The Basics of P-splines

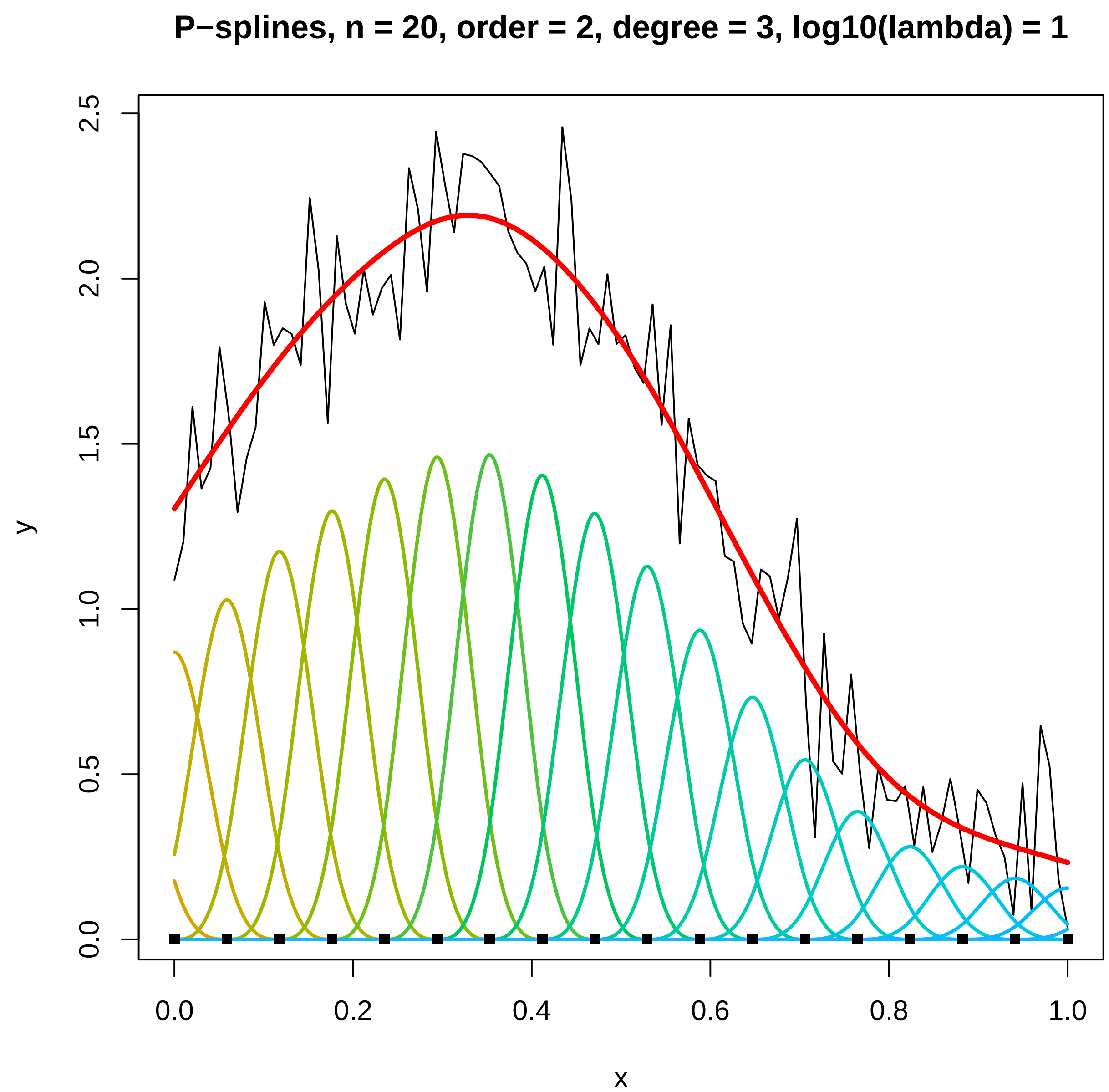
Paul Eilers

Erasmus Medical Center, Rotterdam

What are P-splines?

- A flexible tool for smoothing
- Based on regression with local basis functions: B-splines
- No efforts to optimize the basis
- Just a large number of B-splines
- And a penalty to tune smoothness
- (Software demo: `PSPlay_psplines`)

Plot from `PSPlay_psplines` program



The roots of P-splines

- Eilers and Marx: *Statistical Science*, 1996
- In fact not a very revolutionary proposal
- A simplification of O'Sullivan's ideas
- But the time seemed right
- Now over 1500 citations (in Web of Science)
- Many from applied areas (that's what really counts)
- I will show some theory and examples today

Discrete smoothing

- Given: data series y_i, $i = 1, \ldots, m$
- Wanted: a smooth series z
- Two (conflicting) goals: fidelity to y *and* smoothness of z
- Fidelity, sum of squares: $S = \sum_i (y_i - z_i)^2$
- How to quantify smoothness?
- Use roughness instead: $R = \sum_i (z_i - z_{i-1})^2$
- Simplification of Whittaker's (1923) "graduation"

Penalized least squares

- Combine fidelity and roughness

$$Q = S + \lambda R = \sum_i (y_i - z_i)^2 + \lambda \sum_i (z_i - z_{i-1})^2$$

- Parameter λ sets the balance

- Operator notation: $\Delta z_i = z_i - z_{i-1}$

$$Q = \sum_i (y_i - z_i)^2 + \lambda \sum_i (\Delta z_i)^2$$

Matrix-vector notation

- Penalized least squares objective function

$$Q = ||y - z||^2 + \lambda ||Dz||^2$$

- Differencing matrix D, such that $Dz = \Delta z$

$$D = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

- Explicit solution: $\hat{z} = (I + \lambda D'D)^{-1} y$

Implementation in R

```
m <- length(y)
E <- diag(m)        # Identity matrix
D <- diff(E)        # Difference operator
G <- E + lambda * t(D) %*%  D
z <- solve(G, y)  # Solve the equations
```

Notes on computation

- Linear system of equations
- m equations in m unknowns
- Practical limit with standard algorithm: $m \approx 4000$
- Computation time proportional to m^3
- But the system is extremely sparse (bandwidth = 3)
- Specialized algorithms easily handle $m > 10^6$ (package `spam`)
- Computation time then linear in m
- One million observations smoothed in one second

Sparse implementation in R

```
library(spam)
m <- length(y)
E <- diag.spam(m)      # Identity matrix
D <- diff(E)      # Difference operator
G <- E + lambda * t(D) %*%  D
z <- solve(G, y)  # Solve the equations
```

Higher order penalties

- Second order differences are easily defined
- Notation: $\Delta^2 z_i = \Delta(\Delta z_i) = (z_i - z_{i-1}) - (z_{i-1} - z_{i-2})$
- Second order differencing matrix

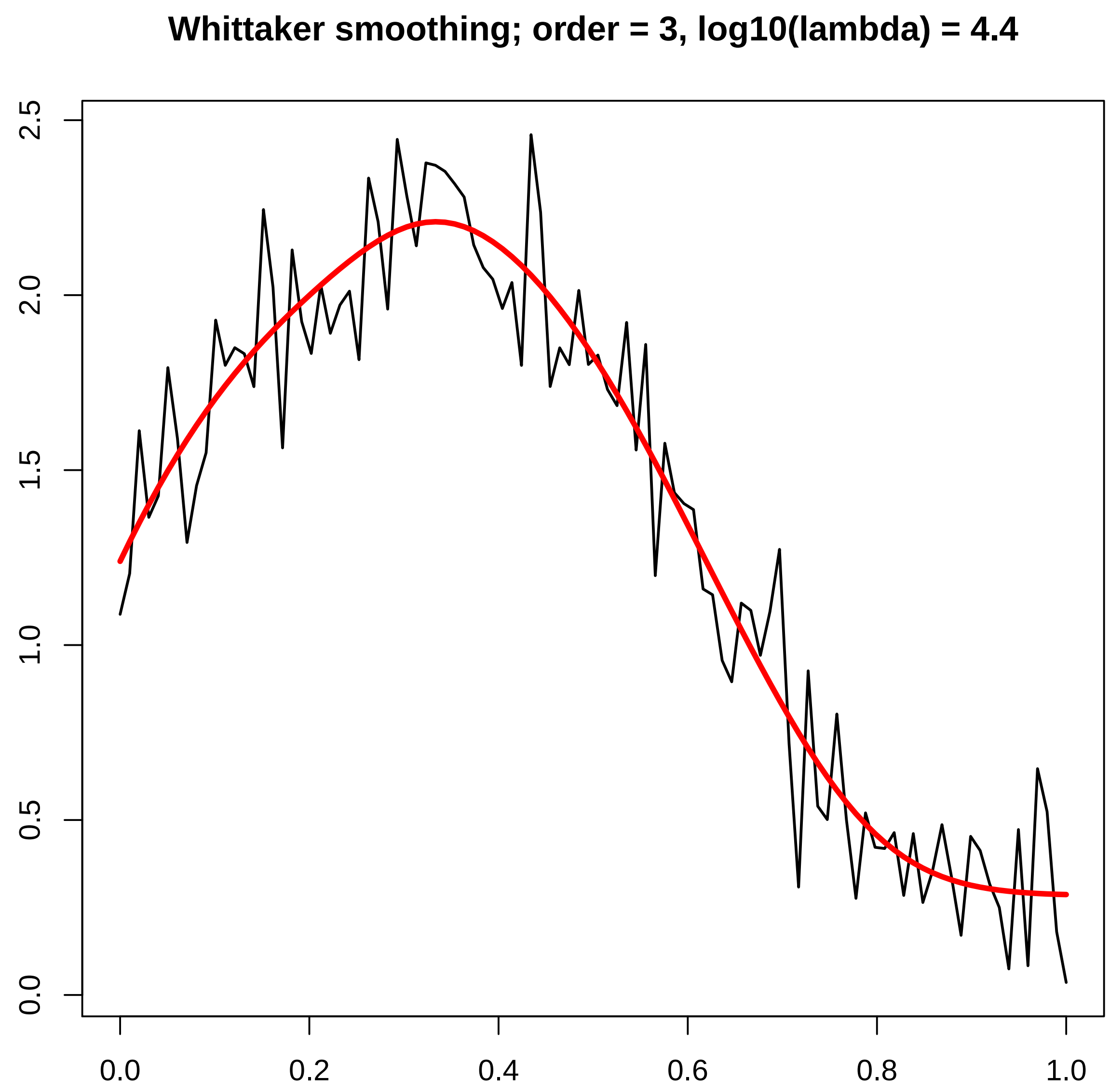
$$D = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}$$

- Higher orders are straightforward
- In R: `D = diff(diag(m), diff = d)`

The effects of higher orders

- Smoother curves
- Polynomial limits for large λ
- Degree of interpolation
- Degree of extrapolation
- Conservation of moments (will be explained later)
- (Software demo: `PSPlay_discrete`)

Plot from `PSPlay_discrete` program



Limits

- Consider large λ in $Q = ||y - z||^2 + \lambda||Dz||^2$
- Penalty is overwhelming, hence essentially $Dz = \Delta z = 0$
- This is the case if $z_i - z_{i-1} = 0$, hence $z_i = c$, a constant
- Generally: $\Delta^d z = 0$ if z is order $d - 1$ polynomial in i
- Linear limit when $d = 2$, quadratic when $d = 3$, ...
- It is also the least squares polynomial
- In the limit we have essentially a parametric model

Interpolation and extrapolation

- Let y_i be missing for some i
- Use weights w_i (0 if missing, 1 if not)
- Fill in arbitrary values (say 0) for missing y
- Minimize, with $W = \text{diag}(w)$

