Moving Beyond Linearity

ddebarr@uw.edu

2017-02-16

xkcd.com

Course Outline

1. Introduction to Statistical Learning
2. Linear Regression
3. Classification
4. Resampling Methods
5. Linear Model Selection and Regularization
6. Moving Beyond Linearity
7. Tree-Based Methods
8. Support Vector Machines
9. Unsupervised Learning
10. Neural Networks and Genetic Algorithms

Agenda

Modeling Nonlinear Relationships

- Polynomial regression extends the linear model by adding extra predictors
- Step functions cut the range of a variable into “k” distinct regions
- Regression splines are more flexible than polynomials and step functions
- Smoothing splines include a smoothness penalty
- Local regression makes use of distance information to perform regression
- Generalized Additive Models (GAMs) allow us to extend the above approaches to deal with multiple predictors

Polynomial Regression

- Simple Linear Regression

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

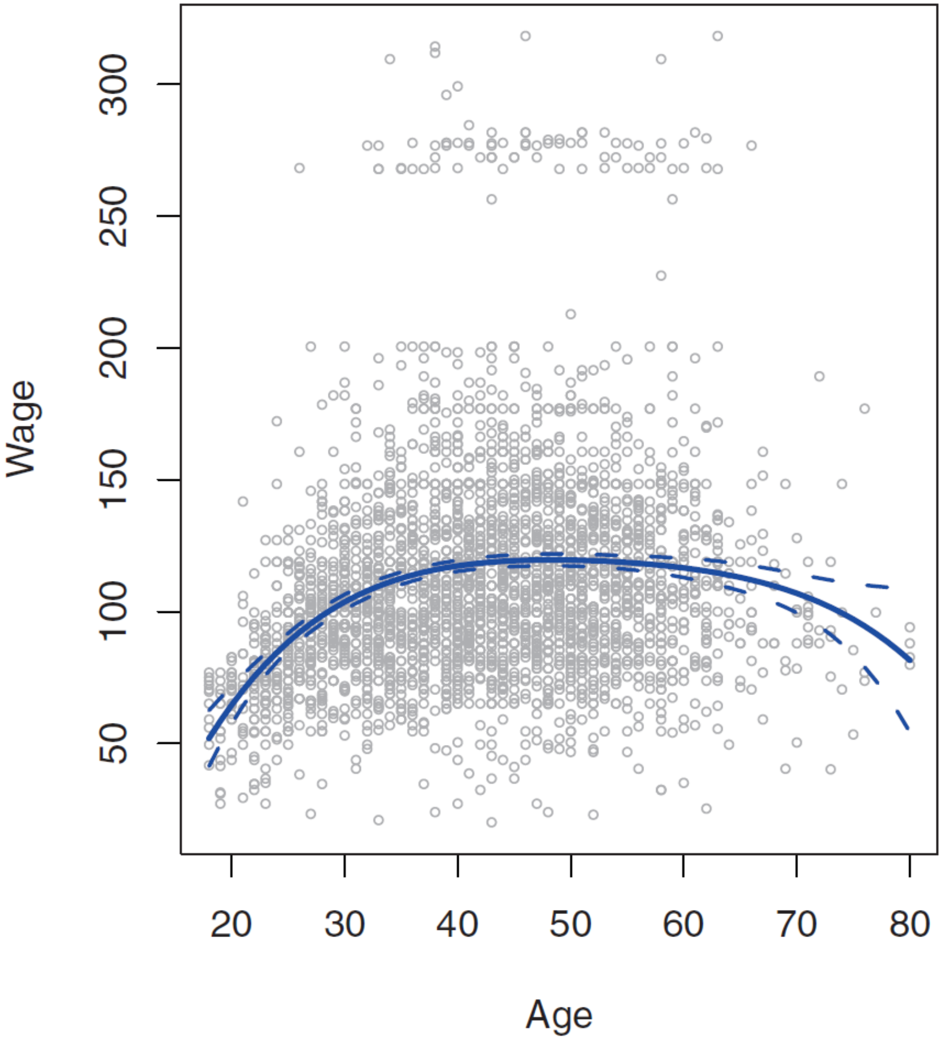
- Polynomial Linear Regression [still only one predictor]

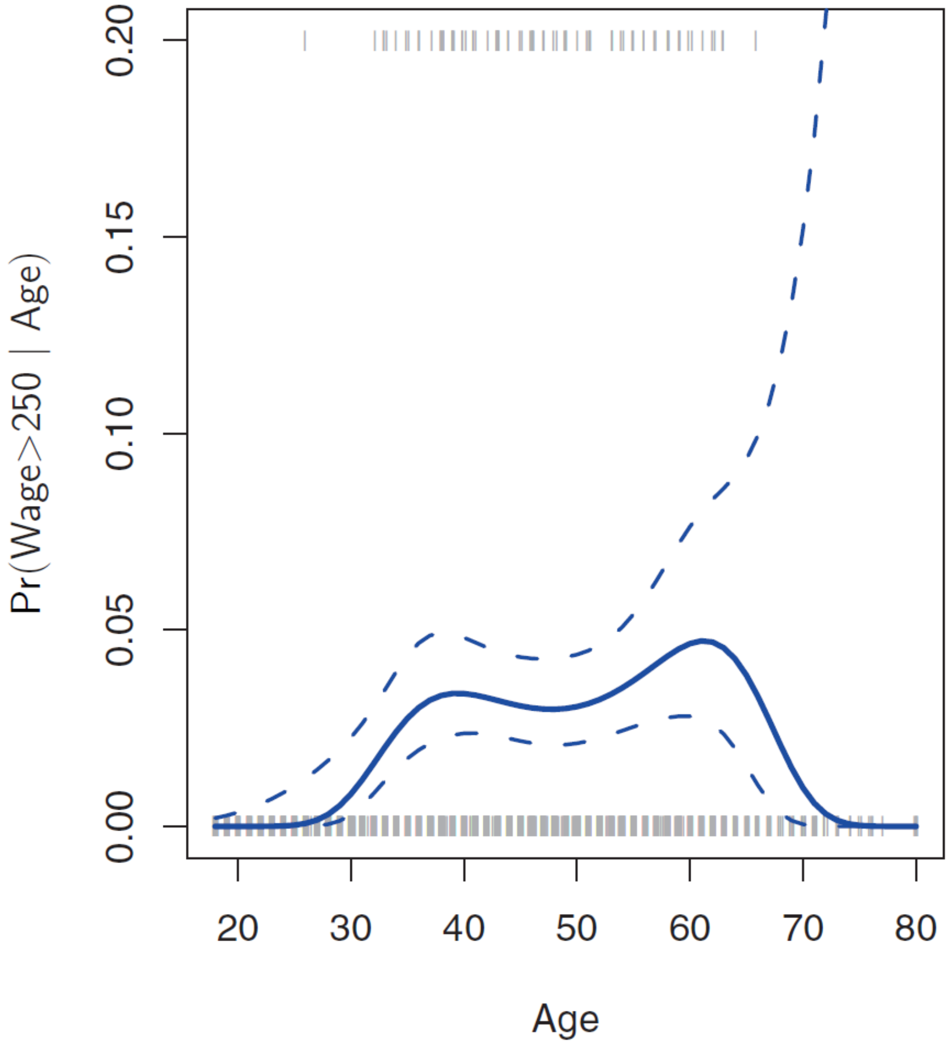
$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_d x_i^d + \epsilon_i$$

