Moving Beyond Linearity

- Basic non-linear models

  one input feature:

  - polynomial regression
  - step functions
  - splines
  - smoothing splines
  - local regression.

  more features:

  - generalized additive models.

Polynomial Regression

- Fit a polynom:
  - linear regression

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_d x_i^d + \epsilon_i$$

  - logistic regression

$$\Pr(y_i > 250 | x_i) = \frac{\exp(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots + \beta_d x_i^d)}{1 + \exp(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots + \beta_d x_i^d)}$$

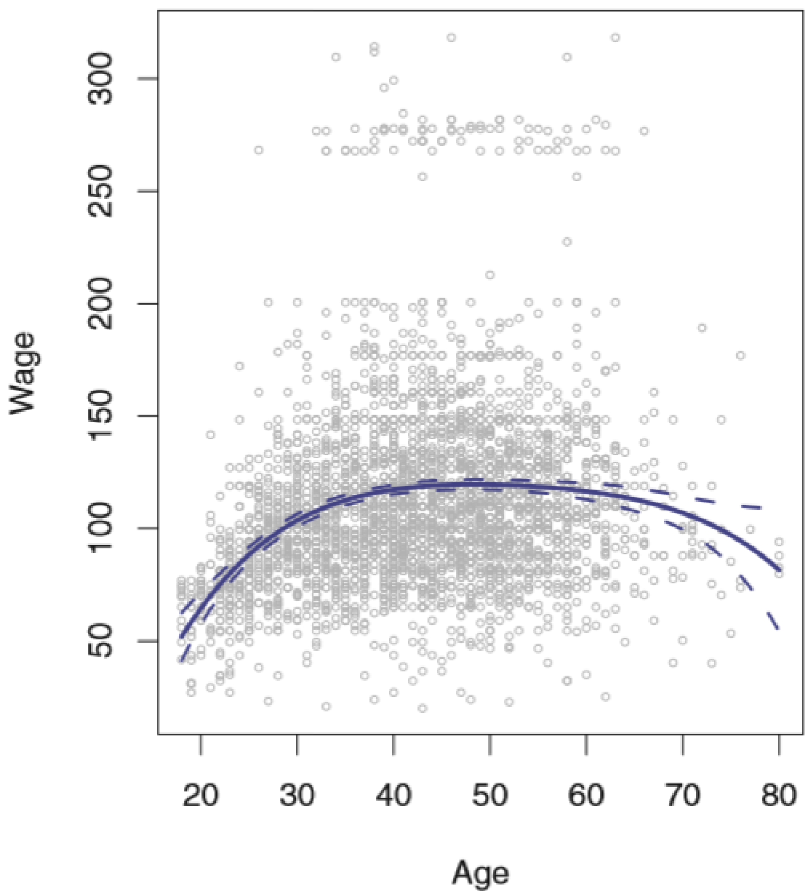
  - (variance)

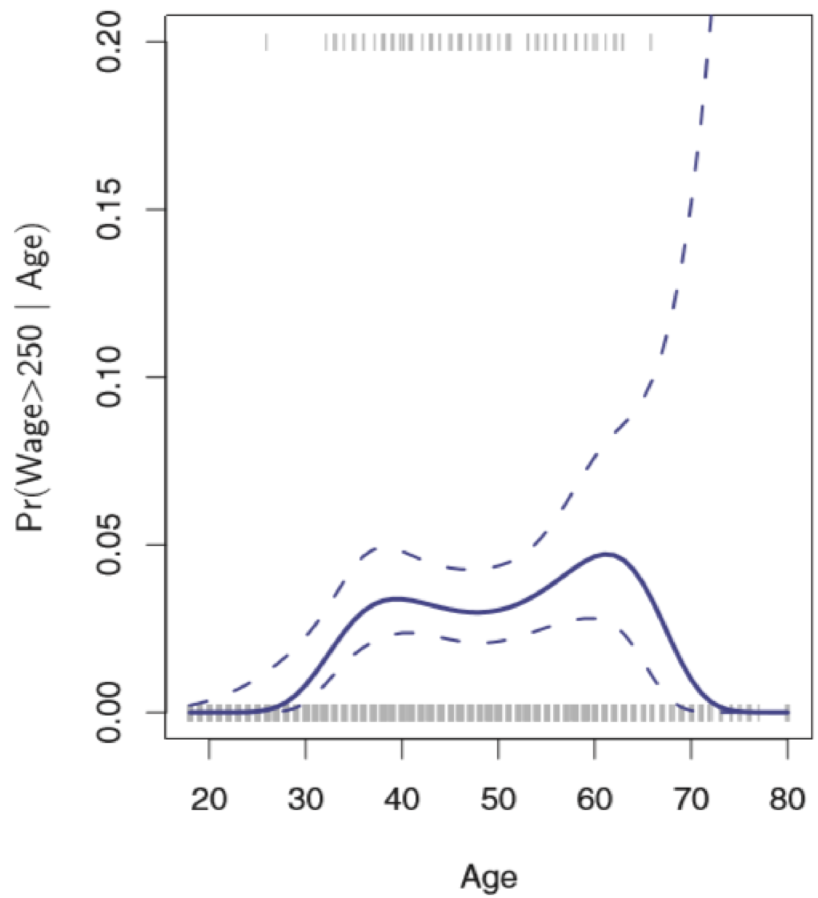
$$\mathrm{Var}[\hat{f}(x_0)] = \ell_0^T \hat{\mathbf{C}} \ell_0 \qquad \ell_0^T = (1, x_0, x_0^2, x_0^3, x_0^4)$$

$\hat{\mathbf{C}}$ is the 5×5 covariance matrix of the $\hat{\beta}_j$

Wage Data Example

- more appropriate than linear fit (ANOVA)
- (usually) high variance close to borders of 'x'

