Moving Beyond Linearity (Chapter 7)

Set up: The usual binary (0/1) classification and regression setting with response Y and p predictors $X_1, \ldots, X_p$. The data consist of $(Y_i, X_{i1}, \ldots, X_{ip})$, $i = 1, \ldots, n$, from n independent subjects.We model $E(Y|X) = f(X)$ in linear regression and $P(Y = 1|X) = p(X)$ in logistic regression. The standard approach assumes **linearity** in X_j, i.e.,

$$f(x) \text{ or } \text{logit}\{p(x)\} = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p.$$

Issue: The assumption of linearity in X_j is not always reasonable. We would like to allow non-linear effects.

Goal: Use a **linear modeling** approach where $f(x)$ or $\text{logit}\{p(x)\}$ is linear in β_j but may be non-linear in X_j via:

- **polynomials** — add polynomials in X_j as predictors
- **step functions** — categorize X_j and fit a piecewise const
- **splines** — fit piecewise polynomials that are constrained
- **local regression** — similar to splines but works differently
- **generalized additive models** — multiple predictors

To begin, we will assume only one predictor X.

Polynomial regression: Assume a degree-d polynomial in X. The model becomes:

$$f(x) \text{ or } \text{logit}\{p(x)\} = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_d x^d.$$

- As these models are linear in the β coefficients, they can be fit using the usual approach — least squares for linear regression and maximum likelihood for logistic regression. They also provide SEs and confidence intervals.
- Choosing d — we can use a hypothesis testing approach or a cross-validation or validation set approach
- In practice, $d > 3$ or 4 is rare
- Expanding the feature space by adding higher order terms
- Imposes a *global* structure on the relationship

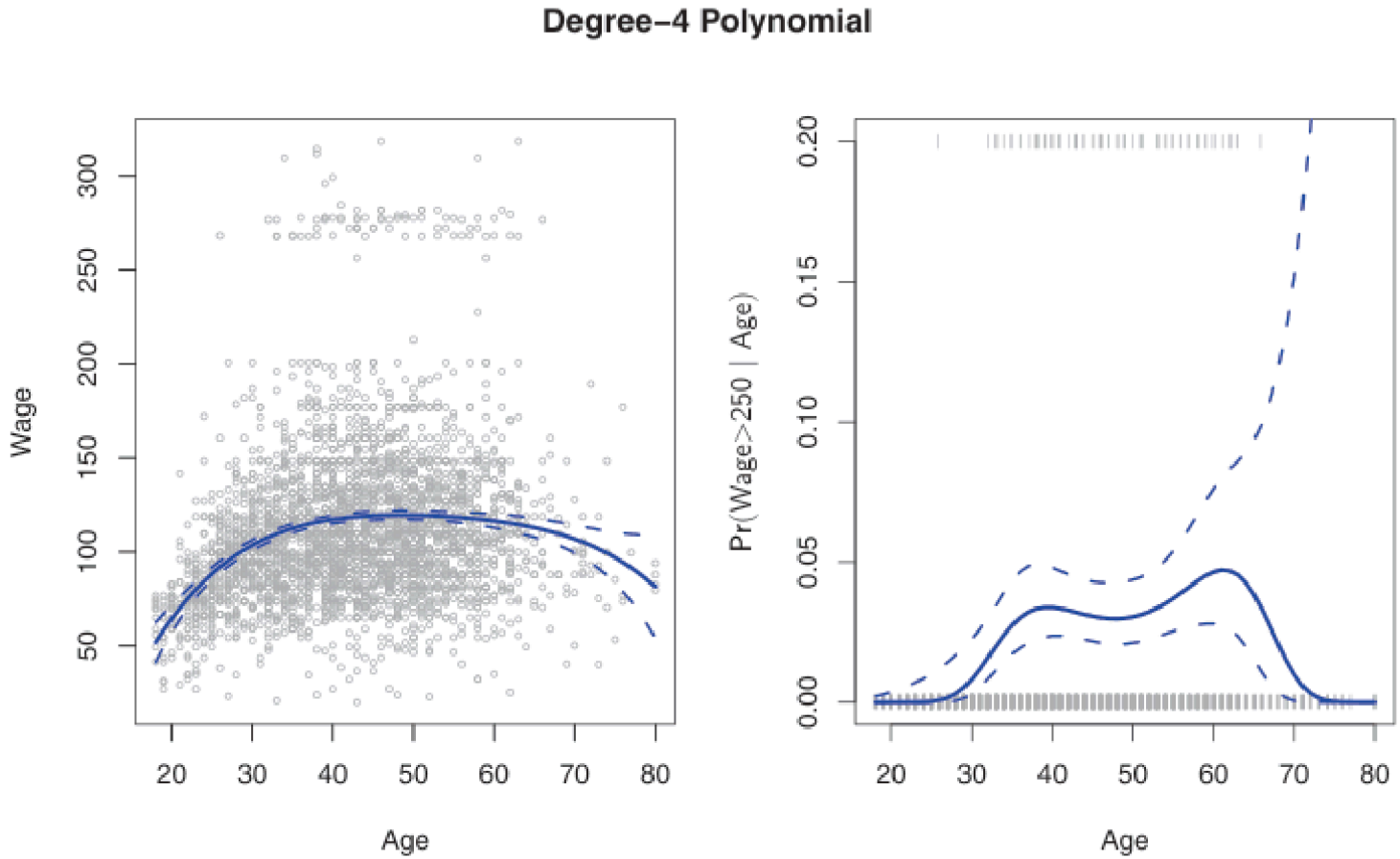

FIGURE 7.1. *The* `Wage` *data.* Left: *The solid blue curve is a degree-4 polynomial of* `wage` *(in thousands of dollars) as a function of* `age`*, fit by least squares. The dotted curves indicate an estimated 95 % confidence interval.* Right: *We model the binary event* `wage>250` *using logistic 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.*

Source: ISL

Regression with step functions: Divide the range of X into bins and fit a constant in each bin. To be specific, let $c_1, c_2, \ldots, c_K$ be the cut points that divide the range of X into $K+1$ bins. Associate a *dummy* indicator variable to each bin:

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

Fit the model:

$$f(x) \text{ or } \text{logit}\{p(x)\} = \beta_0 + \beta_1 C_1(x) + \ldots + \beta_K C_K(x),$$

which is a *piecewise constant function.*

- Treating X as a qualitative variable with $K+1$ levels and $\{X < c_1\}$ category as the base category
- $\beta_0 =$ function value when $X < c_1$, $\beta_j =$ change in function value for $c_j \leq X < c_{j+1}$ relative to the base category
- Models are linear in β coefficients — can be fit in usual way

