol{\mathcal{B}}$ and $\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{S}_x\boldsymbol{\alpha}$.
- The tensor product basis construction gives:

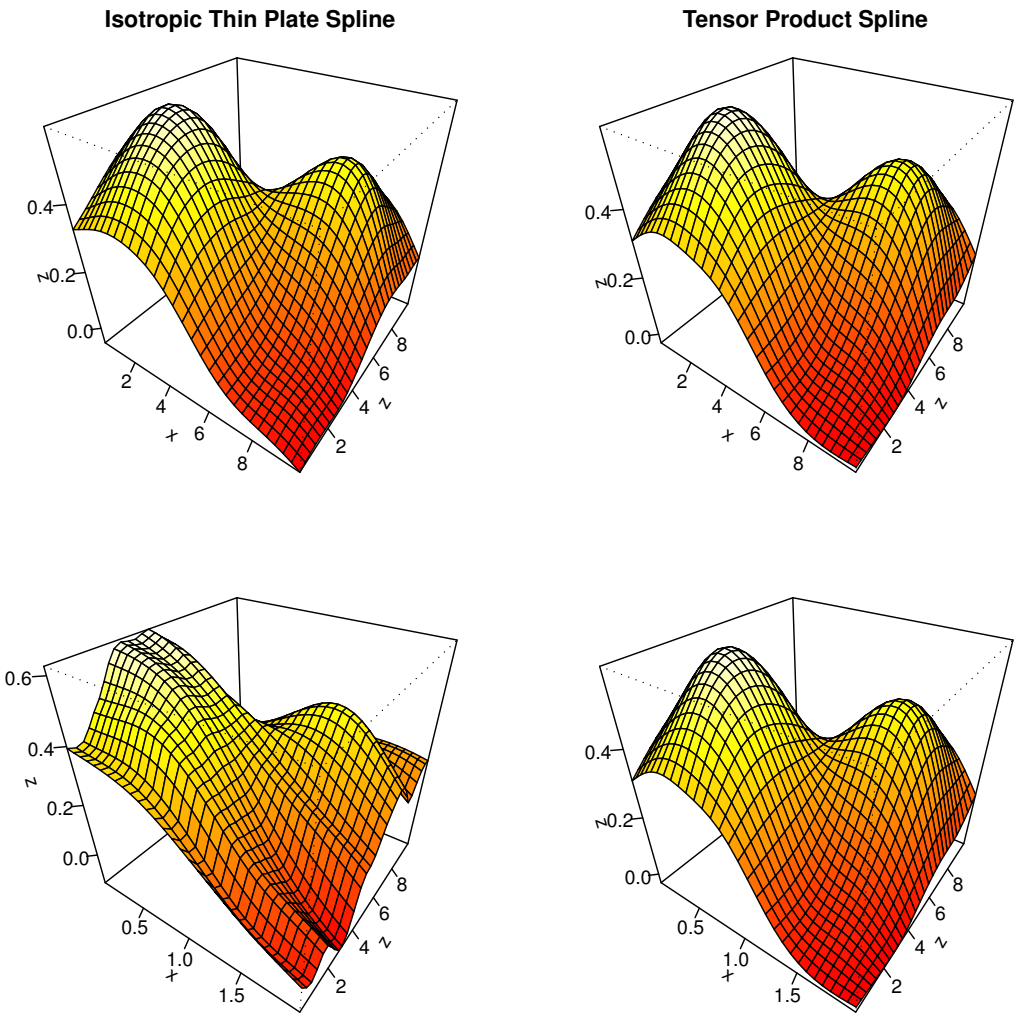
$$f(x,z) = \sum\sum \beta_{ij} b_j(z) a_i(x)$$

- With two penalties (requiring 2 smoothing parameters)

$$J_z^*(f) = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{I}_I \otimes \mathbf{S}_z\boldsymbol{\beta} \text{ and } J_x^*(f) = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{S}_x \otimes \mathbf{I}_J\boldsymbol{\beta}$$

- The construction generalizes to any number of marginals and multi-dimensional marginals.

