Lecture 8

Beyond Linearity

ISLR 7, ESL 5,6,9

Jilles Vreeken
Krikamol Muandet

UNIVERSITÄT DES SAARLANDES

CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

Moving Beyond Linearity

There are several ways of extending linear models

1. **polynomial regression**, with polynomial basis functions
   - e.g., (simple) cubic regression uses basis functions X, X^2, X^3
2. **step functions** decompose the value range into K distinct regions
   - the effect is to fit a piecewise constant function (k-nearest neighbor models)
3. **regression splines** combine the two approaches
   - they divide the variable range into K regions,
   - they fit polynomials in each region, and
   - they force smoothness at region boundaries (knots)
4. **smoothing splines** are splines with many knots
   - they fit the RSS subject to a smoothness penalty
5. **local regression** is similar to splines
   - but allows the regions overlap in a smooth fashion

Generalized additive models allow for dealing with multiple predictors

Polynomial Regression

Standard linear model

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

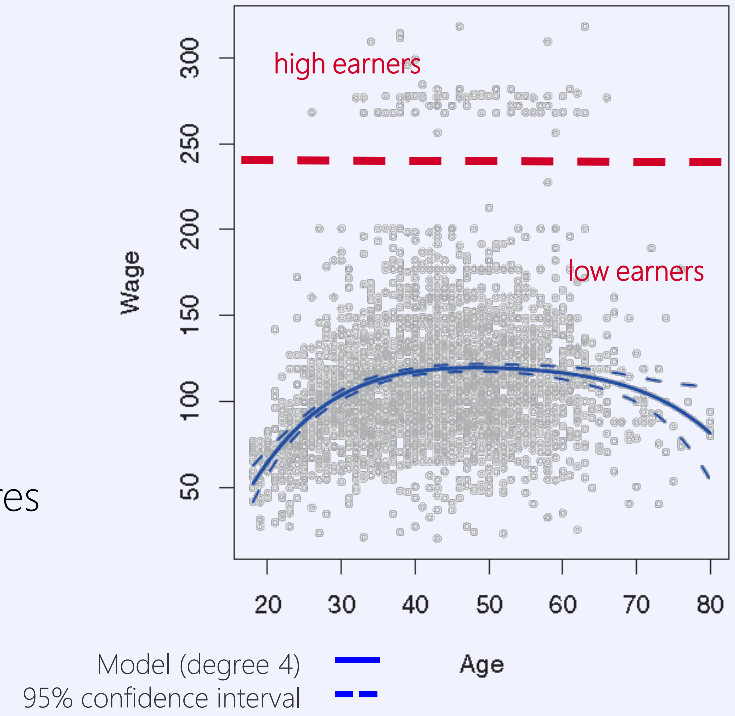
Polynomial regression

$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d + \epsilon$

Model is still linear in the coefficients β_i!

- compute confidence bounds as before using pointwise variance from least squares

example regression on wage data



Polynomial Regression

Standard linear model

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

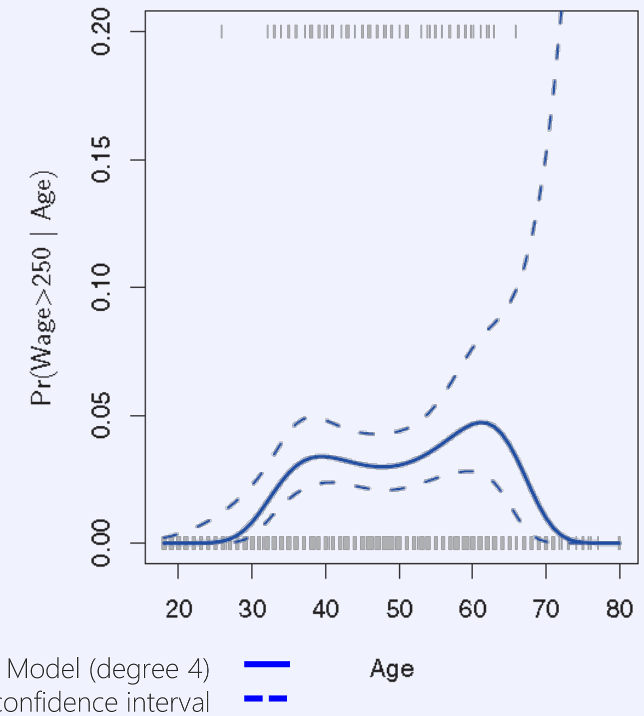
Polynomial **logistic** regression

$$\Pr(y_i > 250 \mid x_i) = \frac{\exp\left(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d\right)}{1 + \exp\left(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d\right)}$$

Model is still linear in the coefficients β_i!

- compute confidence bounds as before using pointwise variance from least squares
- bands are wide because there are only few (79 out of 3000) high earners

logistic regression wage>250



(ISLR 7.1)

Choosing the Degree

Unusual to use d greater than $3 \text{ or } 4$

Polynomial of degree n can perfectly fit n observations with different inputs

- $n + 1$ if we also include the bias/intercept
- risk of overfitting

In practice you can just use cross validation

Issue: Notorious tail behavior – bad for extrapolation

Step Functions

We convert a **continuous** to an **ordered categorical** variable (ordinal)

- create cutpoints $c_1, c_2, \dots, c_K$ in the range of X
- construct $K+1$ new variables