Degree=4 Polynomial Examples

$$\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 x_0^2 + \hat{\beta}_3 x_0^3 + \hat{\beta}_4 x_0^4$$

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





Step Functions

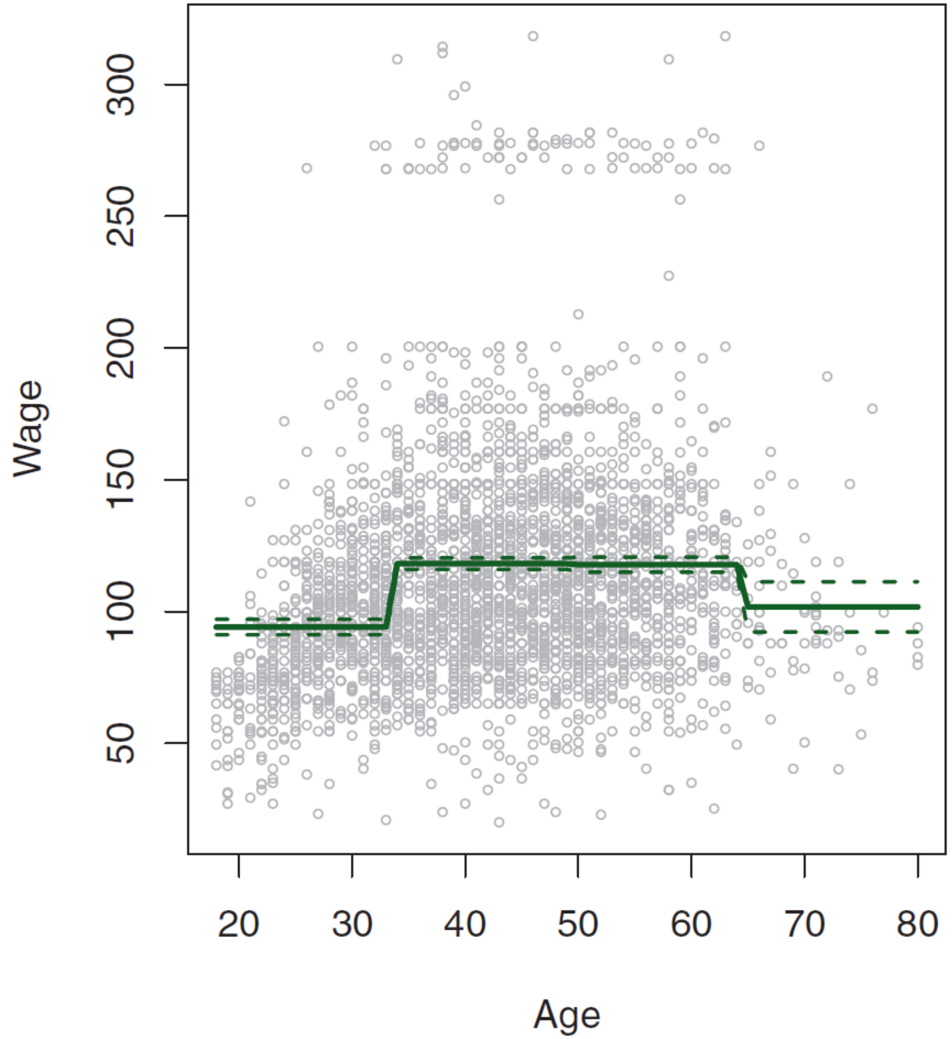
$$
\begin{aligned}
C_0(X) &= I(X < c_1) \\
C_1(X) &= I(c_1 \leq X < c_2) \\
C_2(X) &= I(c_2 \leq X < c_3) \\
&\vdots \\
C_{K-1}(X) &= I(c_{K-1} \leq X < c_K) \\
C_K(X) &= I(c_K \leq X)
\end{aligned}
$$

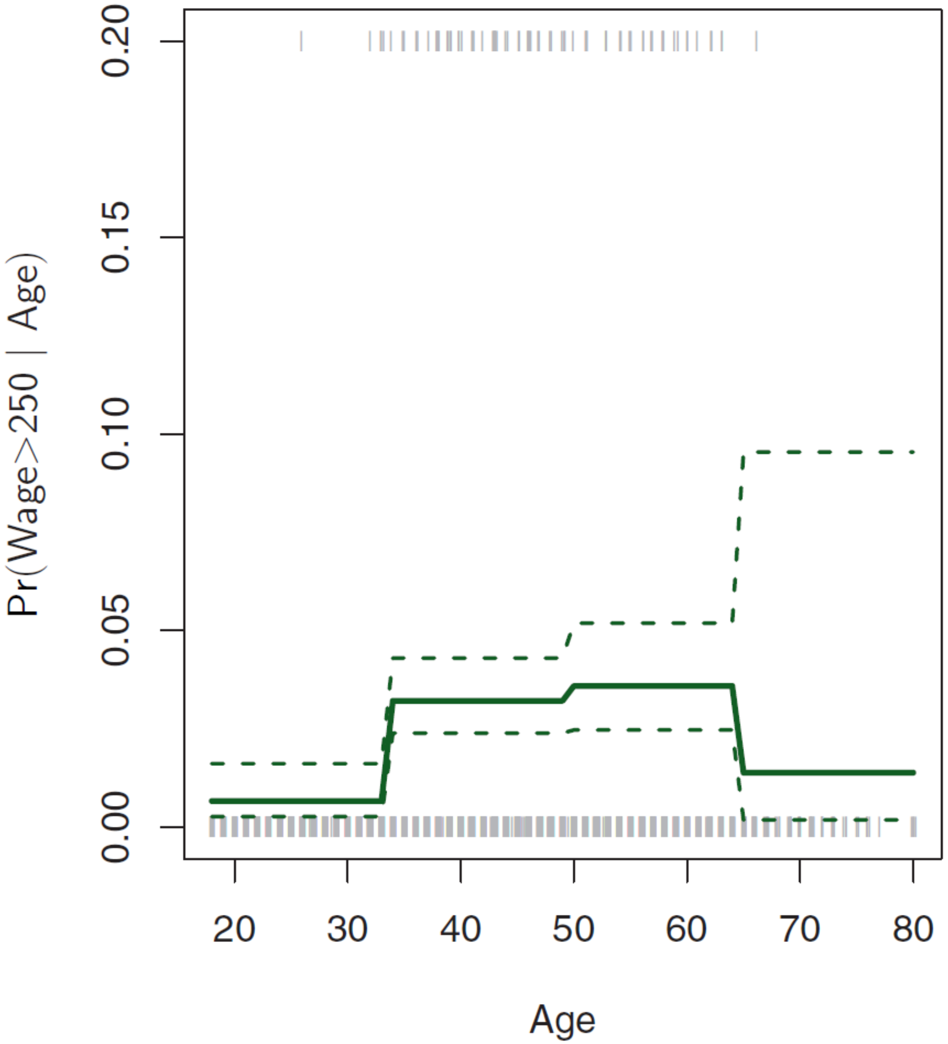
$I(c_K \leq X)$ equals 1 if $c_K \leq X$, and equals 0 otherwise

