Chapter7 Moving Beyond Linearity

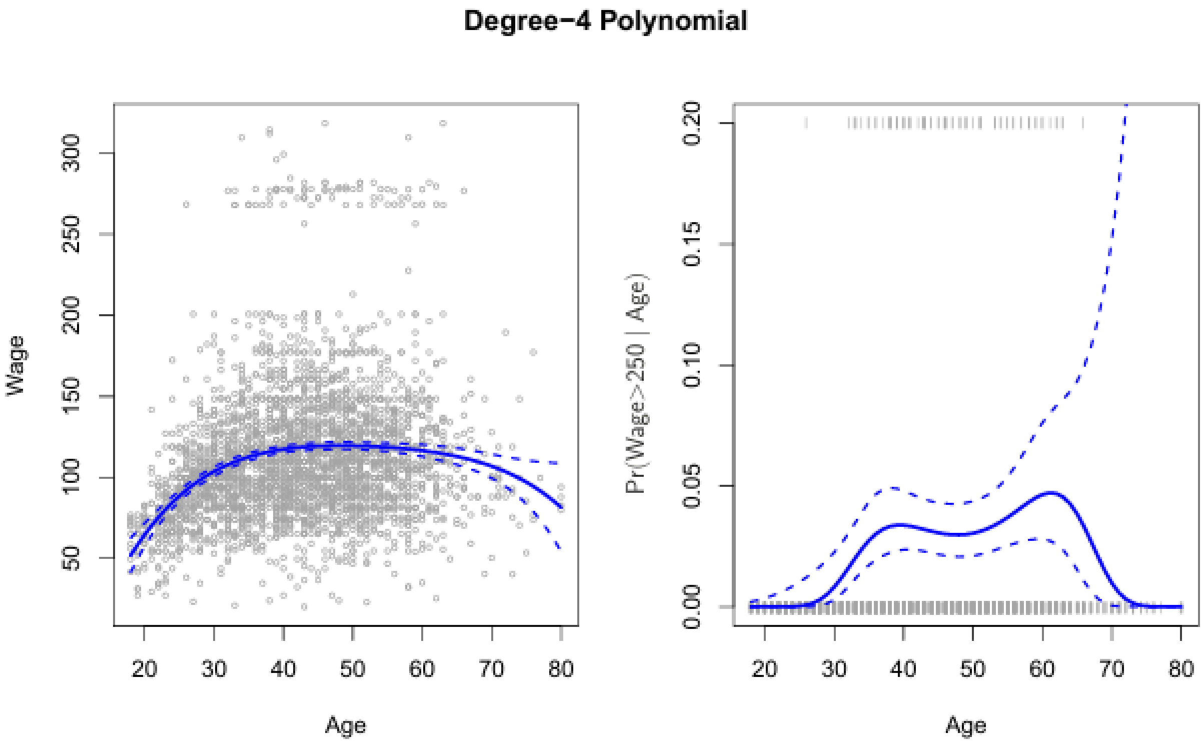
The truth is never linear or you can say almost never !
But often linearity assumption is good enough. When its not

- polynomials,
- step functions,
- splines,
- local regression and,
- generalized additive models
  offer a lot of flexibility , without losing the case and interpretability of linear models.

Polynomial Regression

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

- Shown below is the Wage data.
- Left: The solid blue curve is a degree-4 polynomial of wage (in thousands of dollars) as a function of age, fit by least squares. The dashed curves indicate an estimated 95% confidence interval.
- Right: We model the binary event $wage > 250$ using logistic regression, again with a degree-4 polynomial. The fitted posterior probability of wage exceeding $ $250,000$ is shown in blue, along with an estimated 95% confidence interval.



- Create new variables $X_1 = X, X_2 = X^2$, etc and then treat as multiple linear regression.
- Not really interested in the coefficients ; more interested in the fitted function values at any value x_0:

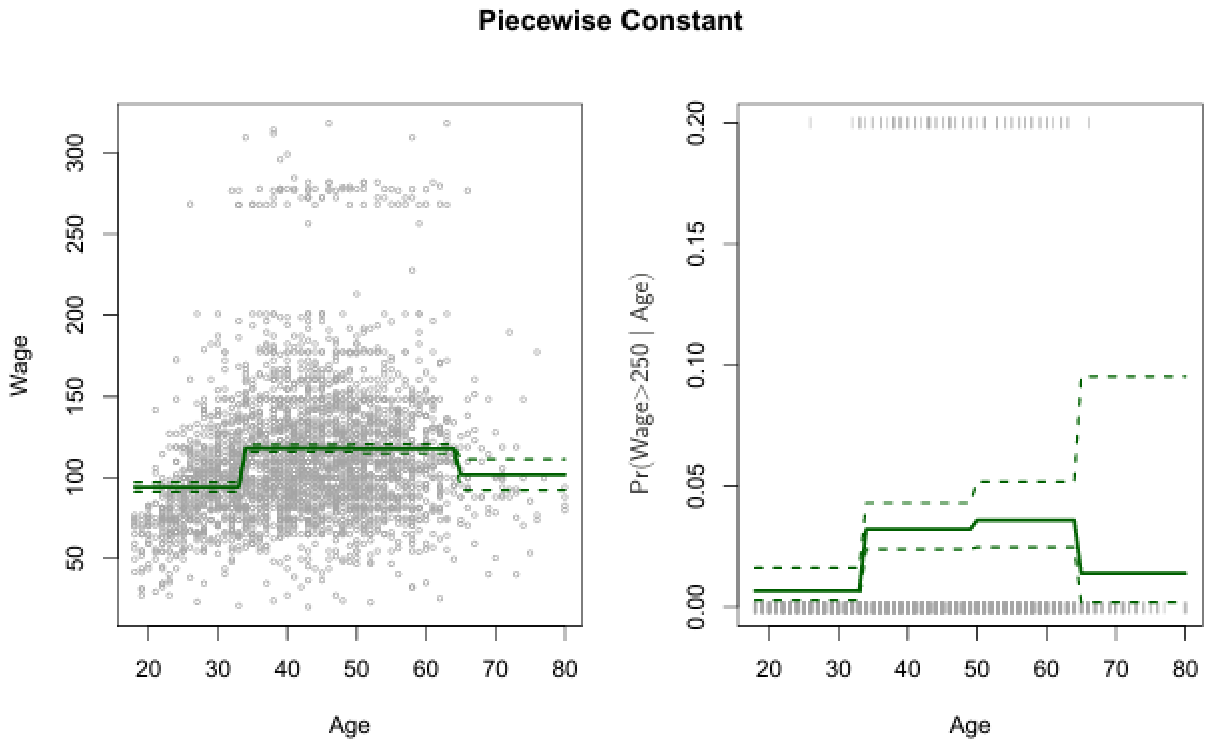
$$\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 x_0^2 + \hat{\beta}_3 x_0^3 + \hat{\beta}_4 x_0^4$$

- Since $\hat{f}(x_0)$ is a linear function of the $\hat{\beta}_l$, can get a simple expression for **pointwise-variances** . $Var[\hat{f}(x_0)]$ at any value x_0 . In the figure , we have computed the fit and pointwise standard errors on a grid of values for x_0 . We show $\hat{f}(x_0) \pm 2 \cdot se[\hat{f}(x_0)]$.
- We either fix the degree d at some reasonable low value, or use cross-validation to choose d.
- Logistic regression follows naturally. For example , in the figure we model

$$Pr(y_i > 250|x_i) = \frac{exp(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \cdots + \beta_d x_i^d)}{1 + exp(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \cdots + \beta_d x_i^d)}$$

- To get the confidence intervals , compute upper and lower bounds on the **logit scale** , and then invert to get on probability scale.
- Can do separately on several variables - just stack the variables into one matrix, and separate out the pieces afterwards.
- Caveat : polynomials have notorious tail behavior - very bad for extrapolation
- Can fit using $y \sim poly(x, degree = 3)$ in formula.

Step functions



- Again using the wage data.
- Another way of creating transformations of a variable - cut the variable into distinct regions .
  $C_1(X) = I(X < 35), C_2(X) = I(35 \leq X < 65), \ldots, C_3(X) = I(X \geq 65)$
- Easy to work with .Create a series of dummy variables representing each group.
- Useful way of creating interactions that are easy to interpret. For example, interaction effect of **Year** and **Age** :