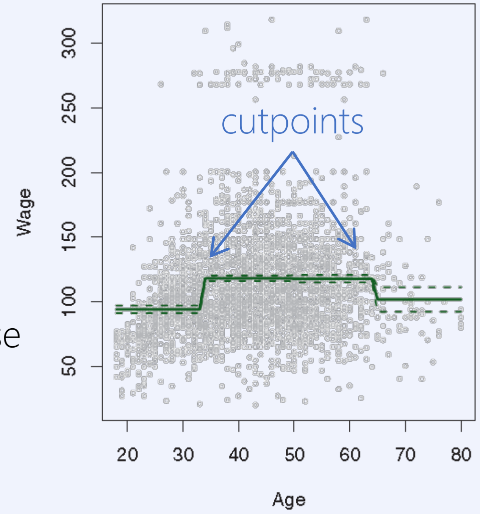
dummy variables

$$C_0(X) = I(X < c_1),$$
$$C_1(X) = I(c_1 \le X < c_2),$$
$$\dots$$
$$C_{K-1}(X) = I(c_{K-1} \le X < c_K),$$
$$C_K(X) = I(c_K \le X)$$

$$\sum_{i=0}^{K} C_i = 1$$

- $I(\cdot)$ is the indicator function: 1 if its argument is true and zero otherwise

wage regression



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

- β_0 is the average of Y for all $X < c_1$
- β_j is the average increase in Y over β_0 for $c_j < X < c_{j+1}$

Step Functions

We convert a **continuous** to an **ordered categorical** variable (ordinal)

- create cutpoints $c_1, c_2, \ldots, c_K$ in the range of X
- construct $K+1$ new variables

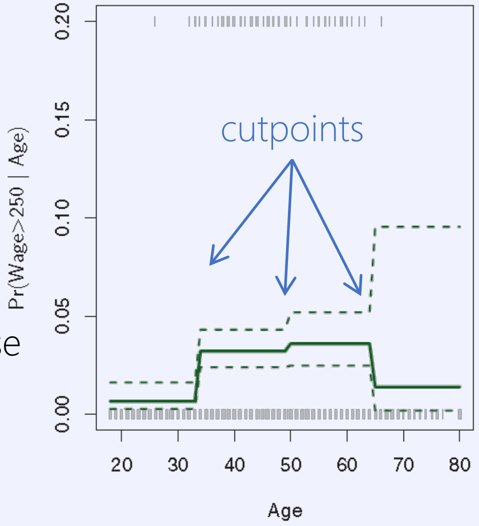
dummy variables

$$C_0(X) = I(X < c_1),$$
$$C_1(X) = I(c_1 \leq X < c_2),$$
$$\ldots$$
$$C_{K-1}(X) = I(c_{K-1} \leq X < c_K),$$
$$C_K(X) = I(c_K \leq X)$$

$$\sum_{i=0}^{K} C_i = 1$$

- $I(\cdot)$ is the indicator function: 1 if its argument is true and zero otherwise

wage classification

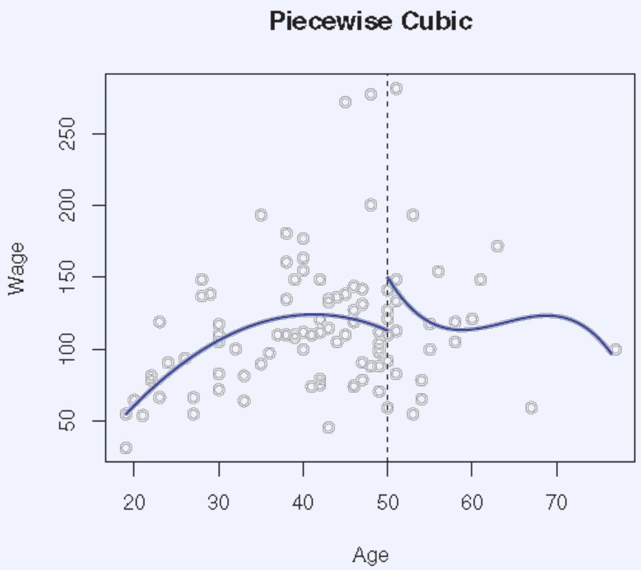


Classification $\Pr(y_i > 250 \mid x_i) = \frac{\exp(\beta_0 + \beta_1 C_1(x_i) + \cdots + \beta_K C_K(x_i))}{1 + \exp(\beta_0 + \beta_1 C_1(x_i) + \cdots + \beta_K C_K(x_i))}$

Cutpoints need to be placed wisely, lest the model misses the action!

Regression Splines

Instead of fitting one high-degree polynomial, we fit a low-degree polynomial *per region* of X

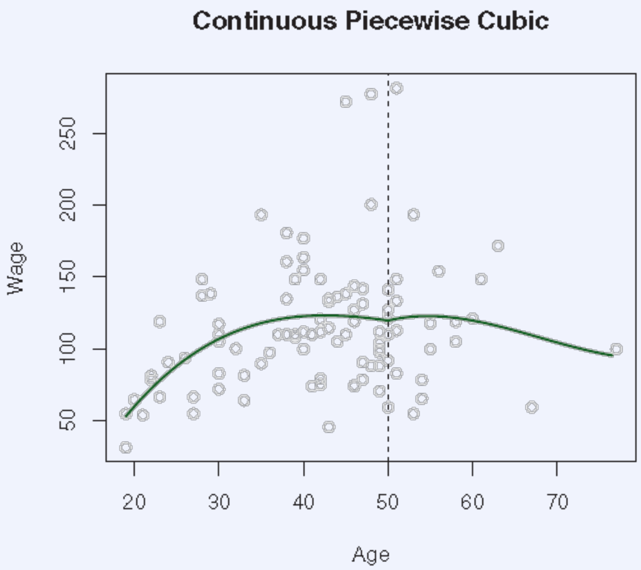


single cutpoint at ***age=50***

Regression Splines

Instead of fitting one high-degree polynomial,
we fit a low-degree polynomial *per region* of X

- make sure that the model is smooth at region boundaries
- that is, continuous and $d-1$ times continuously differentiable, where d is the degree of the polynomial