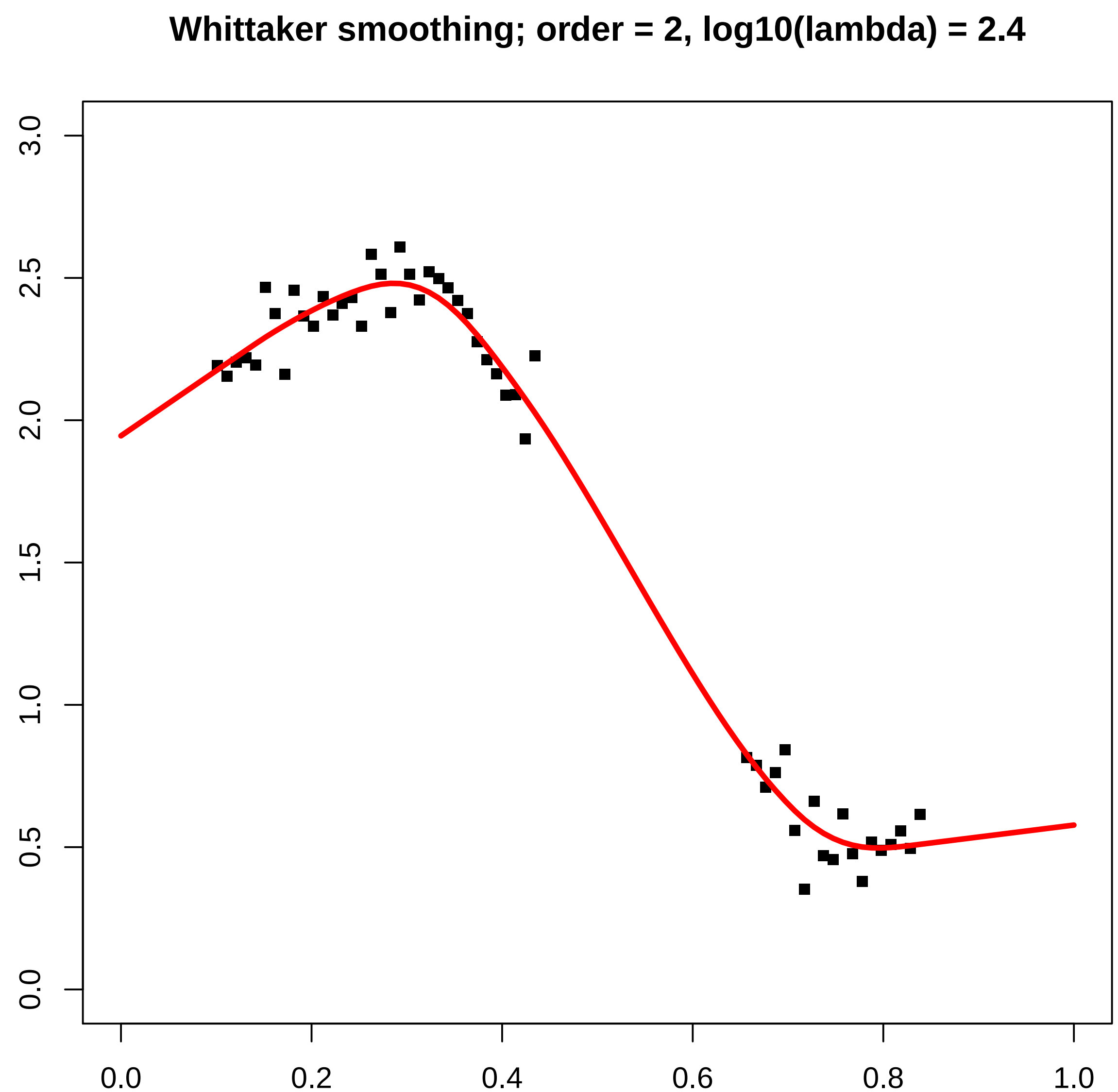
$$Q = (y - z)'W(y - z) + \lambda||Dz||^2$$

- Trivial changes: $\hat{z} = (W + \lambda D'D)^{-1}Wy$

Interpolation and extrapolation, continued

- Interpolation is by polynomial in i
- Order $2d - 1$
- Extrapolation: introduce "missing" data at the end(s)
- Extrapolation is by polynomial in i
- Order $d - 1$
- (Software demo: `PSPlay_interpolation`)

Plot from `PSPlay_interpolate` program



Non-normal data

- We measured fidelity by the sum of squares of residuals
- This is reasonable for (approximately) normal data
- Which means: trend plus normal disturbances
- How will we handle counts?
- Or binomial data?
- Use penalized (log-)likelihood
- Along the lines of the generalized linear model (GLM)

Smoothing of counts

- Given: a series y of counts
- We model a smooth linear predictor η
- Assumption: $y_i \sim \text{Pois}(\mu_i)$, with $\eta_i = \log \mu_i$
- The roughness penalty is the same
- But fidelity now measured by deviance (-2 LL):

$$Q = 2 \sum_i (\mu_i - y_i \eta_i) + \lambda \sum_i (\Delta^d \eta_i)^2$$

Linearization and weighted least squares

- Derivatives of Q give penalized likelihood equations

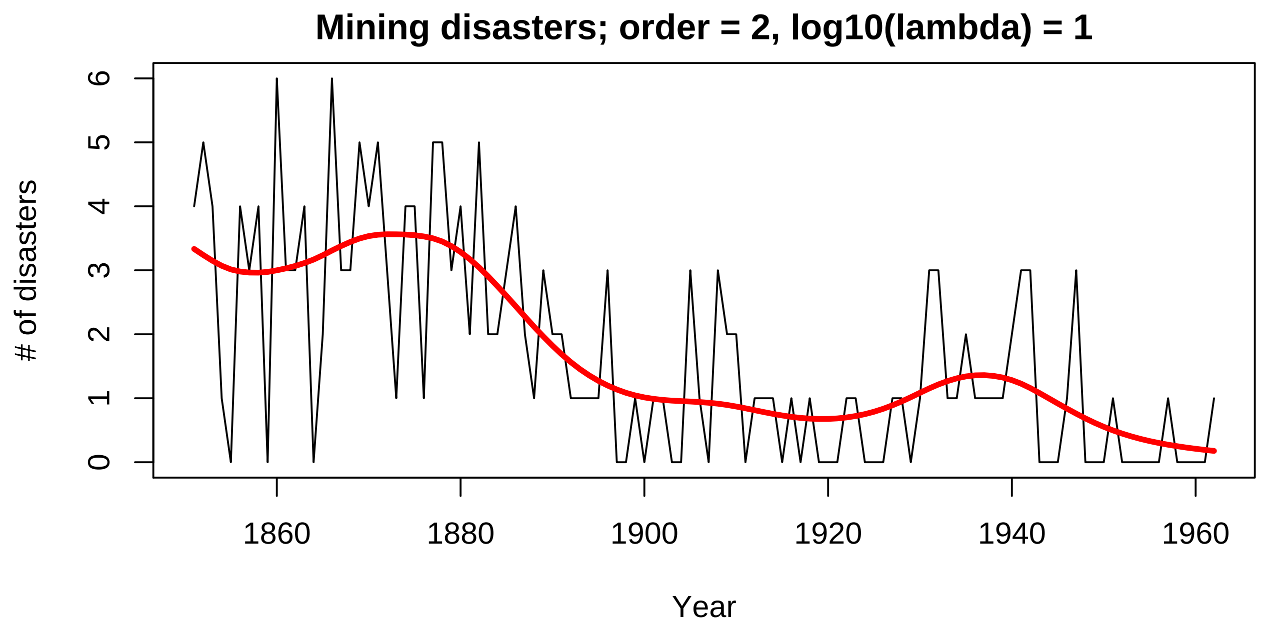
$$\lambda D'D\eta = y - e^{\eta} = y - \mu$$

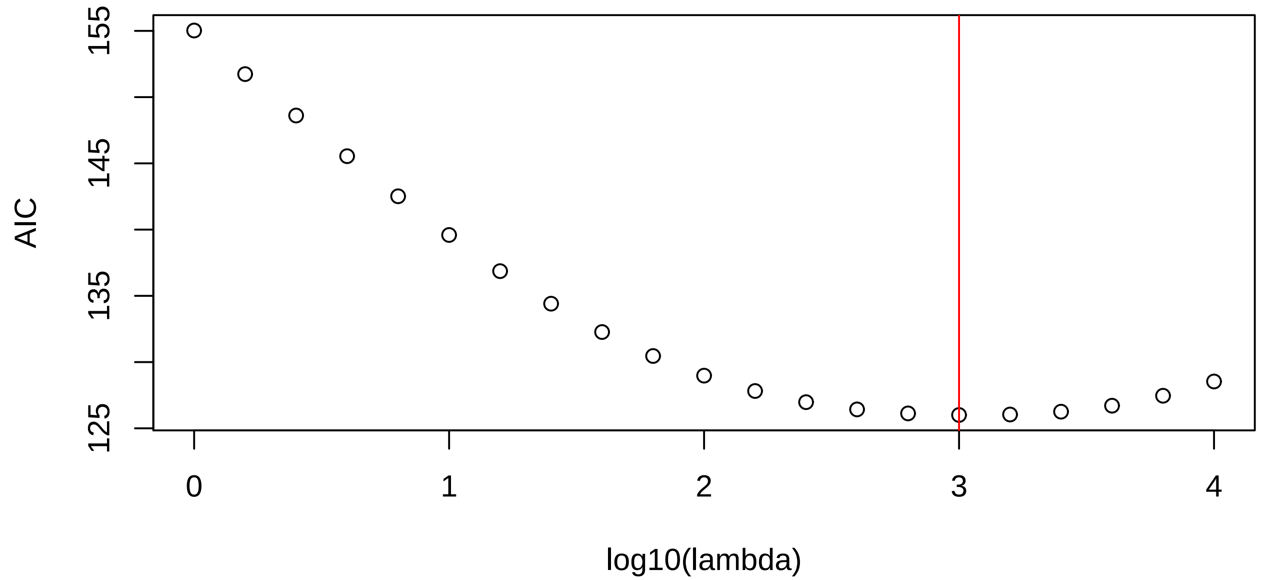
- Non-linear system, but the Taylor approximation gives

$$(\tilde{M} + \lambda D'D)\eta = y - \tilde{\mu} + \tilde{M}\tilde{\eta}$$

- Current approximation $\tilde{\eta}$, and $\tilde{M} = \mathrm{diag}(\tilde{\mu})$
- Repeat until (quick) convergence
- Start from $\tilde{\eta} = \log(y + 1)$