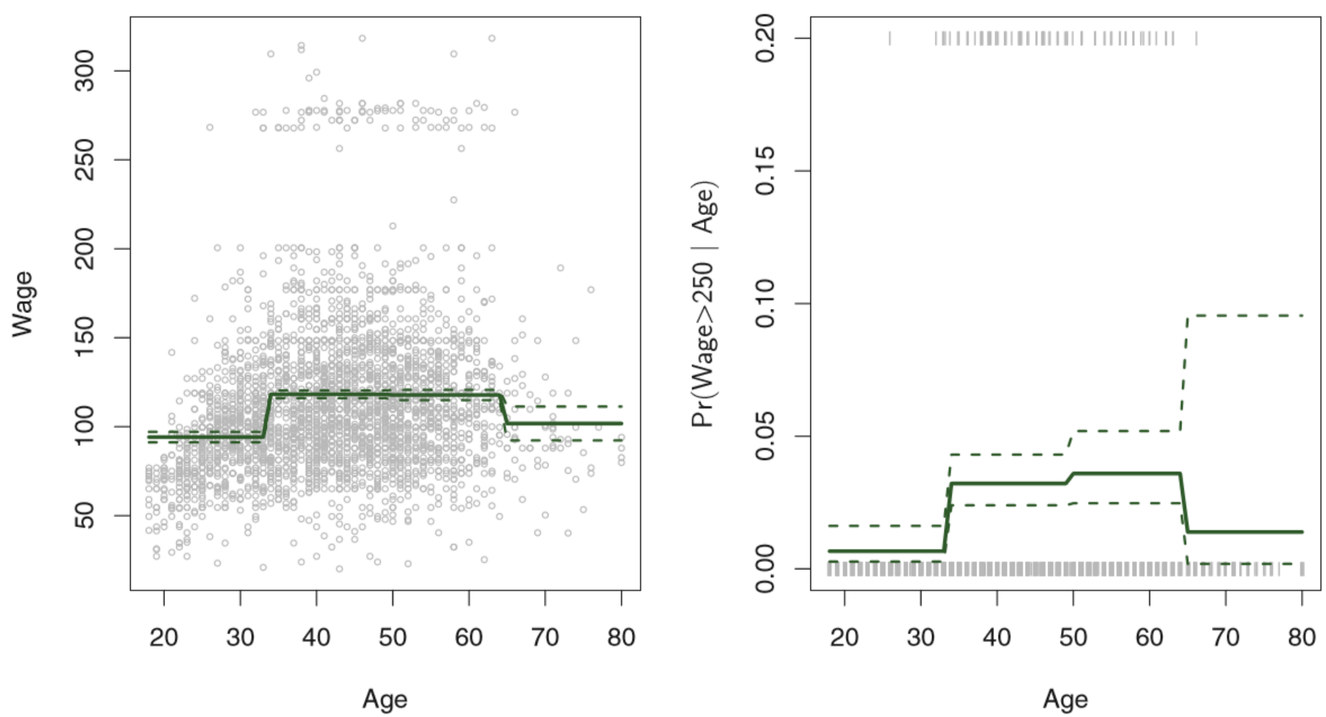




Step Function

- transform continuous predictor to discrete bins,
- fit model using bins:

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + \ldots + \beta_K C_K(x_i) + \epsilon_i$$



Basis Functions

- our model is in the form: $$f_\theta(x) = \sum_{m=1}^{M} \theta_m h_m(x)$$

  for any set of functions $\{h_m(x)\}$

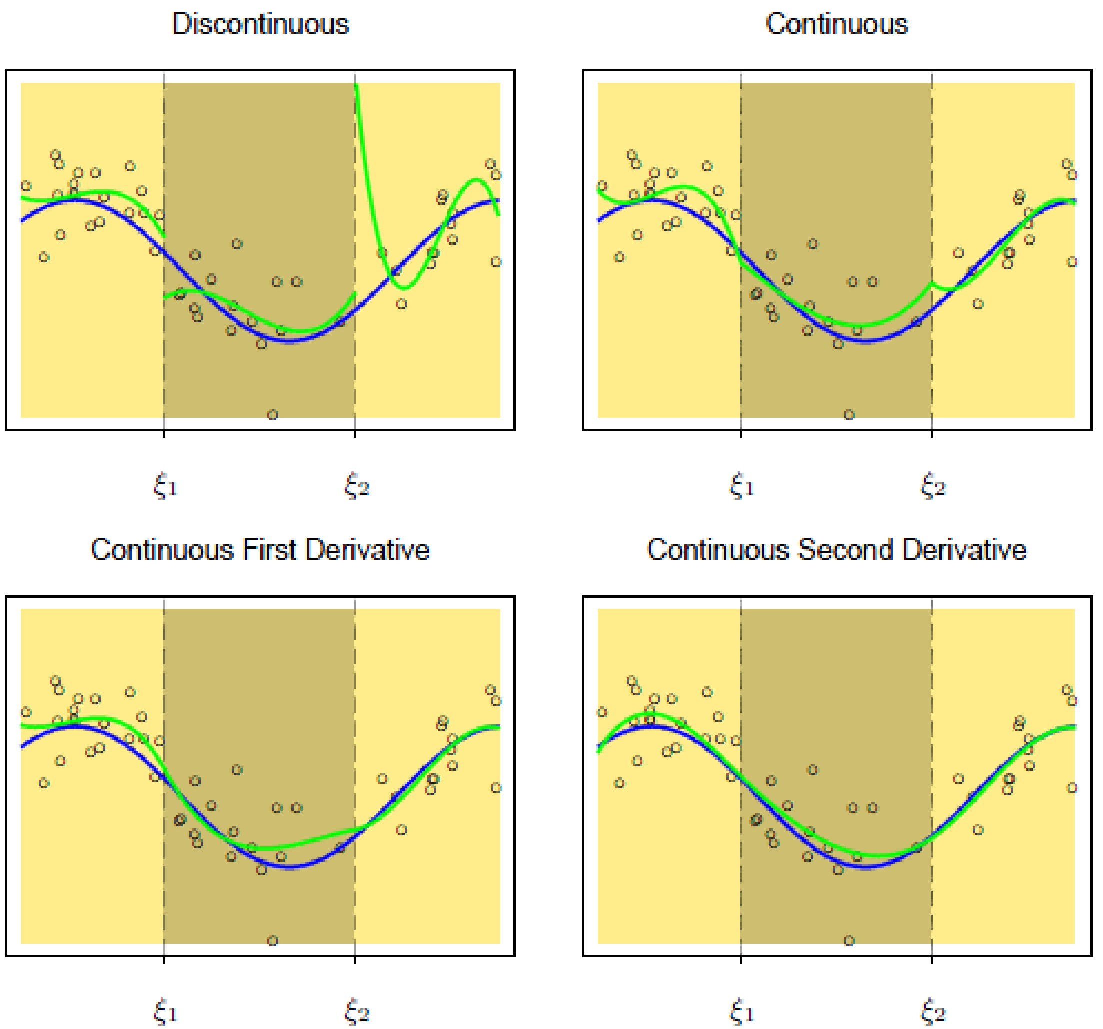
- For example:
  - powers of x
  - interval indicators
  - gaussian kernels
  - truncated polynomials
- Not too many of them to avoid overfitting!

Regression Splines

- Split feature range into intervals
  - knots – split points
  - default: quantiles of data.
- Fit d=3 degree polynomial in each interval
- require to all derivatives up to (d-1) continuous.