  $$I(Year < 2005) \cdot Age, I(Year \geq 2005) \cdot Age$$

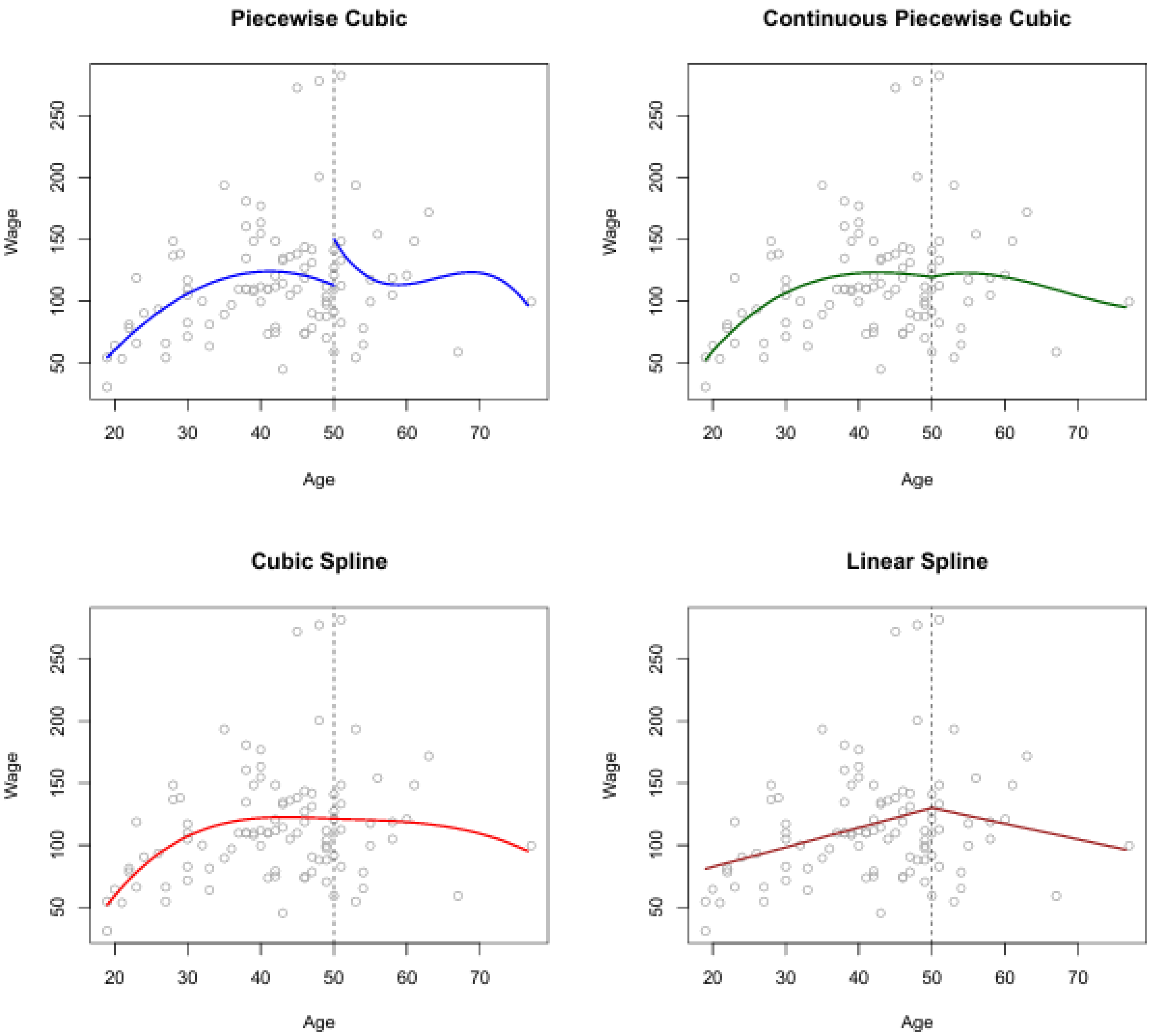
  would allow for different linear functions in each age category.
- Choice of cut points or **knots** can be problematic . For creating non-linearities , smoother alternatives such as **splines** are available.

Piecewise Polynomials

- Instead of a single polynomial in X over its whole domain, we can rather use different polynomials in regions defined by knots. E.g. (see figure)

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

- Better to add constraints to the polynomials, e.g. continuity. The maximum order of the derivative whose continuity you can enforce is one less than the order of the polynomial. e.g. linear function is a 1-degree polynomial so you can enforce the continuity of only $f^0(x)$ but for piecewise cubic you can enforce continuity of $f^0(x), f^1(x)$, and $f^2(x)$.
- **Splines** have the "maximum" amount of continuity.