Example: severe coal mining accidents in UK

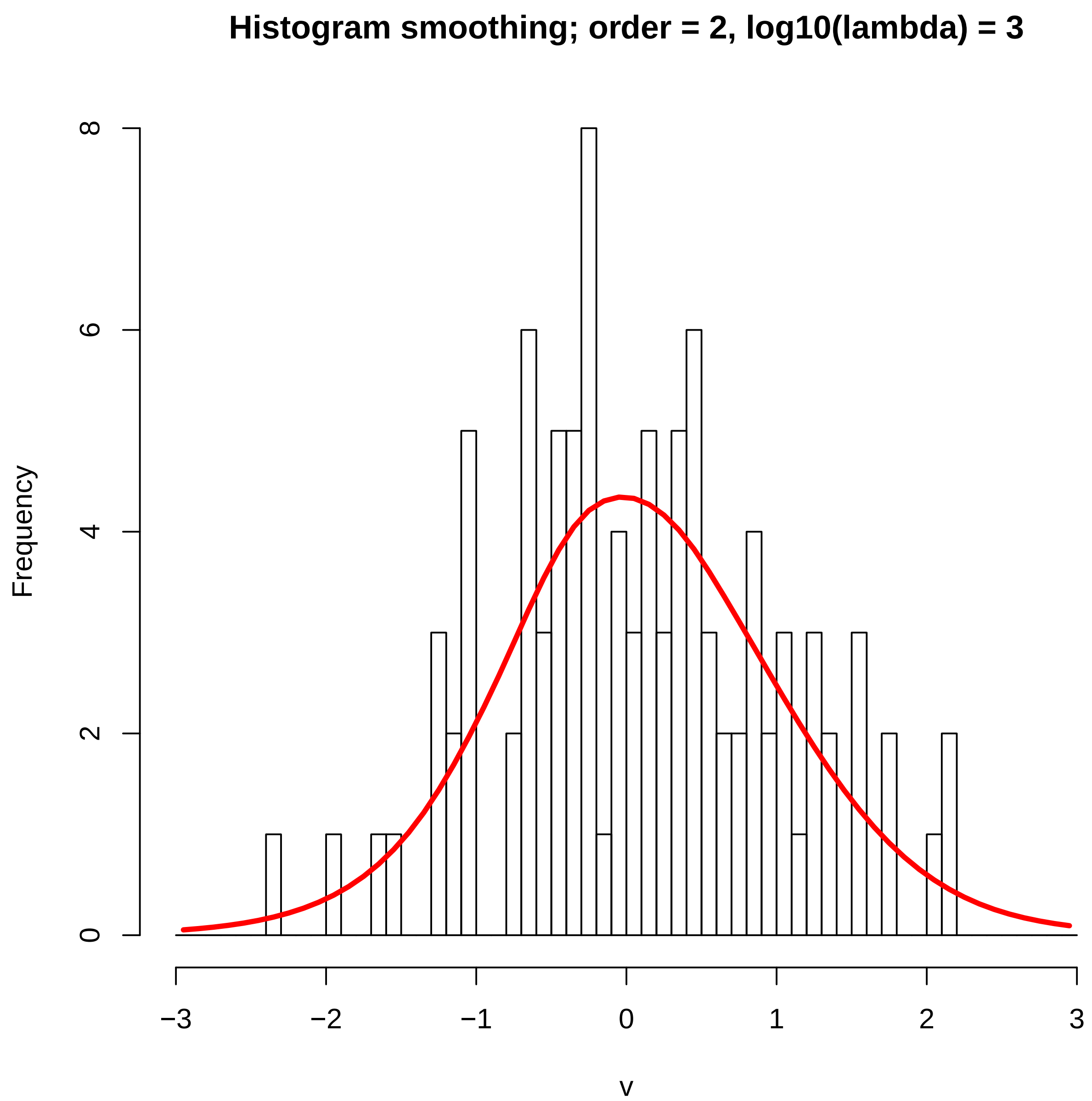




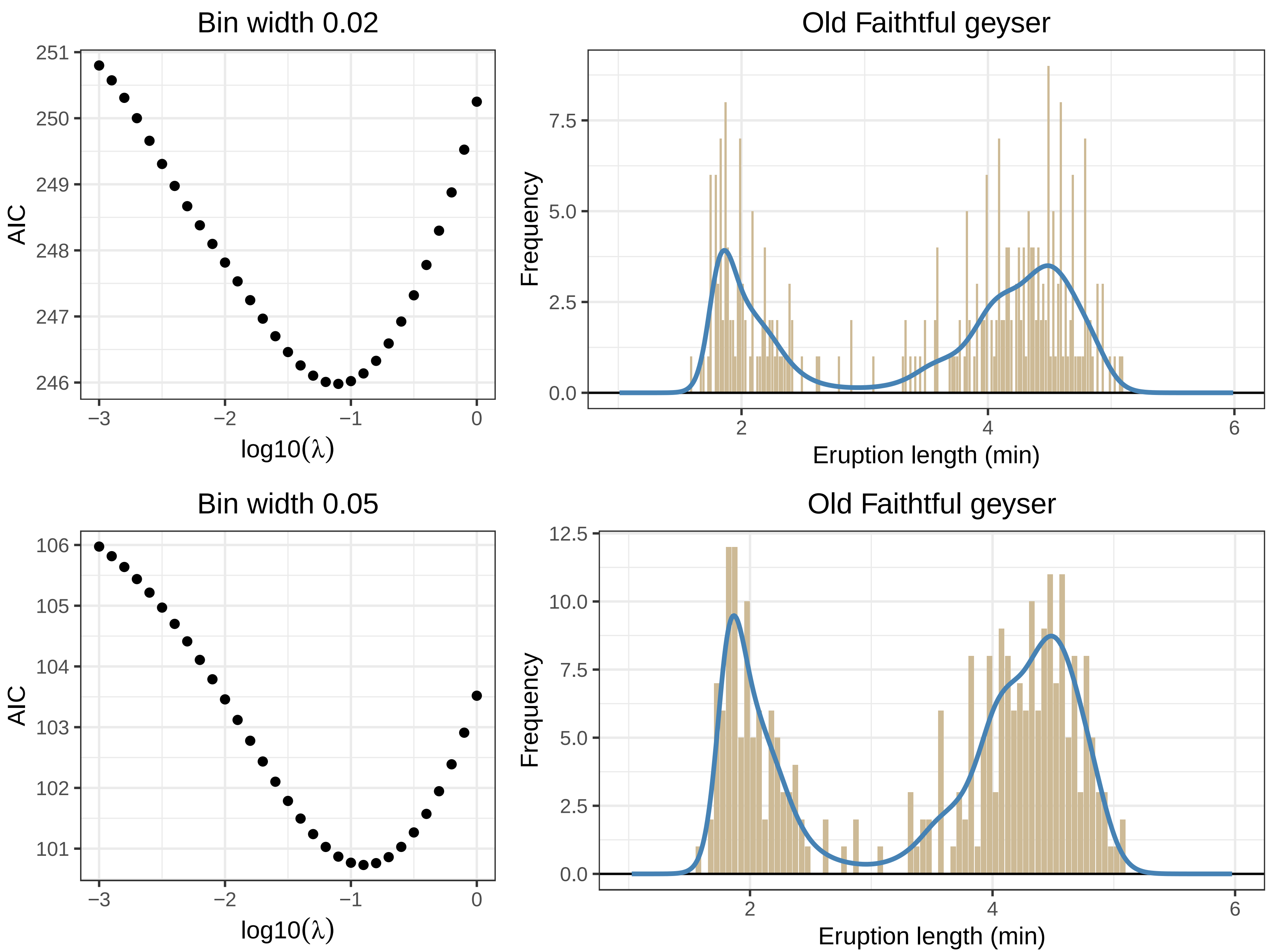
A useful application: histogram smoothing

- The "Poisson smoother" is ideal for histograms
- Bins can be very narrow
- Still a smooth realistic (discretized) density estimate
- Conservation of moments
- $\sum_i y_i x_i^k = \sum_i \hat{\mu}_i x_i^k$ for integer $k < d$ (bin midpoints in x)
- With $d = 3$, mean and variance don't change
- Whatever the amount of smoothing
- (Software demo: `PSPlay_histogram`)

Plot from `PSPlay_histogram` program



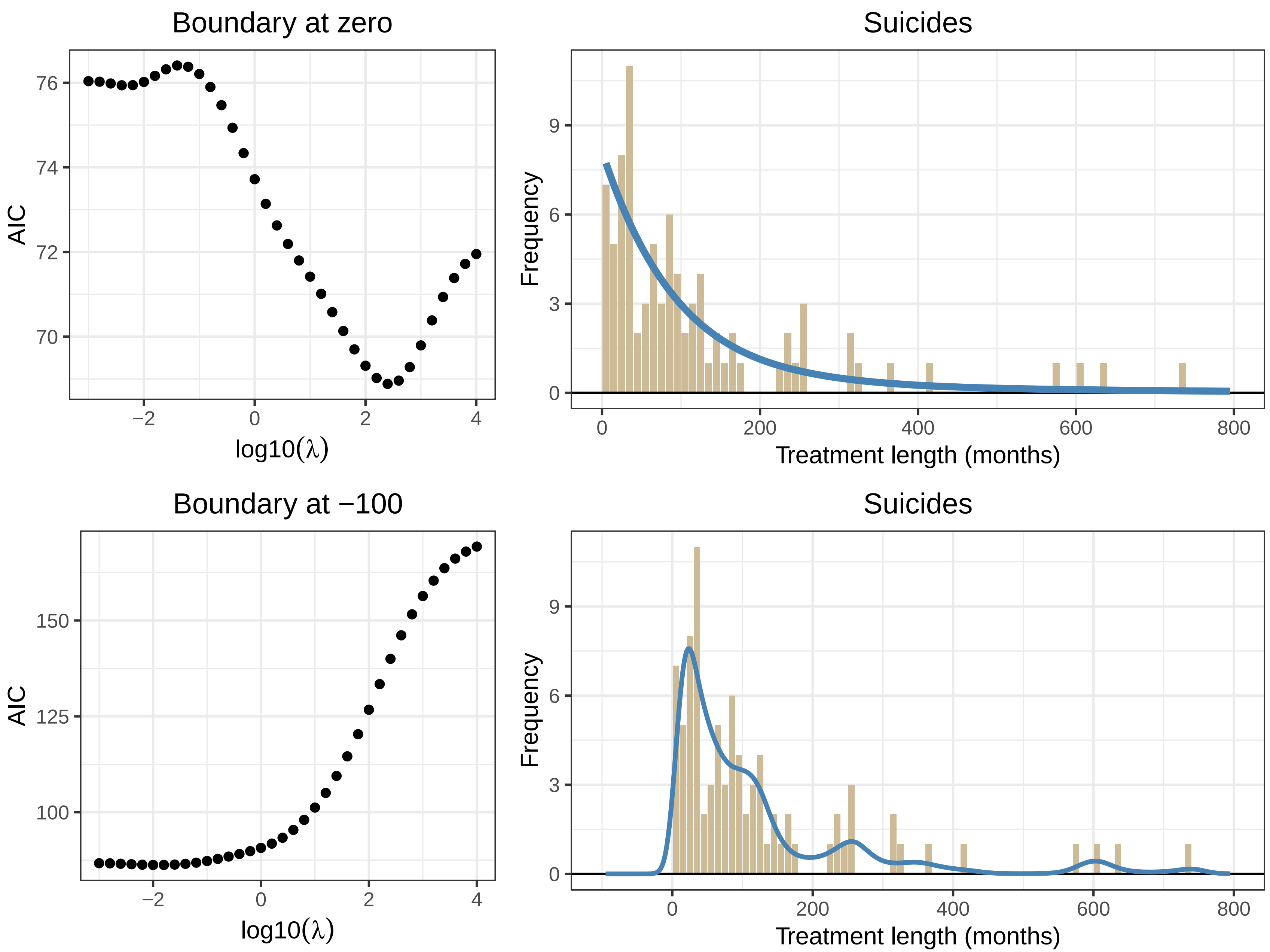
Smoothing old Faithful



Pay attention to the boundaries

- Extend the histogram with enough zero counts
- But some data are inherently bounded
- Non-negative, or between 0 and 1
- Then you should limit the domain accordingly
- Otherwise you will smooth in the "no go" area
- Example: suicide treatment data
- Inherently non-negative durations of treatment spells