- Examples:
  - piecewise constant is spline degree d=0;
  - polynomial fit is a spline without any knot
    (just endpoints)

Continuous Derivative Requirement



The Spline Basis Representation

- A cubic spline with K knots:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_{K+3} b_{K+3}(x_i) + \epsilon_i$$

- truncated power basis function:

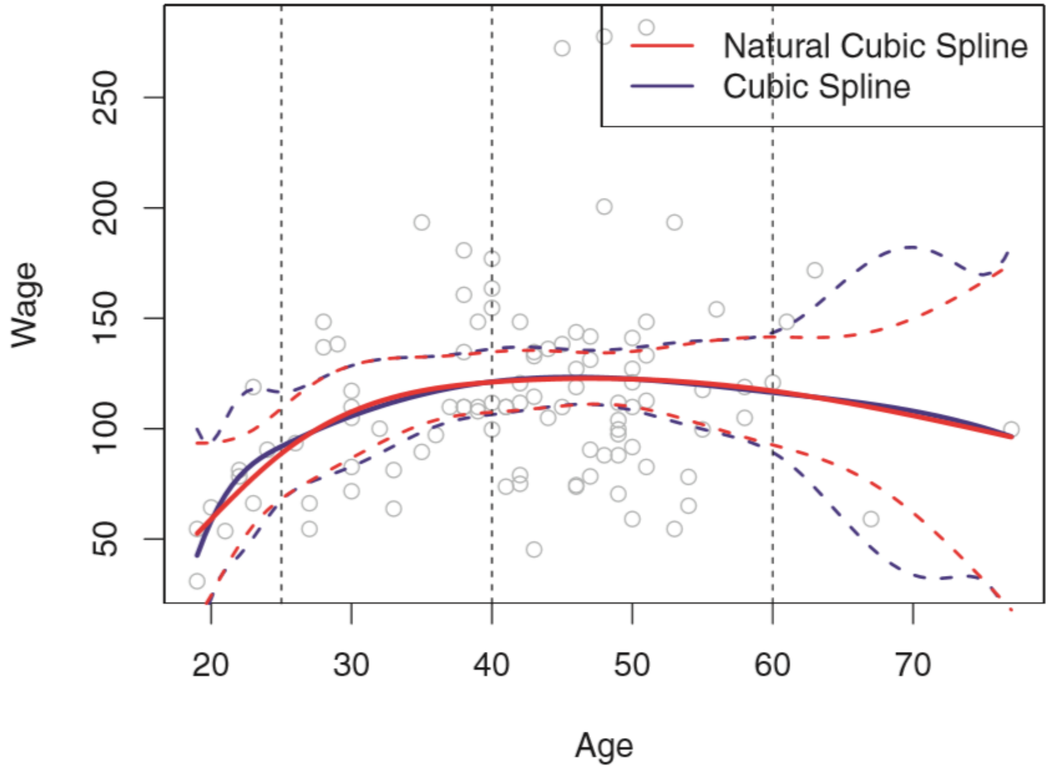
$$h(x, \xi) = (x - \xi)^3_+ = \begin{cases} (x-\xi)^3 & \text{if } x > \xi \\ 0 & \text{otherwise,} \end{cases}$$

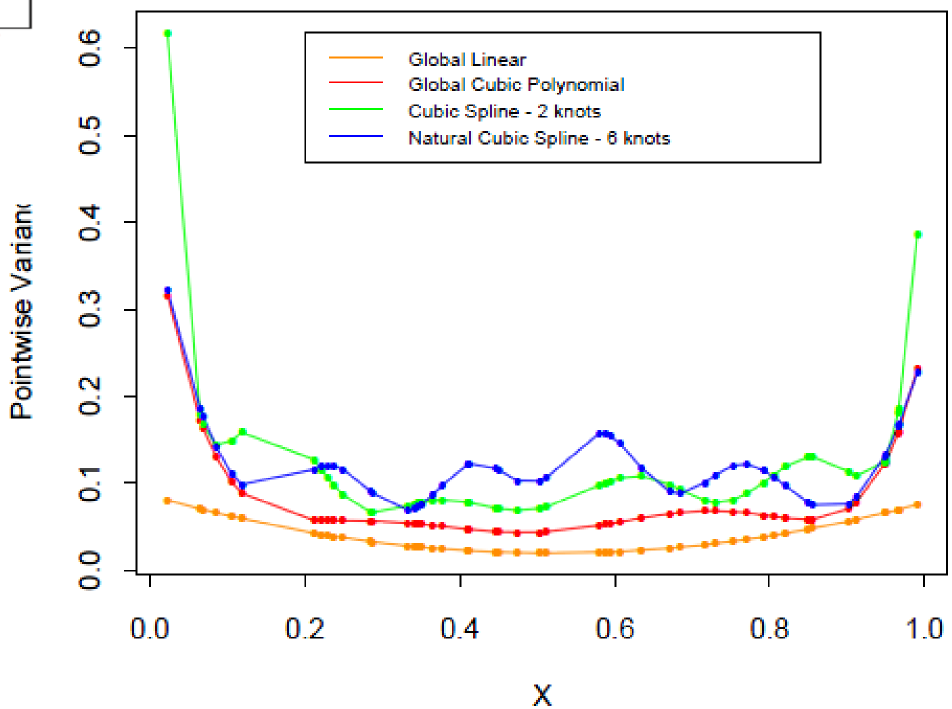
- basis functions:

$$\begin{aligned} h_1(X) &= 1, & h_3(X) &= X^2, & h_5(X) &= (X - \xi_1)^3_+, \\ h_2(X) &= X, & h_4(X) &= X^3, & h_6(X) &= (X - \xi_2)^3_+. \end{aligned}$$

  - powers of X up to degree d=3
  - truncated power basis d=3 for each knot
- standard linear regression to 'new' data matrix.

Spline Example, Pointwise Variance



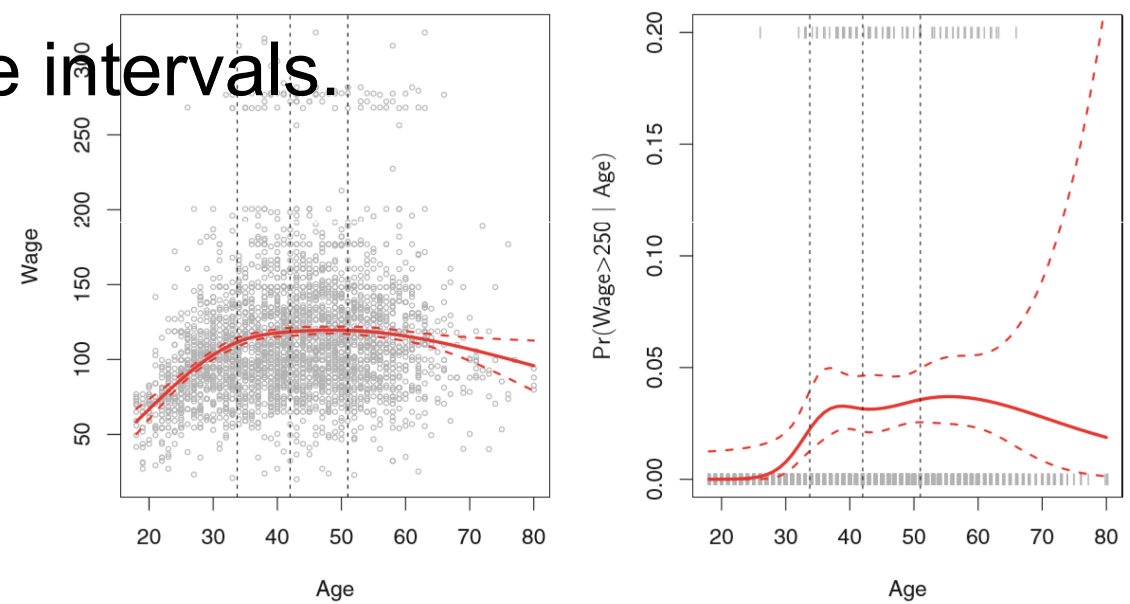


Natural Spline

- Additional constrains: linear on boundaries
  - 2*(d-1) constrains
  - added knots min(x) and max(x),
- generally, more stable estimates at the boundaries,
  - narrower confidence intervals.
- Logistic regression