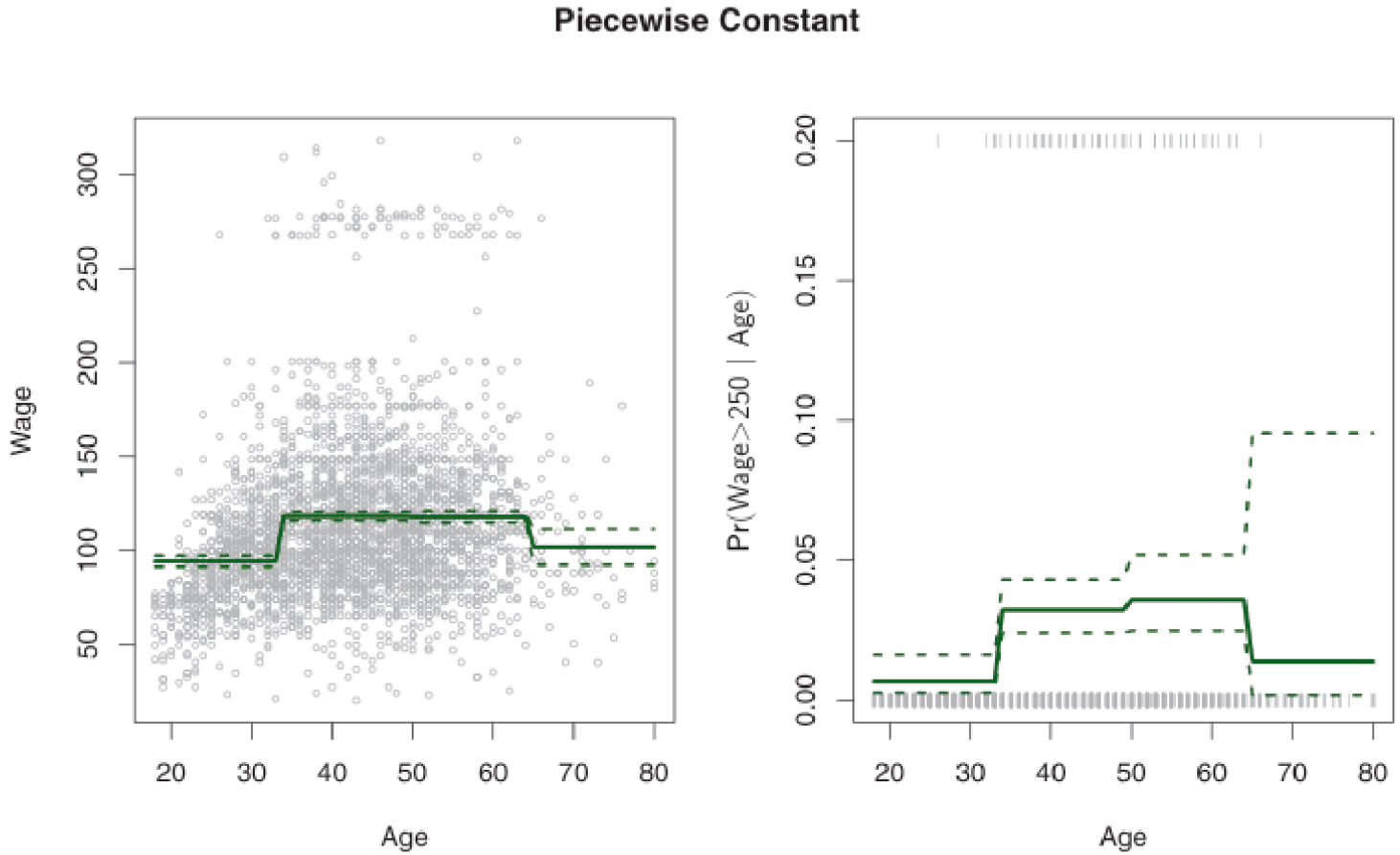

FIGURE 7.2. *The* `Wage` *data.* Left: *The solid curve displays the fitted value from a least squares regression of* `wage` *(in thousands of dollars) using step functions of* `age`*. The dotted curves indicate an estimated 95 % confidence interval.* Right: *We model the binary event* `wage>250` *using logistic regression, again using step functions of* `age`*. The fitted posterior probability of* `wage` *exceeding* \$250,000 *is shown, along with an estimated 95 % confidence interval.*

Source: ISL

Regression with basis functions: Consider a family of functions $b_1(X), \ldots, b_K(X)$ — aka *basis functions* — whose linear combination approximates the true underlying function being modeled. Fit the model

$$f(x) \text{ or } \text{logit}\{p(x)\} = \beta_0 + \beta_1 b_1(x) + \ldots + \beta_K b_K(x).$$

- The basis functions are known and specified beforehand.
- A general approach of which both polynomial and step function models are special cases.
- **Polynomial model ($K = d$)**: $b_1(x) = x, \ldots, b_d(x) = x^d$
- **Step function**: $b_j(x) = I(c_j \leq X < c_{j+1})$, $j = 1, \ldots, K$
- Many alternative basis functions are available — e.g., wavelets, Fourier functions, and spline functions (our focus)
- Models are linear in β coefficients — can be fit in usual way

Regression Splines

To understand this concept, let's start with regression and consider a specific situation. Divide the range of X into two parts — $X < c$ and $X \geq c$. The cut point c is called a *knot* (or an *interior knot*, to be more precise). Consider fitting a separate cubic polynomial in each part:

$$f(x) = \begin{cases} f_1(x) = \beta_{01} + \beta_{11}x + \beta_{21}x^2 + \beta_{31}x^3, & \text{if } x < c, \\ f_2(x) = \beta_{02} + \beta_{12}x + \beta_{22}x^2 + \beta_{32}x^3, & \text{if } x \geq c. \end{cases}$$

This is a *piecewise cubic* regression model. It has 8 df.

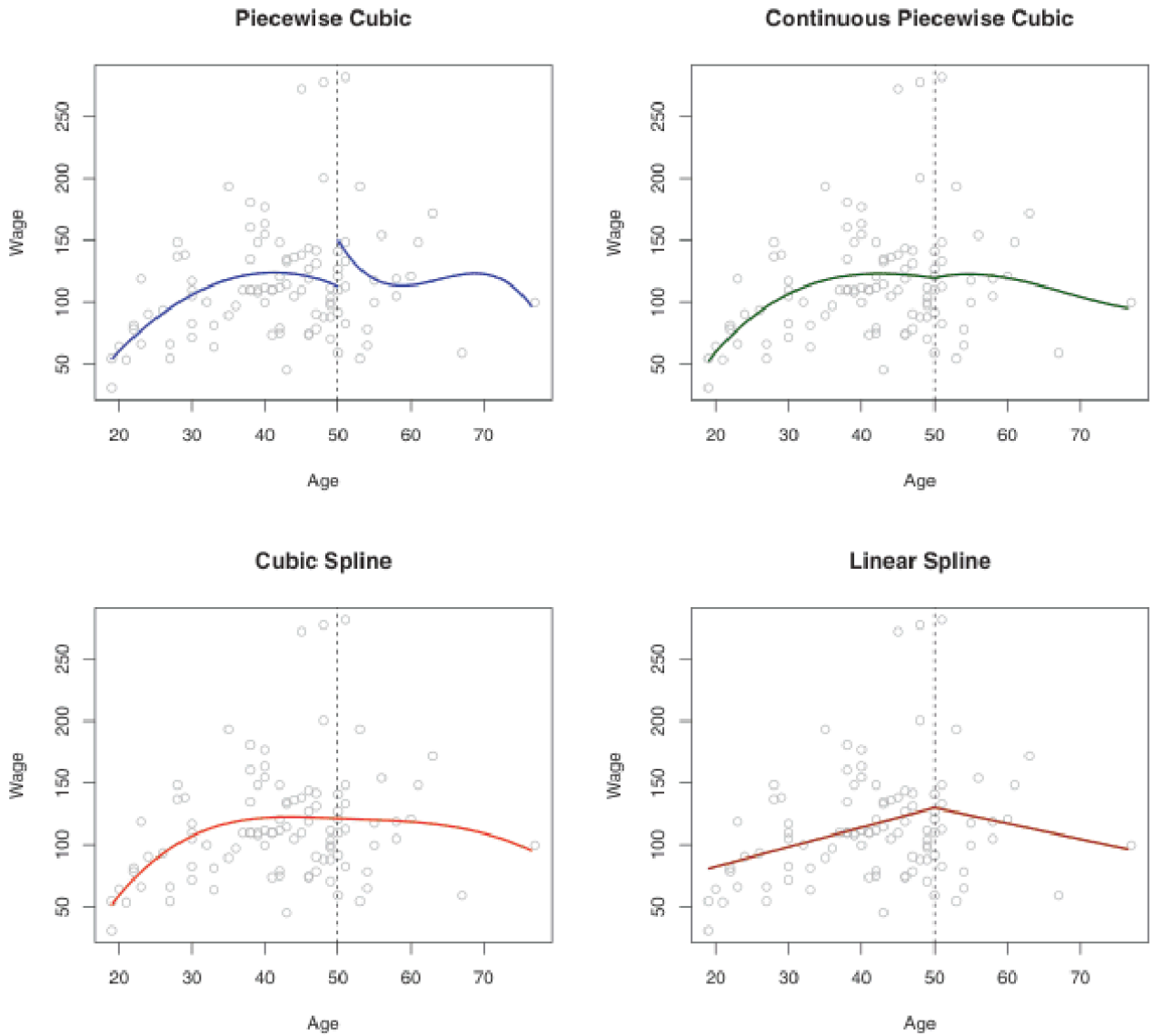

FIGURE 7.3. *Various piecewise polynomials are fit to a subset of the* `Wage` *data, with a knot at* `age=50`. Top Left: *The cubic polynomials are unconstrained.* Top Right: *The cubic polynomials are constrained to be continuous at* `age=50`. Bottom Left: *The cubic polynomials are constrained to be continuous, and to have continuous first and second derivatives.* Bottom Right: *A linear spline is shown, which is constrained to be continuous.*

Q: Should the cubics be constrained? How about requiring:

- them to be continuous at c, i.e., $f_1(c) = f_2(c)$ — frees 1 df
- their first derivatives to be continuous at c, i.e., $f_1'(c) = f_2'(c)$ — frees 1 df
- their second derivatives to be continuous at c, i.e., $f_1''(c) = f_2''(c)$ — frees 1 df

The constrained piecewise cubic model has 3 linear constraints, resulting in $8 - 3 = 5$ df, and it is smooth at c.