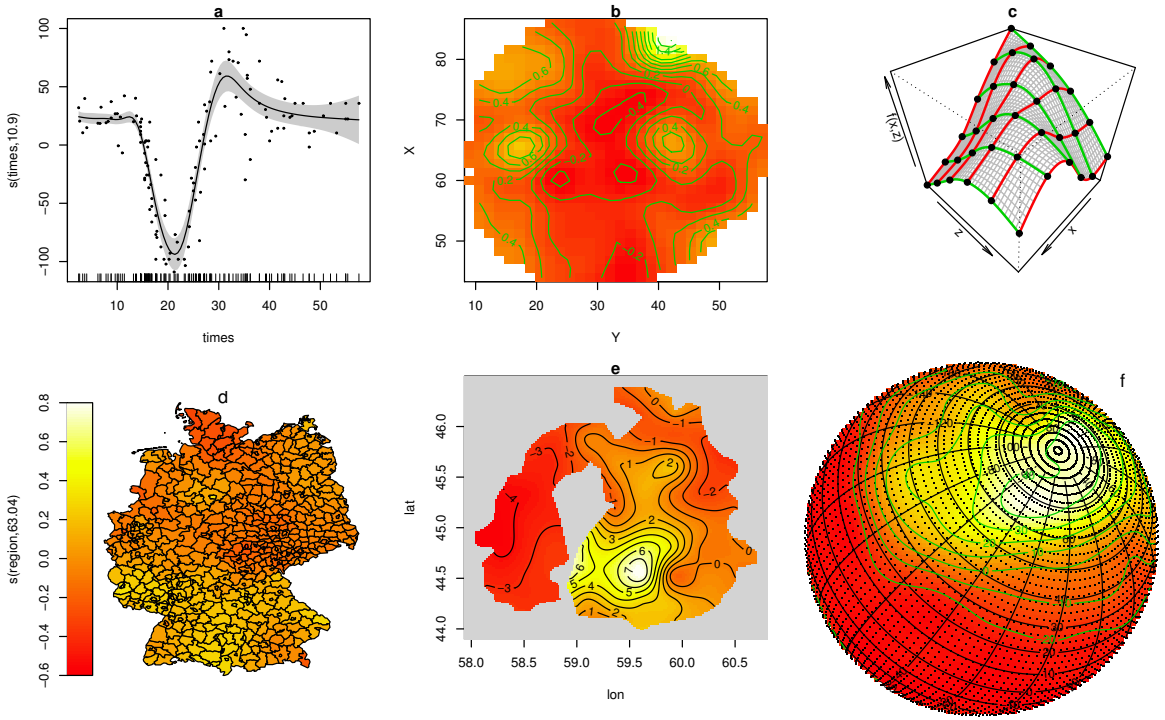
Isotropic vs. tensor product comparison



. . . each figure smooths the same data. The only modification is that x has been divided by 5 in the bottom row.

Smooth model components

- A rich range of basis-penalty smoothers is possible. . .



- When combining several smooths in one model, we often need an identifiability constraint $\sum_i f(x_i) = 0$. Re-write as $\mathbf{1}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta} = 0$ and absorb by reparameterization.

Representing a GAM

- Consider the GAM $y_i \sim \text{EF}(\mu_i, \phi)$, $g(\mu_i) = \beta_0 + \sum_j f_j(x_{ji})$.
- Each $f_j(x)$ has a basis - penalty representation, say $\mathbf{X}^j\boldsymbol{\beta}^j$, $\boldsymbol{\beta}^{j\mathrm{T}}\mathbf{S}^j\boldsymbol{\beta}^j$ (with constraints absorbed).
- So the GAM becomes

$$g(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\beta},$$

  where $\mathbf{X} = [\mathbf{1} : \mathbf{X}^1 : \mathbf{X}^2 : \ldots]$ and $\boldsymbol{\beta}^{\mathrm{T}} = [\beta_0, \boldsymbol{\beta}^{1\mathrm{T}}, \boldsymbol{\beta}^{2\mathrm{T}}, \ldots]$.
- Penalty is now

$$\sum \lambda_j \boldsymbol{\beta}^{\mathrm{T}}\mathbf{S}_j\boldsymbol{\beta},$$

  where $\mathbf{S}_j$ is such that $\boldsymbol{\beta}^{\mathrm{T}}\mathbf{S}_j\boldsymbol{\beta} \equiv \boldsymbol{\beta}^{j\mathrm{T}}\mathbf{S}^j\boldsymbol{\beta}^j$.
- Can easily include parametric components and linear functionals of smooths.