  transformed input

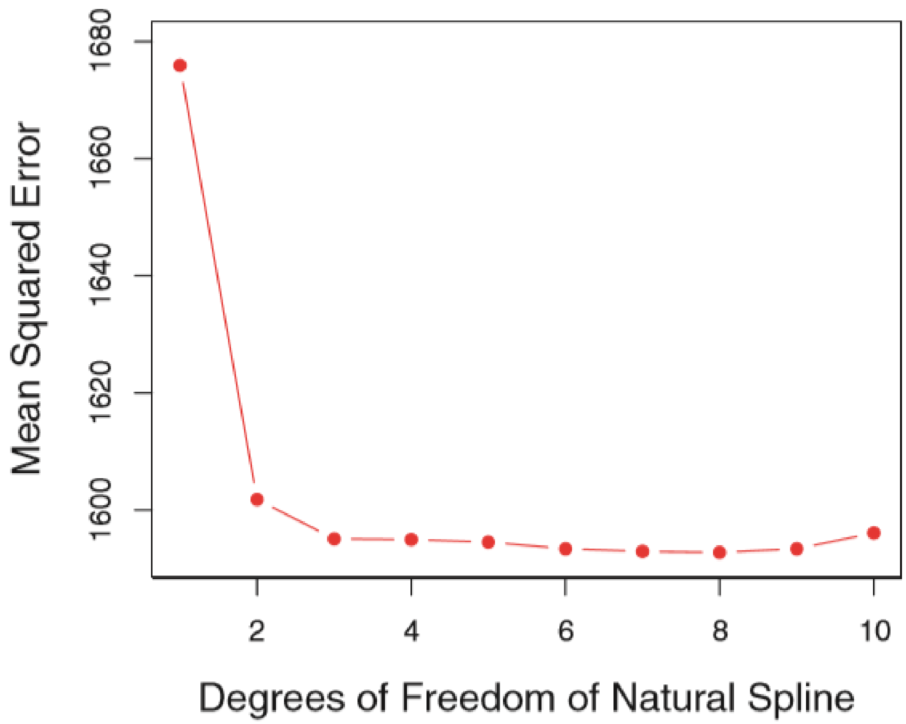
  simillarly.