Q: Why restrict to only one knot? We can choose K knots in the range of X, fit a separate cubic in each region; and require continuity, continuity of the first derivative, and continuity of the second derivative of the resulting piecewise cubic function at each knot. This model is called a **cubic spline model** and it has $4 + K$ df.

Regression spline: A regression spline of degree d is a piecewise degree-d polynomial, with continuity of function and continuity in derivative of degrees $1, \ldots, d-1$ at each knot.

Linear spline: A line in each region of X defined by the knots, requiring continuity at the knots — **broken stick model**. It has $2 + K$ df.

Quadratic spline: A quadratic in each region of X defined by the knots, requiring continuity and continuity of first derivative at the knots. It has $3 + K$ df.

Cubic spline: A cubic in each region of X, requiring continuity and continuity of first and second derivatives at the knots. It has $4 + K$ df. These are popular as the discontinuity at the knots in the third derivative is hard to detect by human eyes — this is a statistical folklore.

In practice, we rarely use $d > 3$. Moreover, the df for the intercept is often counted separately. Therefore, we may associate a cubic spline with $3 + K$ df, assuming another df for the intercept. This is indeed the case in our figures and in `Python`

Basis Representation of a Spline

For an appropriate choice of basis functions $b_1(x), \ldots, b_{K+d}(x)$, a degree-$d$ regression spline with K knots can be represented as

$$f(x) = \beta_0 + \beta_1 b_1(x) + \ldots + \beta_{K+d} b_{K+d}(x).$$

Many choices exist for the basis functions — just like the case of polynomials.

Truncated polynomial basis: Start with a basis for a degree-d polynomial, namely, $x, x^2, \ldots, x^d$, and then add one *truncated power basis function* $\{(x-c)_+\}^d$ for each knot c. Here

$$(x-c)_+ = \begin{cases} 0, & \text{if } x < c, \\ x-c, & \text{if } x \geq c, \end{cases}$$

is called the *positive* part of $(x-c)$. Thus, a degree-d regression spline can be represented as

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_d x^d + \sum_{j=1}^{K} \beta_{d+j} \{(x-c_j)_+\}^d.$$

- This model has $1 + d + K$ df.
- Satisfies the definition of a regression spline — f is continuous and has continuous derivatives up to order $d - 1$ at each knot.
- Model is linear in β coefficients — can be fit using the usual least squares
- Nonparametric modeling of f — flexibility helps model non-linear relationships

B-spline basis: Messy to write but more numerically stable than the truncated polynomial basis.

Note: Often, in practice, we represent the mean function using the truncated polynomial basis but do the actual numerical calculations using an equivalent B-spline basis, obtained via a linear transformation. The software packages may do this automatically.

Issue: A regression spline can have high variance near the boundaries — **boundary effect**.

Natural spline: A natural spline is a regression spline with two additional *boundary constraints*: the function is required to be linear in the region where X is smaller than the smallest knot or larger than the largest knot.

- "Natural constraints" — they generally lead to more stable estimates at the boundaries
- A cubic regression spline model has $4 + K$ df, whereas a natural cubic spline model has $K + 2$ df (including intercept).

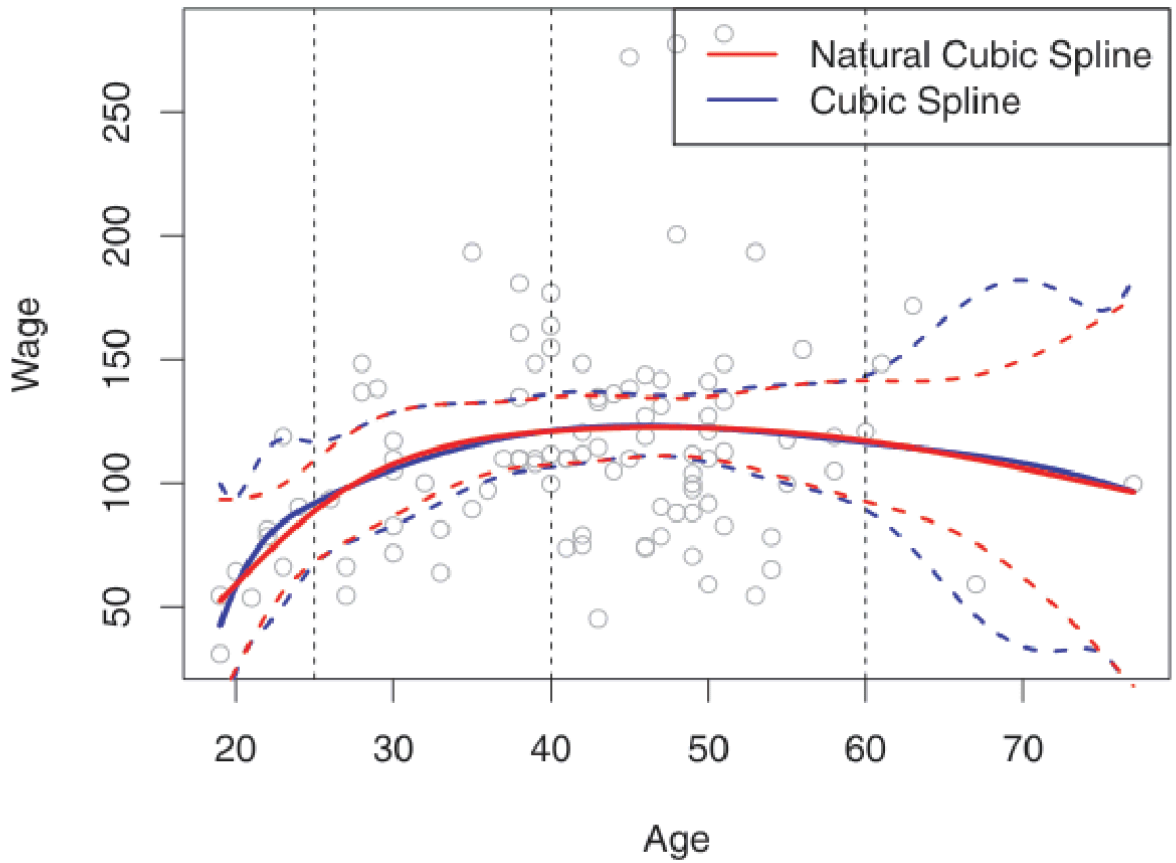

FIGURE 7.4. *A cubic spline and a natural cubic spline, with three knots, fit to a subset of the* `Wage` *data.*

Source: ISL

Choosing the number and location of the knots:

Number of knots K: This is tied to the df of the model, e.g., a cubic regression spline model has $4 + K$ df.

- Choose a large K and use regularization to fit the model — **penalized regression spline** (not in this course).
- Choose K by cross-validation (our focus).
- Need K to be moderately large if many "peaks" and "valleys" are seen in the scatterplot.

Location of knots: We would like the spline to be more flexible in the regions where the underlying function changes rapidly. So, we can place more knots in such regions, and fewer knots in regions where the function is stable. In practice, however, we tend to place knots at the quantiles of X. For example, if $K = 3$, the knots are taken be the quartiles of X.

Splines in case of logistic regression: All the ideas apply with $f(x)$ replaced by $\text{logit}\{p(x)\}$ and fitting done by maximum likelihood instead of least squares.