$\hat{\beta}$ given λ

- Let l be the model log-likelihood and use penalized MLE

$$\mathcal{L}(\beta) = l(\beta) - \frac{1}{2}\sum_j \lambda_j \beta^{\mathrm{T}} \mathbf{S}_j \beta, \quad \hat{\beta} = \underset{\beta}{\text{argmax}} \ \mathcal{L}(\beta)$$

- Optimize by Newton's method (penalized IRLS for EF).
- Bayes motivation. Prior: $\pi(\beta) = N(\mathbf{0}, (\sum \lambda_j \mathbf{S}_j)^-)$; likelihood, $\pi(\mathbf{y}|\beta)$: as is. Then MAP estimate is $\hat{\beta}$.
- Further, in large sample limit, if $\mathcal{I}$ is information matrix at $\hat{\beta}$,

$$\beta|\mathbf{y} \sim N(\hat{\beta}, (\mathcal{I} + \sum \lambda_j \mathbf{S}_j)^{-1})$$

  which can be used to construct well calibrated CIs.
- Note link to Gaussian random effects!

AIC, effective degrees of freedom, p-values

- Following through the derivation of AIC in the penalized case, yields

$$\text{AIC} = -2l(\hat{\beta}) + 2\text{tr}(\mathbf{F})$$

  where $\mathbf{F} = (\mathcal{I} + \sum \lambda_j \mathbf{S}_j)^{-1}\mathcal{I}$.
- Better still $\mathbf{F} = \mathbf{V}_\beta \mathcal{I}$, where $\mathbf{V}_\beta$ is an approximate posterior covariance matrix for β, corrected for λ uncertainty (see Greven and Kneib, 2010, Biometrika for why).
- tr($\mathbf{F}$) is the *effective degrees of freedom* of the model.
- p-values for testing smooth terms or variance components for equality to zero can also be obtained (see Wood 2013a,b Biometrika).
- But we still need smoothing parameter (λ) estimates.

Smoothing parameter estimates

- Let $\boldsymbol{\rho} = \log \boldsymbol{\lambda}$ and $\pi(\boldsymbol{\rho}) =$ constant, then the *marginal likelihood* is

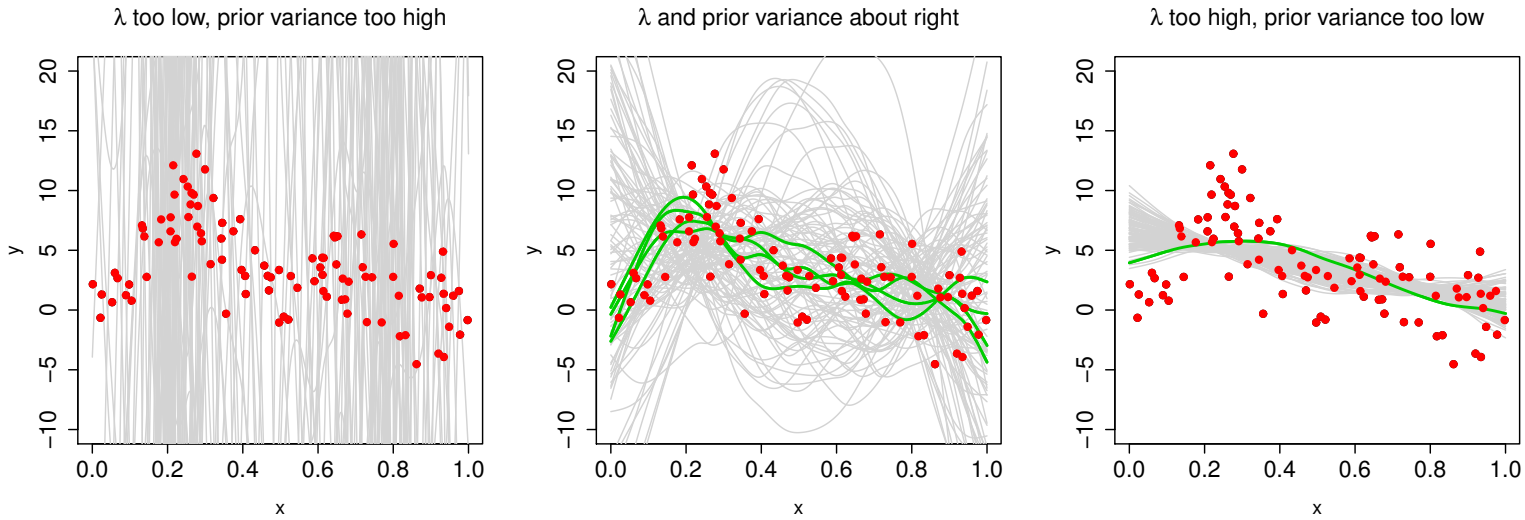
$$\pi(\boldsymbol{\rho}|\mathbf{y}) = \int \pi(\mathbf{y}|\boldsymbol{\beta})\pi(\boldsymbol{\beta}|\boldsymbol{\rho})d\boldsymbol{\beta}$$