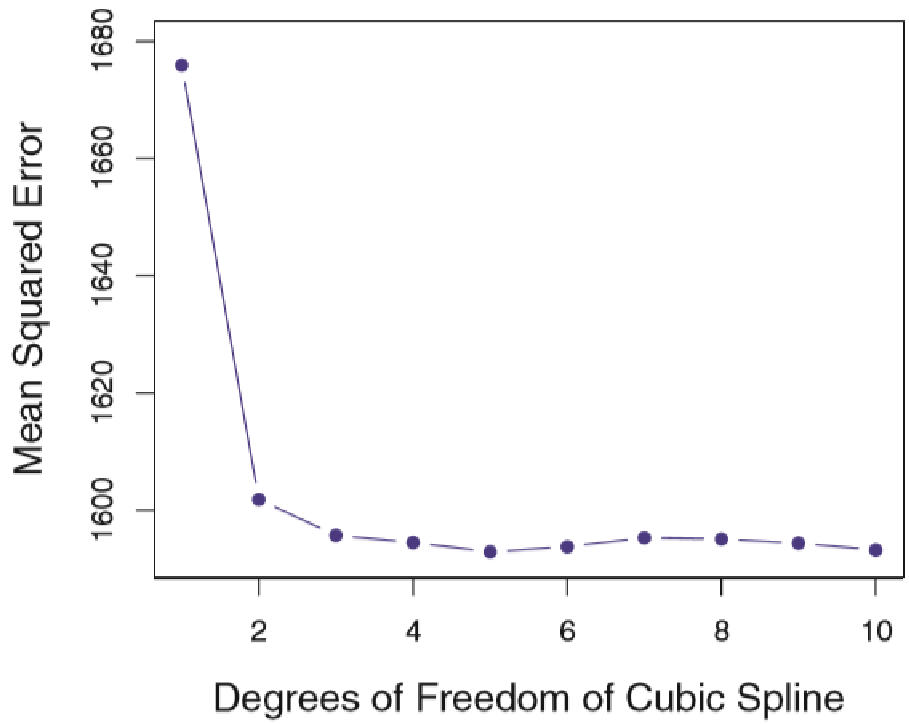


Number and Locations of the Knots

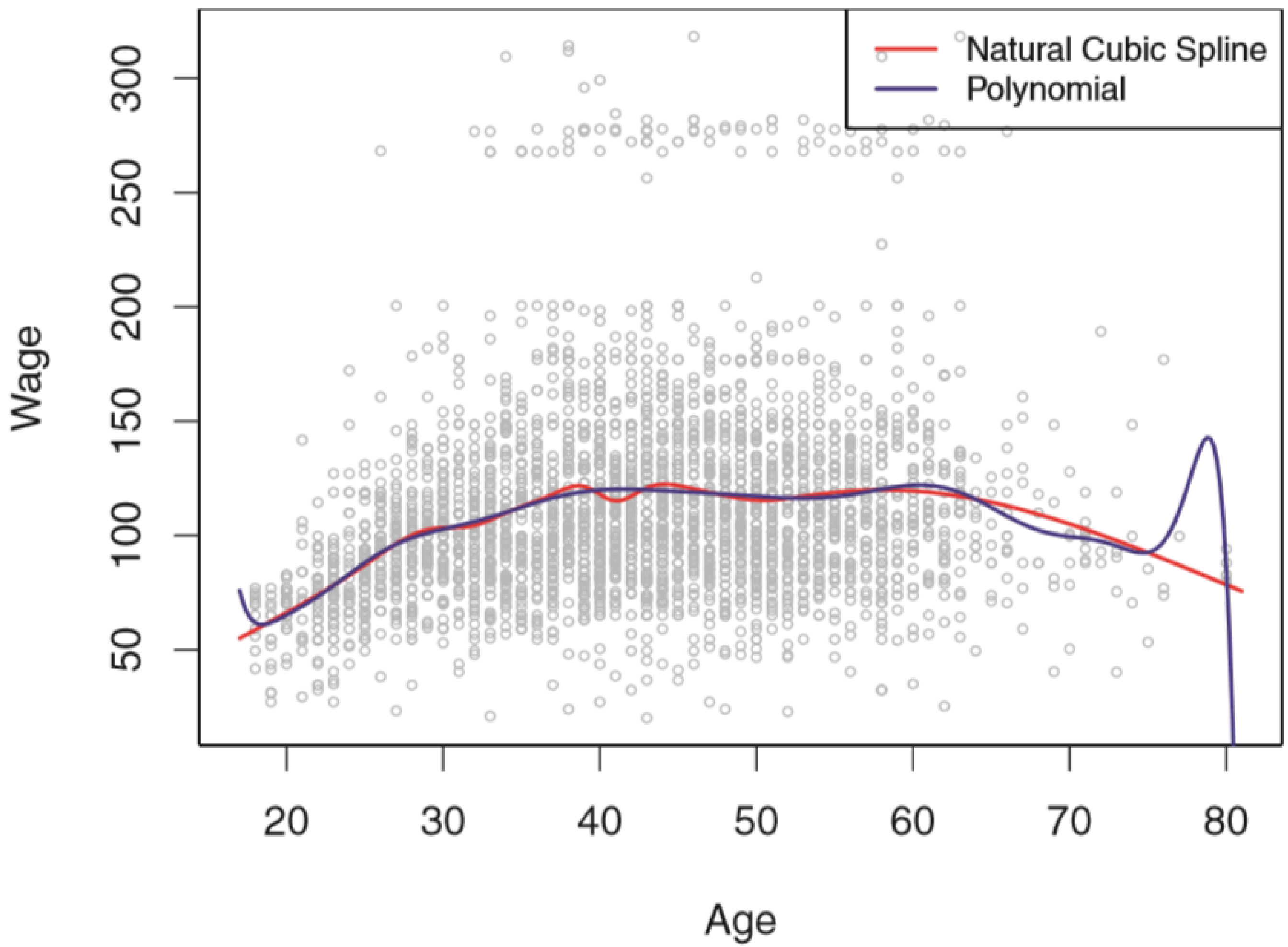
- locations:
  - usually split data to K+1 quantiles
    - (equally sized bins)
  - you may try: more knots in more varying parts
- Number of knots
  - crossvalidation
  - degrees of freedom
    - cubic spline: K knots, K+1 intervals, 3+K parameters
    - natural cubic spline: 4 restrictions, 2 knots more = K+1 p.

CV – Degrees of Freedom





Splines vs. Polynomial Regression



Smoothing Splines

- penalyzation method
- we search g that minimizes:

$$\sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$$

- lambda is a tuning parameter,
  - lambda 0 – interpolation
  - lambda infinity – strait line, linear regression.
- g(x) is a natural cubic spline with knots x_1, ...,x_n
  - shrunken version of that from previous section caused by lambda.

Choosing the Smoothing Parameter

- crossvalidation
  - one-leave-out can be done efficiently
- g can be expressed as (S is nxn matrix):

$$\hat{\mathbf{g}}_\lambda = \mathbf{S}_\lambda \mathbf{y}$$

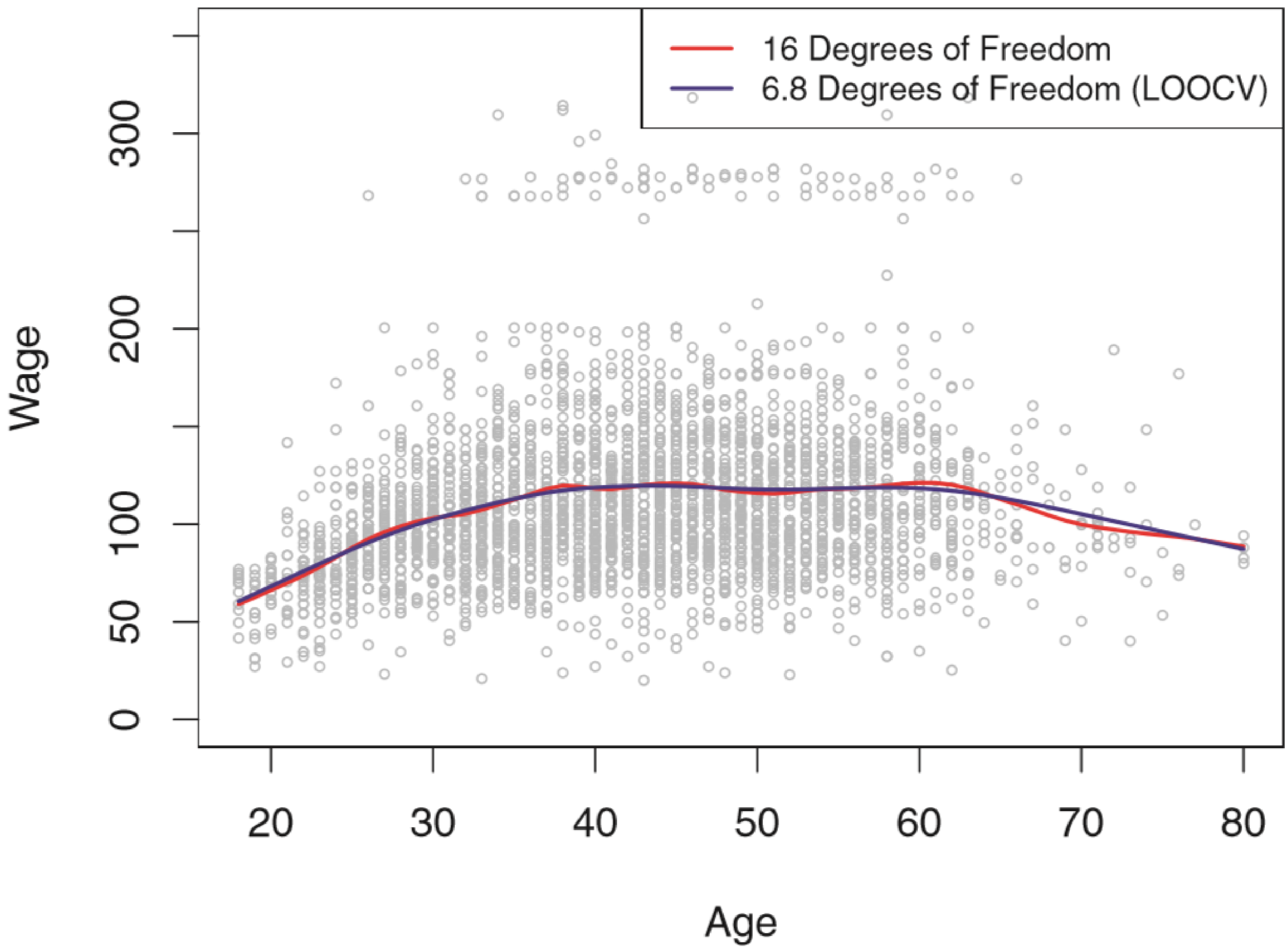
- Residual sum of squares:

$$\mathrm{RSS}_{cv}(\lambda) = \sum_{i=1}^{n} (y_i - \hat{g}_\lambda^{(-i)}(x_i))^2 = \sum_{i=1}^{n} \left[\frac{y_i - \hat{g}_\lambda(x_i)}{1 - \{\mathbf{S}_\lambda\}_{ii}} \right]^2$$