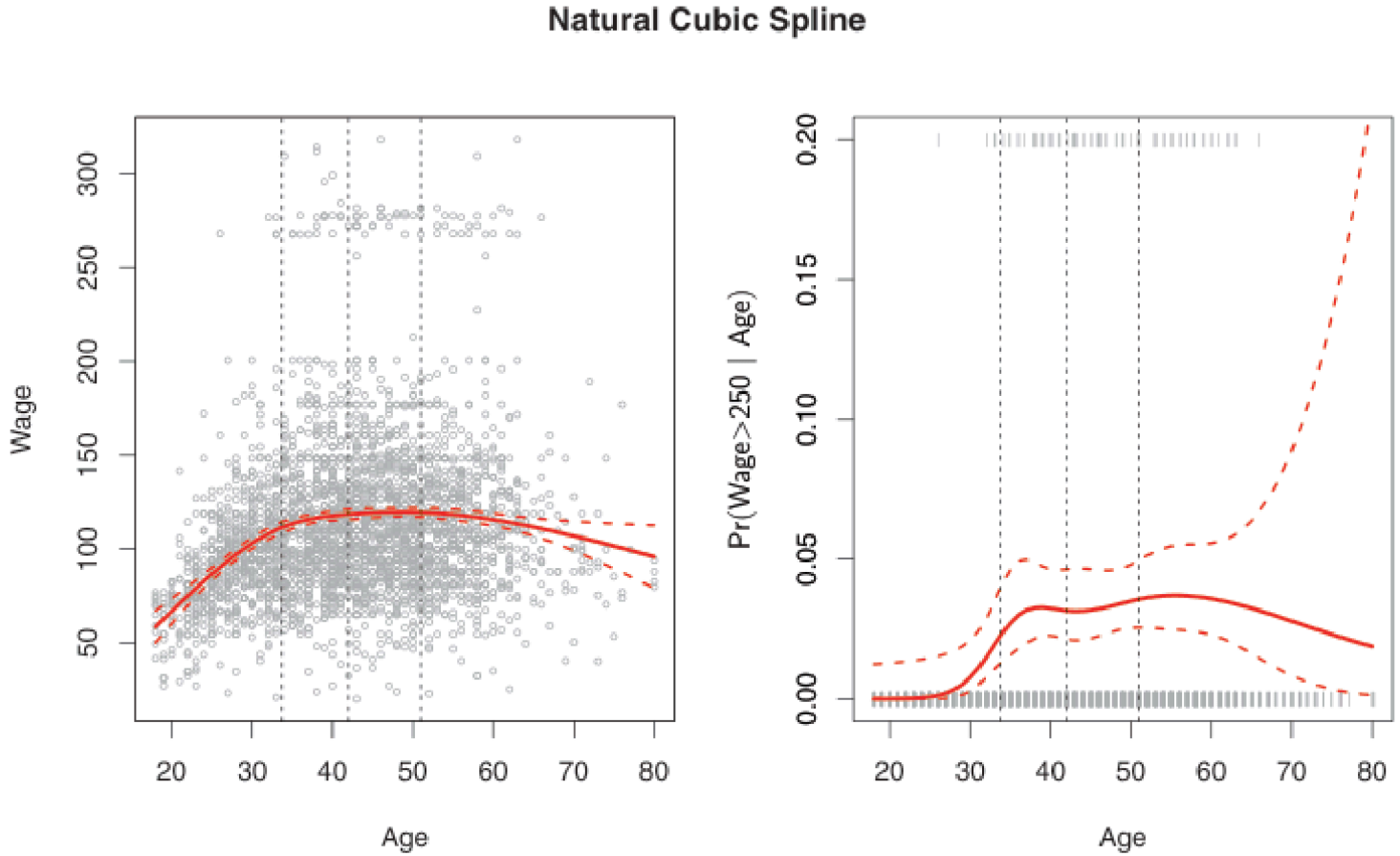

FIGURE 7.5. *A natural cubic spline function with four degrees of freedom is fit to the* `Wage` *data.* Left: *A spline is fit to* `wage` *(in thousands of dollars) as a function of* `age`. Right: *Logistic regression is used to model the binary event* `wage>250` *as a function of* `age`. *The fitted posterior probability of* `wage` *exceeding* $250,000 *is shown.*

Source: ISL

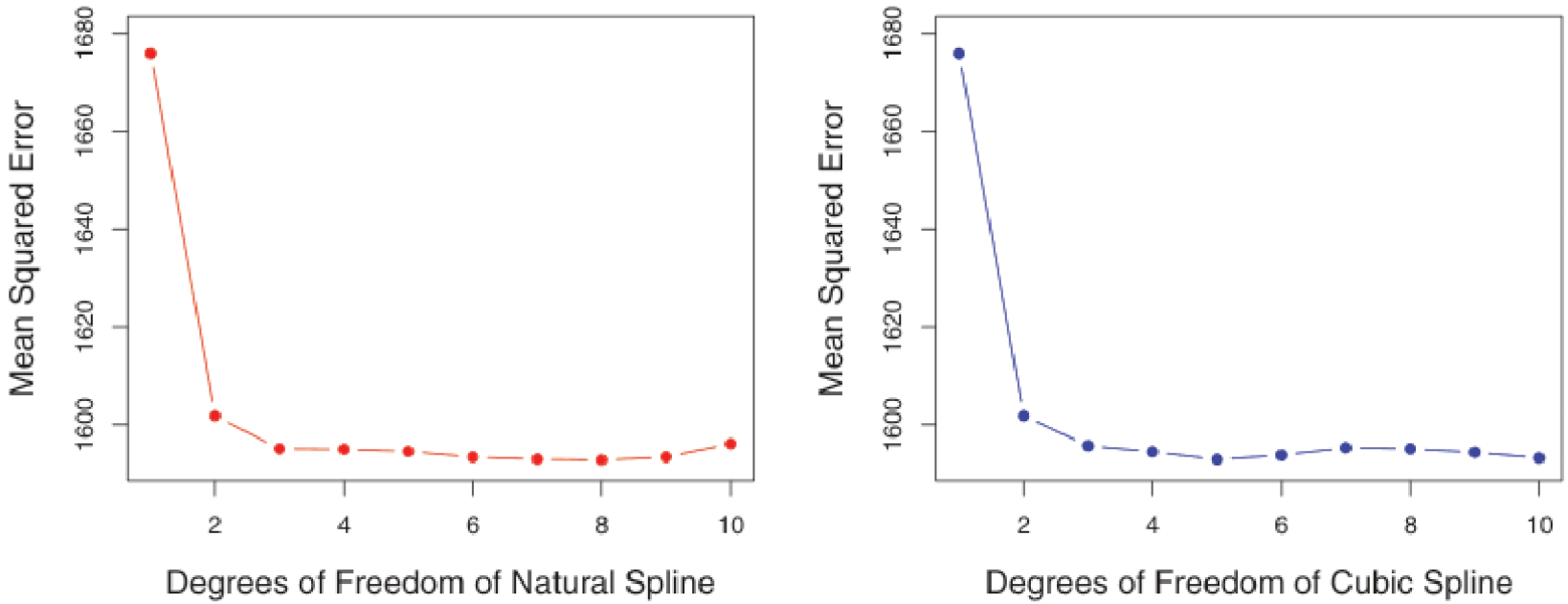

FIGURE 7.6. *Ten-fold cross-validated mean squared errors for selecting the degrees of freedom when fitting splines to the* `Wage` *data. The response is* `wage` *and the predictor* `age`. Left: *A natural cubic spline.* Right: *A cubic spline.*

Source: ISL

Regression splines vs polynomial regression:

- The polynomial regression requires the degree d to be large to produce a flexible fit. But this may result in a very unstable fit, especially near the boundaries.
- Regression splines introduce flexibility by fixing the degree d but by increasing the number of knots. Additional flexibility in placement of knots allows fitting "peaks" and "valleys" in the scatterplot well. Overfitting is avoided by choosing the number of knots by cross-validation (or by fitting the model using a regularization method).
- Generally regression splines are a much better option than a polynomial regression with a large d.