Piecewise Constant: Step Function Example

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

$$\Pr(y_i > 250|x_i) = \frac{\exp(\beta_0 + \beta_1 C_1(x_i) + \ldots + \beta_K C_K(x_i))}{1 + \exp(\beta_0 + \beta_1 C_1(x_i) + \ldots + \beta_K C_K(x_i))}$$





Basis Functions

- Basis Functions Model

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \beta_3 b_3(x_i) + \ldots + \beta_K b_K(x_i) + \epsilon_i$$

- Polynomial Regression Example

$$b_j(x_i) = x_i^j$$

- Step Function Example

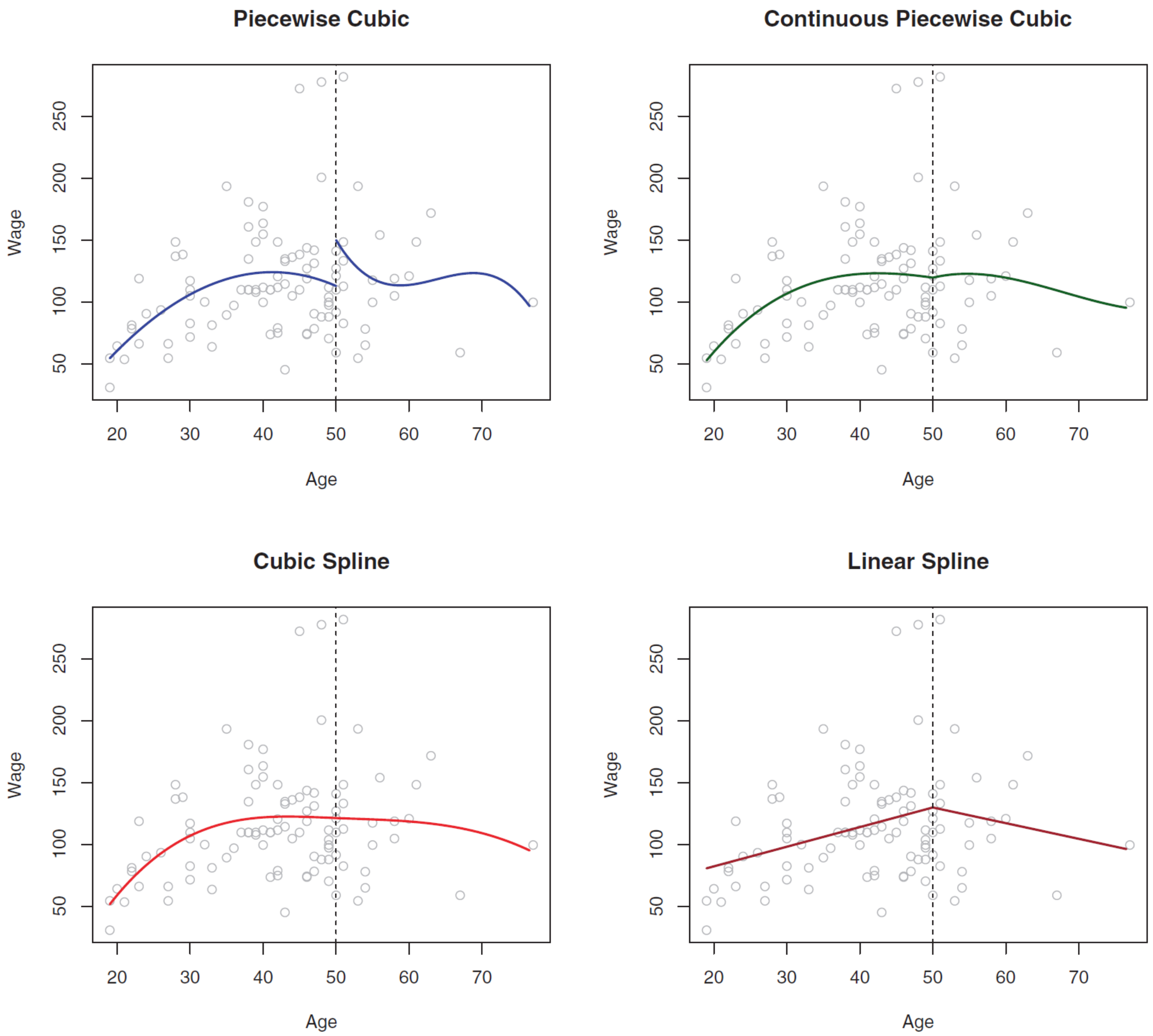
$$b_j(x_i) = I(c_j \leq x_i < c_{j+1})$$

Regression Splines: Piecewise Polynomials

$$y_i = \begin{cases} \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \epsilon_i & \text{if } x_i < c \\ \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \epsilon_i & \text{if } x_i \geq c \end{cases}$$