Smoothing the suicide treatment data



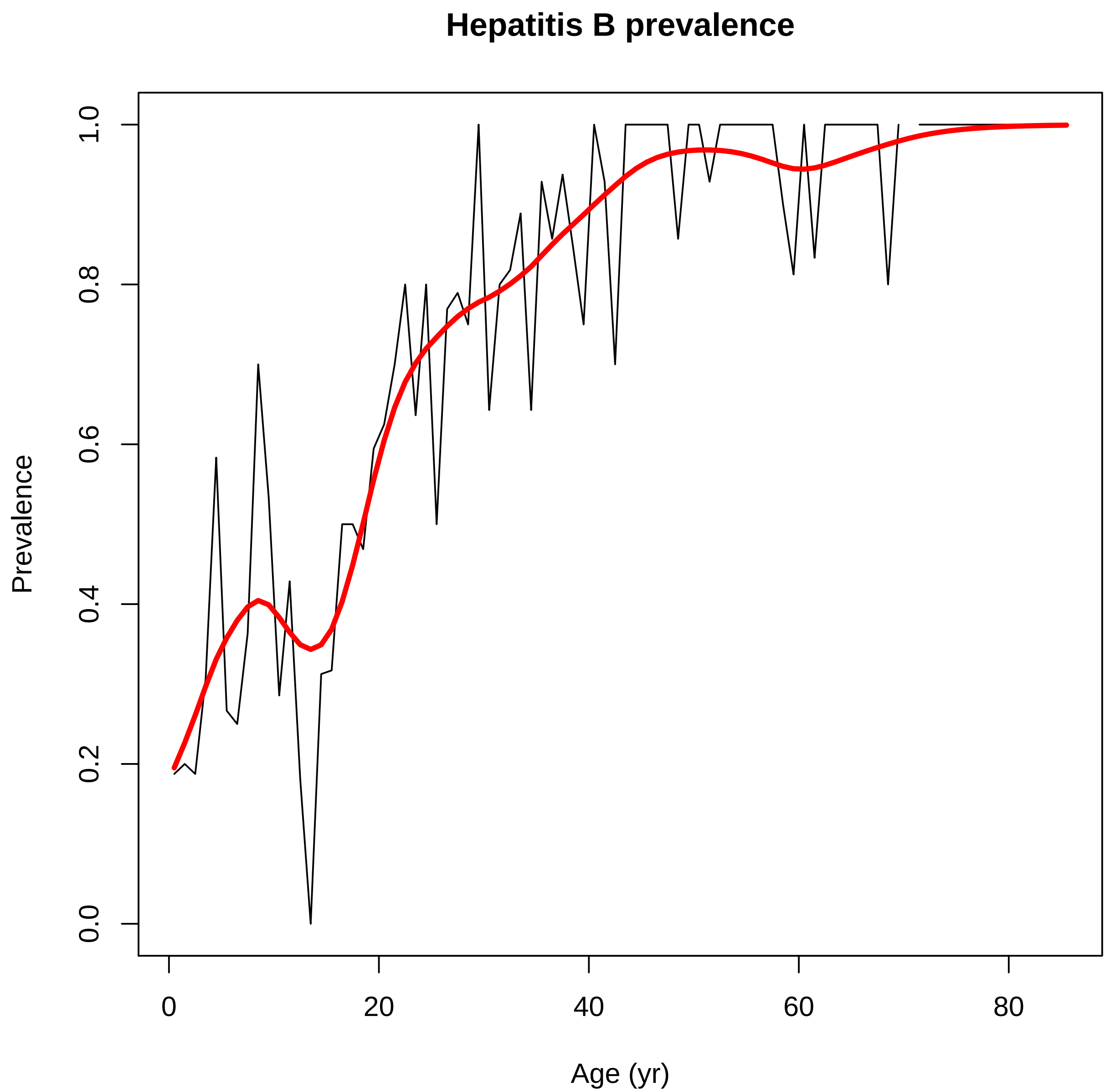
Binomial data

- Given: sample sizes s, "successes" y
- Smooth curve wanted for p, probability of succes
- We model the logit:

$$\eta = \log \frac{p}{1-p}; \quad p = \frac{e^{\eta}}{1+e^{\eta}} = \frac{1}{1+e^{-\eta}}$$

- Linearization as for counts
- Start from logit of $(y+1)/(s+2)$
- No surprises, details skipped

Example: hepatitis B prevalence (Keiding)



Optimal smoothing

- We can smooth almost anything (in the GLM sense)
- How much should we smooth?
- Let the data decide
- Cross-validation, AIC (BIC)
- Essentially we measure prediction performance
- On new or left-out data

Leave-one-out cross-validation

- Leave out y_i (make w_i zero)
- Interpolate a value for it: $\hat{y}_{-i}$
- Do this for all observations in turn
- You get a series of "predictions" $\hat{y}_{-i}$
- How good are they?
- Use $CV = \sum(y_i - \hat{y}_{-i})^2$, or $RMSCV = \sqrt{CV/m}$
- Search for λ that minimizes CV

Speeding up the computations

- LOO CV looks expensive (repeat smoothing m times)
- It is, if done without care
- But there is a better way
- We have $\hat{y} = (W + \lambda D'D)^{-1}Wy = Hy$
- We call H the hat matrix; property: $h_{ij} = \partial \hat{y}_i / \partial y_j$
- One can prove: $y_i - \hat{y}_{-i} = (y_i - \hat{y}_i)/(1 - h_{ii})$
- Smooth once (for each λ), compute all $\hat{y}_{-i}$ at the same time

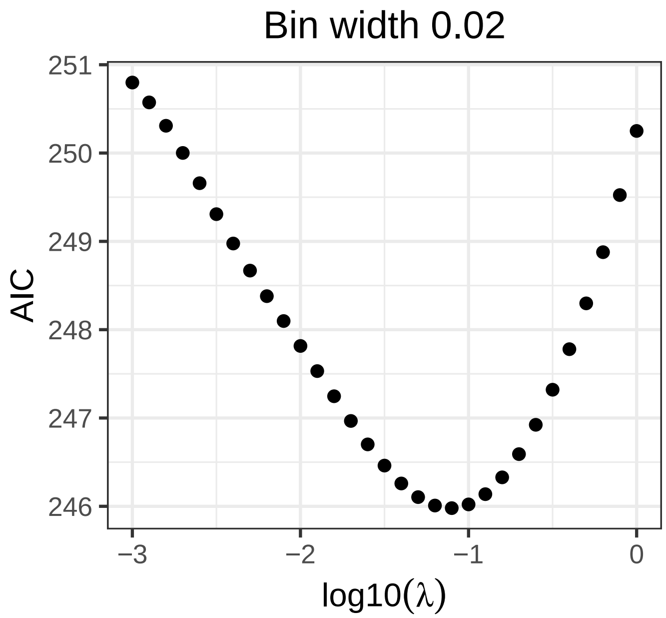
Akaike's information criterion

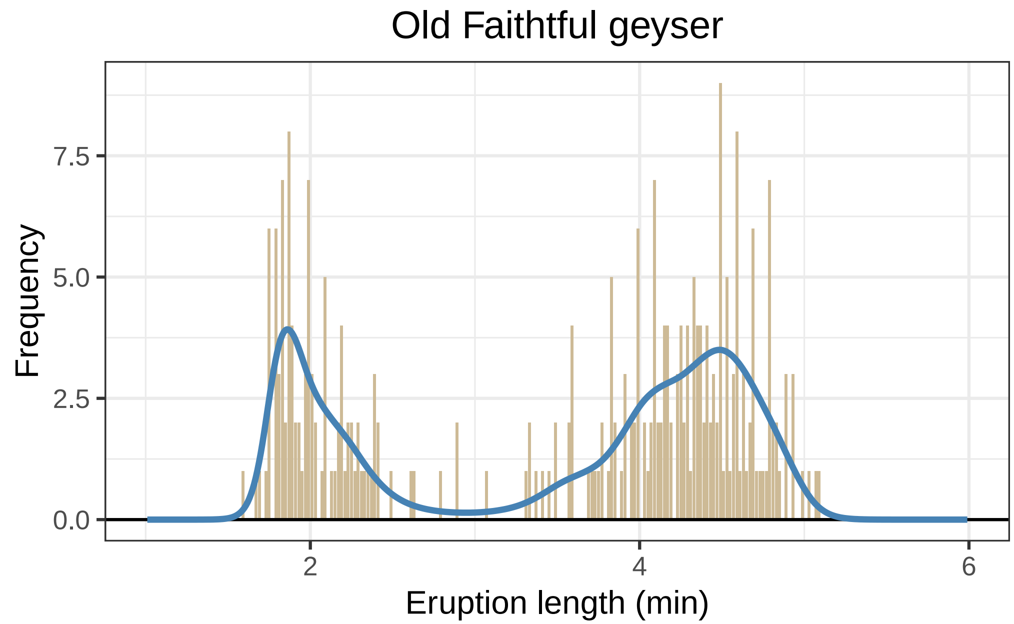
- Definition: $AIC = \text{Deviance} + 2ED = -2LL + 2ED$
- Here ED is the effective model dimension
- Useful definition:

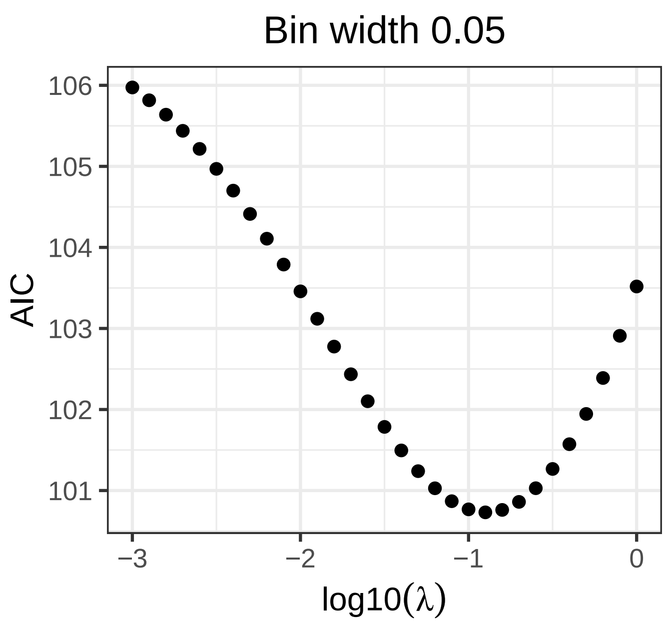
$$ED = \sum_i \partial \hat{\mu}_i / \partial y_i = \sum_i h_{ii} = \text{tr}(H)$$

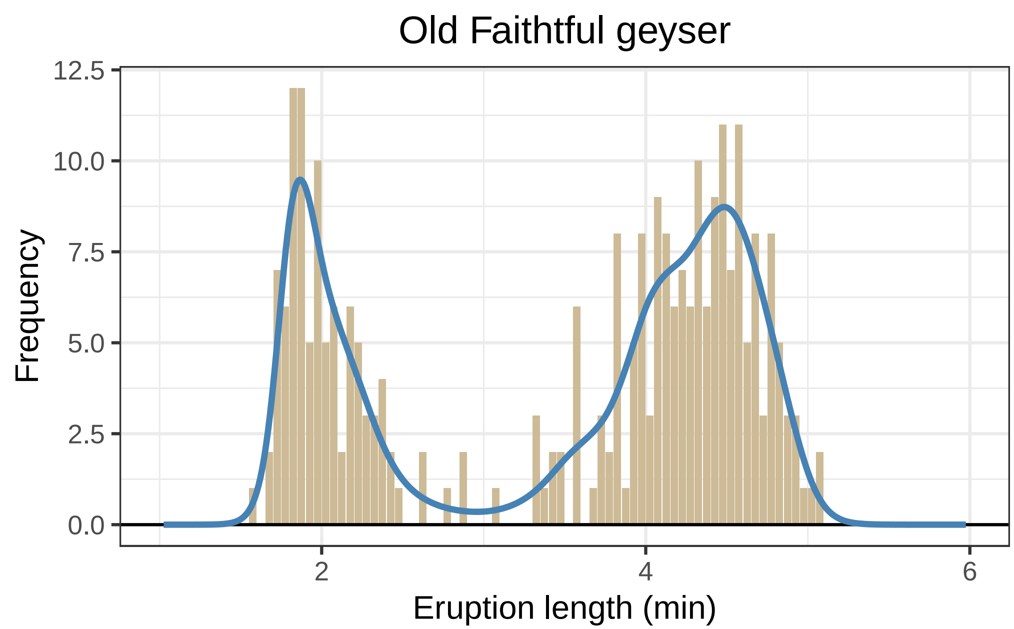
- This defines a hat matrix for generalized linear smoothing
- Vary λ on a grid to find minimum of AIC
- Minimization routine can be used too
- But it is useful to see the curve of AIC vs. $\log \lambda$