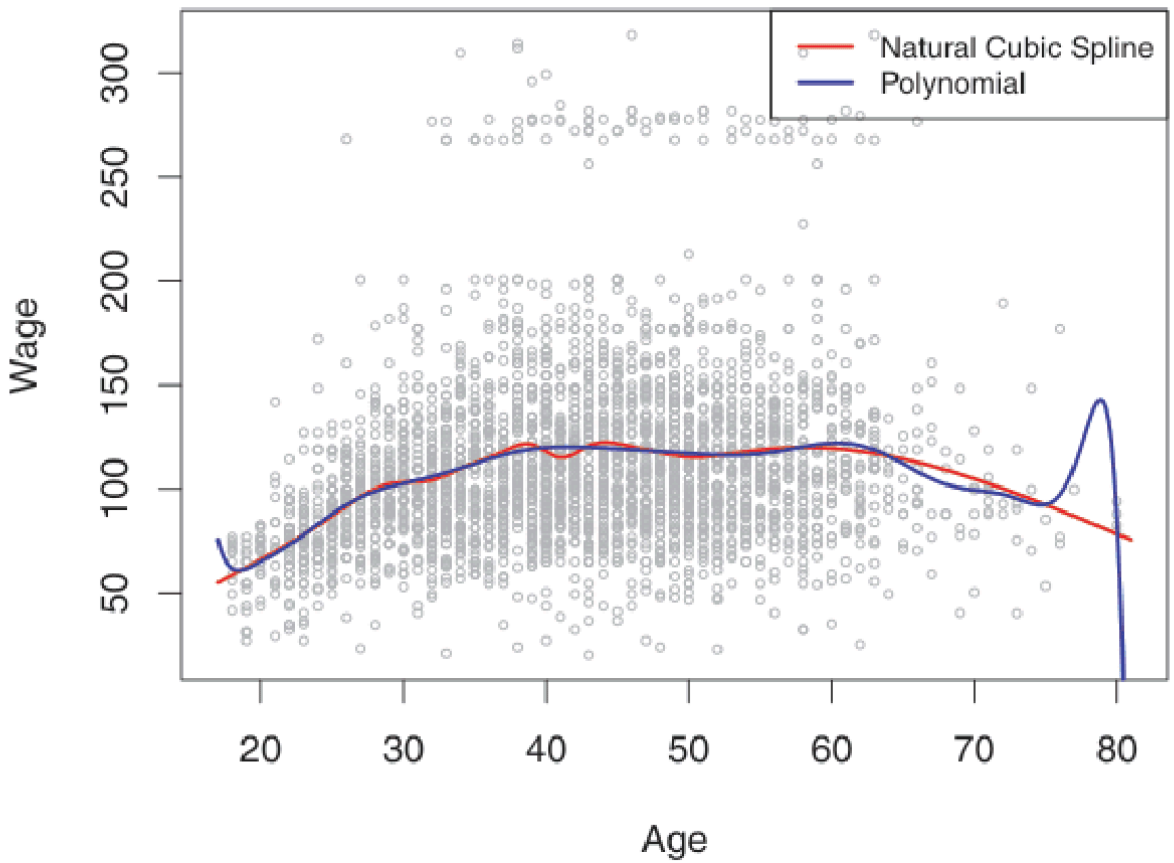

FIGURE 7.7. *On the* `Wage` *data set, a natural cubic spline with 15 degrees of freedom is compared to a degree-*15 *polynomial. Polynomials can show wild behavior, especially near the tails.*

Source: ISL

Smoothing Splines

Let's take an alternative route to the problem of fitting a curve f to the scatterplot. We would like the curve to be *smooth*, i.e., f should have continuous derivatives up to a desired order. In practice, we generally ask for continuous second derivative.

Q: How about f that minimizes the RSS $= \sum_{i=1}^{n}(Y_i - f(X_i))^2$?

A: This f will simply interpolate the points (i.e., zero RSS); it won't be smooth — overfitting.

Instead, take the usual "loss + penalty" approach and find f that minimizes the objective function:

$$\text{RSS} + \lambda \int \{f''(x)\}^2,$$

where λ is a non-negative *tuning* parameter.

- The second derivative $f''(x)$ is a measure of *roughness* (or wiggliness or jumpiness) of the function f at x.
- The linear function $f(x) = \beta_0 + \beta_1 x$ has $f''(x) = 0 \implies$ a straight line is perfectly smooth
- The integral $\int \{f''(x)\}^2$ provides an overall measure of roughness of f on the entire range of X
- $\lambda = 0$: No roughness penalty, f simply interpolates the points — **zero smoothing**
- $\lambda \to \infty$: Infinitely large roughness penalty, f will be the least squares line — **perfect smoothing**
- Larger λ = smoother f
- λ controls the bias-variance tradeoff — as $\lambda \uparrow$, bias $\uparrow$ and variance $\downarrow$

Result: For a given λ, the f that minimizes the objective function is a natural cubic spline with knots at the unique values of $x_1, \ldots, x_n$ — aka **smoothing spline**. The resulting $n \times 1$ vector of fitted values is:

$$\hat{\mathbf{Y}}_\lambda = \mathbf{S}_\lambda \mathbf{Y},$$

where $\mathbf{S}_\lambda$ is a $n \times n$ *smoother matrix* that is available in a closed form and $\mathbf{Y}$ is the $n \times 1$ response vector.

- The smoothing spline is not the same as a natural cubic spline that we get by applying the basis function approach with knots at $x_1, \ldots, x_n$ — it's a *shrunken* (or regularized) version of such a natural cubic spline
- How to count df when a "loss + penalty" approach is used? Simply counting the number of regression coefficients as in a linear model won't work as the coefficients are shrunken.
- Use *effective* df defined as $\text{df}_\lambda = \text{trace}(\mathbf{S}_\lambda)$

In a smoothing spline, there is no need to select the number or location of the knots. By definition, there will be a knot at each unique $x_1, \ldots, x_n$. However, we need to choose λ.

- Choose λ by cross-validation or a validation set approach.
- LOOCV estimate of λ has an explicit formula that only involves the original fit to all of the data.
- Alternatively, we may specify df_λ and find λ by solving $\mathrm{df}_\lambda = \mathrm{trace}(\mathbf{S}_\lambda)$.
- Nonparametric modeling of f — flexibility helps model non-linear relationships

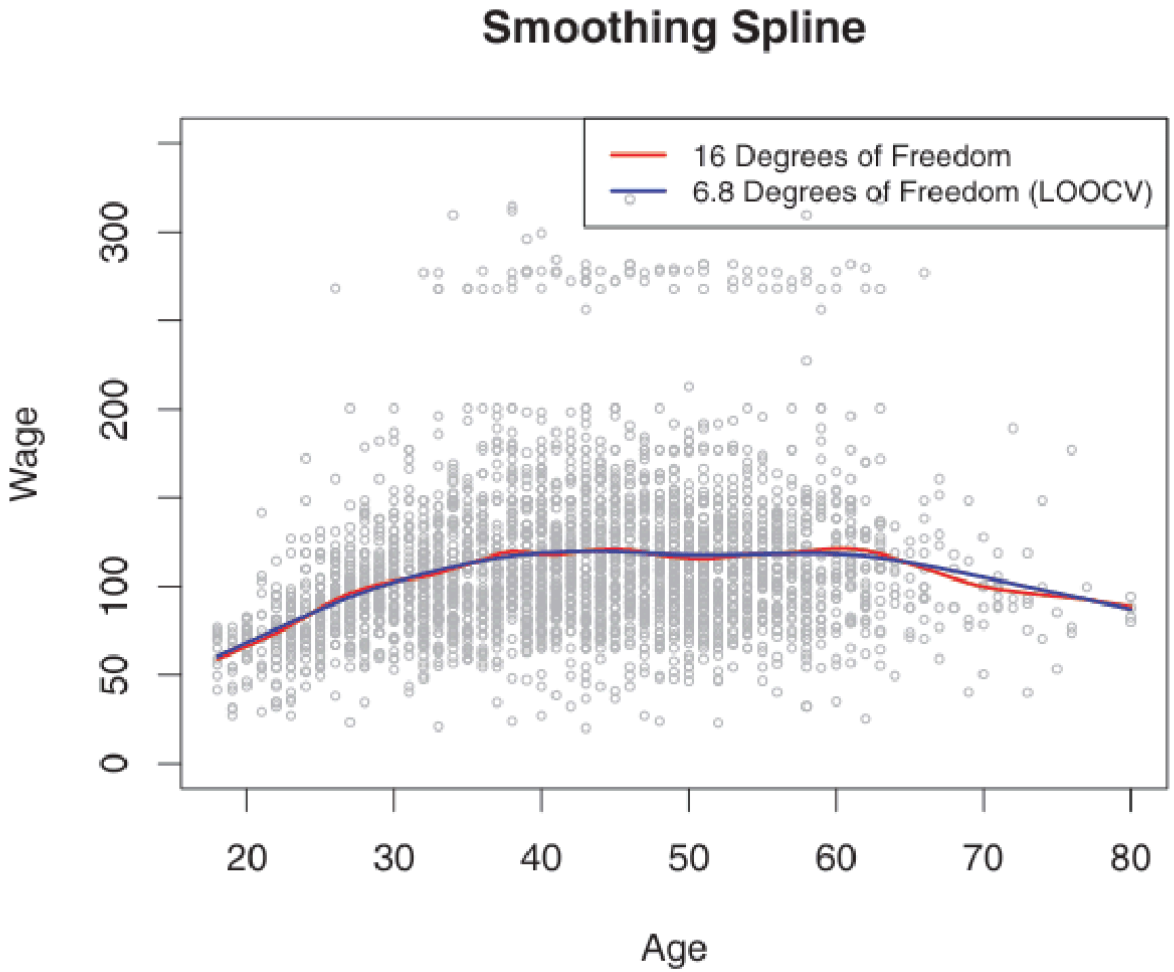

FIGURE 7.8. *Smoothing spline fits to the* `Wage` *data. The red curve results from specifying* 16 *effective degrees of freedom. For the blue curve,* λ *was found automatically by leave-one-out cross-validation, which resulted in* 6.8 *effective degrees of freedom.*

Source: ISL

Local Regression

Local regression provides an alternative to splines for fitting flexible non-linear functions.

- It is *local* in that it computes the fit at a target point x_0 using only the observations near x_0 — aka **span** of x_0.
- It also associates a weight to the observations that depends upon the distance from x_0. Among the observations in the span, the observation closest to x_0 has the highest weight and the observation furthest from x_0 has zero weight. The observations outside the span have zero weight.
- Similar to KNN regression in its use of only the nearby observations but the two methods are different
- Nonparametric estimate of f

Algorithm 7.1 *Local Regression At* $X = x_0$

1. Gather the fraction $s = k/n$ of training points whose x_i are closest to x_0.

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 $\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$.