‘c’ is called a knot

Constraints and Splines



Spline Basis Representation

- Model for a cubic spline with ‘k’ knots

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

- One truncated power basis function per knot: 4 + K degrees of freedom

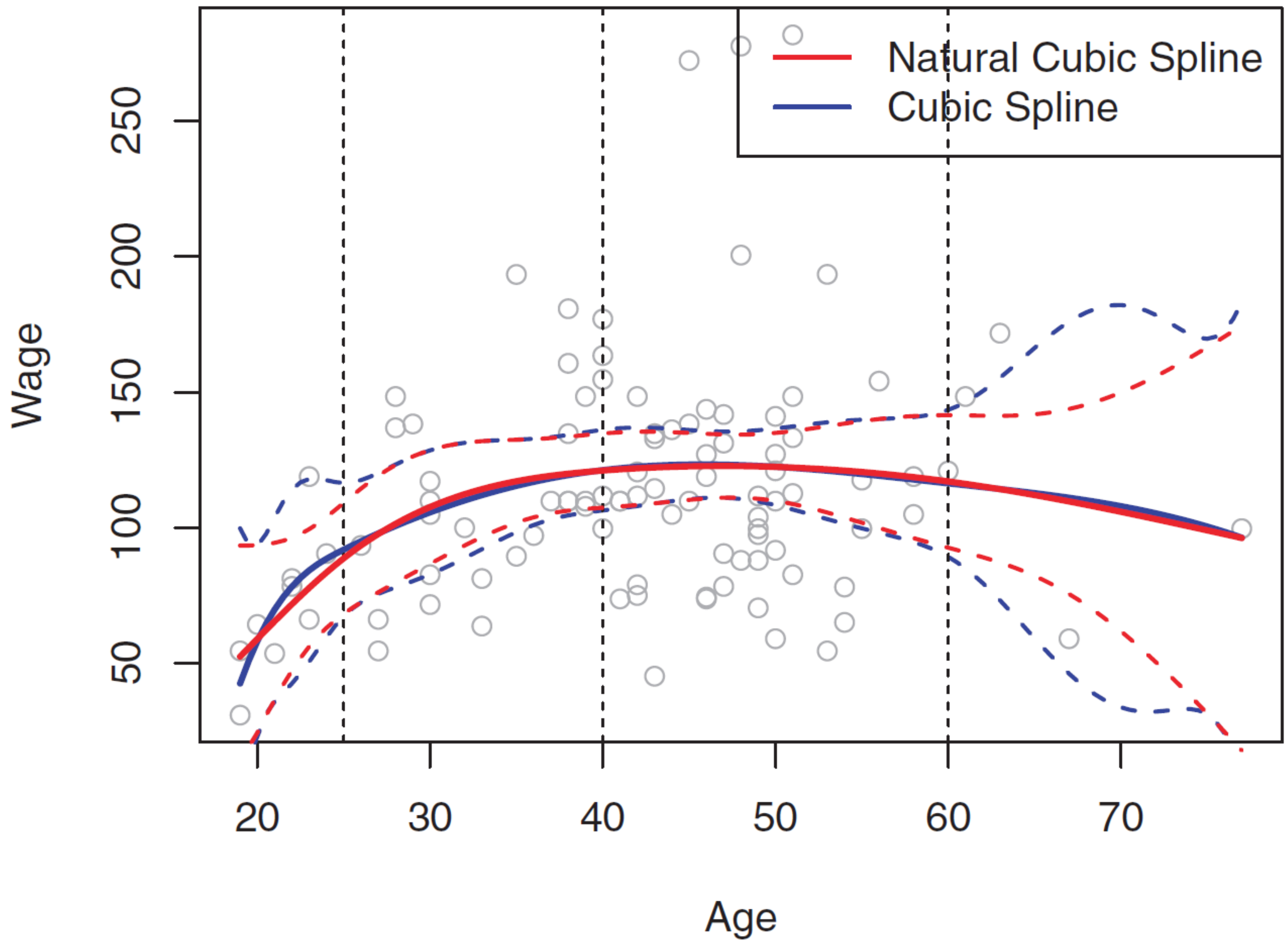
$$h(x,\xi) = (x-\xi)^3_+ = \begin{cases} (x-\xi)^3 & \text{if } x > \xi \\ 0 & \text{otherwise} \end{cases}$$

The Greek letter ξ is pronounced zi
[looks cooler than using ‘c’?]

Basis Splines Example

```
> library(ISLR)
> age.limits =range(Wage$age)
> age.grid = seq(from = age.limits[1], to = age.limits[2])    # 18 .. 80
>
> library(splines)
> model1 = lm(wage ~ bs(age, knots = c(25, 40, 60)), data = Wage)
> predictions1 = predict(model1, Wage)
> predictions1[1:5]
   231655     86582    161300    155159     11443
 60.49371  82.84196 119.39567 118.91764 119.41254
>
> X = cbind(Wage$age,
+           Wage$age^2,
+           Wage$age^3,
+           ifelse(Wage$age > 25, (Wage$age - 25)^3, 0),
+           ifelse(Wage$age > 40, (Wage$age - 40)^3, 0),
+           ifelse(Wage$age > 60, (Wage$age - 60)^3, 0))
> model2 = lm(Wage$wage ~ X)
> predictions2 = predict(model2, data.frame(X))
> predictions2[1:5]
        1         2         3         4         5
 60.49371  82.84196 119.39567 118.91764 119.41254
```

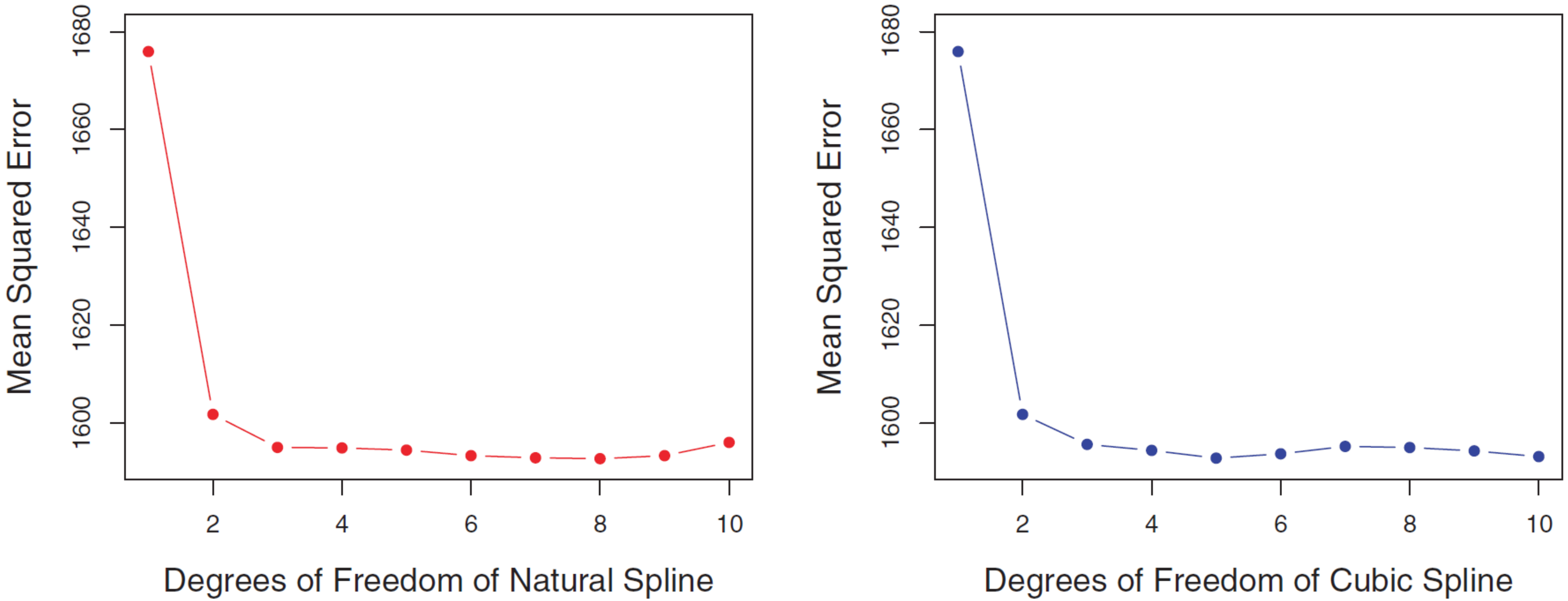
Natural Cubic Spline versus Cubic Spline