1. **Linear Splines**

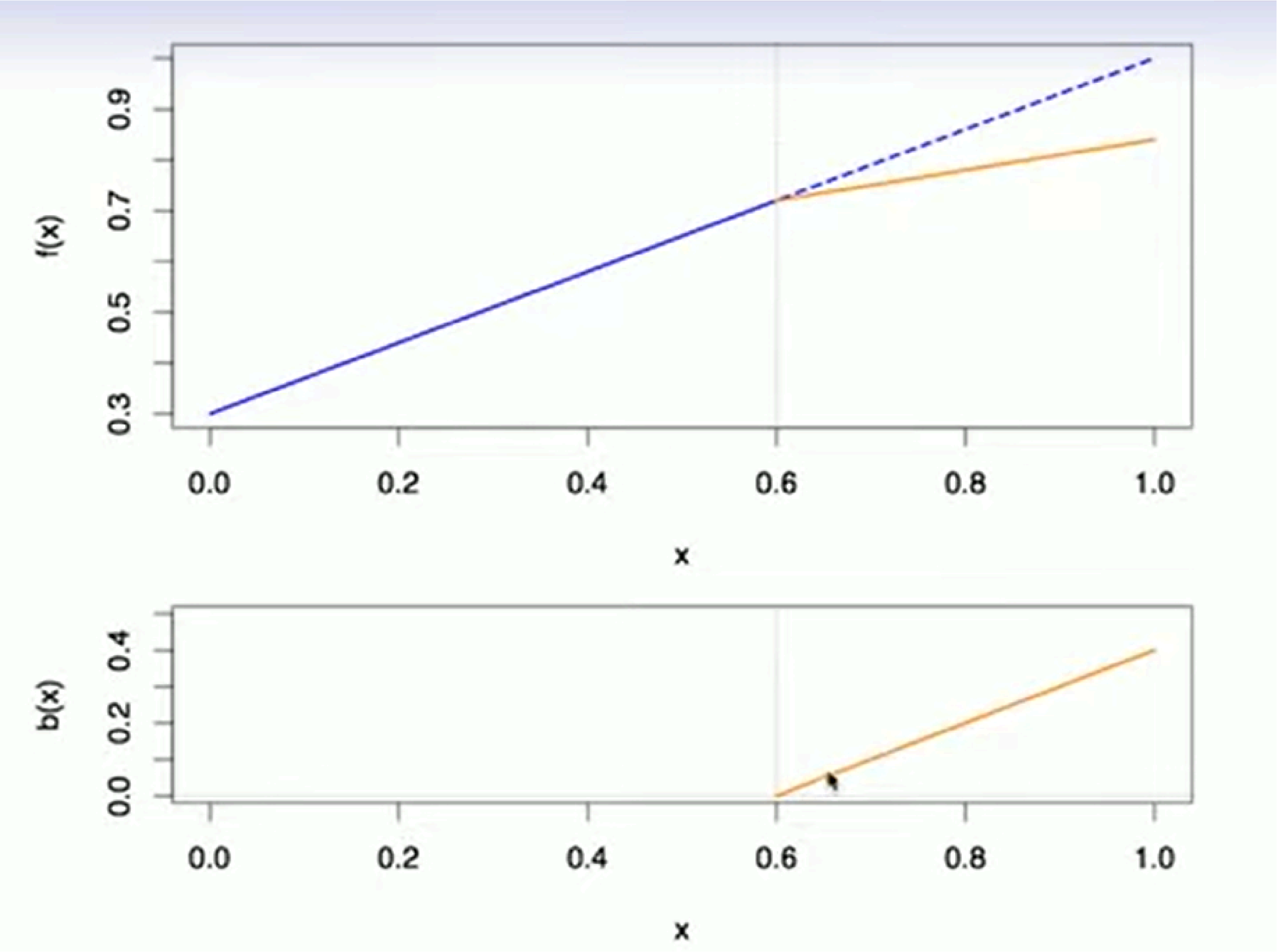
   - **A linear spline with knots at ξ_k, $k = 1, \ldots, K$ is a piecewise linear polynomial continuous at each knot.**
   - We can represent this model as

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

where the b_k are **basis functions**.

- $$b_1(x_i) = x_i$$
$$b_{k+1}(x_i) = (x_i - \xi_k)_+, \quad k = 1, \ldots, K$$
- Here the $()_+$ means **positive part**; i.e.

$$(x_i - \xi_k)_+ = \begin{cases} x_i - \xi_k & \text{if } x_i > \xi_k \\ 0 & \text{otherwise} \end{cases}$$



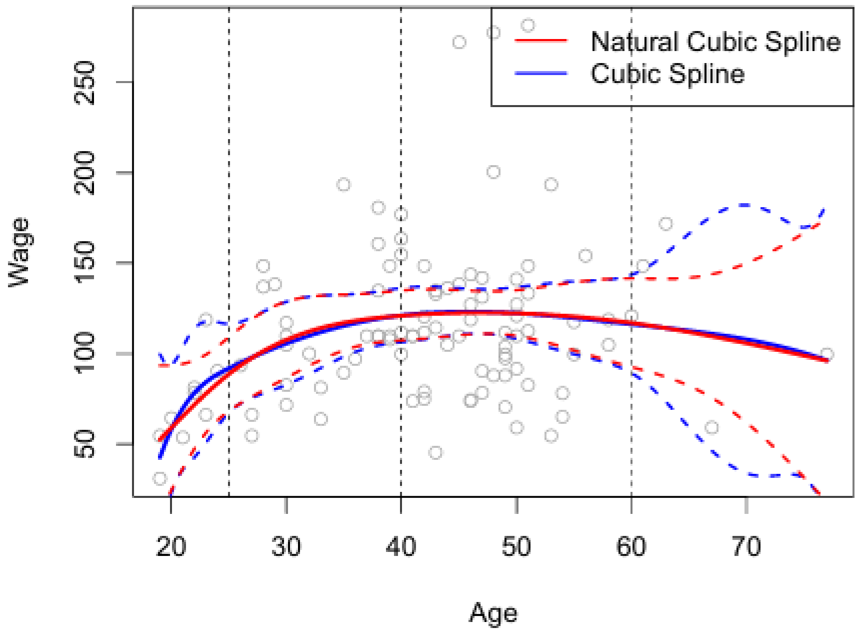
2. **Cubic Splines**
    - **A cubic spline with knots at ξ_k, $k = 1, \ldots, K$ is a piecewise cubic polynomial with continuous derivatives up to order 2 at each knot.**
    - Again we can represent this model with truncated power basis functions