- As in KNN, we need all the training data each time we wish to compute a prediction

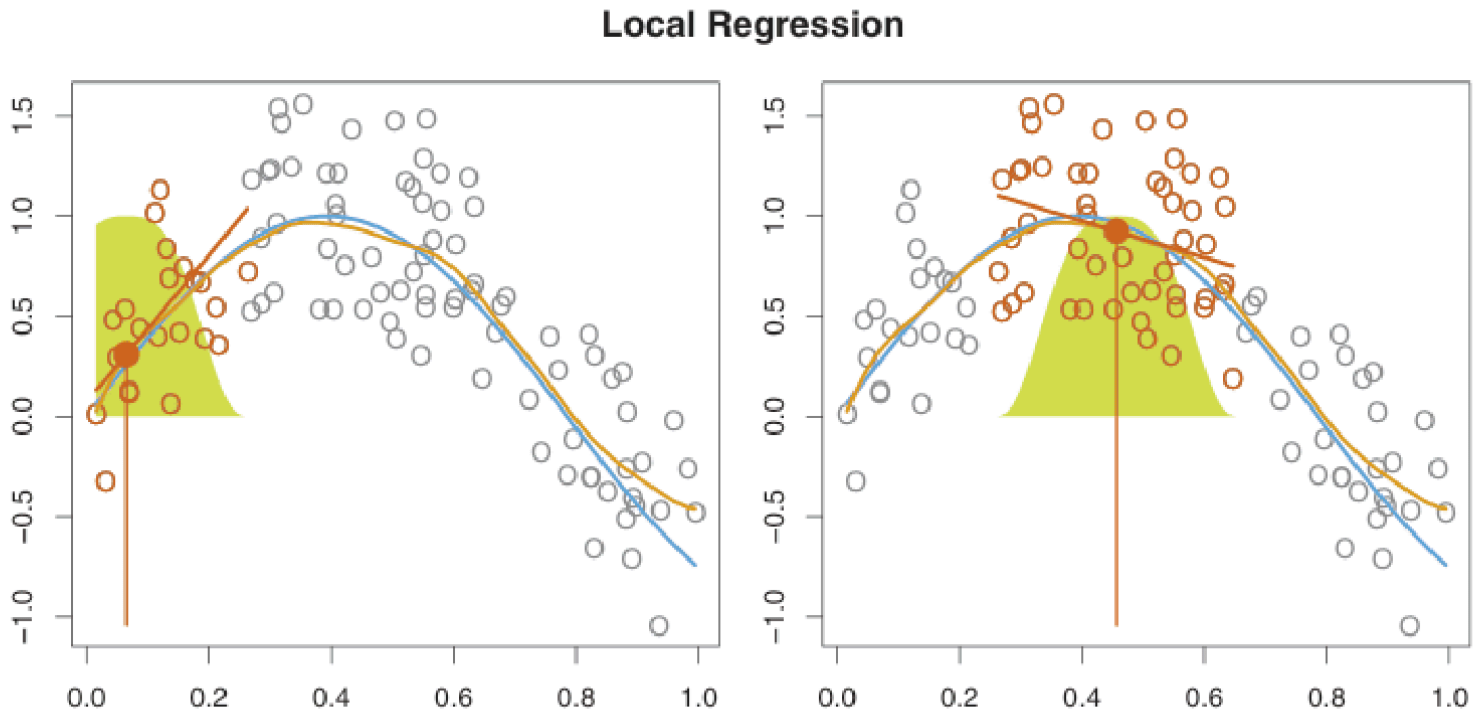

FIGURE 7.9. *Local regression illustrated on some simulated data, where the blue curve represents $f(x)$ from which the data were generated, and the light orange curve corresponds to the local regression estimate $\hat{f}(x)$. The orange colored points are local to the target point x_0, represented by 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 $\hat{f}(x_0)$.*

Source: ISL

Local regression involves making three choices — span s (step 1), weighting (or kernel) function K (step 2), and order of the polynomial to be fit (step 3).

- **Span s is the most important choice**. It plays the role of tuning parameter λ in smoothing splines, and controls the bias-variance tradeoff.
- Smaller s = regression more local (low bias) = more wiggly fit (high variance)
- Larger s = smoother fit
- A very large s = global weighted least squares fit
- Use cross-validation to choose s or specify it directly — experiment with different choices and pick the one that gives the most aesthetically pleasing fit
- A popular choice for the weight function K is the Gaussian density with mean x_0
- We can use cubics for a smoother fit

Generalized Additive Models

So far: Flexible modeling with only one predictor X

Now: Flexible modeling with p predictors $X_1, \ldots, X_p$, denoted by vector X, using **generalized additive models**. Let's first consider the regression case with $f(x) = E(Y|X = x)$. Let $f_j(x_j)$ be the function that captures the effect of X_j on Y. These f_j serve as the *building blocks*.

Generalized additive model (GAM):

$$f(x) = \beta_0 + f_1(x_1) + \ldots + f_p(x_p)$$

- Have a separate f_j for each X_j and simply add them up — **additive model**
- Extends the linear model $f(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$ by replacing $\beta_j x_j$ with $f_j(x_j)$
- **Interpretation**: $f_j(x_j)$ represents the effect of X_j *by holding all other predictors fixed*

- **Qualitative X_j:** Use the dummy variable approach — represent f_j as a linear combination of indicators
- **Quantitative X_j:** May specify f_j parametrically (e.g., as a polynomial or a piecewise constant function in x_j)or nonparametrically (e.g., via splines or local regression in x_j with a specified level of smoothness) — flexibility to capture non-linear effect of X_j

How to fit a GAM?:

Case 1: The entire model can be written as a big linear model in the form $E(Y) = X\beta$ — use least squares. For example, this is the case when the f_j for a quantitative X_j is specified using polynomials or regression and natural splines, but not when it is specified using local regression or smoothing splines — **Why?**

Case 2: Use the *backfitting* algorithm — repeatedly update the fit for each predictor in turn, using partial residual and whatever be the appropriate method for that predictor and holding the fits of the other predictors fixed, until convergence (see book and Ex 7.11-12)