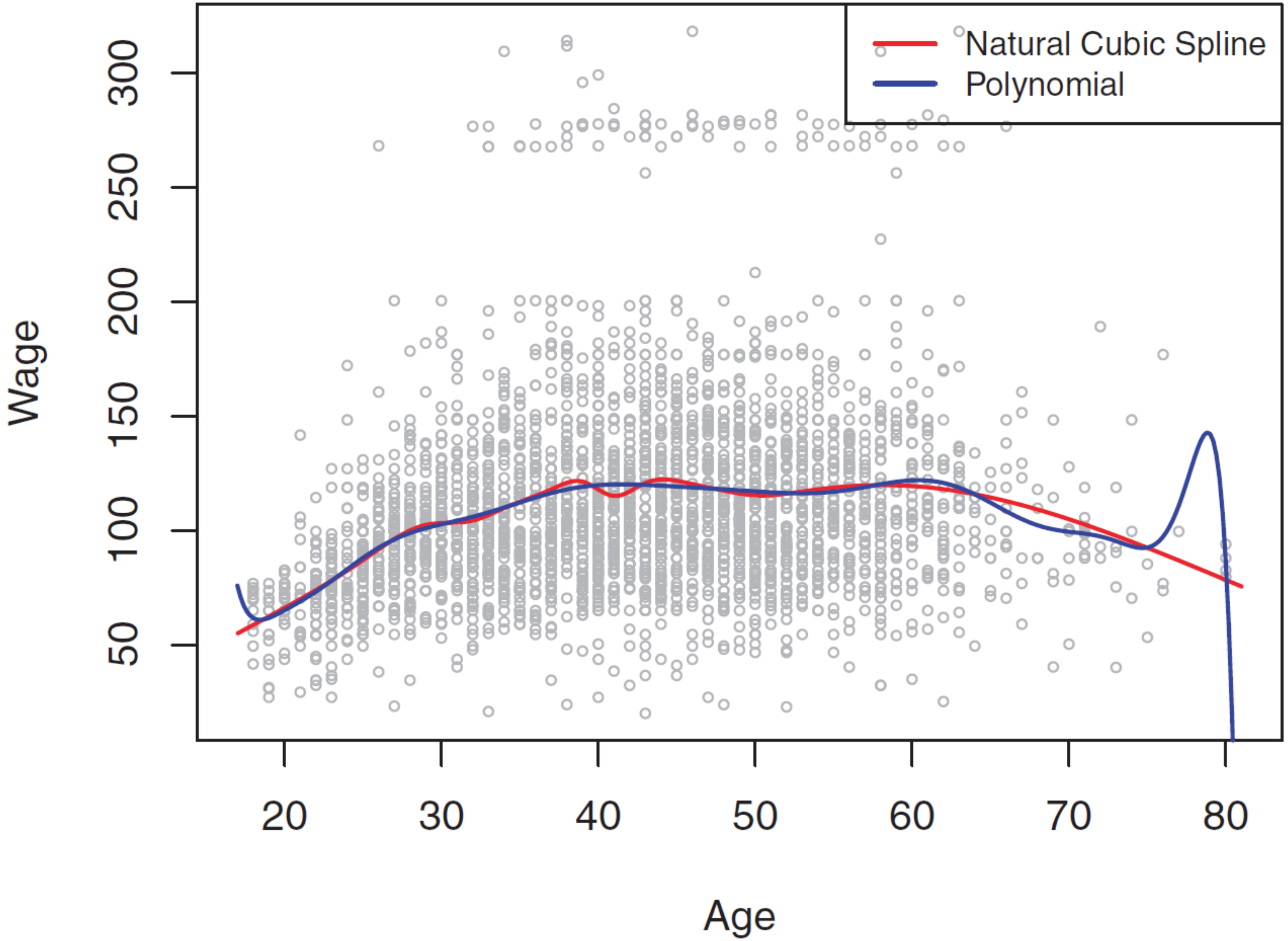
- Natural: the function is required to be linear at the boundary (when smaller than the smallest knot or larger than the largest knot)
- Note the width of the confidence intervals

Choosing the Number and Location of the Knots

- This is model selection, and cross validation is our friend …



Comparison to Polynomial Regression



Smoothing Splines

- Penalizing the squared second derivative of the prediction function

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

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

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

Cross validation used to select effective degrees of freedom

Smoother Matrix

From Chapter 5 of The Elements of Statistical Learning …

$$\mathrm{RSS}(\theta,\lambda) = (\mathbf{y}-\mathbf{N}\theta)^T(\mathbf{y}-\mathbf{N}\theta) + \lambda\theta^T\boldsymbol{\Omega}_N\theta, \qquad (5.11)$$

where $\{\mathbf{N}\}_{ij} = N_j(x_i)$ and $\{\boldsymbol{\Omega}_N\}_{jk} = \int N_j''(t)N_k''(t)dt$. The solution is easily seen to be

$$\hat{\theta} = (\mathbf{N}^T\mathbf{N} + \lambda\boldsymbol{\Omega}_N)^{-1}\mathbf{N}^T\mathbf{y}, \qquad (5.12)$$

a generalized ridge regression. The fitted smoothing spline is given by

$$\hat{f}(x) = \sum_{j=1}^{N} N_j(x)\hat{\theta}_j. \qquad (5.13)$$

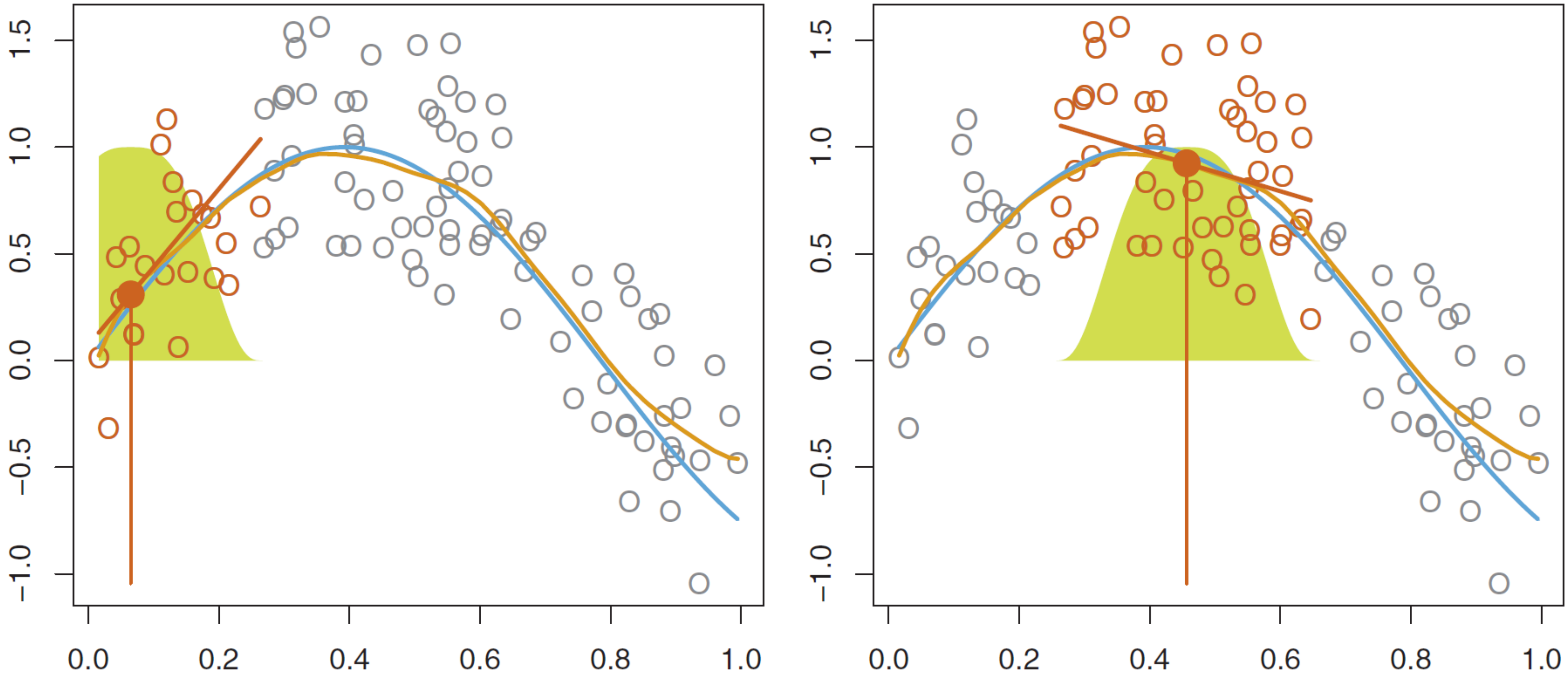
$$\begin{aligned}\hat{\mathbf{f}} &= \mathbf{N}(\mathbf{N}^T\mathbf{N} + \lambda\boldsymbol{\Omega}_N)^{-1}\mathbf{N}^T\mathbf{y} \\ &= \mathbf{S}_\lambda\mathbf{y}. \qquad (5.14)\end{aligned}$$