Old Faithful again





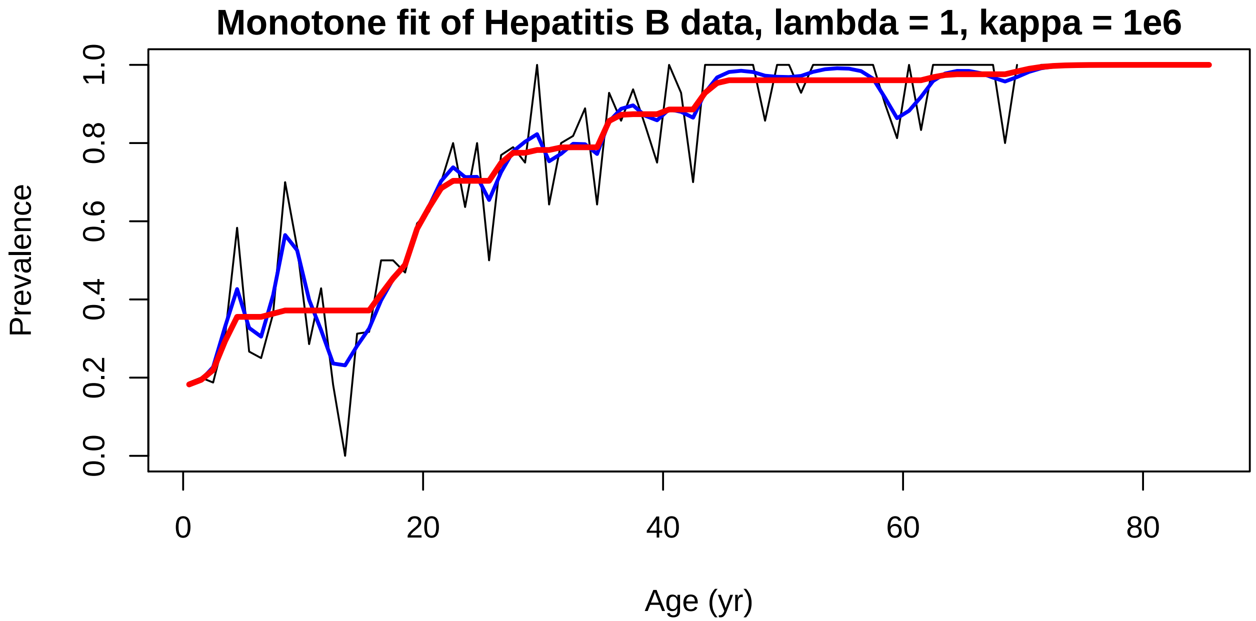


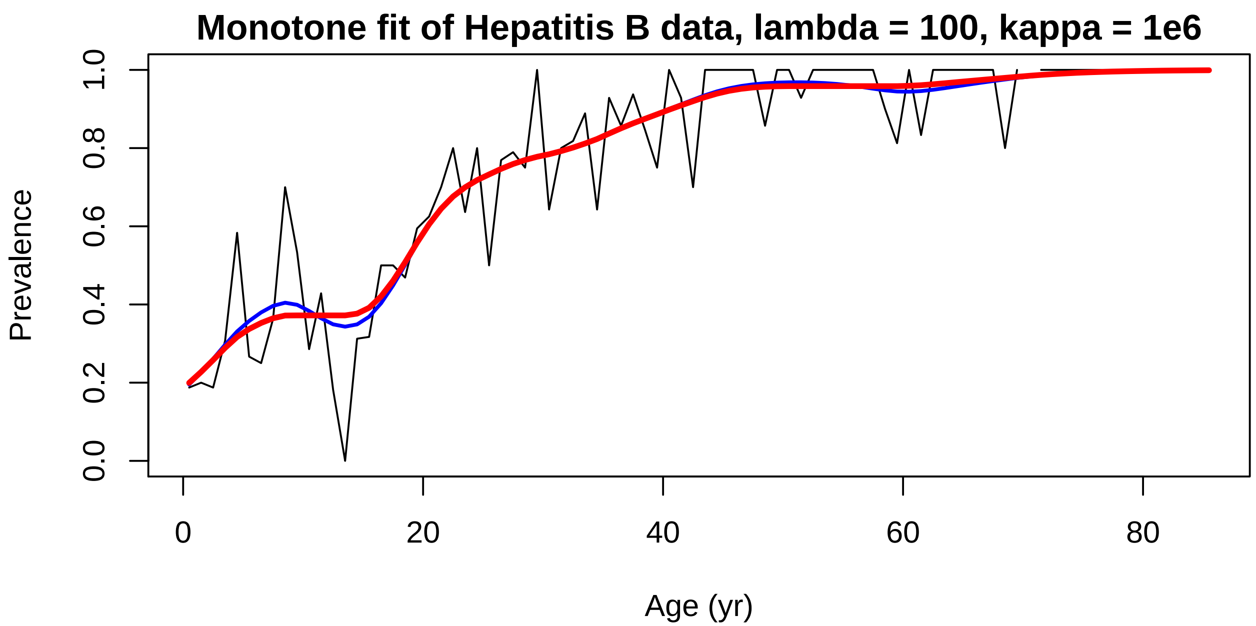


Asymmetric penalties and monotone smoothing

- Sometimes we want a smooth increasing result
- Smoothing alone does not guarantee a monotone shape
- We need a little help
- Additional asymmetric penalty $P = \kappa \sum_i v_i(z_i - z_{i-1})^2$
- With $v_i = 1$ if $z_i < z_{i-1}$ and $v_i = 0$ otherwise
- The penalty only works where monotonicity is violated
- With large κ we get the desired result
- This idea also works for convex smoothing

Example of monotone smoothing



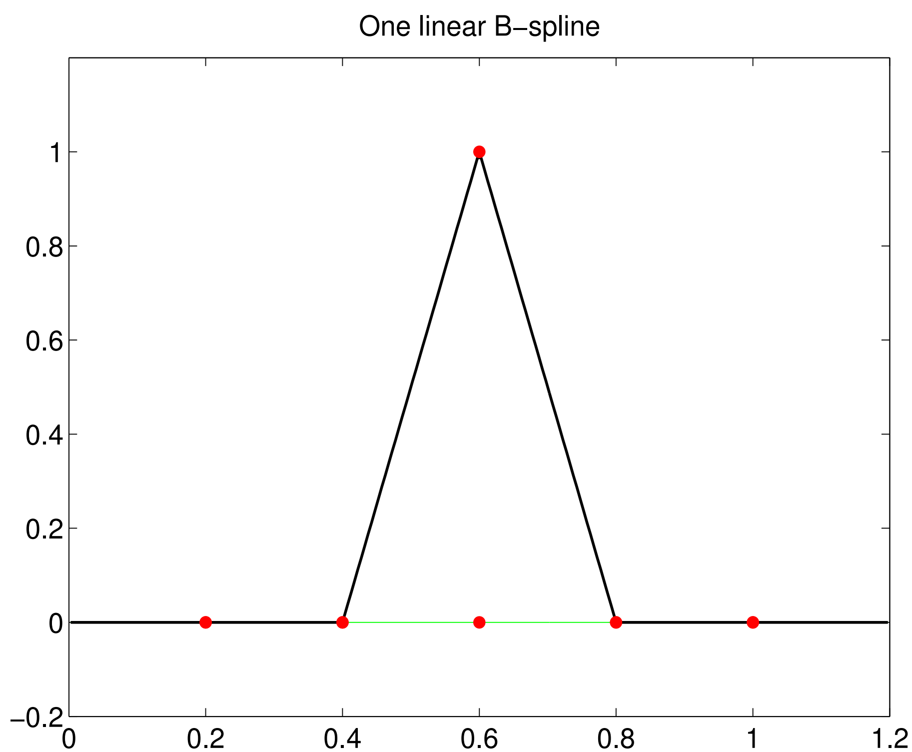


Limitations of the Whittaker smoother

- The xs of the observations must be equally spaced
- Multiple y for one x need extra work
- Inefficient computation in complex models
- Solution: P-splines
- Combine Whittaker's penalty with regression on B-splines

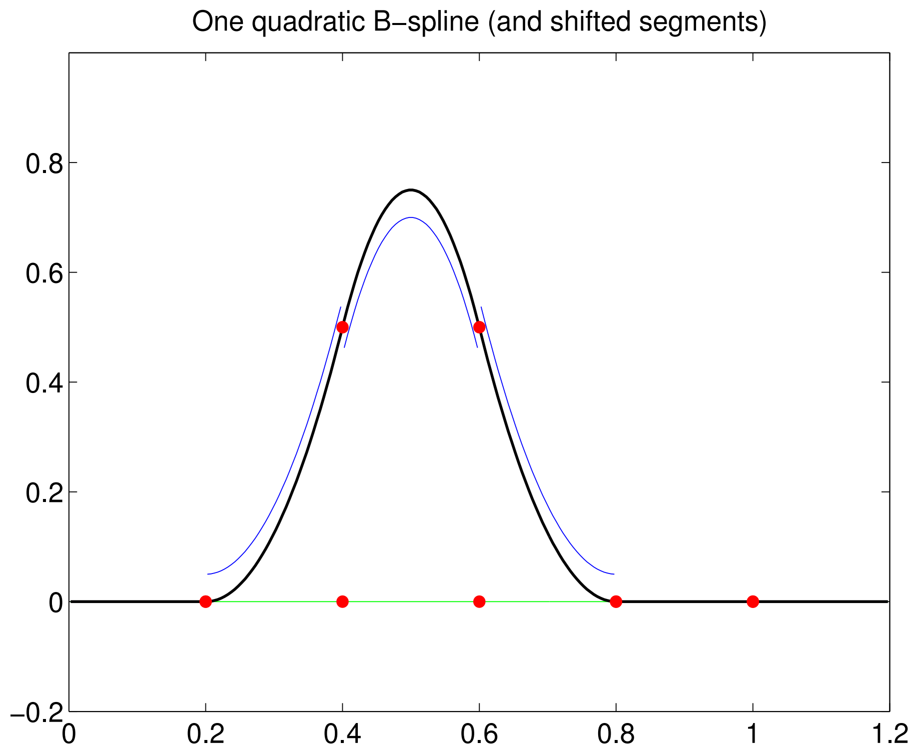
One linear B-spline

- Two pieces, each a straight line, everything else zero
- Nicely connected at knots (t_1 to t_3) same value
- Slope jumps at knots