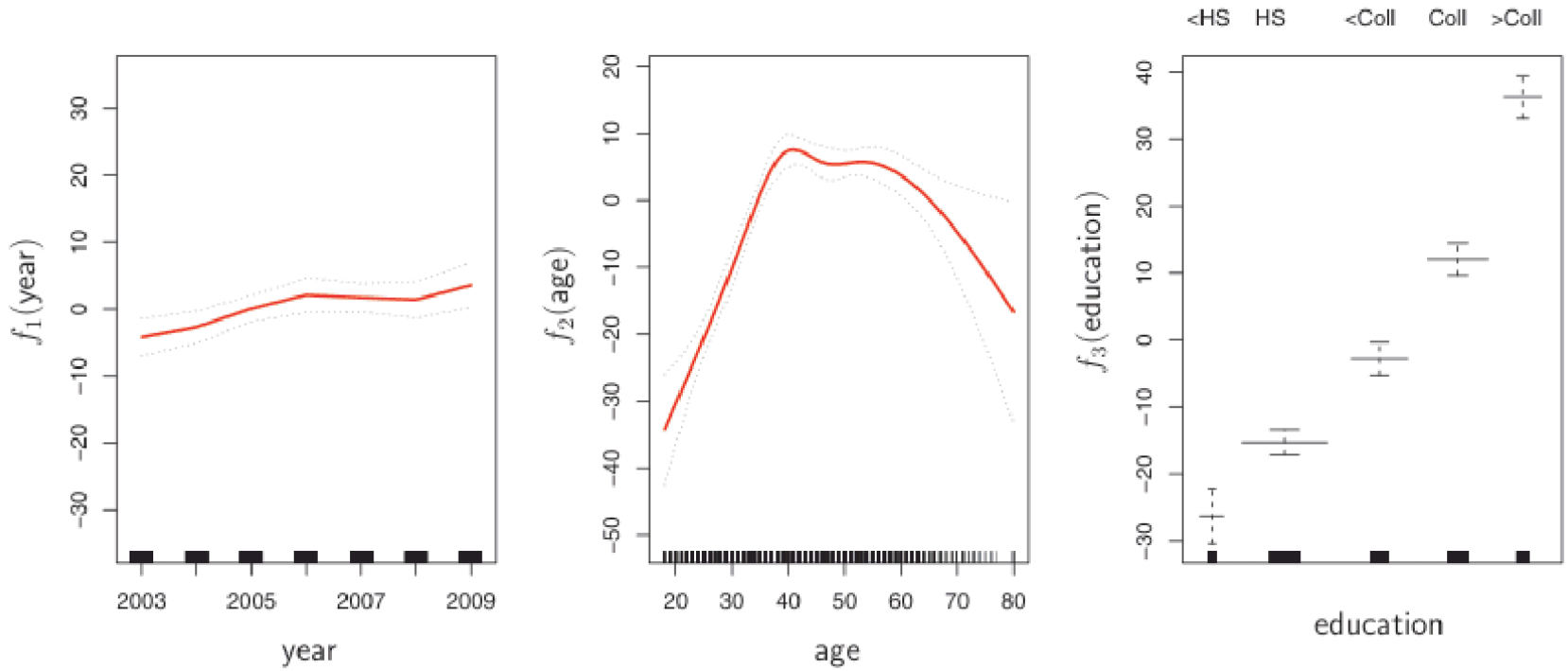

FIGURE 7.11. *For the* `Wage` *data, plots of the relationship between each feature and the response,* `wage`*, in the fitted model (7.16). Each plot displays the fitted function and pointwise standard errors. The first two functions are natural splines in* `year` *and* `age`*, with four and five degrees of freedom, respectively. The third function is a step function, fit to the qualitative variable* `education`*.*

- This model is fit using least squares

Source: ISL

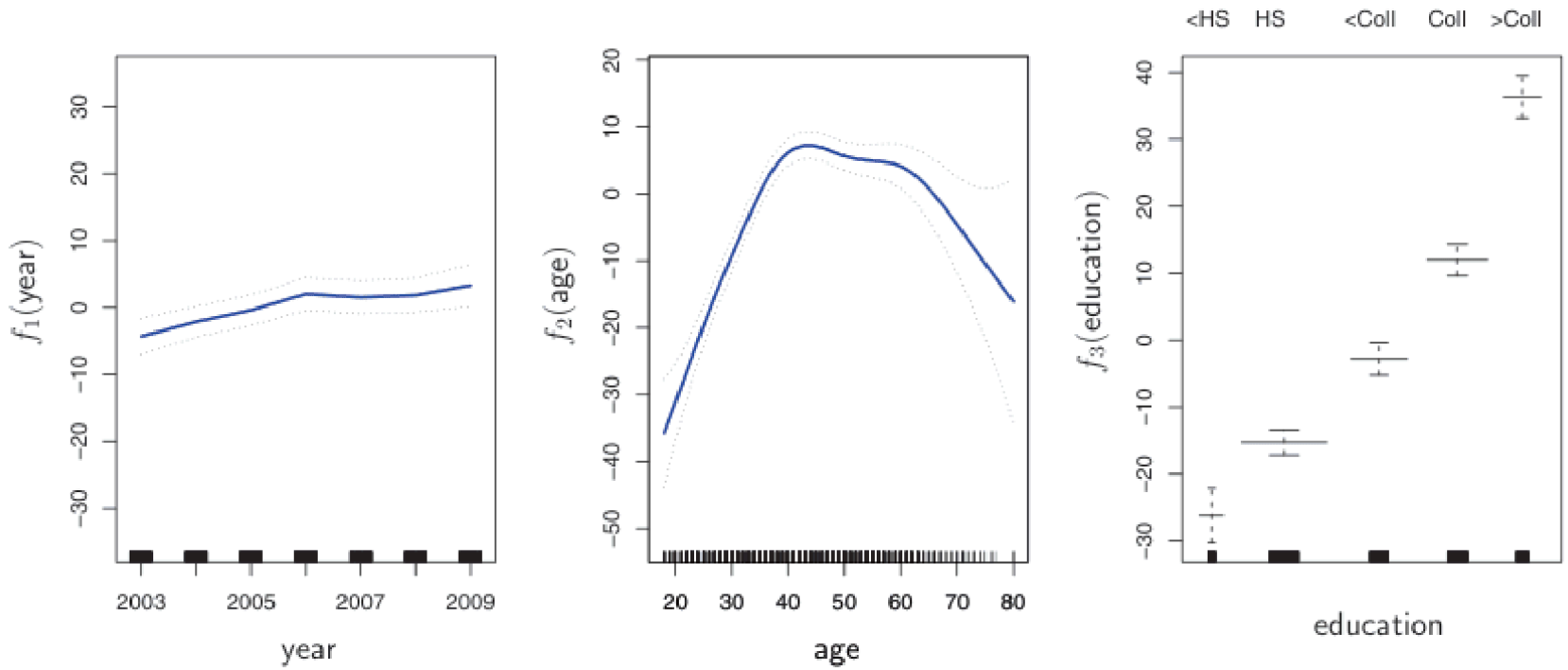

FIGURE 7.12. *Details are as in Figure 7.11, but now f_1 and f_2 are smoothing splines with four and five degrees of freedom, respectively.*

- This model is fit using the backfitting algorithm

Source: ISL

GAM for a Classification Problem

Similar approach as in the regression problem but with $f(x)$ replaced by $\text{logit}\{p(x)\}$, where $p(x) = P(Y = 1|X = x)$. In other words,

$$\text{logit}\{p(x)\} = \beta_0 + f_1(x_1) + \ldots + f_p(x_p).$$

The model is fit using maximum likelihood/backfitting.

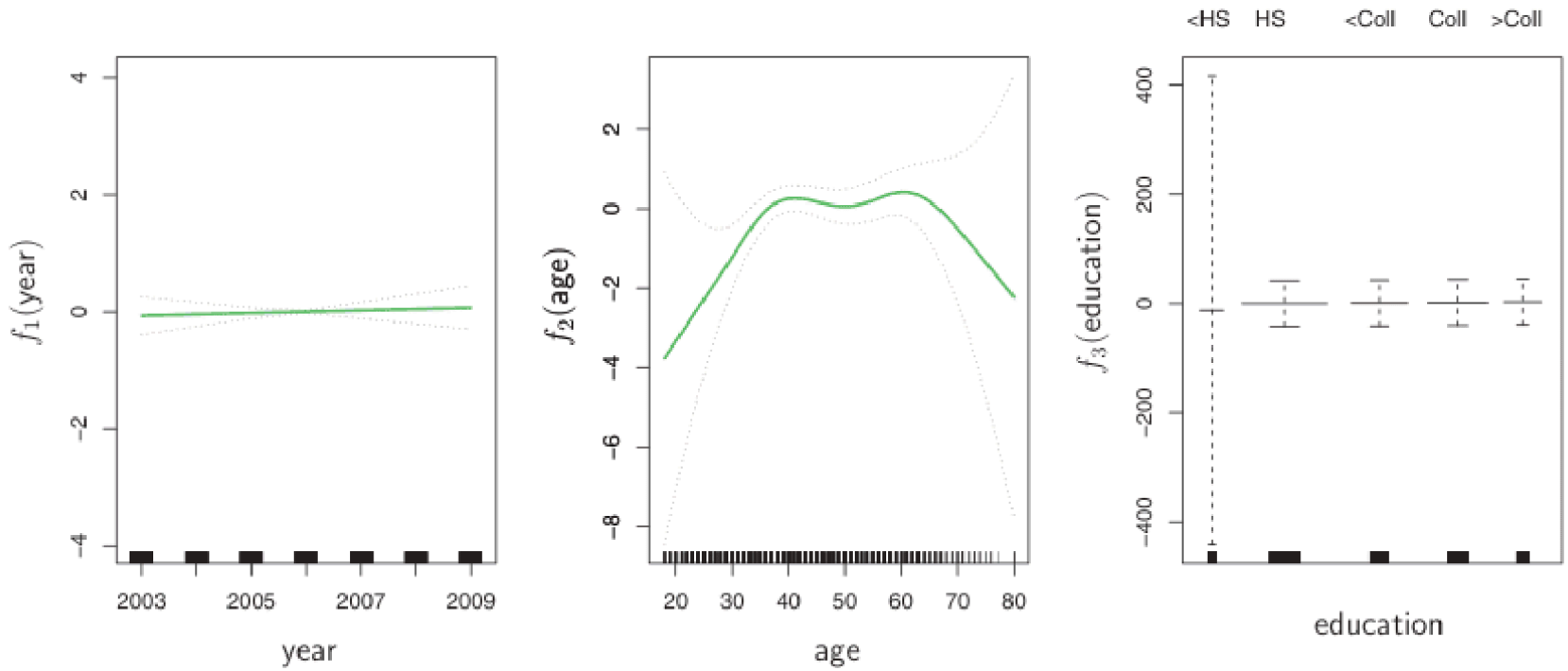

FIGURE 7.13. *For the* `Wage` *data, the logistic regression GAM given in (7.19) is fit to the binary response* `I(wage>250)`*. Each plot displays the fitted function and pointwise standard errors. The first function is linear in* `year`*, the second function a smoothing spline with five degrees of freedom in* `age`*, and the third a step function for* `education`*. There are very wide standard errors for the first level* `<HS` *of* `education`*.*