Simple Smoothing Spline Examples

Simple Smoothing Spline Example

```
> library(ISLR)
> library(splines)
> Smoother.Matrix = function(x, df) {
+     n = length(x)
+     S = matrix(0, n, n)
+     for(i in 1:n) {
+         y = rep(0, n)
+         y[i] = 1
+         S[,i] = predict(smooth.spline(x, y, df = df), x)$y
+     }
+     return((S + t(S)) / 2)
+ }
> S = Smoother.Matrix(Wage$age, df = 6.8)
> model = smooth.spline(Wage$age, Wage$wage, df = 6.8)
> model$df
[1] 6.801142
> sum(diag(S))
[1] 6.801142
> estimates = S %*% Wage$wage
> predictions = predict(model, Wage$age)$y
> estimates[1:5]
[1]  60.46695  83.49226 119.53504 119.70016 117.82229
> predictions[1:5]
[1]  60.46695  83.49226 119.53504 119.70016 117.82229
```

Local Regression (Loess)



Local Regression Algorithm

1. Gather the fraction $s = k/n$ of training points whose x_i are closest to x_0. ['s' is the span parameter of the loess() function]

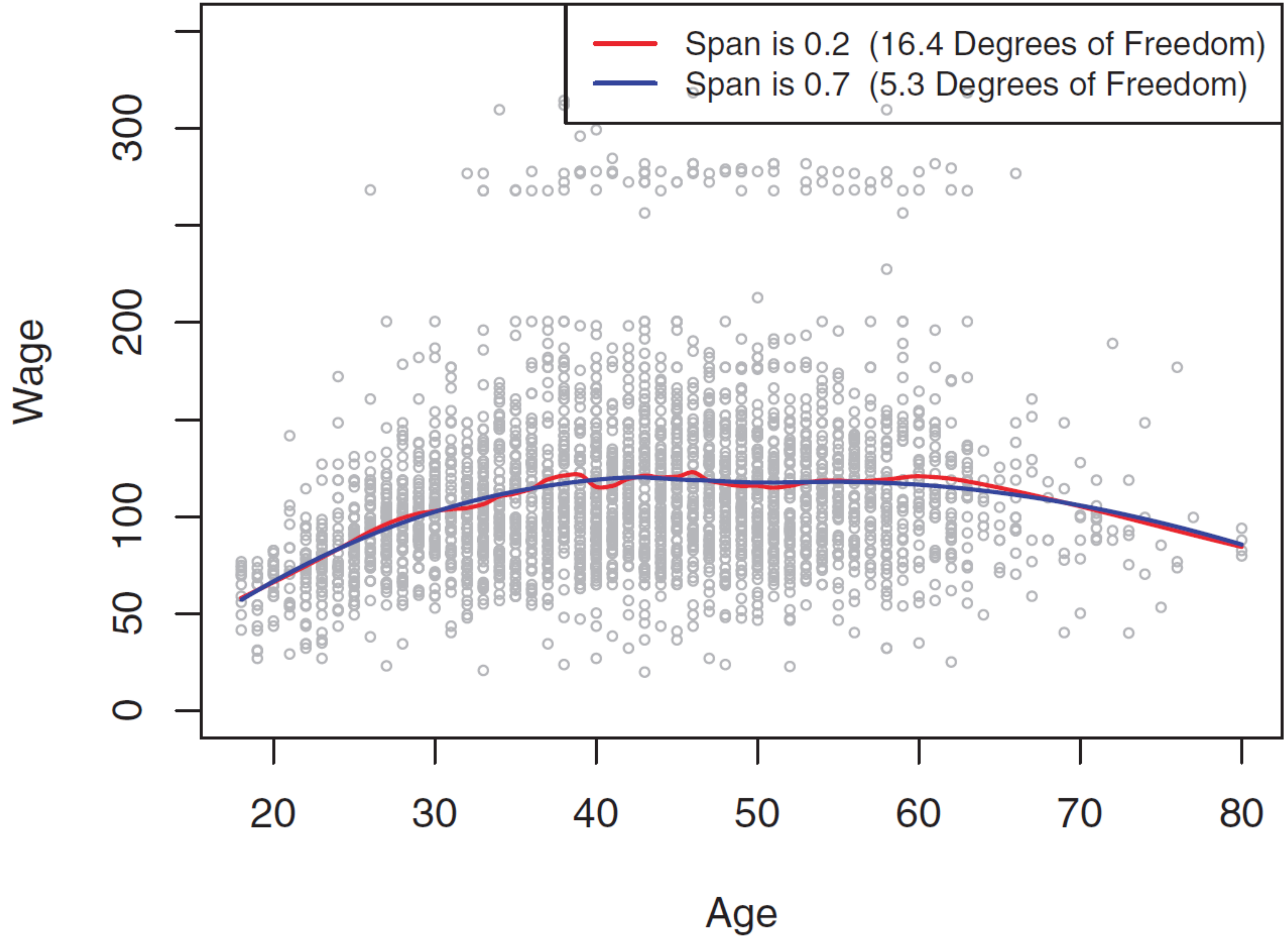
2. Assign a weight $K_{i0} = K(x_i, x_0)$ to each point in this neighborhood, so that the point furthest from x_0 has weight zero, and the closest has the highest weight. All but these k nearest neighbors get weight zero.