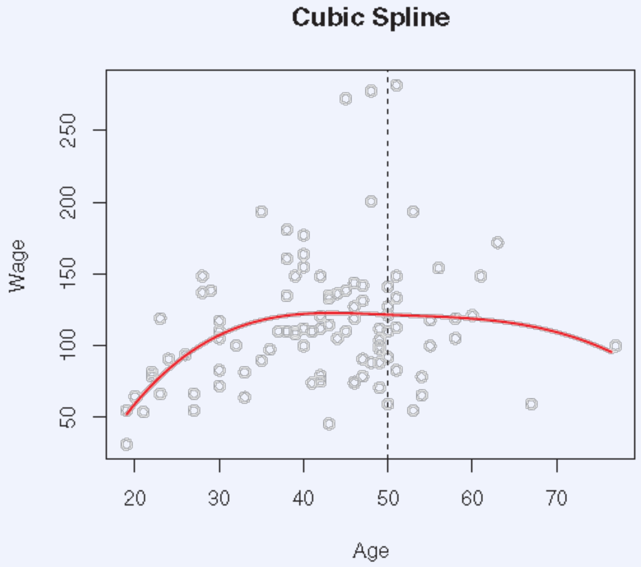


single cutpoint at ***age=50***

Regression Splines

Instead of fitting one high-degree polynomial,
we fit a low-degree polynomial *per region* of X

- make sure that the model is smooth at region boundaries
- that is, continuous and $d-1$ times continuously differentiable, where d is the degree of the polynomial
- $d = 3$ is a popular choice, it appears to be the right compromise between nonlinearity and smoothness
- the more regions, the more flexibility in the model
  - with K cutpoints (knots) fit $K+1$ polynomials

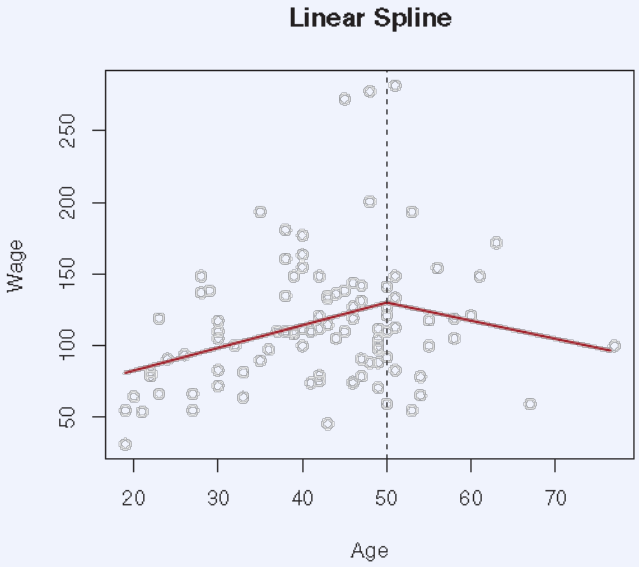


single cutpoint at ***age=50***

Regression Splines

Instead of fitting one high-degree polynomial,
we fit a low-degree polynomial *per region* of X

- make sure that the model is smooth at region boundaries
- that is, continuous and $d-1$ times continuously differentiable, where d is the degree of the polynomial
- $d = 3$ is a popular choice, it appears to be the right compromise between nonlinearity and smoothness
- the more regions, the more flexibility in the model
  - with K cutpoints (knots) fit $K+1$ polynomials
- regression splines of degree d with K knots form a vector space with dimension $d + K + 1$



single cutpoint at ***age=50***

Spline Bases of the Cubic Splines

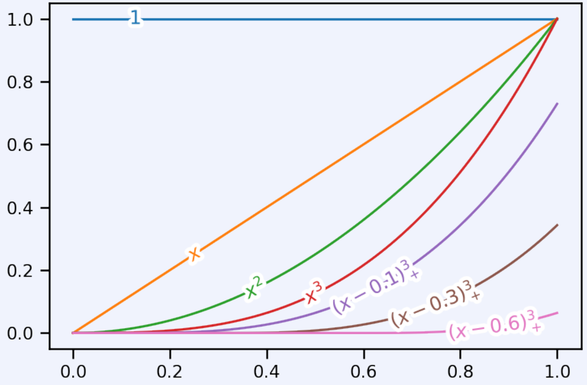
The vector space of **cubic splines** ($d = 3$) with K knots has dimension $K + d + 1 = K + 4$

The truncated cubic function is defined as

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

The functions $1, X, X^2, X^3, h(X,\zeta_1), h(X,\zeta_2), h(X,\zeta_K)$ form the canonical basis of the vectors space of cubic splines with K knots

- $\zeta_1, \zeta_2, \dots, \zeta_K$ are the positions of the knots
- every cubic spline with K knots is a unique linear combination of the basis functions



basis functions of cubic splines with 3 knots at 0.1, 0.3, and 0.6

Natural Cubic Splines

Cubic splines have high variance at the boundaries

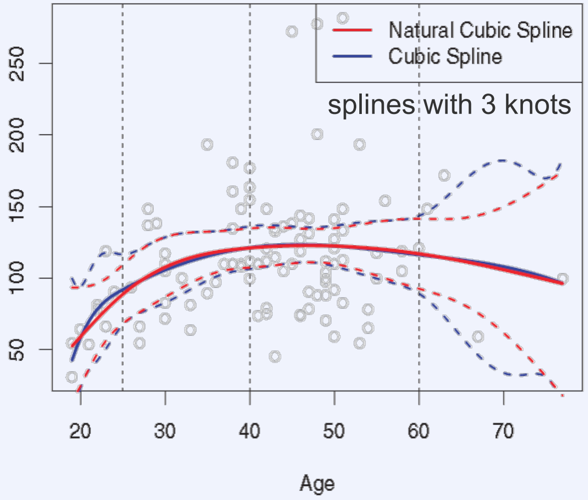
- because information comes from only one side

Idea: make spline simpler at the boundaries (linear splines)

The resulting **natural cubic splines** have a different basis

$N_1(X) = 1,\ \ N_2(X) = X,\ \ N_{k+2}(X) = d_k(X) - d_{(K-1)}(X),$

where $d_k(X) = \frac{(X-\zeta_k)_+^3 - (X-\zeta_K)_+^3}{\zeta_K - \zeta_k}$ for $k = 1, \dots, K-2$



The vector space of natural cubic splines with K knots has dimension K

- lost two degrees of freedom at each boundary region – square and cubic coefficients are zero
- natural splines have less variance at the boundaries