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_{K+3} b_{K+3}(x_i) + \epsilon_i,$$
$$b_1(x_i) = x_i$$
$$b_2(x_i) = x_i^2$$
$$b_3(x_i) = x_i^3$$

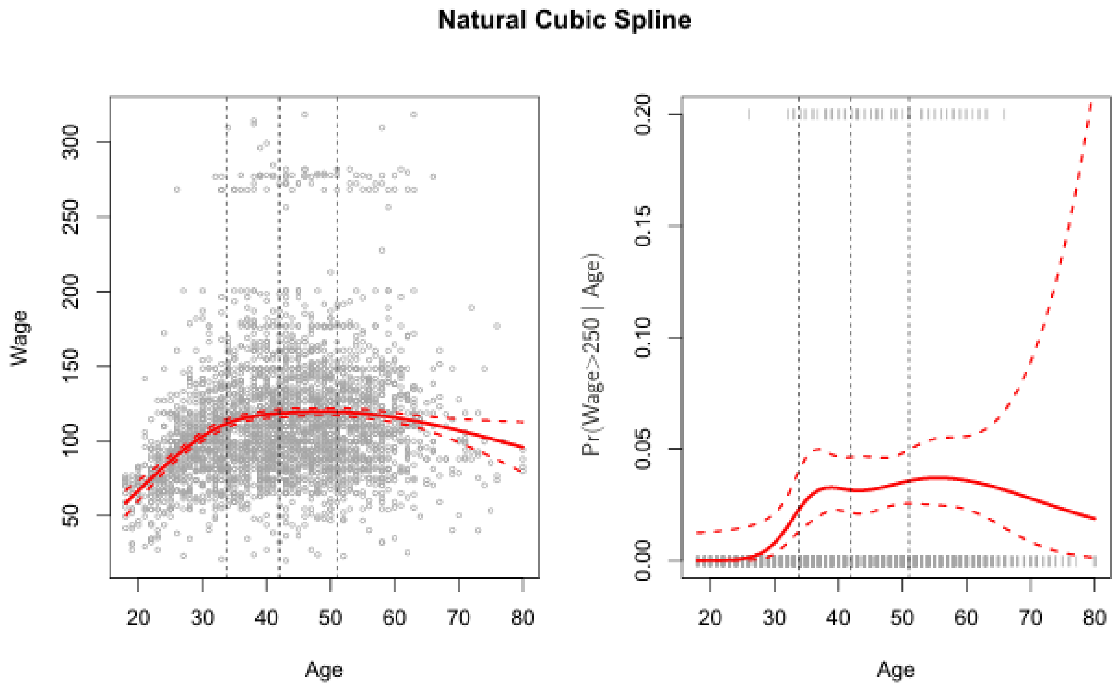
$$b_{k+3}(x_i) = (x_i - \xi_k)_{+}^3, \quad k = 1, \ldots, K$$

```
        where
        $$(x_i - \xi_k)_{+}^3 = \begin{cases} (x_i - \xi_k)^3 & \text{if } x_i > \xi_k \\ 0 & \text{otherwise} \end{cases}$$



        ![[Pasted image 20240830111231.png]]
- Orange curve in the second figure is the truncated power basis function that is zero at the knot, the first and second derivative are also zero at the knot.
- Blue is the global cubic polynomial in the first figure, orange represents the changed polynomial.
```

3. Natural Cubic Splines



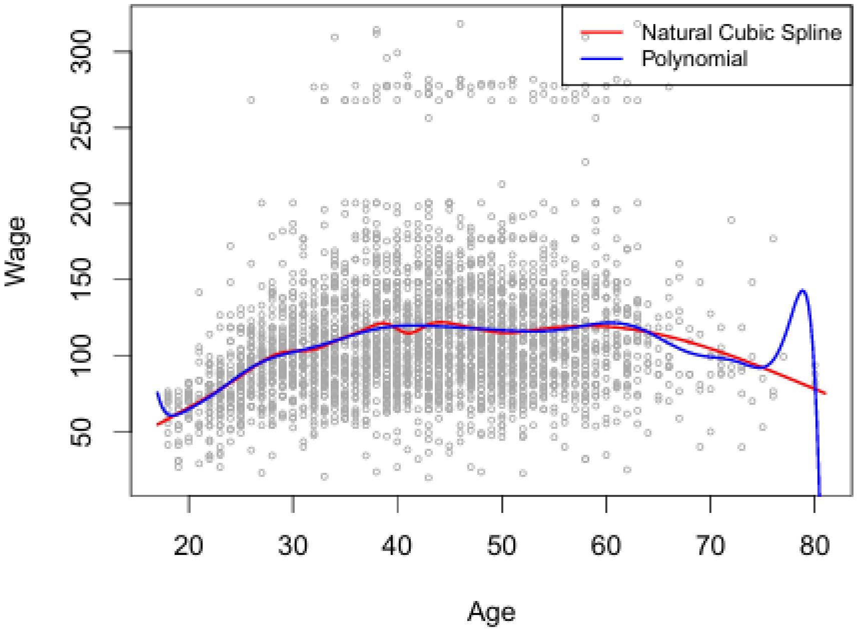
- A natural cubic spline extrapolates linearly beyond the boundary knot. This adds $4 = 2 * 2$ extra constraints, and allows us to put more internal knots for the same degrees of freedom as a regular cubic spline.
- What this means is that the ends of the spline aren't left unconstrained , as we saw for the cubic polynomial that the tail could wag a lot.
- You can see that fits don't have much difference but the standard errors are very different.