3. Fit a *weighted least squares regression* of the y_i on the x_i using the aforementioned weights, by finding $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize

$$\sum_{i=1}^{n} K_{i0}(y_i - \beta_0 - \beta_1 x_i)^2. \tag{7.14}$$

4. The fitted value at x_0 is given by $\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$.

Local Regression Example

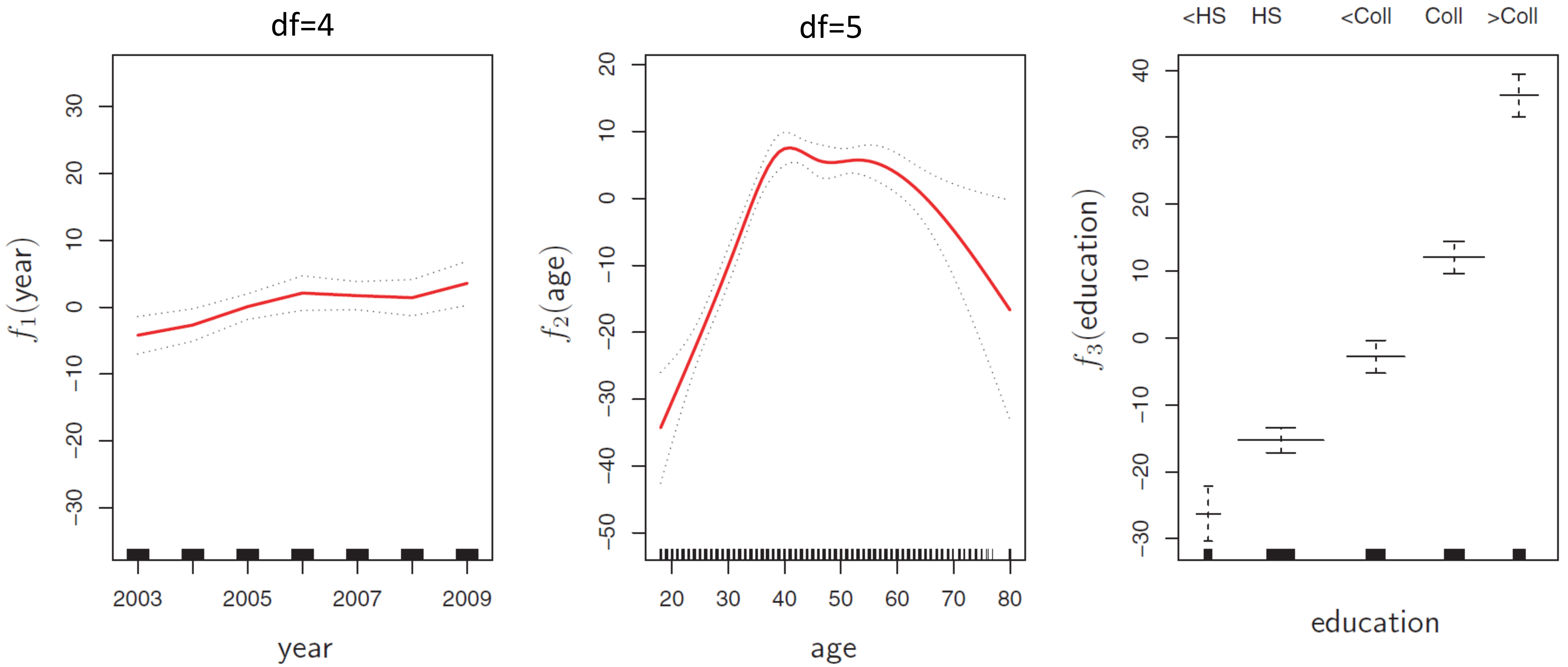


Generalized Additive Model (GAM)

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

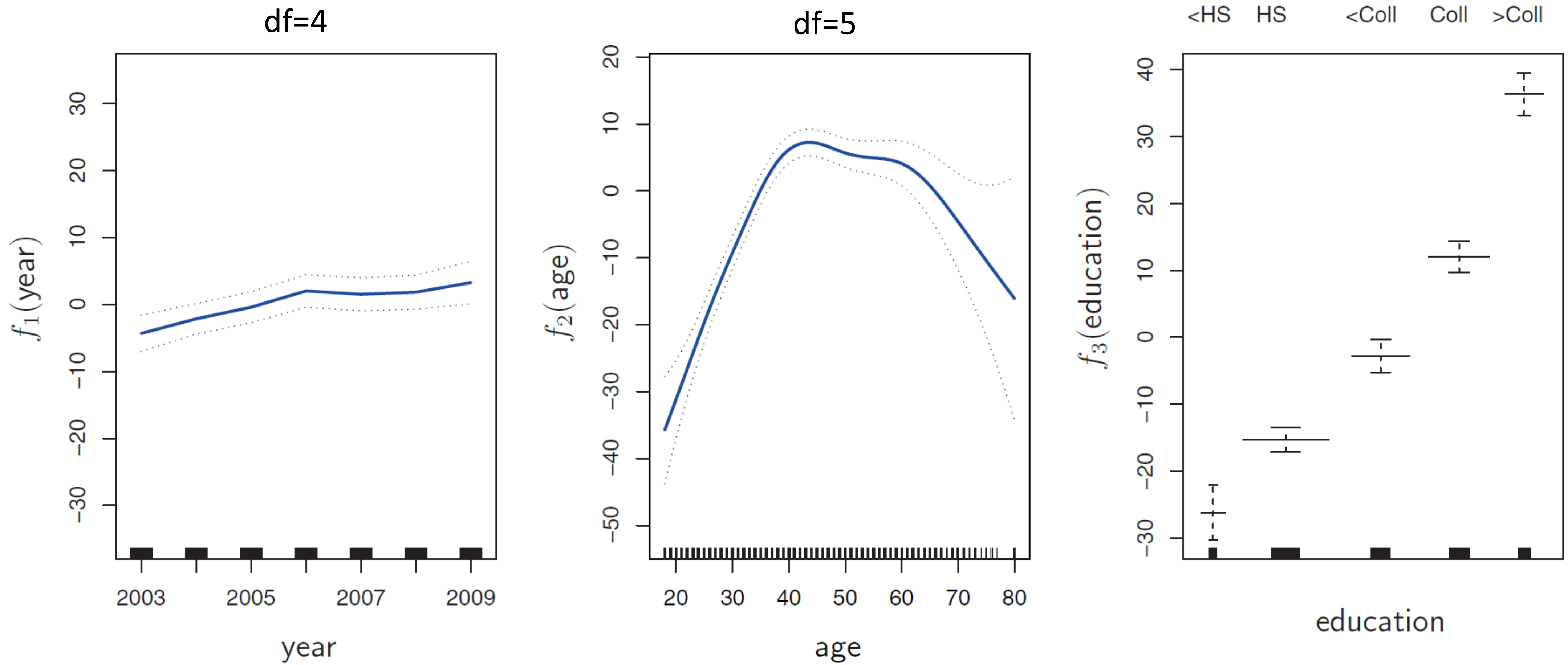
GAM Example with Natural Splines

$$\texttt{wage} = \beta_0 + f_1(\texttt{year}) + f_2(\texttt{age}) + f_3(\texttt{education}) + \epsilon$$



GAM Example with Smoothing Splines

$$\texttt{wage} = \beta_0 + f_1(\texttt{year}) + f_2(\texttt{age}) + f_3(\texttt{education}) + \epsilon$$



The Pros and Cons of GAMs

- Pro: allows us to fit a non-linear f() to each predictor, so we can automatically model non-linear relationships that standard linear regression will miss
- Pro: potentially more accurate predictions
- Pro: the model is additive so we can examine the effect of each predictor [while holding the other predictors fixed]
- Pro: the smoothness of the f() for each predictor, can be summarized by the degrees of freedom
- Con: the model is restricted to be additive