On the Number and Location of Knots

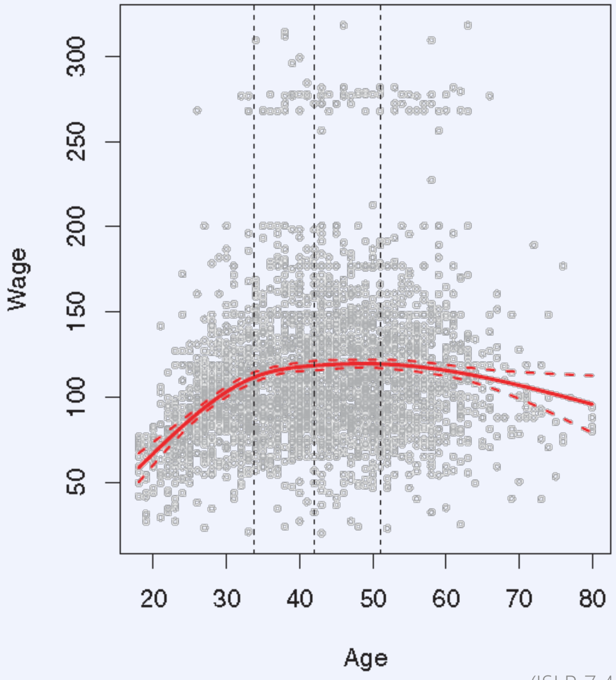
Location of the knots

- equidistant in values range of input
  - common approach
- according to quantiles in the data set
  - more information on response in input regions of high data density
  - thus the knots can be placed more densely,
    affording higher model flexibility in these regions

Number of knots

- directly related to degrees of freedom (dof) of the model
- software often lets you choose the dof

wage regression with
natural cubic splines with 3 dof
(3 knots at the three quartiles)



On the Number and Location of Knots

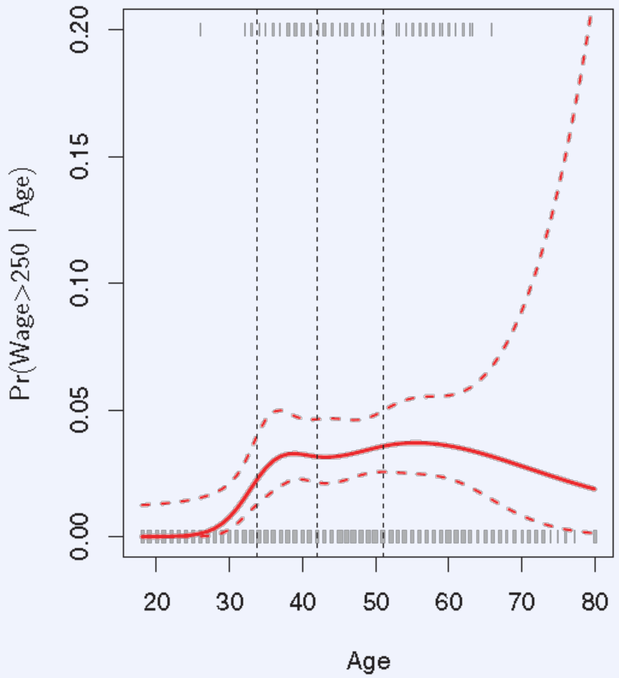
Location of the knots

- equidistant in values range of input
  - common approach
- according to quantiles in the data set
  - more information on response in input regions of high data density
  - thus the knots can be placed more densely,
    affording higher model flexibility in these regions

Number of knots

- directly related to degrees of freedom (dof) of the model
- software often lets you choose the dof

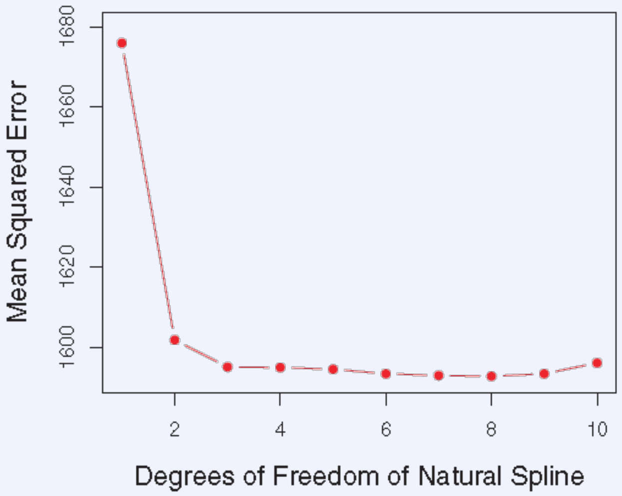
wage classification with natural cubic splines with 3 dof (3 knots at the three quartiles)

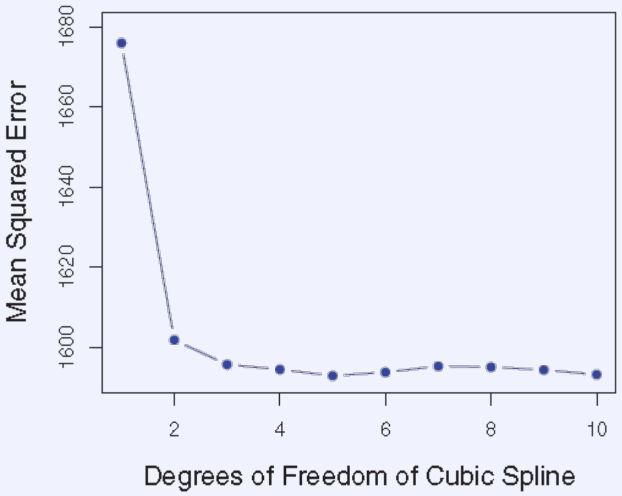


On the Number and Location of Knots

Model selection for splines

- the degree and kind of splines, and the number of knots
- number of knots can be chosen by cross validation