- Degrees of freedom:

$$df_\lambda = \sum_{i=1}^{n} \{\mathbf{S}_\lambda\}_{ii}$$

Smoothing Spline Example

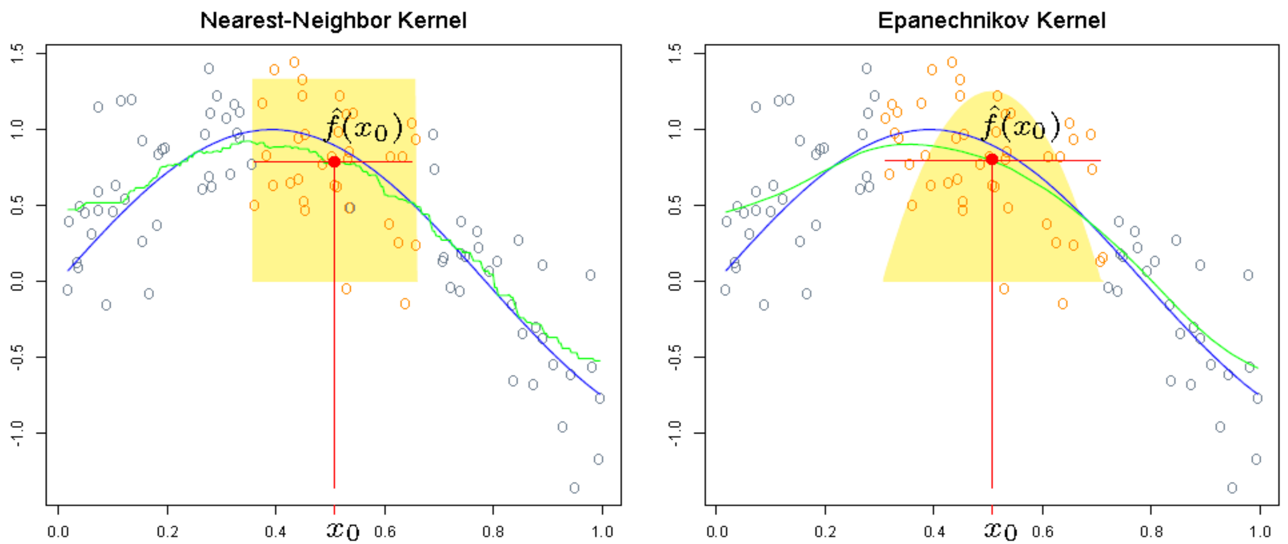


Kernel Methods

- generalization of nearest-neighbour method
- kernel function defines weights of neighbour examples
- prediction is the weighted average

  (Nadaraya-Watson)

$$\hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)}$$



Common Kernels

- Gaussian

$$K_\lambda(x_0, x) = \frac{1}{\lambda} \exp\left[-\frac{||x - x_0||^2}{2\lambda}\right]$$

- Epanechnikov

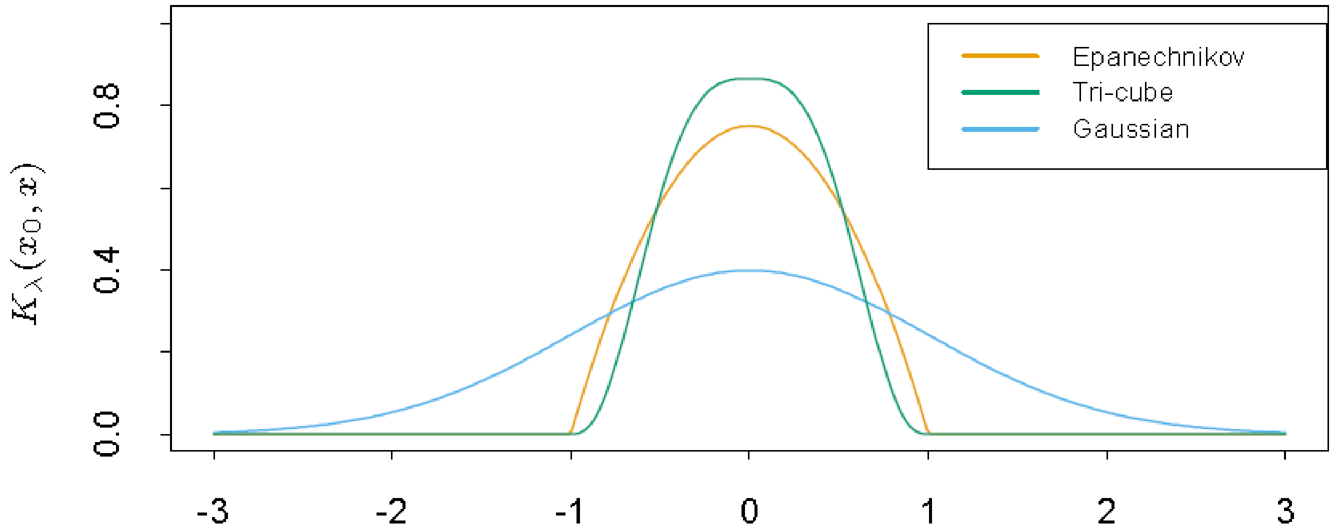
$$K_\lambda(x_0, x) = D\left(\frac{|x - x_0|}{\lambda}\right),$$

with

$$D(t) = \begin{cases} \frac{3}{4}(1 - t^2) & \text{if } |t| \le 1; \\ 0 & \text{otherwise.} \end{cases}$$