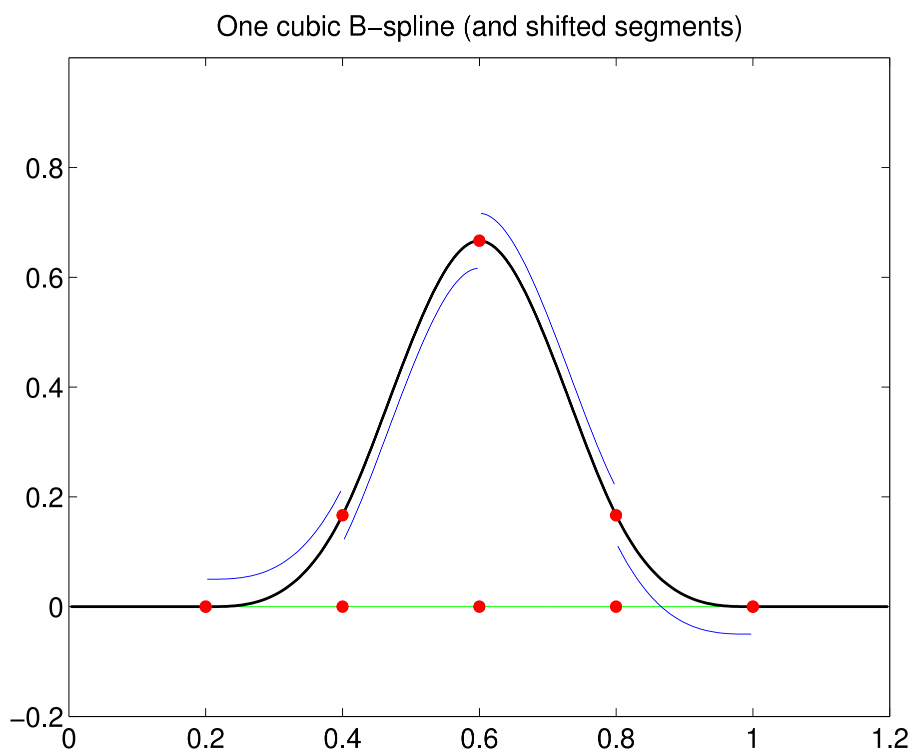
One quadratic B-spline

- Three pieces, each a quadratic segment, rest zero
- Nicely connected at knots (t_1 to t_4): same values and slopes
- Shape similar to Gaussian

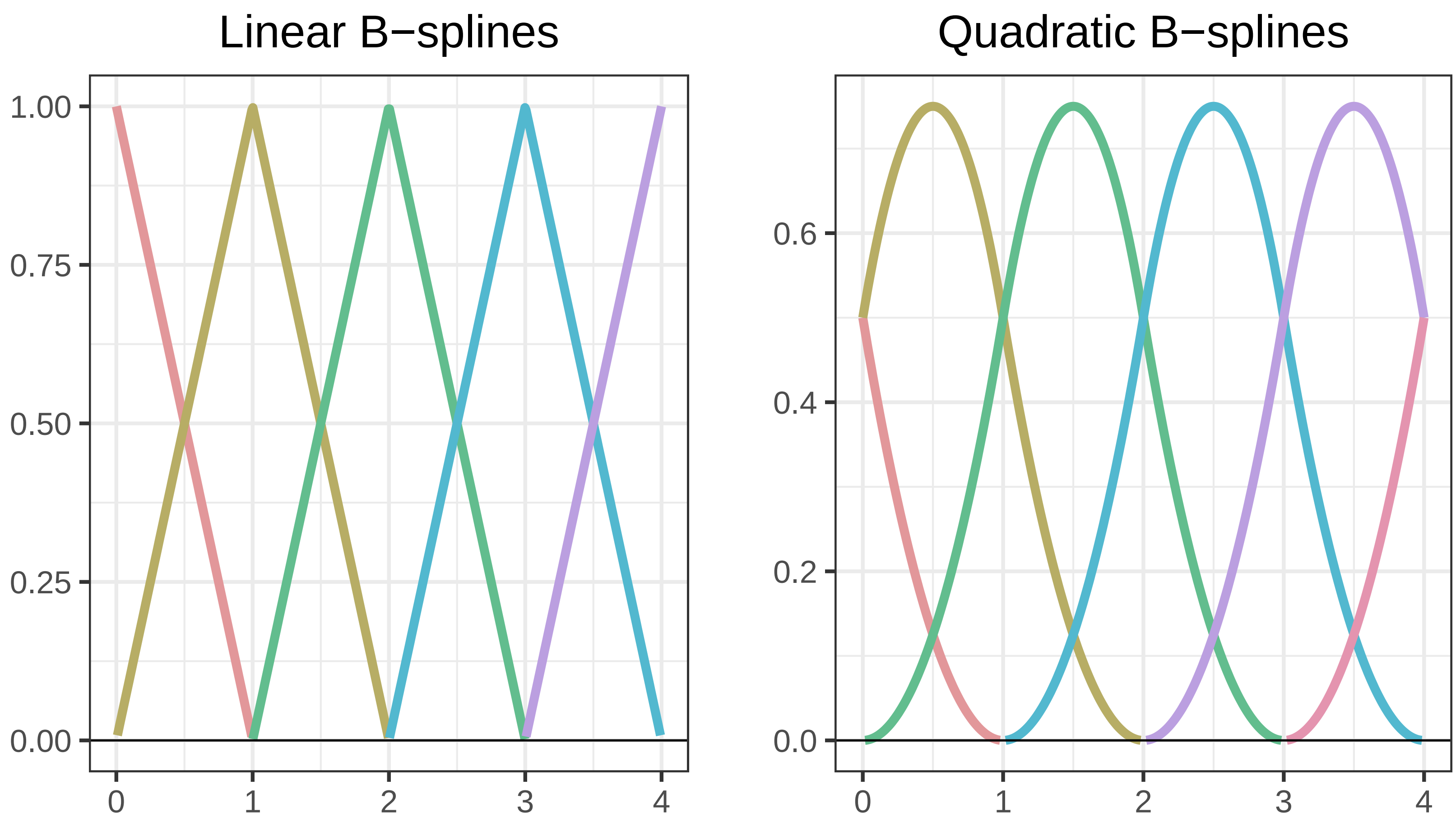


One cubic B-spline

- Four pieces, each a cubic segment, rest zero
- At knots (t_1 to t_5): same values, first & second derivatives
- Shape more similar to Gaussian



Sets of linear and cubic B-splines



B-spline basis

- Basis matrix B

- Columns are B-splines

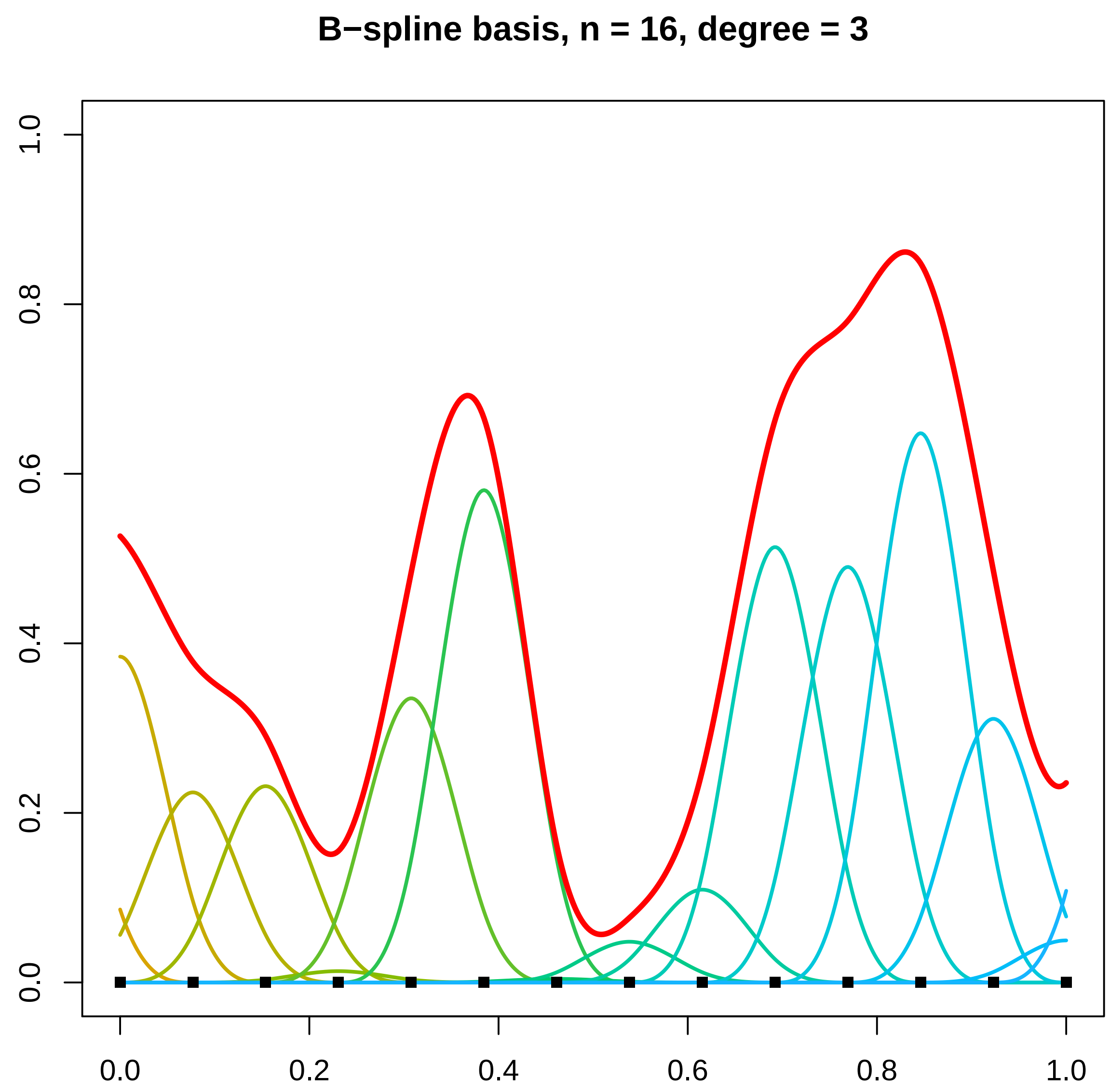
$$\begin{bmatrix} B_1(x_1) & B_2(x_1) & B_3(x_1) & \dots & B_n(x_1) \\ B_1(x_2) & B_2(x_2) & B_3(x_2) & \dots & B_n(x_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ B_1(x_m) & B_2(x_m) & B_3(x_m) & \dots & B_n(x_m) \end{bmatrix}$$

- In each row only a few non-zero elements (degree plus one)