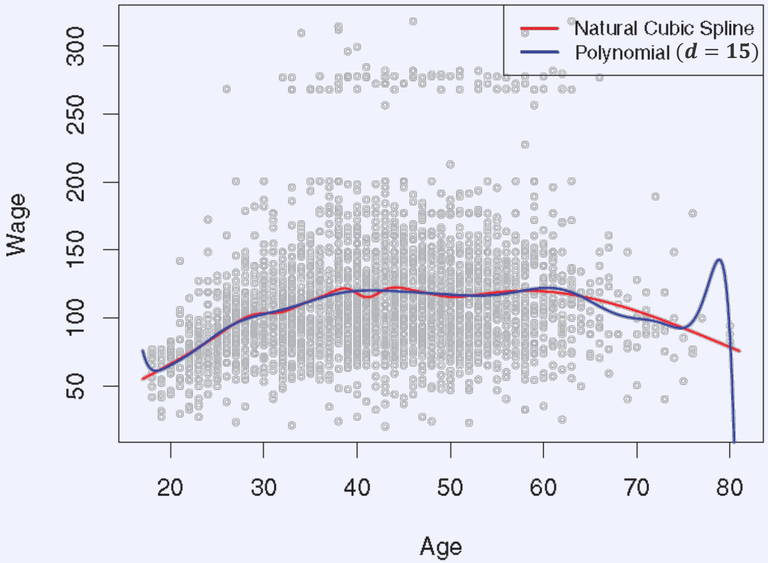
regression to the wage data

models with small degrees of freedom are low-degree polynomials without knots

On the Number and Location of Knots

Splines tend to be superior to polynomials

- splines are smoother than polynomials because of their low degree
- polynomials can be very wiggly, especially at the value space boundaries
- knots can be placed flexibly to account for non-uniform data density



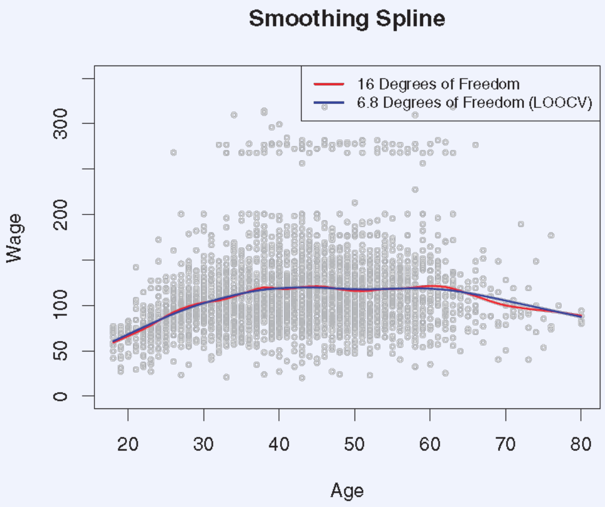
polynomial and spline both with 15 degrees of freedom

Smoothing Splines

Reduces RSS while keeping the curve smooth, i.e. non-wiggly

- nonparametric approach
- wigglyness is quantified in terms of second derivative $g''(x)$
- introduce a penalty on the size of the second derivative
- we minimize the following (analogous to ridge regression):

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



Important theorem: it can be shown that the function that minimizes the above equation **(*)** is a natural cubic spline with knots at the inputs of all data points

- this is not the same spline we get when we use the natural cubic spline basis
- rather, it is a shrunken version of that spline with restrictions on its second derivative

How to choose λ?

Increasing $\boldsymbol{\lambda}$ shrinks the spline, reducing its effective degrees of freedom

- same notion of effective degrees of freedom as in ridge regression
- let $\hat{\boldsymbol{g}}_\lambda = \boldsymbol{S}_\lambda \boldsymbol{y}$
- here $\hat{\boldsymbol{g}}_\lambda$ is the vector containing the fitted outputs at the training inputs $x_1, \ldots, x_n$ for a particular choice of λ
- this vector is a linear function of $\boldsymbol{y}$ denoted $\boldsymbol{S}_\lambda \boldsymbol{y}$
- in Chapter 6 we defined the effective degrees of freedom as $\mathbf{tr}(\boldsymbol{S}_\lambda)$

λ can be chosen by cross validation

- we can use the formula for generalized LOOCV from Chapter 5

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

Local Regression

Extension of k-nearest neighbors

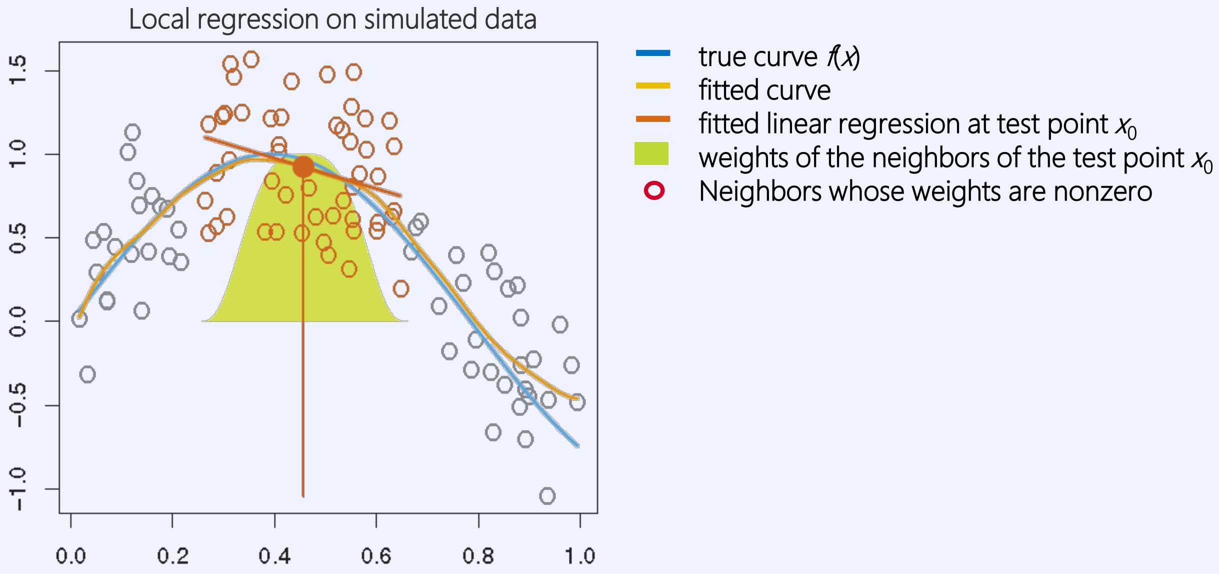
- fits not constant, but polynomial models based on the nearest neighbors of a test point
- weighs the contribution of neighbors by their distance to test point