- A natural cubic spline function with four degrees of freedom is fit to the Wage data.
- Left : A spline is fit to wage(in thousands of dollars)as a function of age.
- Right : Logistic regression is used to model the binary event $wage > 250$ as a function of age. The fitted posterior probability of wage exceeding $250,000 is shown. The dashed lines denote the knot locations.

But where to place the knots?

- One strategy is to decide K, the no, of knots , and then place them at appropriate quantiles of observed X.
- A cubic spline with K knots has $K + 4$ parameters or degrees of freedom.
- A natural spline with K knots has K degrees of freedom.



- On the Wage data set, a natural cubic spline with 15 degrees of freedom is compared to a degree-15 polynomial .Polynomials can show wild behavior, especially near the tails.

4. **Smoothing Splines**
   The take away message from smoothing splines is a way to fit splines without worrying about the knots. But it approaches a problem from a completely different pov.
   Consider this criterion for fitting a smooth function $g(x)$ to some data:

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

- The first term is RSS and tries to make $g(x)$ match the data at each x_i
- The second term constrains the functions over we search to smooth. It is called **roughness penalty** and controls how wiggly $g(x)$ is.
  - It is modulated by the tuning parameter $\lambda \geq 0$. The smaller λ , the more wiggly the function , eventually interpolating y_i when $\lambda = 0$.
  - As $\lambda \rightarrow \infty$, the function $g(x)$ become linear.

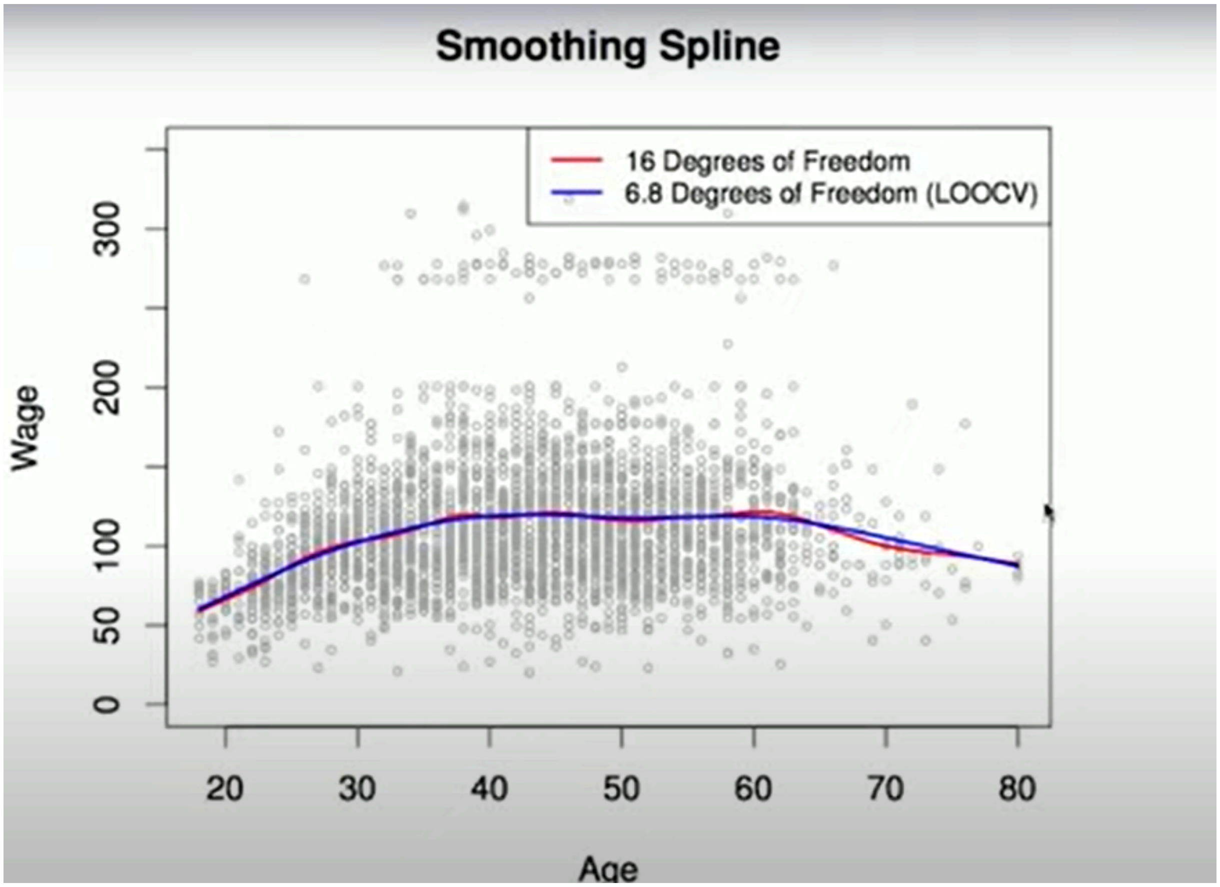
The solution is a natural cubic spline, with a knot at every unique value of x_i . The roughness penalty still controls the roughness via λ .