- Empirical Bayes: $\hat{\boldsymbol{\rho}} = \text{argmax}_{\lambda}\pi(\boldsymbol{\rho}|\mathbf{y})$.
- . . . same as REML in Gaussian random effects context.
- In practice use a Laplace approximation for the integral.

$$\log \pi(\boldsymbol{\rho}|\mathbf{y}) \simeq -\mathcal{L}(\hat{\boldsymbol{\beta}}) - \frac{1}{2}\log|\sum \lambda_j \mathbf{S}_j|_+ + \frac{1}{2}\log|\boldsymbol{\mathcal{H}}| + c$$

$\boldsymbol{\mathcal{H}}$ is Hessian of $-\mathcal{L}$. Log $|\bullet|$ require numerical care.

How marginal likelihood works



- Draw β from prior implied by λ. Find average value of likelihood for these draws.
- Choose λ to maximize this average likelihood.
- i.e. formally, maximize $\int \pi(\mathbf{y}|\beta)\pi(\beta|\lambda)d\beta$ w.r.t. λ.
- Cross validation is an alternative.

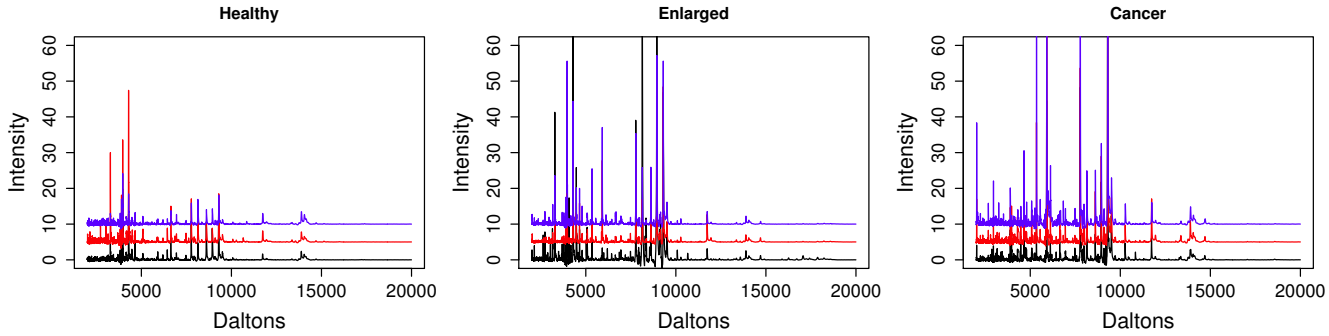
Numerical fitting strategy

- Optimize REML w.r.t. $\boldsymbol{\rho} = \log(\boldsymbol{\lambda})$ by Newton's method. For each trial $\boldsymbol{\rho}$...
  1. Re-parameterize $\boldsymbol{\beta}$ so that $\log|\mathbf{S}|_+$ computation is stable.
  2. Find $\hat{\boldsymbol{\beta}}$ by an inner Newton optimization of $\mathcal{L}(\boldsymbol{\beta})$.
  3. Use implicit differentiation to find first two derivatives of $\hat{\boldsymbol{\beta}}$ w.r.t. $\boldsymbol{\rho}$.
  4. Compute log determinant terms and their derivatives.
  5. Hence compute the REML score and its first two derivatives, as required for the next Newton step.
- For large datasets an alternative is often possible: Find $\hat{\boldsymbol{\beta}}$ by iterative fitting of working linear models, and estimate $\boldsymbol{\rho}$ for the working model at each iteration step.

Where's the exponential family assumption?