(ISLR 7.6)

Local Regression

Extension of k-nearest neighbors

- fits not constant, but polynomial models based on the nearest neighbors of a test point
- weighs the contribution of neighbors by their distance to test point



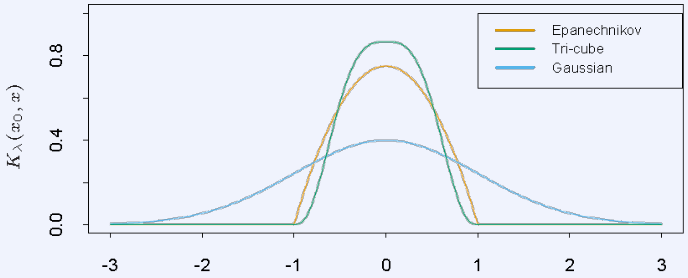
Local Regression

Extension of k-nearest neighbors

- fits not constant, but polynomial models based on the nearest neighbors of a test point
- weighs the contribution of neighbors by their distance to test point

Weights of neighbors are calculated with a **kernel function**

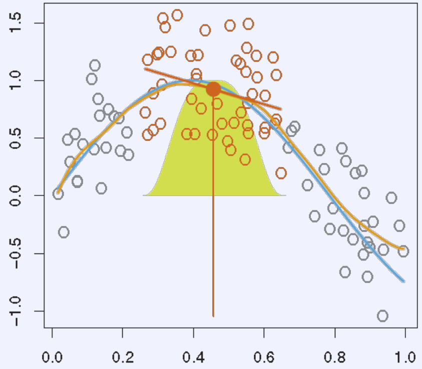
- in the simulated example we used the tri-cube kernel $D(t) = \begin{cases} (1-|t|^3)^3 & \text{if } |t| \le 1 \\ 0 & \text{otherwise} \end{cases}$



Local Regression

Extension of k-nearest neighbors

- fits not constant, but polynomial models based on the nearest neighbors of a test point
- weighs the contribution of neighbors by their distance to test point



Weights of neighbors are calculated with a kernel function

- in the simulated example we used the tri-cube kernel

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

- the width of the kernel is the span, important model parameter, to be chosen by CV
- if the kernel has no compact support all training data is needed for each prediction (high memory)
- local fit is with a linear function

Local Regression – Kernel

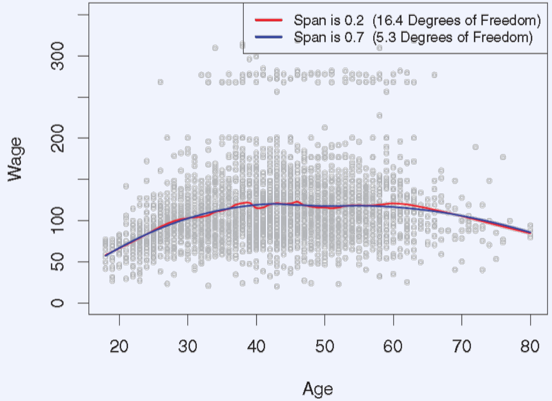
Define the kernel as $K_\lambda(x_0, x) = D\left(\frac{|x-x_0|}{s_\lambda(x_0)}\right)$

- the span $s_\lambda(x_0)$ depends on smoothing parameter λ and on the test point x_0
- large λ implies high bias and low variance
- constant span $s_\lambda(x_0) = \lambda$ leads to metric kernels
  - bias is constant over data range
  - variance is inversely proportional to the local density
- nearest-neighbor window width $s_k(x_0) = |x_0 - x_{[k]}|$ displays the opposite behavior
  - $x_{[k]}$ is the k-th closest neighbor
  - variance is constant over data range
  - bias is inversely proportional to local density

Local (Linear) Regression

Local regression at $X = x_0$

1. assign weight $K_{i0} = K(x_i, x_0)$ to each training point via the kernel
2. fit a weighted least-squares regression model, i.e. find $\hat{\beta}_0, \hat{\beta}_1$ that minimize $\sum_{i=1}^{n} K_{i0}(y_i - \beta_0 - \beta_1 x_i)^2$
3. the fitted value at x_0 is given by $\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_i$



local linear regression on wage with nearest-neighbor kernel
span = fraction of the data used to fit each target

Easily generalizes to multivariate, for higher dimensions $(p > 3,4)$ data sparsity can be an issue

Generalized Additive Models (GAM)

General framework for including nonlinear basis functions into linear multivariate models

- generalize the linear model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon_i$$