- The smoothing splines avoid the knot selection issue , leaving a single λ to be chosen.
- The vector of n fitted values can be written as $\hat{g}_\lambda = S_\lambda y$,where S_λ is a n X n matrix (determined by the x_i and λ).
- The effective degrees of freedom are given by

$$df_\lambda = \sum_{i=1}^{n} (S_\lambda)_{ii}$$

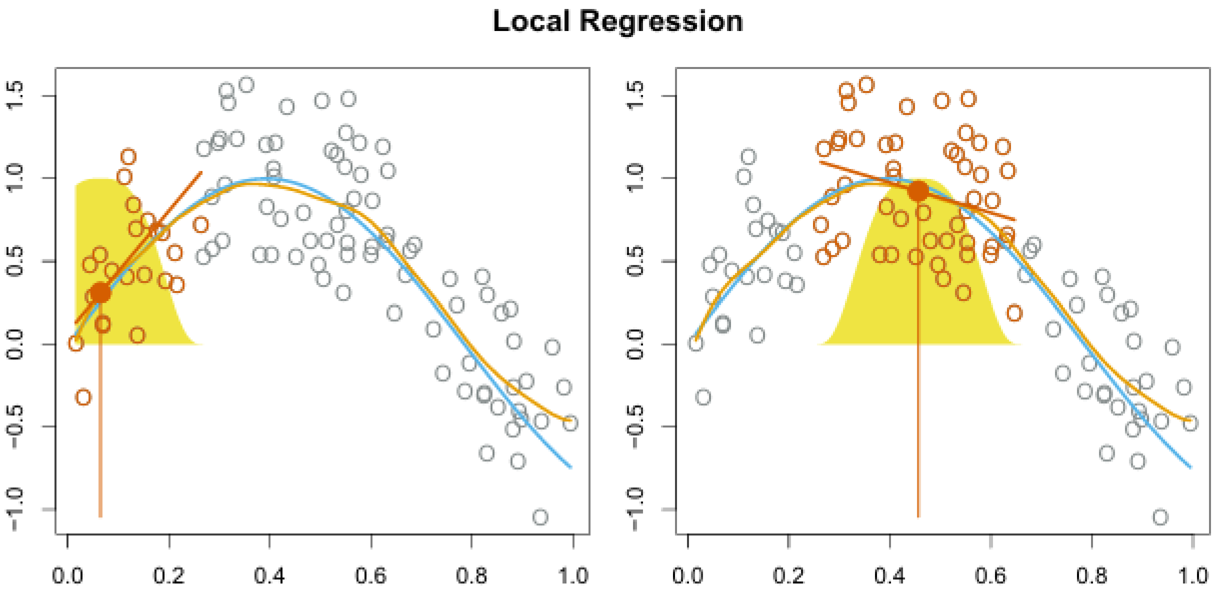
- We can now specify df rather than λ .
- If you don't want to fix the λ , you can actually determine it using the LOOCV :

$$textRSS_{\text{CV}}(\lambda) = \sum_{i=1}^{n} \left(y_i - \hat{g}_\lambda^{(-i)}(x_i)\right)^2 = \sum_{i=1}^{n} \left[\frac{y_i - \hat{g}_\lambda(x_i)}{1 - \{S_\lambda\}_{ii}}\right]^2$$



- So here's an example, the two functions almost the same but one is using 16 fixed degrees of freedom and other got calculated from cross-validation and ended up being 6.8 dof . But why decimal? the effective dof formula doesn't guarantees you integers.