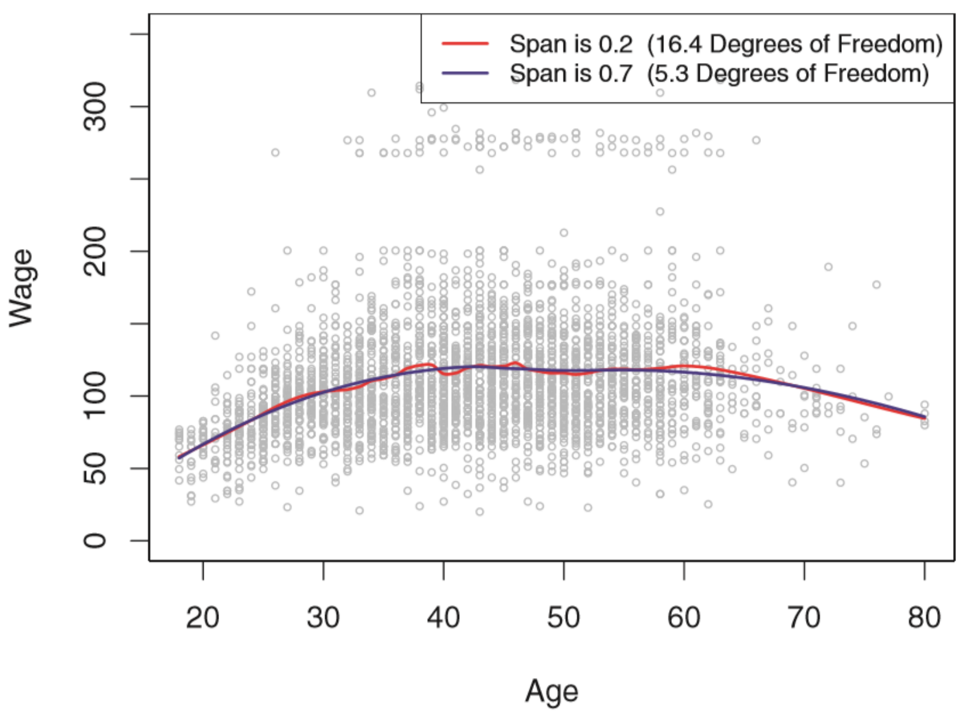
Tri-cube

$$D(t) = \begin{cases} (1 - |t|^3)^3 & \text{if } |t| \le 1; \\ 0 & \text{otherwise} \end{cases}$$



Local Regression

- fit least squares regression on weighted RSS, minimizing: $$\sum_{i=1}^{n} K_{i0}(y_i - \beta_0 - \beta_1 x_i)^2$$
- Fitted value is given by: $\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$
- Span of the kernel
  - 'inverse' degrees of freedom.



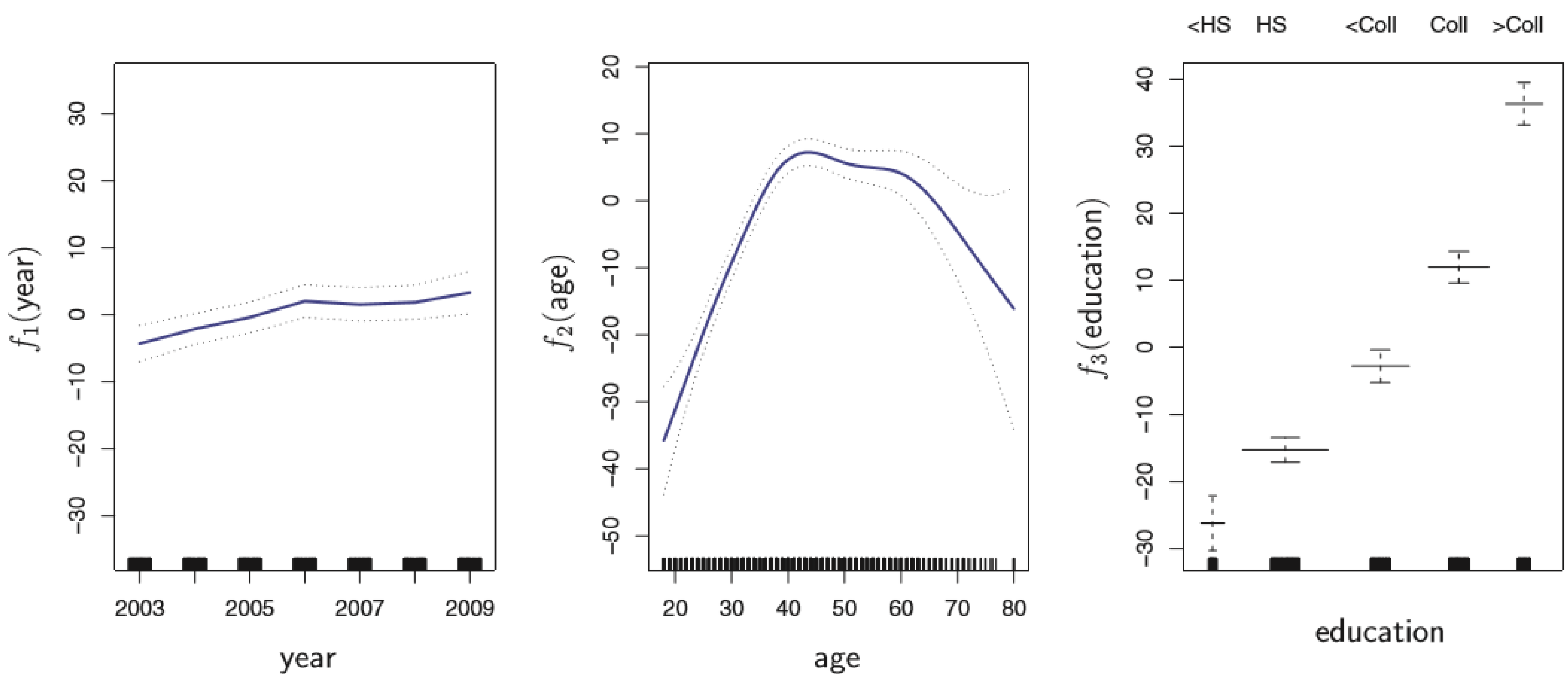
Generalized Additive Models (GAM)

- we handele multiple predictors,
- allow non-linear transformation of them,
- maintain additivity w.r.t. parameters, i.e.

$$\begin{aligned} y_i &= \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \epsilon_i \\ &= \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \epsilon_i \end{aligned}$$

GAM Example

- Model: $\texttt{wage} = \beta_0 + f_1(\texttt{year}) + f_2(\texttt{age}) + f_3(\texttt{education}) + \epsilon$
  - year: smoothing spline df=4
  - age: smoothing spline df=5
  - education: categorical (standard dummy vars.)