Source: ISL

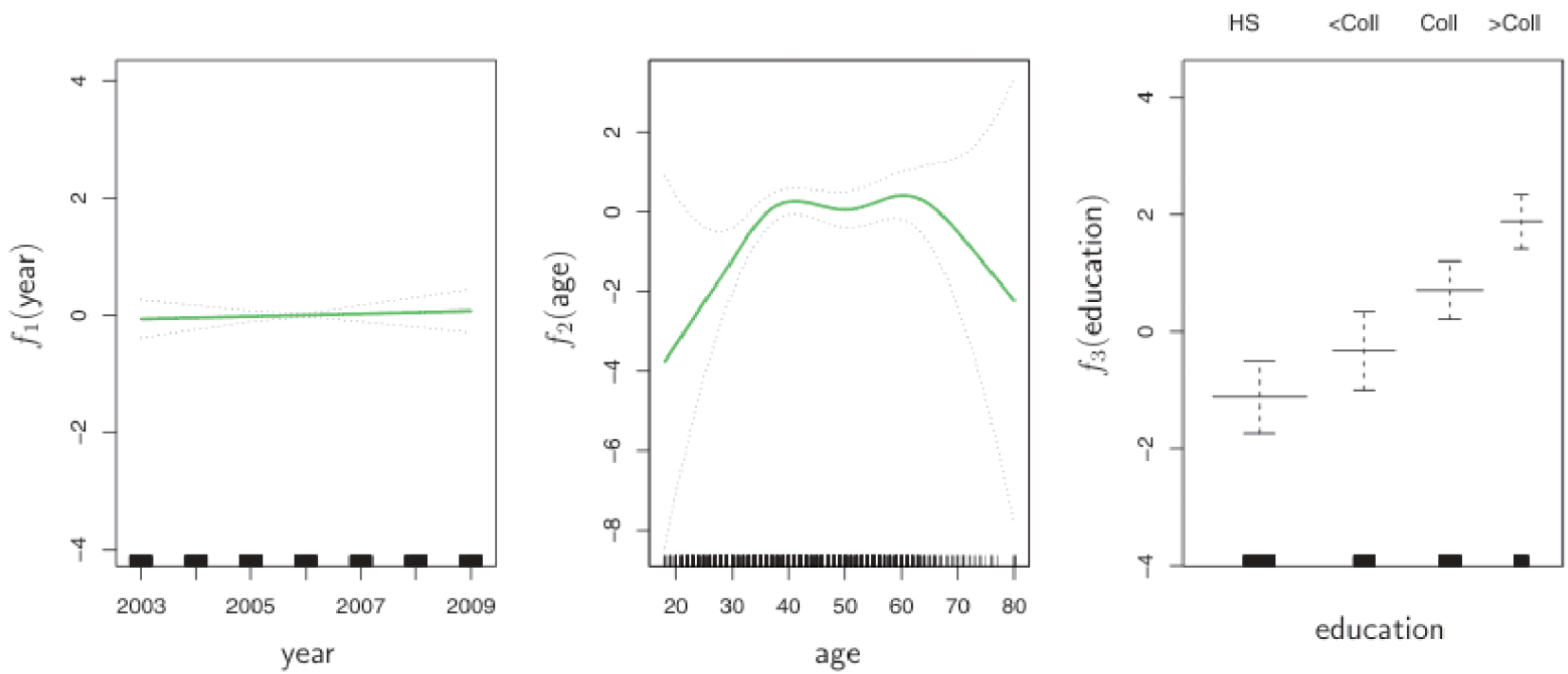

FIGURE 7.14. *The same model is fit as in Figure 7.13, this time excluding the observations for which* `education` *is* `<HS`*. Now we see that increased education tends to be associated with higher salaries.*

Source: ISL

Pros and Cons of GAMs

Pros:

- Easy to model non-linear relationship for each X_j via f_j — no need to try transformation
- Non-linear fits may provide more accurate predictions
- Easy interpretation of f_j as the effect of X_j by holding all other predictors fixed
- Model comparison can be done in the usual way

Cons: The model is restricted to be additive. Therefore, with many predictors, important interactions may be missed.

Incorporating interactions in a GAM: Add to the model:

- predictors of the form $X_j \times X_k$ (as in linear regression)
- low-dimensional interaction functions of the form $f_{jk}(x_j, x_k)$ that are fit using multi-dimensional splines or local regression

On the whole, GAMs provide a compromise between linear and fully nonparametric models (e.g., random forests and boosting).

A Final Note

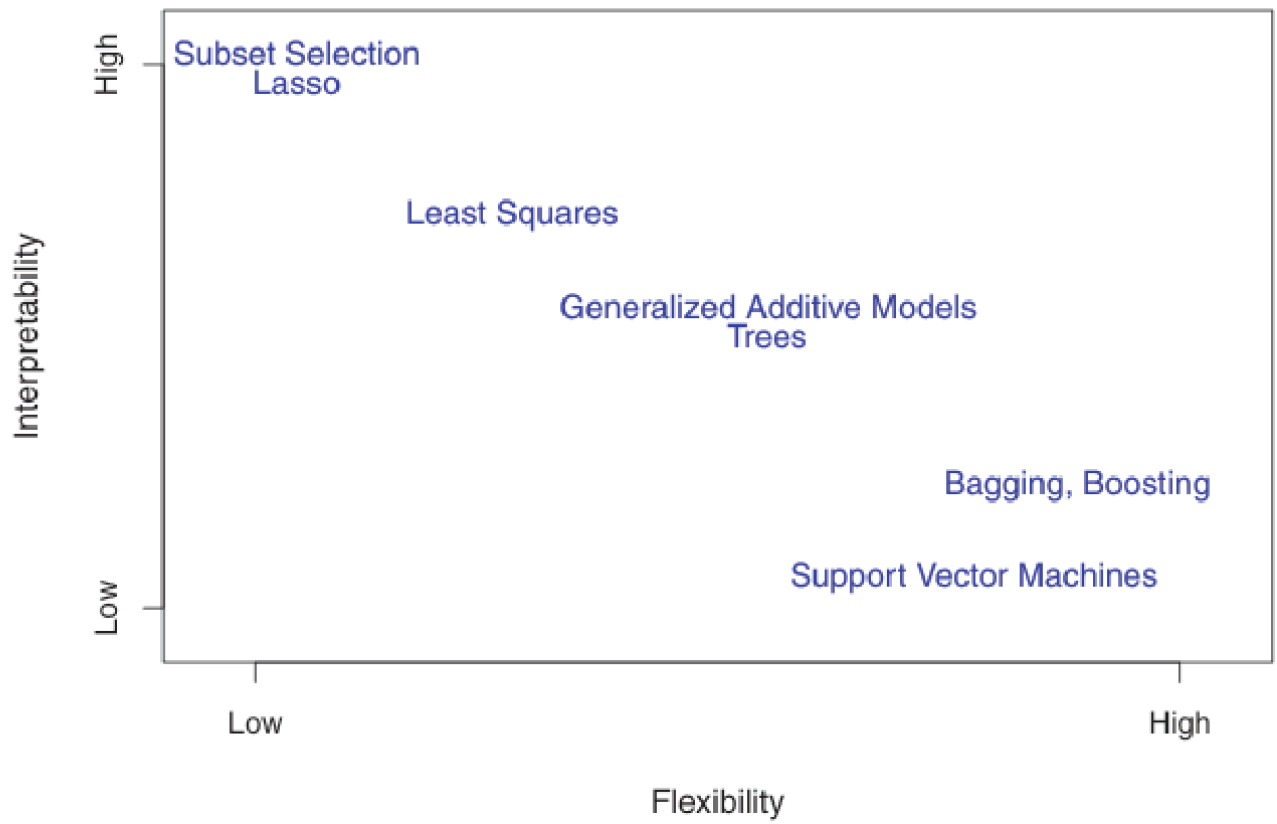

FIGURE 2.7. *A representation of the tradeoff between flexibility and interpretability, using different statistical learning methods. In general, as the flexibility of a method increases, its interpretability decreases.*

Source: ISL

Non-linear modeling & GAMs

Python applications

- Polynomial Regression & Step functions
- Splines
- Smoothing Splines & GAMs
- Local Regression