to

$$y_i = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \epsilon_i$$

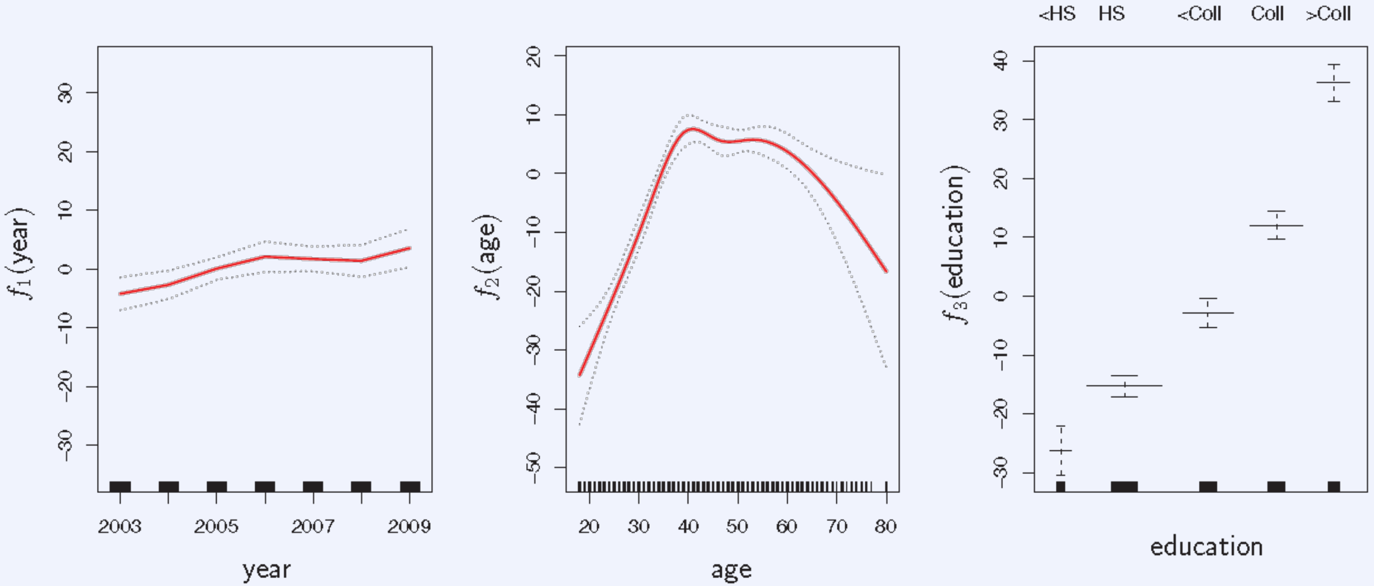
All methods we discussed so far can be plugged into this scheme (!)

For example:

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

- **`year`**, **`age`** continuous, fitted with natural splines (4 and 5 dof, respectively)
- **`education`** has categories **`<HS`**, **`HS`**, **`<Coll`**, **`Coll`**, **`>Coll`** fitted with constants per dummy variable

Generalized Additive Models (GAM)

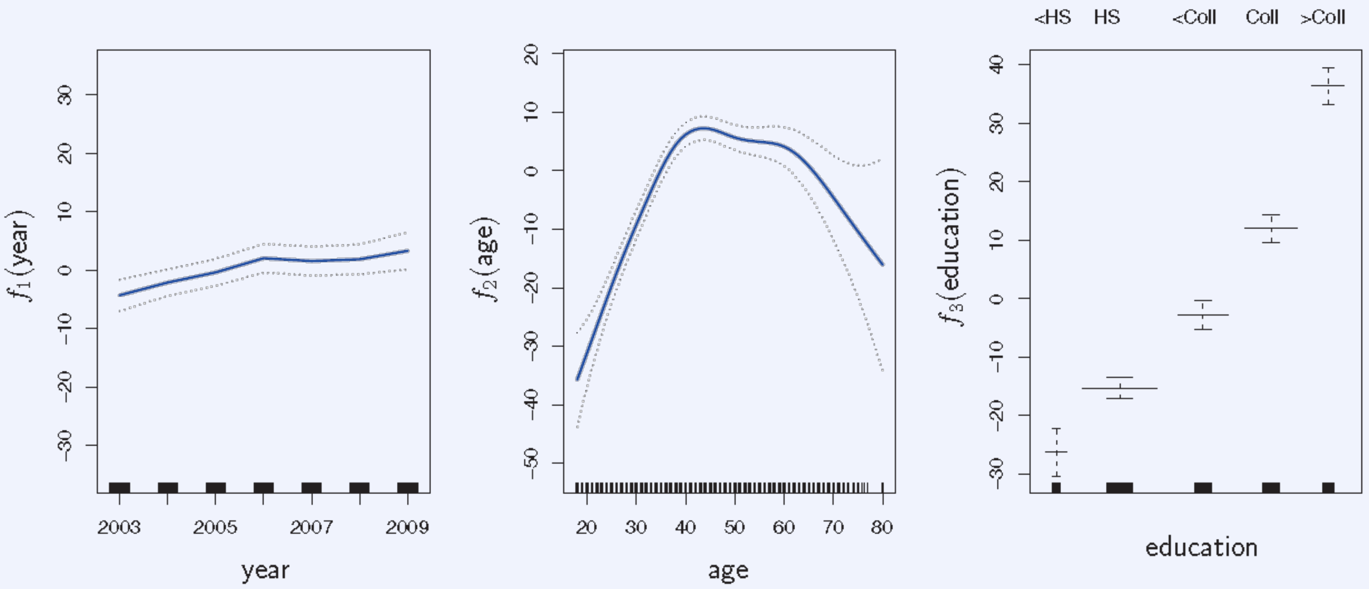


natural spline 4 dof

natural spline 5 dof

constants for dummies

Generalized Additive Models (GAM)



smoothing spline 4 dof | smoothing spline 5 dof | constants for dummies

Fitting Additive Models With Linear Smoothers

Simple iterative solution procedure (backfitting)

1. Initialize

$$\hat{\beta}_0 = \frac{1}{N}\sum_{i=1}^{N} y_i\,, \qquad \hat{f}_j \equiv 0\ \forall_{i,j}$$

The nonlinear terms average to zero over the data

2. Cycle $j = 1, 2, \ldots, p, 1, 2, \ldots, p, \ldots,$

Run until convergence

$$\hat{f}_j \leftarrow S_j\left[\left\{y_i - \hat{\beta}_0 - \sum_{k\neq j}^{p} \hat{f}_k(x_{ik})\right\}_{i=1}^{N}\right]$$

Cubic smoothing spline fit to the residual as a function of the j-th input only, Applicable also to other linear smoothing operators

$$\hat{f}_j \leftarrow \hat{f}_j - \frac{1}{N}\sum_{i=1}^{N} \hat{f}_j(x_{ij})$$

Secure mean zero,
Not necessary, in theory, since the smoothing spline for a mean zero response has mean zero, good, in practice, to counteract slippage caused by machine rounding