- . . . the single parameter independent EF assumption of GAMs is barely used.
- It just simplifies some of the numerical computations (and adds to numerical robustness).
- Actually we can use most of the apparatus just described with almost any regular likelihood, provided its practical to compute with, and the first three or four derivatives w.r.t. to the model coefficients can also be computed . . .

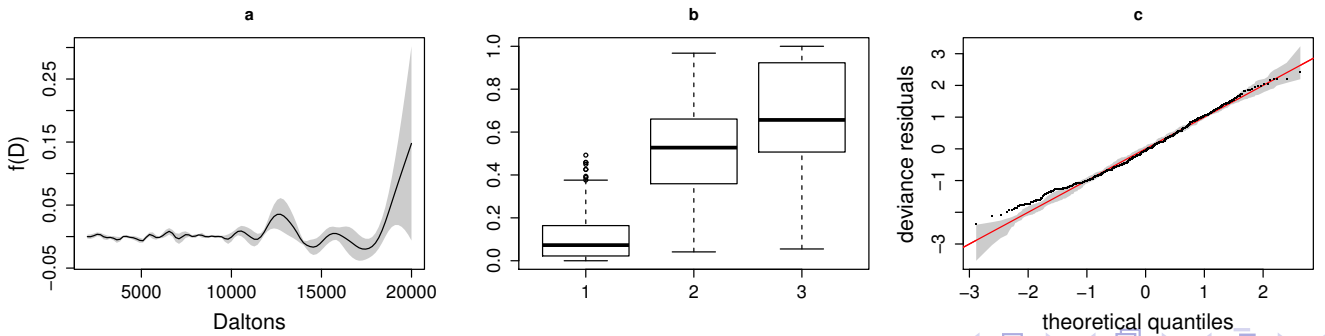
Example: predicting prostate status



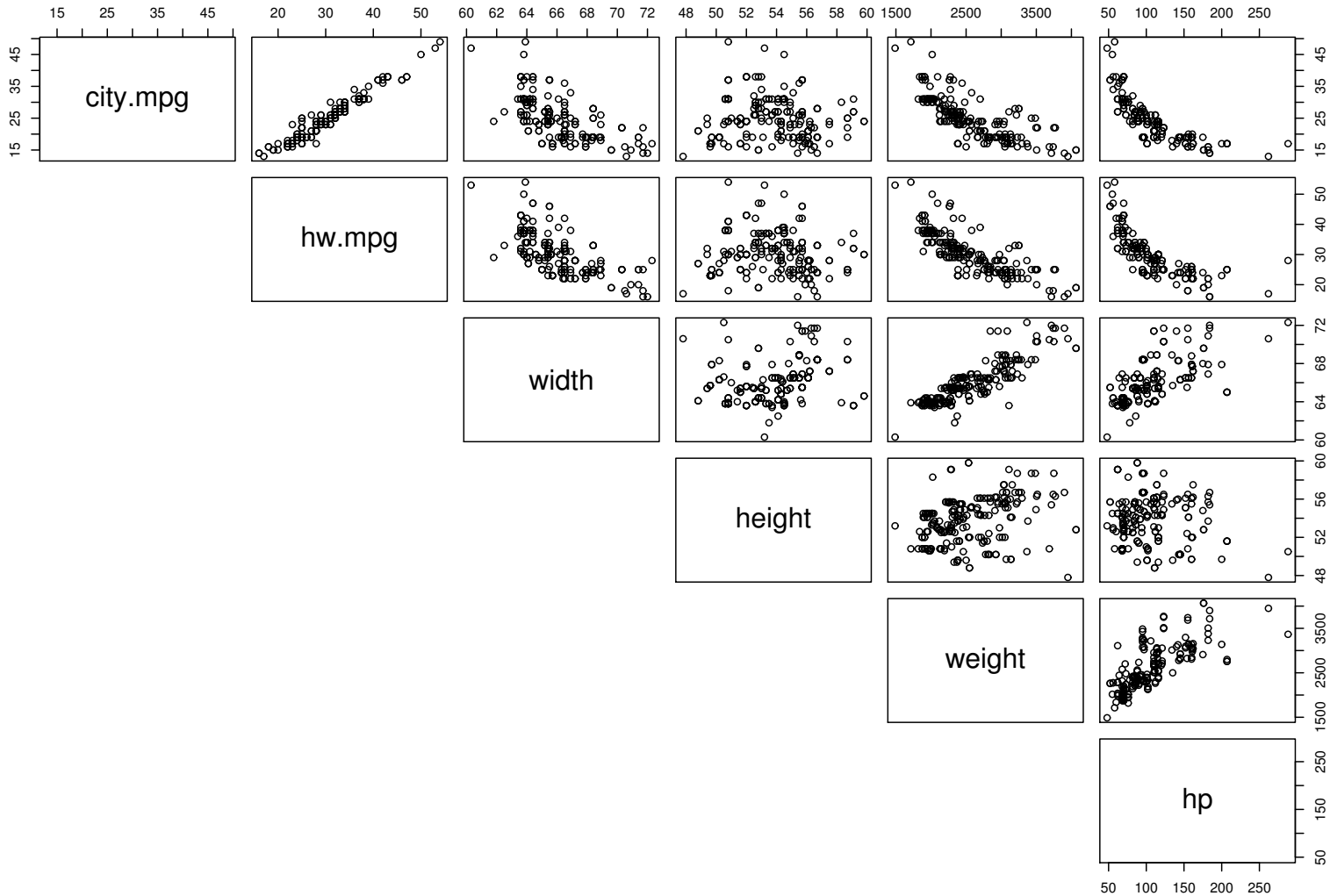
- Model: ordered category (benign/enlarged/cancer) predicted by logistic latent random variable with mean