GAMs Can Also Be Used for Classification

- Model

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

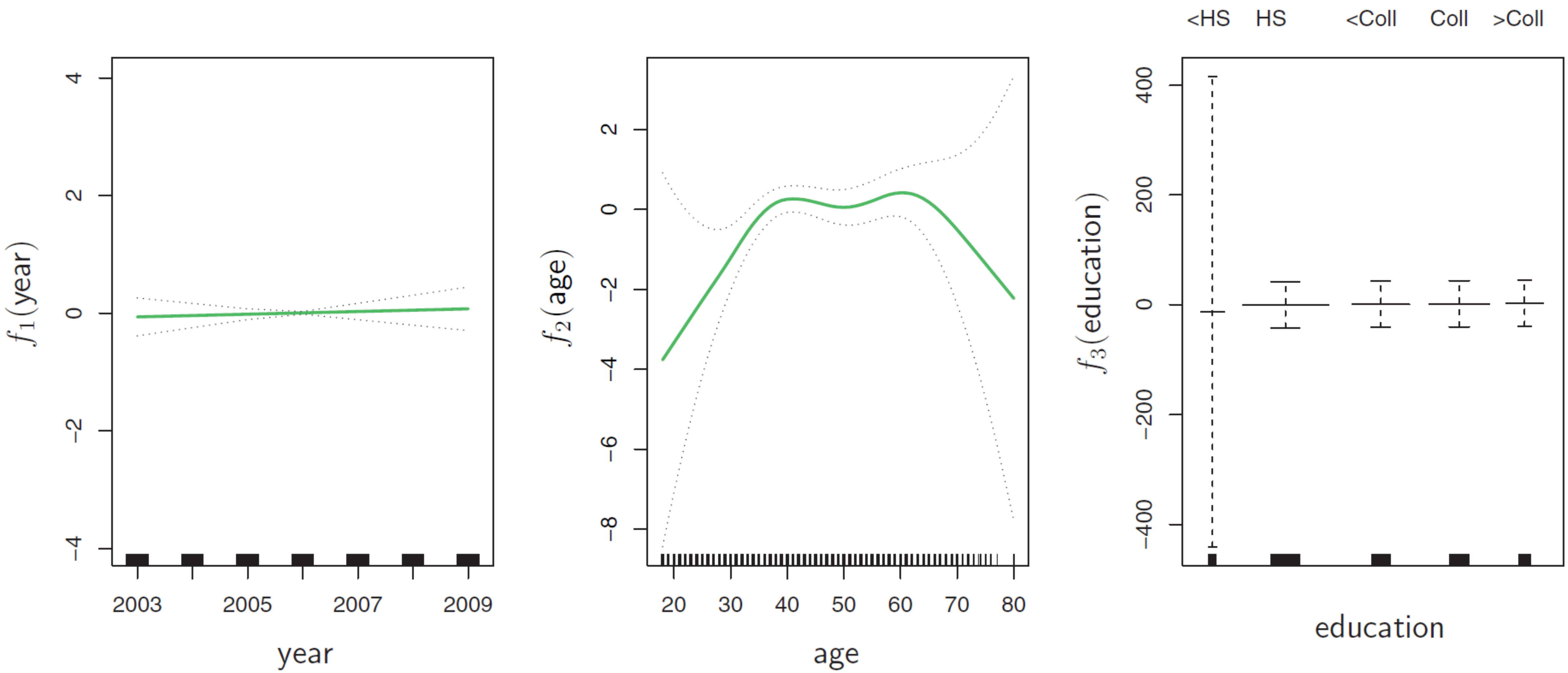
- Example

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

$$p(X) = \Pr(\texttt{wage} > 250 | \texttt{year}, \texttt{age}, \texttt{education})$$

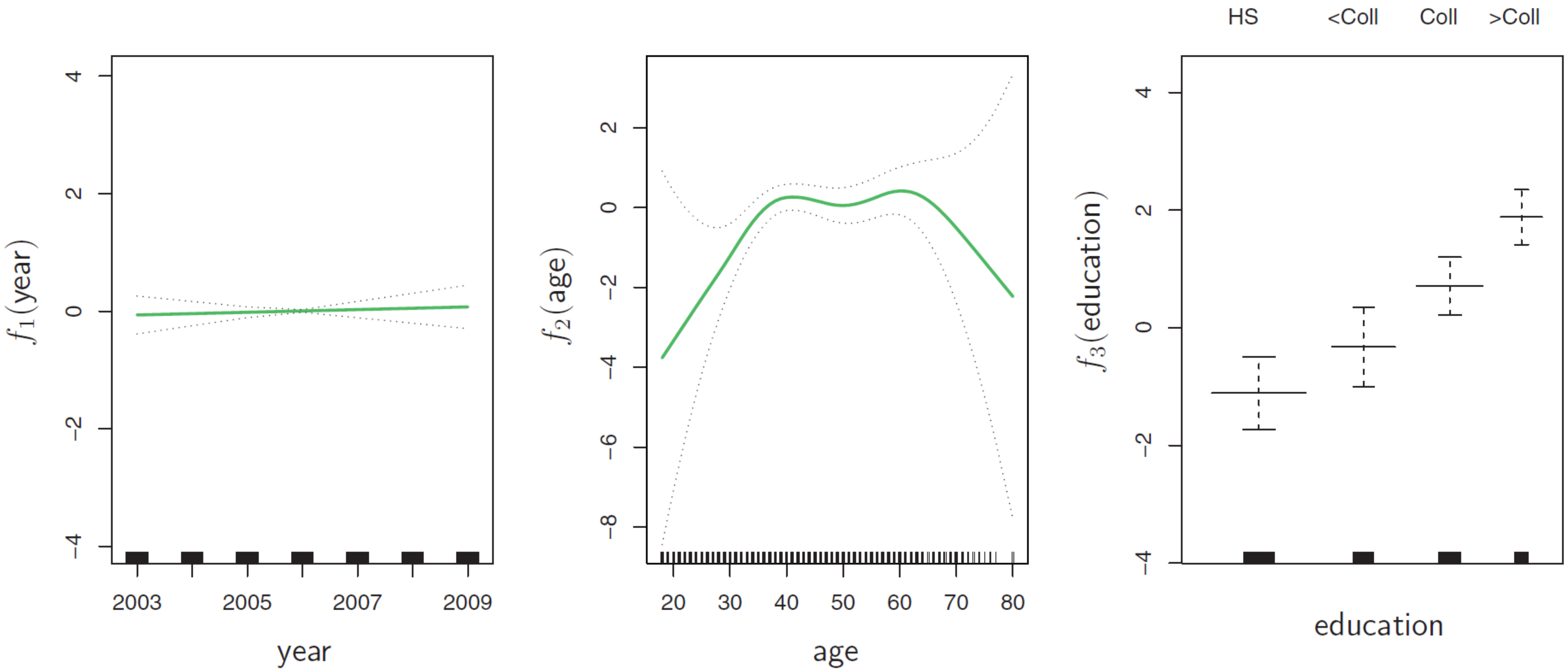
GAM Example for Classification

Check out the confidence interval for Education < HS …



GAM Example for Classification

After removing the observations for which Education < HS …



Agenda