For many linear smoothers backfitting is the same as the Gauss-Seidel algorithm for solving linear systems of equations

Generalized Additive Models (GAM)

Pros and cons of GAMs

- ➕ nonparametric, no need of trying out different model assumptions
- ➕ nonparametric, can afford more accurate predictions
- ➕ since model is additive, we can assess the influence of a variable while holding the other variables fixed
- ➕ smoothness of function f_j for variable X_j can be summarized via degrees of freedom
- ➖ restriction of the model to be additive, this can miss important interactions
  - but, we can add predictors like $X_j \times X_k$ fitted with e.g. two-dimensional splines

Generalized Additive Models (GAM)

GAMs for classification

- use logistic regression
- linear model $\log\left(\frac{p(x)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$
- generalized additive 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 on the wage data: $p(X) = \Pr(\texttt{wage} > 250 \mid \texttt{year}, \texttt{age}, \texttt{education})$

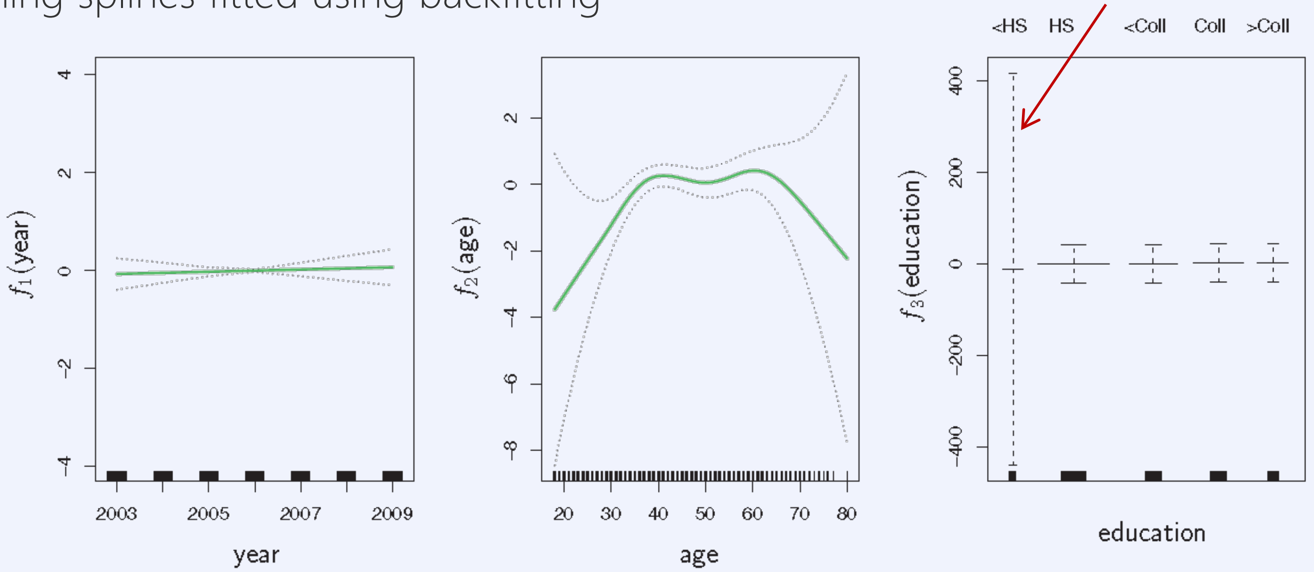
- the GAM takes the form

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + f_1(\texttt{year}) + f_2(\texttt{age}) + f_3(\texttt{education})$$

Generalized Additive Models (GAM)

Smoothing splines fitted using backfitting

Due to no individuals without high school education earning more than $250K per year



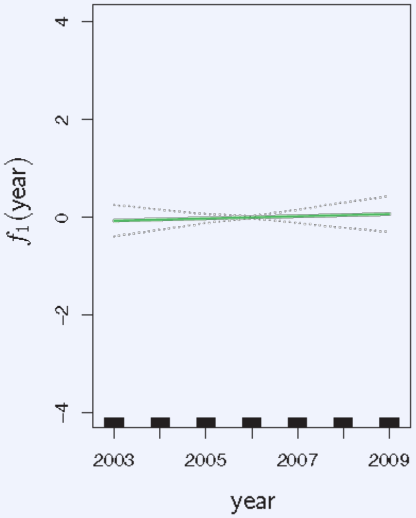
linear function in year

smoothing spline 5 dof

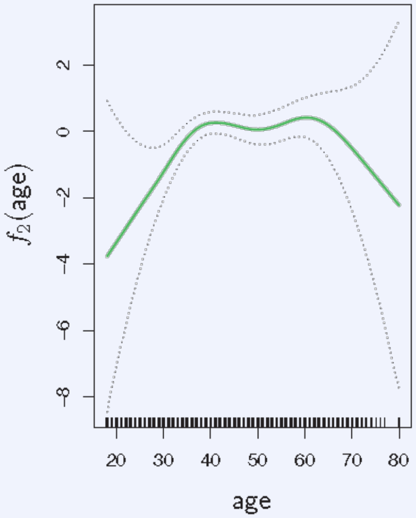
constants for dummies

Generalized Additive Models (GAM)

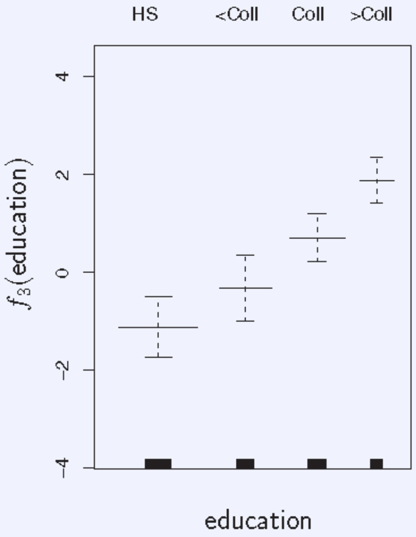
Refit excluding people without high school education



linear function in year



smoothing spline 5 dof



constants for dummies