Local Regression

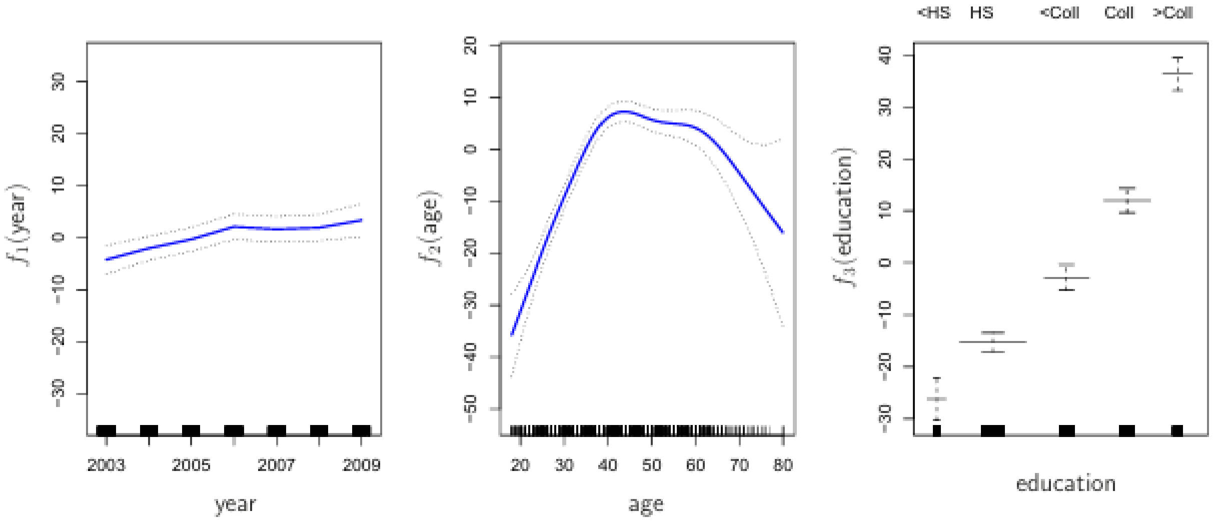


- Local regression illustrated on some simulated data, where the blue curve represents $f(x)$ from which the data were generated, and the light orange curve corresponds to the local regression estimate $\hat{f}(x)$.
- The orange colored points are local to the target point x_0 , represented by the orange vertical line.
- The yellow bell-shape superimposed on the plot indicates weights assigned to each point, decreasing to zero with distance from the target point.
- The fit $\hat{f}(x_0)$ at x_0 is obtained by fitting a weighted linear regression (orange line segment), and using the fitted value at x_0 (orange solid dot) as the estimate $\hat{f}(x0)$.
- Turns out Local Regression and cubic smoothing splines are the two best ways of doing smoothing and if you set the dof to be roughly equal, they look pretty similar too in general.

Generalized Additive Models

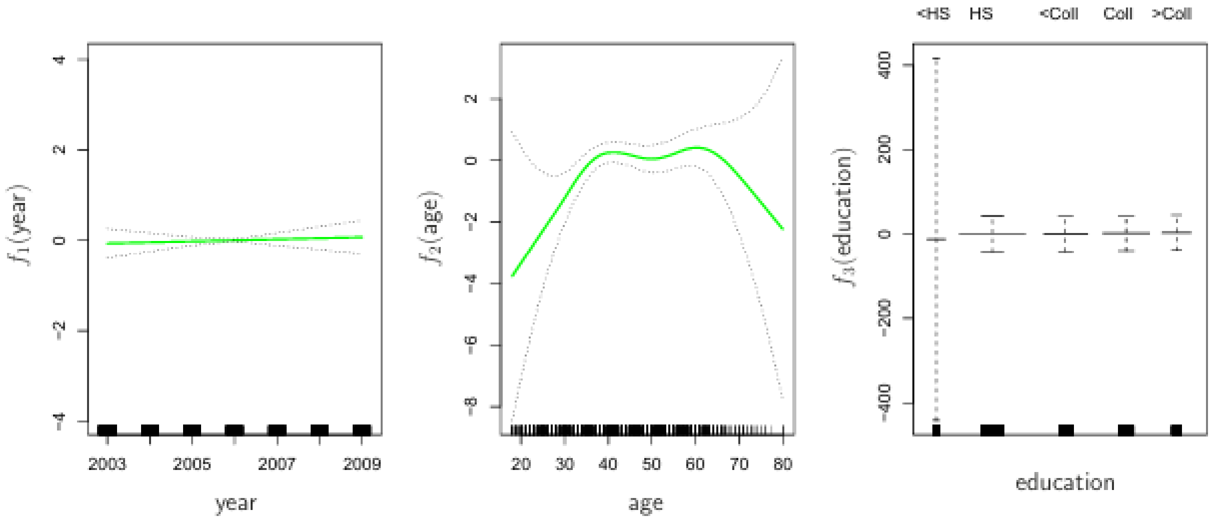
- Allows for flexible non-linearities in several variables, but retains the additive structure of linear models

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



- Can fit a GAM simply using, e.g. natural splines $lm(wage \to ns(year, df = 5) + ns(age, df = 5) + education)$
- Coefficients aren't interesting; fitted functions are.
- Can mix terms- some linear, some non-linear - and use $anova()$ to compare models.
- Can use smoothing splines or local regression as well: $gam(wage \to s(year, df = 5) + lo(age, df = .5) + education)$
- GAMs are additive , although low-order interactions can be included in a natural way using e.g. bivariate smoothers or interactions of the form $ns(age, df = 5) : ns(year, df = 5)$.
- You can fit GAM models for classification

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


$f_1(\text{year})$
year
$f_2(\text{age})$
age
<HS
HS
<Coll
Coll
>Coll
$f_3(\text{education})$
education