$$\mu_i = \int f(D)\nu_i(D)dD, \quad \nu_i(D) \text{ is } i^{\text{th}} \text{ spectrum.}$$

```
gam(ds~s(D,by=K),family=ocat(R=3))
```

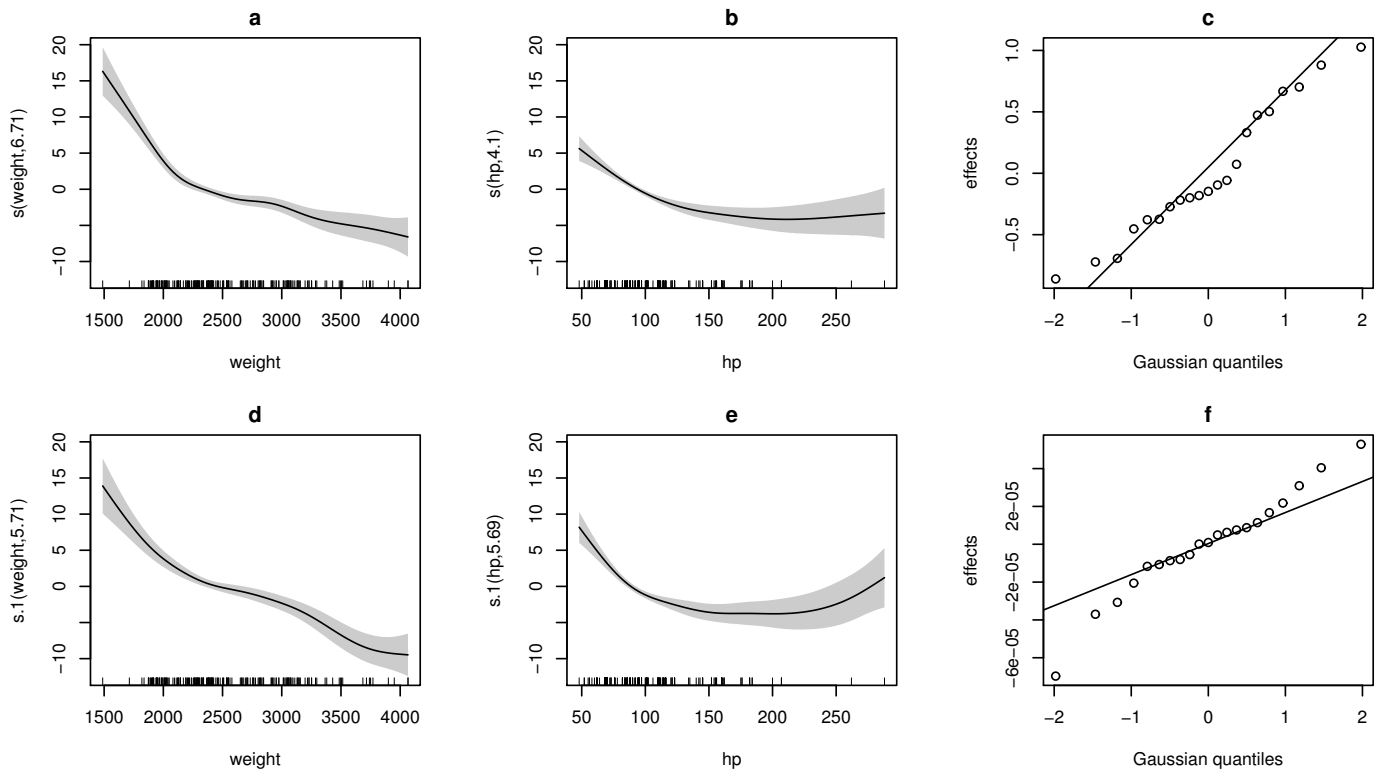


Fuel efficiency of cars



Multivariate additive model

- Correlated bivariate normal response (`hw.mpg`, `city.mpg`).
- Component means given by smooth additive predictors. Best model very simple (and somewhat unexpected)

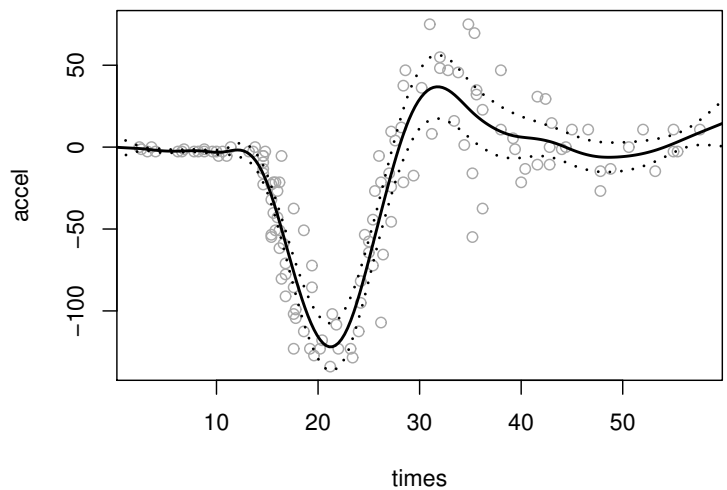


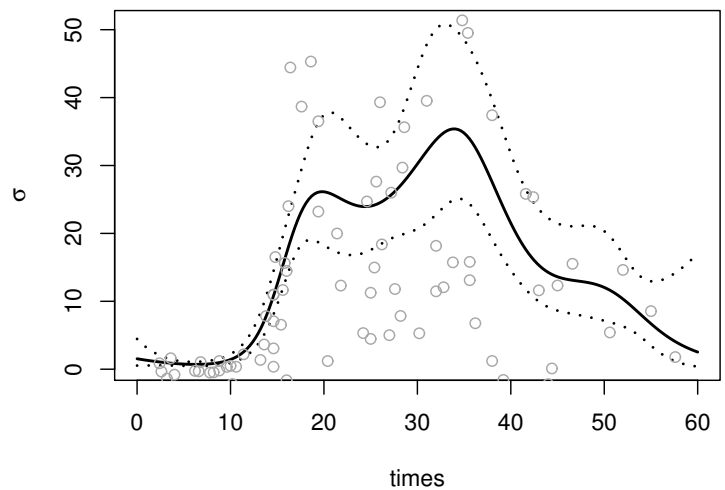
Scale location models

- Can additively model mean and variance (and skew and...)
- Simple example: $y_i \sim N(\mu_i, \sigma_i)$

$$\mu_i = \sum_j f_j(x_{ji}), \qquad \log \sigma_i = \sum_j g_j(z_{ji}).$$

- Here is a simple 1-D smoothing example of this...





R packages

There are many alternative R packages available:

1. `gam` for original backfitting approach[1].
2. `vgam` for vector GAMs and more[1].
3. `gamlss` GAMs for location scale and shape.[2]
4. `mboost` GAMs via boosting.
5. `gss` Smoothing Spline ANOVA.
6. `scam` Shaped constrained additive models.
7. `gamm4` GAMMs using `lme4`.
8. `bayesx` MCMC and likelihood based GAMs[3].
9. Methods discussed here are in R recommended package `mgcv`...

[1] No smoothing parameter selection

[2] Limited smoothing parameter selection

[3] see also `mgcv::jagam`

mgcv package in R