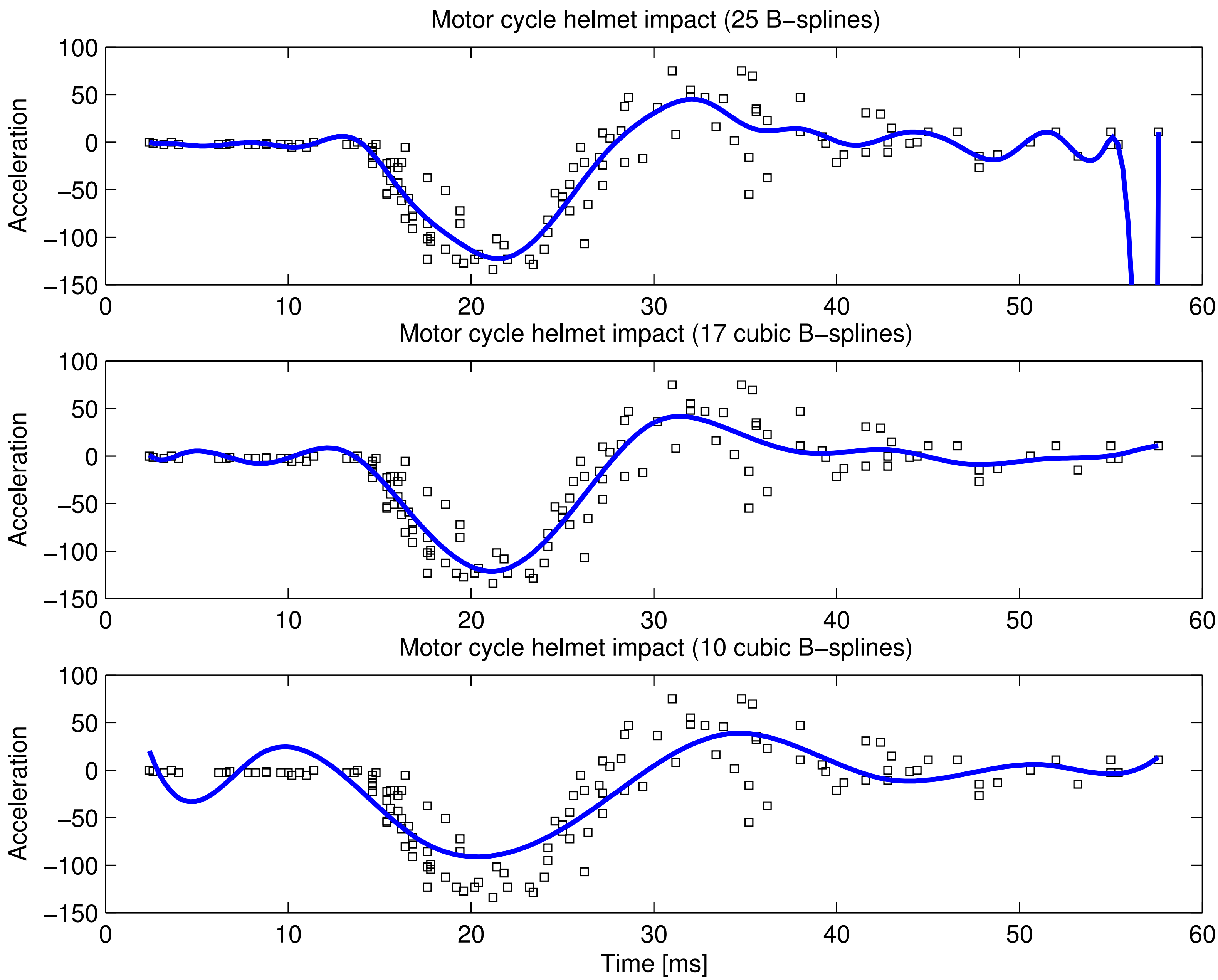
- Only a few basis functions contribute to $\mu_i = \sum b_{ij}\alpha_j = B'_{i\bullet}\alpha$

- (Software demo: `PSPlay_bsplines`)

Plot from `PSPlay_bsplines` program



B-splines fit to motorcycle data



P-splines on one slide

- Do regression on (cubic) B-splines
- Use equally spaced knots
- Take a large number of them (10, 20, 50)
- Put a difference penalty (order 2 or 3) on the coefficients
- Tune smoothness with λ (penalty weight)
- Don't try to optimize the number of B-splines
- Relatively small system of equations (10, 20, 50)
- Arbitrary distribution of x allowed

Technical details of P-splines

- Minimize (with basis B)

$$Q = ||y - B\alpha||^2 + \lambda||D\alpha||^2$$

- Explicit solution:

$$\hat{\alpha} = (B'B + \lambda D'D)^{-1}B'y$$

- Hat matrix $H = (B'B + \lambda D'D)^{-1}B'$
- For a nice curve, compute B^* on nice grid x^*
- Plot $B^*\hat{\alpha}$ vs x^*

Properties of P-splines

- Penalty $\sum_j (\Delta^d \alpha_j)^2$
- Limit for strong smoothing is a polynomial of degree $d - 1$
- Interpolation: polynomial of degree $2d - 1$
- Extrapolation: polynomial of degree $d - 1$
- Conservation of moments of degree up to $d - 1$
- Many more B-splines then observations are allowed
- The penalty does the work!
- (Software demo: `PSPlay_psplines`)

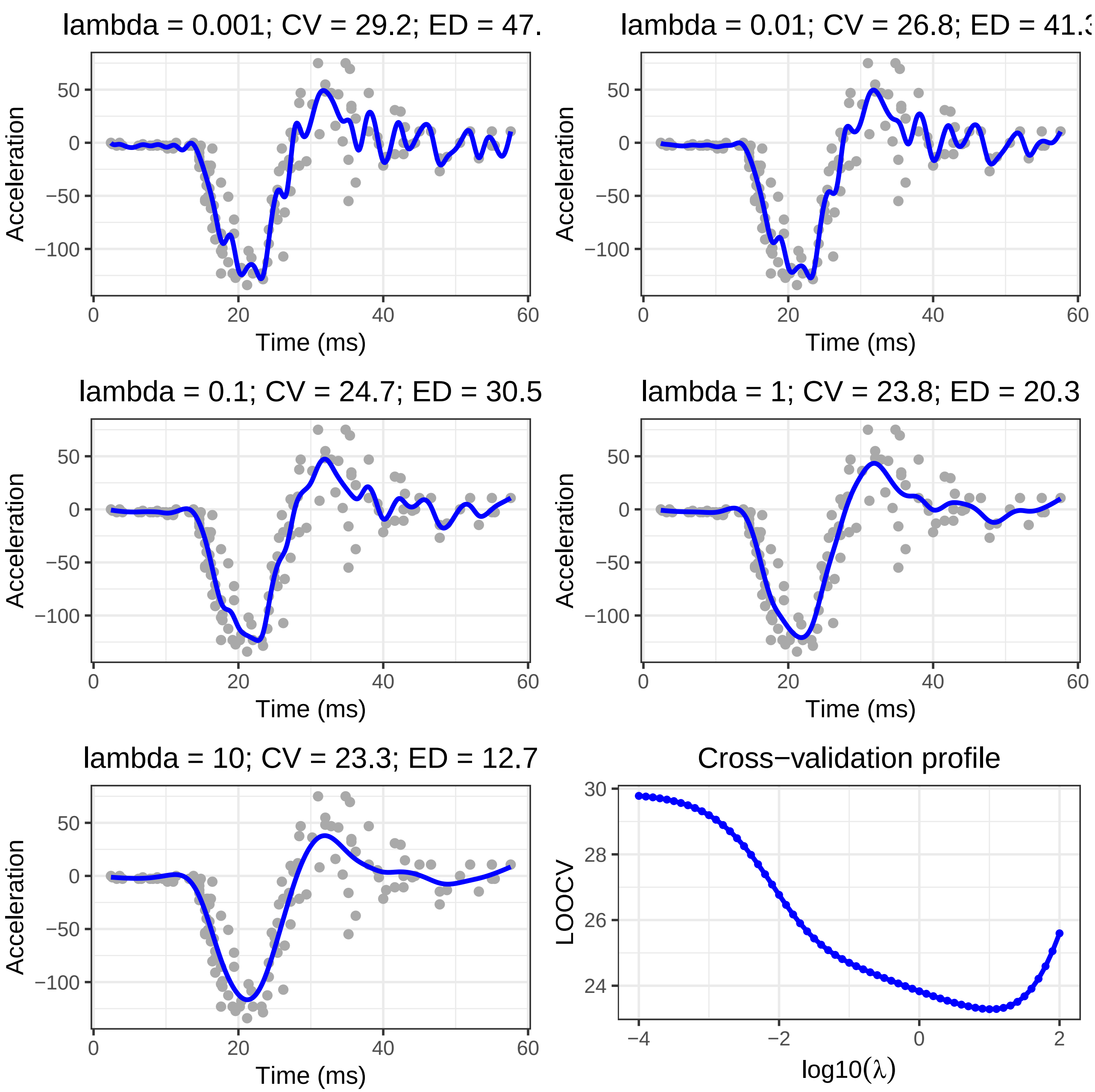
Cross-validation

- The same idea as for Whittaker smoother
- Leave out each observation in turn and predict it: $\hat{y}_{-i}$
- Compute how close they are to observations:

$$CV = \sum_i (y_i - \hat{y}_{-i})^2 = \sum_i r_{-i}^2$$

- Speedy computation with hat matrix: $H = B(B'B + \lambda D'D)^{-1}B'$
- $r_{-i} = y_i - y_{-i} = (y_i - \hat{y}_i)/(1 - h_{ii})$

Motorcycle helmet data



Generalized linear smoothing

- It is just like a GLM (generalized linear model)
- With the penalty sneaked in
- Poisson example for counts y
- Linear predictor $\eta = B\alpha$, expectations $\mu = e^{\eta}$
- Assumption $y_i \sim \text{Pois}(\mu_i)$ (independent)
- From penalized Poisson log-likelihood follows iteration with

$$(B'\tilde{M}B + \lambda D'D)\alpha = B'(y - \tilde{\mu} + \tilde{M}B\tilde{\alpha})$$

- Here $M = \text{diag}(\mu)$

Generalized additive models

- One-dimensional smooth model: $\eta = f(x)$
- Two-dimensional smooth model: $\eta = f(x_1, x_2)$
- General f: any interaction between x_1 and x_2 allowed
- We want to avoid two-dimensional smoothing
- Generalized additive model: $\eta = f_1(x_1) + f_2(x_2)$
- Both f_1 and f_2 smooth (Hastie and Tibshirani, 1990)
- Higher dimensions straightforward

The old way: backfitting for GAM

- Assume linear model: $E(y) = \mu = f_1(x_1) + f_2(x_2)$
- Assume: approximations $\tilde{f}_1$ and $\tilde{f}_2$ available
- Compute partial residuals $r_1 = y - \tilde{f}_2(x_2)$
- Smooth scatterplot of (x_1, r_1) to get better $\tilde{f}_1$
- Compute partial residuals $r_2 = y - \tilde{f}_1(x_1)$
- Smooth scatterplot of (x_2, r_2) to get better $\tilde{f}_2$
- Repeat to convergence

More on backfitting

- Start with $\tilde{f}_1 = 0$ and $\tilde{f}_2 = 0$
- Generalized residuals and weights for non-normal data:
- Any smoother can be used
- Convergence can be proved, but may take many iterations
- Convergence criteria should be strict

PGAM: GAM with P-splines

- Use B-splines: $\eta = f_1(x_1) + f_2(x_2) = B_1\alpha_1 + B_2\alpha_2$

- Combine B_1 and B_2 to matrix, α_1 and α_2 to vector:

$$\eta = [B_1 : B_2]\begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = B^*\alpha^*$$

- Difference penalties on α_1, α_2, in block-diagonal matrix

- Penalized GLM as before: no backfitting

P-GAM fitting (GLM setting)

- Maximize

$$l^* = l(\alpha; B, y) - \frac{1}{2}\lambda_1||D_1\alpha_1||^2 - \frac{1}{2}\lambda_2||D_2\alpha_2||^2$$

- Iterative solution:

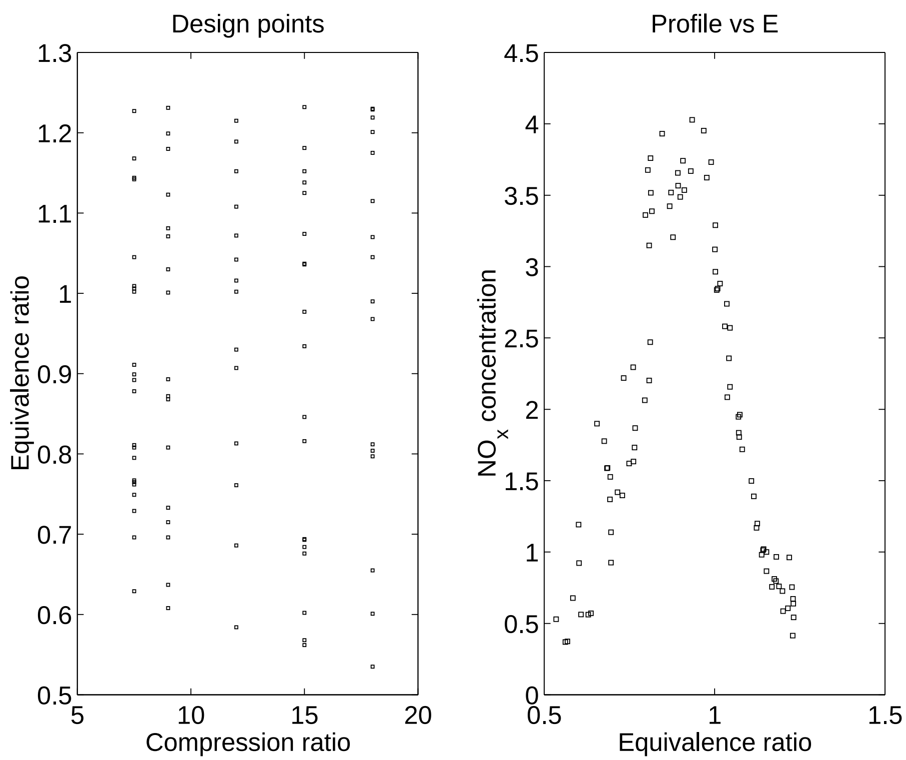
$$\hat{\alpha}_{t+1} = (B'\hat{W}_t B + P)^{-1} B'(y - \tilde{\mu} + \hat{W}_t \hat{\eta}_t^\star)$$

where

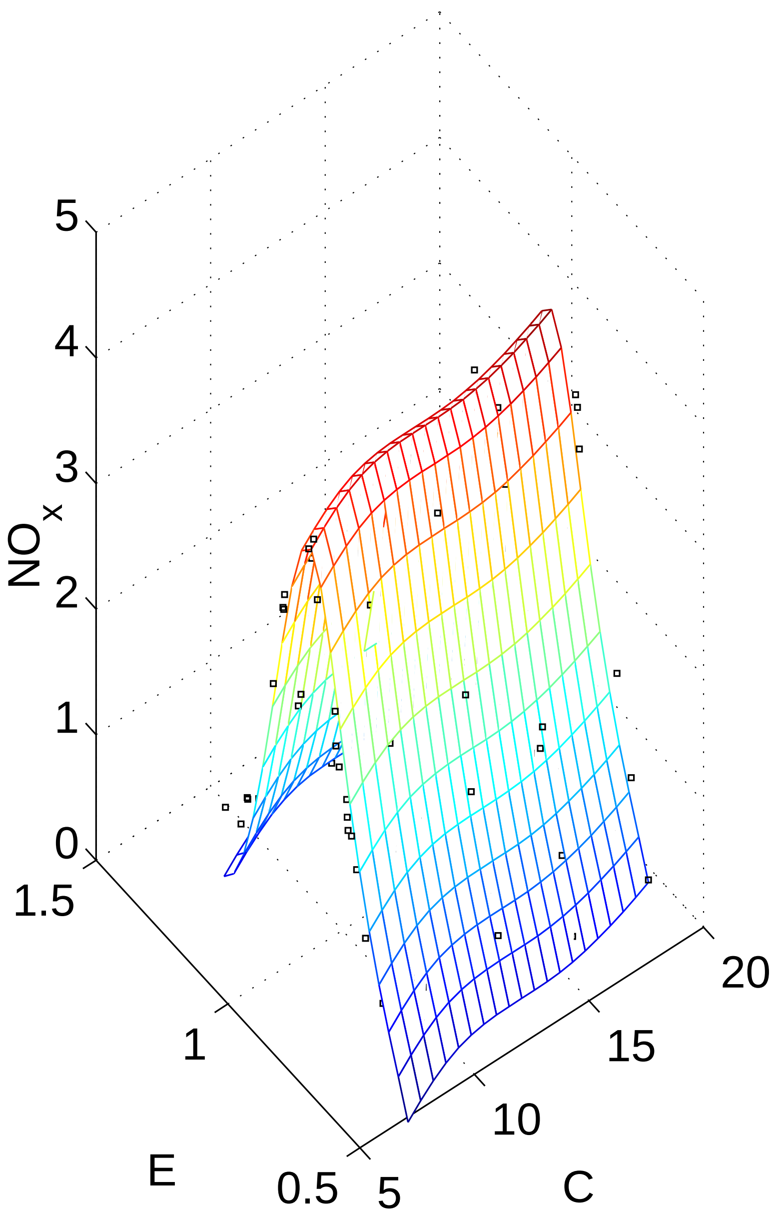
$$P = \begin{bmatrix} \lambda_1 D_1' D_1 & 0 \\ 0 & \lambda_2 D_2' D_2 \end{bmatrix}$$

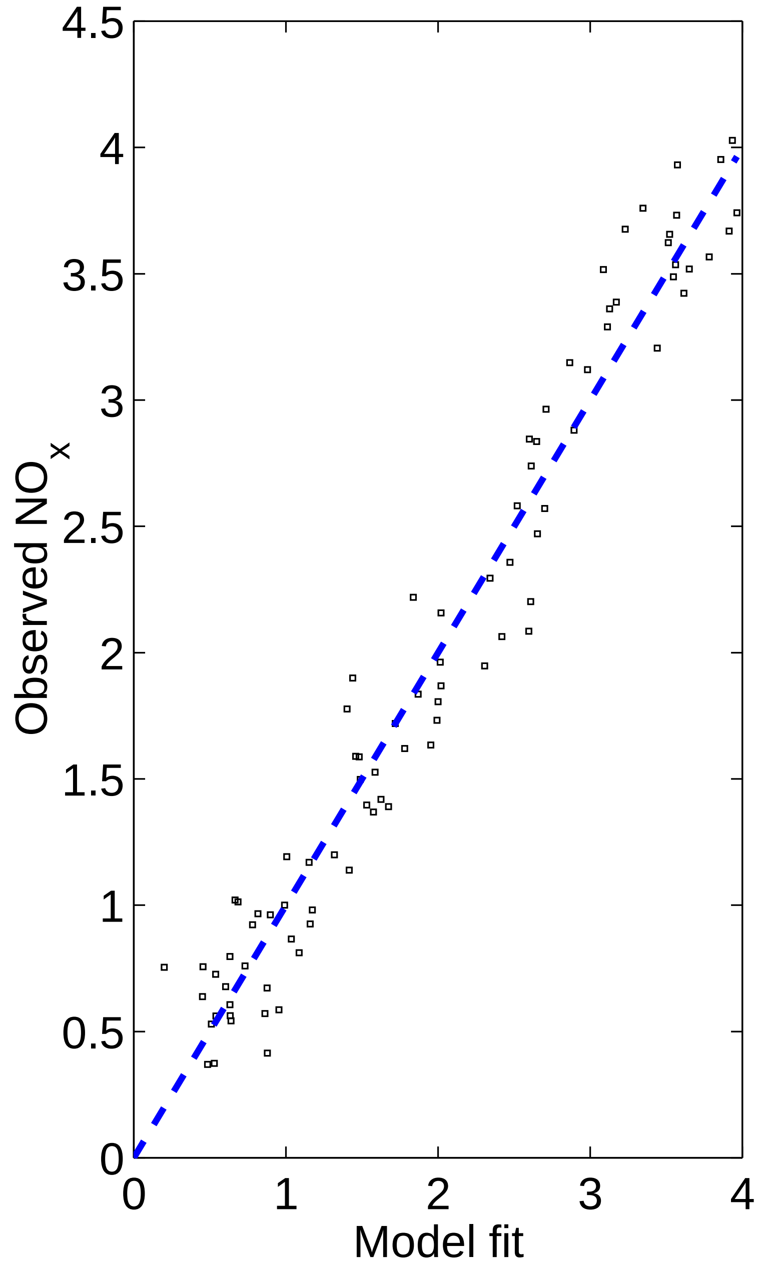
The ethanol data

- Nitrogen oxides in motor exhaust: NO_x (z)
- Compression ratio, C (x), equivalence ratio, E (y)

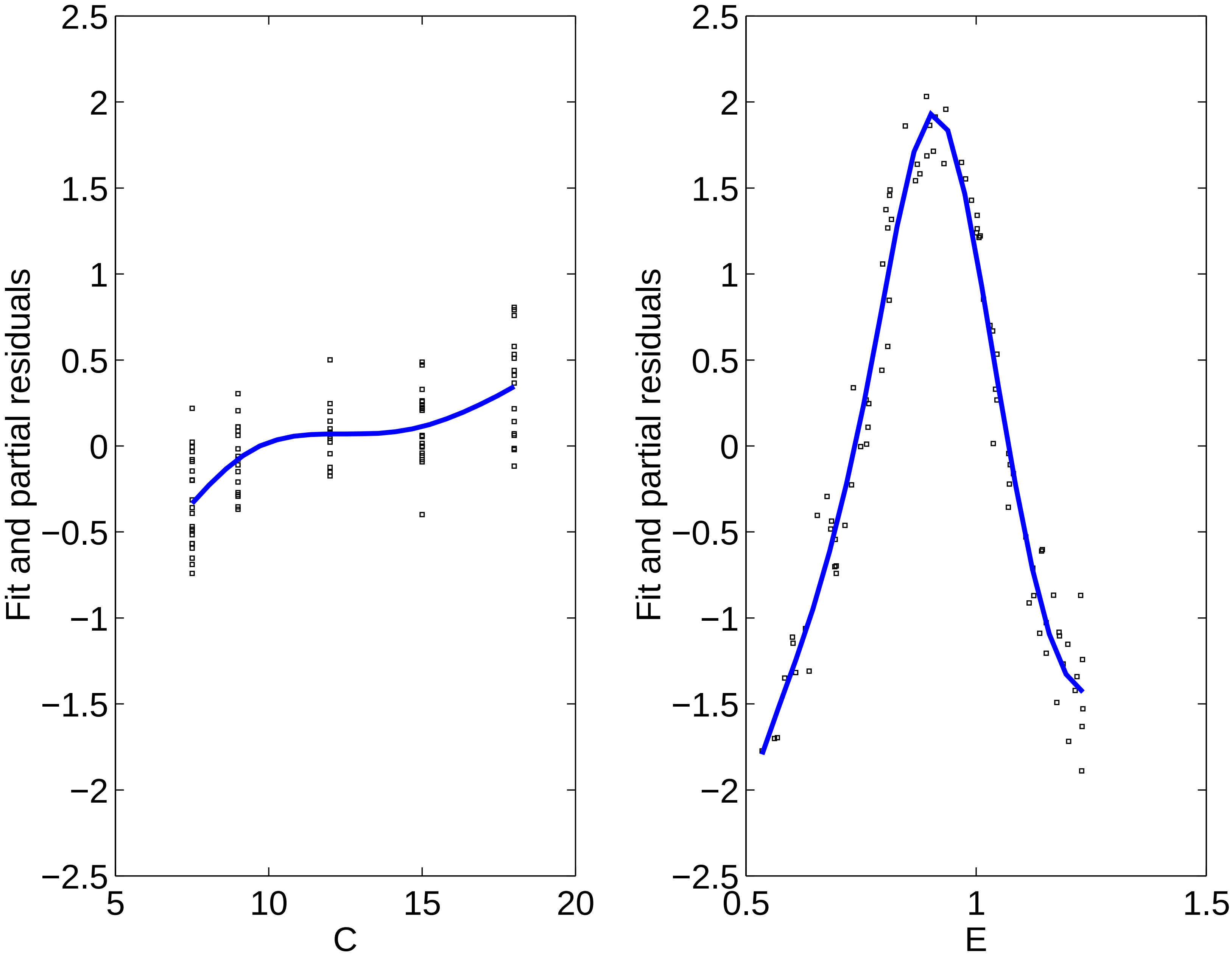


PGAM fit for ethanol data





PGAM components for ethanol data



Wrap-up

- P-splines are useful
- Based on regression, very flexible
- The penalty is the key
- Computation is relatively easy and efficient
- Eilers, PHC and Marx, BD (1996) Flexible smoothing with B-splines and penalties (with Discussion). *Statistical Science* **11**, 89–121.
- Eilers, PHC; Marx, BD and Durbán, M (2015) Twenty years of P-splines. *SORT* **39**, 149–186.