Evaluation: Backfitting

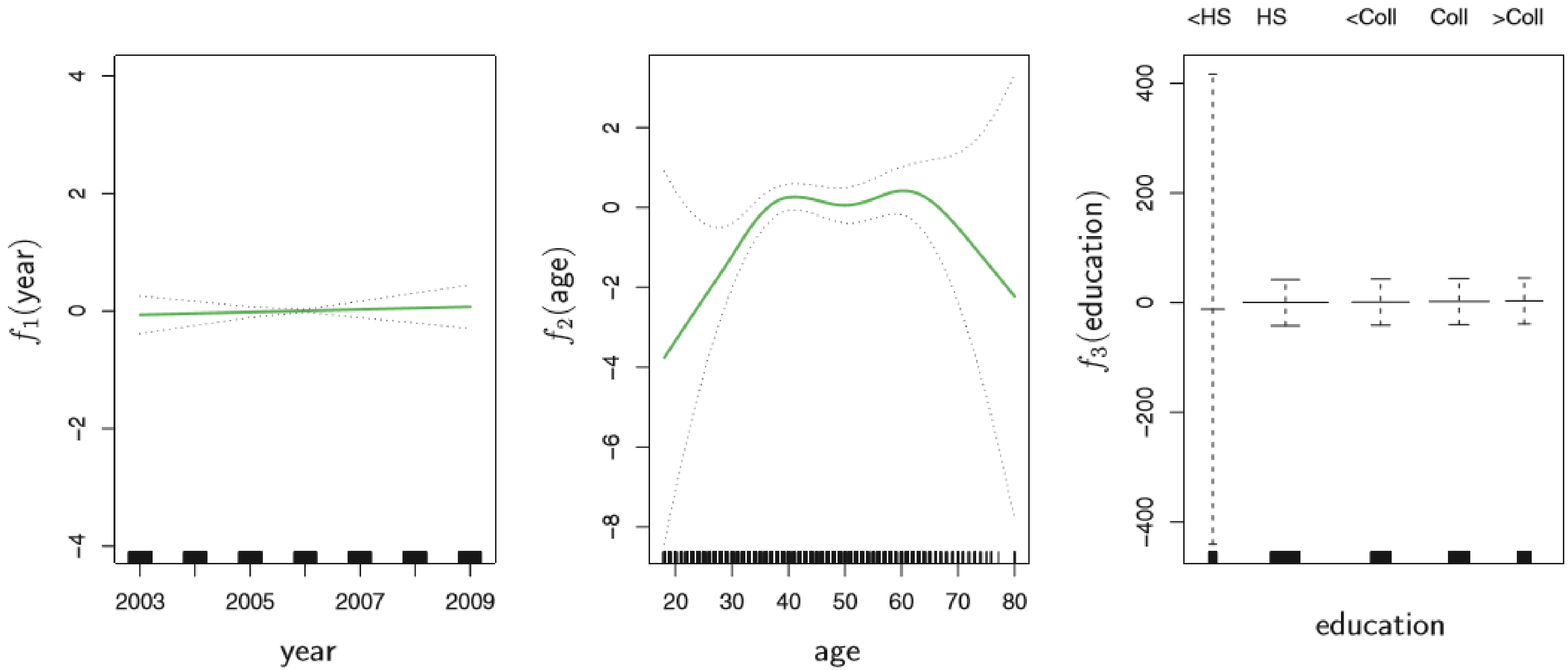
- Natural splines:
  - just join the transformed input matrixes,
  - perform linear regression on transformed input.
- Learning Smooting Splines requires Backfitting:
  - repeatedly update the fit for each predictor in turn
  - holding others fixed
  - apply fitting method to partial residuals:

$$r_i = y_i - f_1(x_{i1}) - f_2(x_{i2})$$

GAM for Classification

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + f_1(X_1) + f_2(X_2) + \cdots + f_p(X_p)$$

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 \times \texttt{year} + f_2(\texttt{age}) + f_3(\texttt{education})$$



Structural Regression Models

- penalized methods, bayesian methods

  Lasso, Ridge reg. $$\mathrm{PRSS}(f;\lambda) = \mathrm{RSS}(f) + \lambda J(f).$$

  smoothing spline $$\mathrm{PRSS}(f;\lambda) = \sum_{i=1}^{N}(y_i - f(x_i))^2 + \lambda \int [f''(x)]^2 dx.$$

- kernal methods a local regression

$$K_\lambda(x_0, x) = \frac{1}{\lambda}\exp\left[-\frac{||x - x_0||^2}{2\lambda}\right] \qquad \hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)}$$

$$\mathrm{RSS}(f_\theta, x_0) = \sum_{i=1}^{N} K_\lambda(x_0, x_i)(y_i - f_\theta(x_i))^2,$$

- dictionary methods, basis functions

$$f_\theta(x) = \sum_{m=1}^{M} \theta_m h_m(x),$$

$$b_1(x) = 1,\ b_2(x) = x, \text{ and } b_{m+2}(x) = (x - t_m)_+,\ m = 1, \dots, M-2,$$

t_m is the mth knot,

Splines in More Dimensions

- Too many products of one-dim. basic functions.
- We need a way to select.

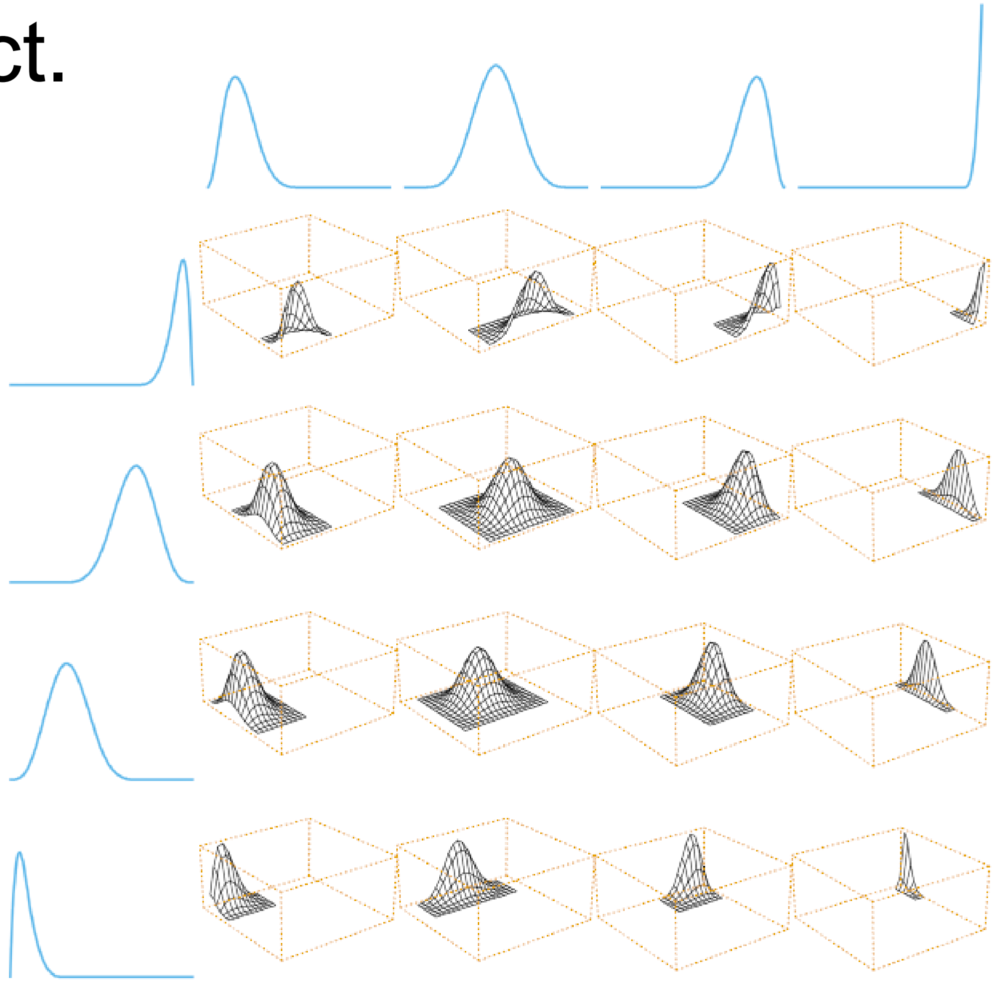


FIGURE 5.10. *A tensor product basis of B-splines, showing some selected pairs. Each two-dimensional function is the tensor product of the corresponding one dimensional marginals.*