Lecture 8
Moving Beyond Linearity

CHUNG-MING KUAN

Department of Finance & CRETA

National Taiwan University

May 3, 2020

Lecture Outline

Introduction

- Linear models are easy to implement and interpret, but they may provide only a poor approximation to the true (conditional mean) function in many applications.
- A specific nonlinear model may be useful for fitting one sample but not the other samples. Moreover, fitting nonlinear models usually requires numerical optimization which may find only a local optimum. As such, nonlinear models do not necessarily outperform linear models.
- In this lecture we relax the linearity assumption and consider different classes of flexible, nonlinear models. We are particularly concerned with their implementability, interpretability, and generalizability.

- We begin with nonlinear models with only one regressor (predictor). We discuss polynomial regressions that provide global approximation to the true function. We also introduce various methods for local approximation, including step functions, regression splines, and local regressions. A complete picture of the function is obtained by piecing together these local functions.
- These nonlinear models are then extended to generalized additive models that admit multiple predictors, each transformed using some nonlinear function. The generalized additive models are useful alternatives to linear models in practice.

Polynomial Regressions

Consider fitting the data (y_i, x_i) with the polynomial with degree k:

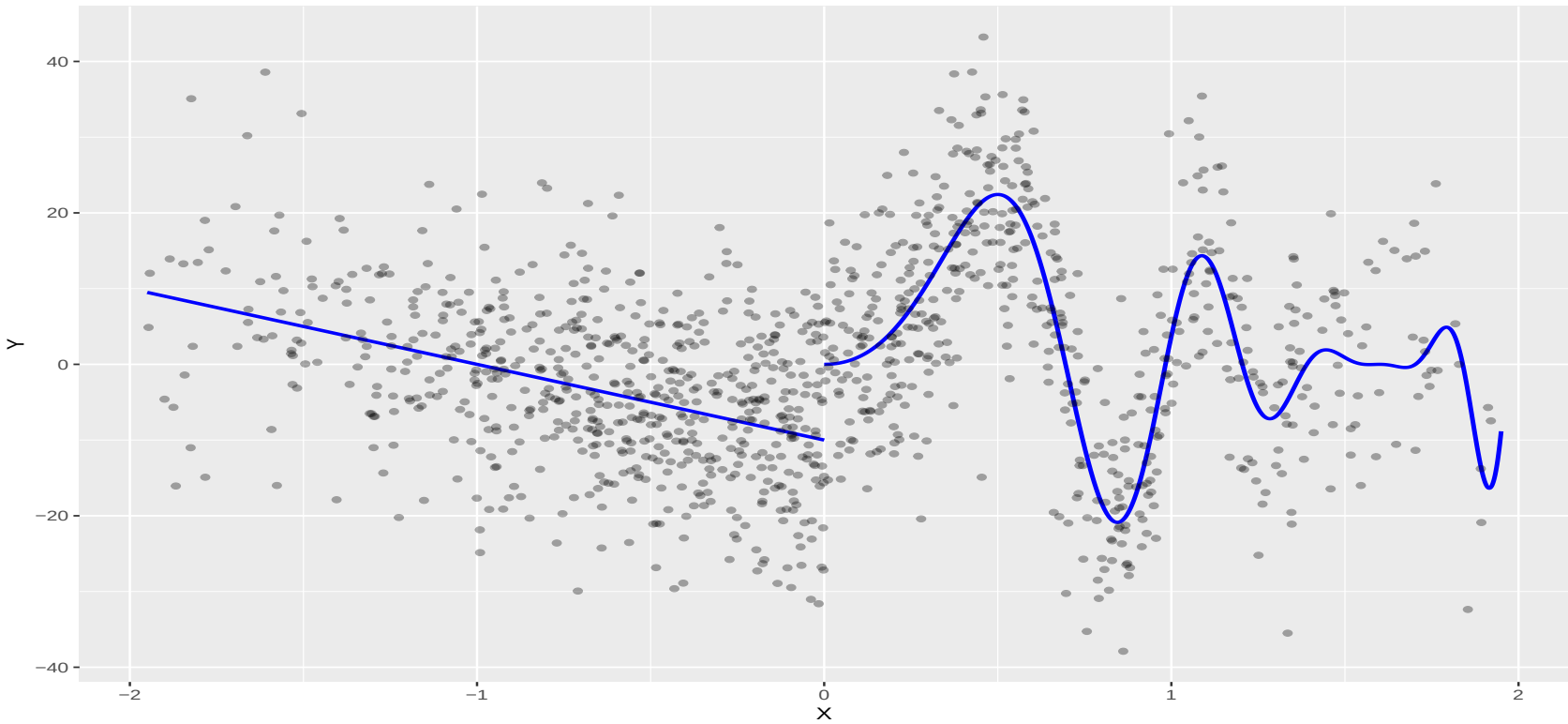
$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \cdots + \beta_k x_i^k + u_i.$$

- The class of polynomials is flexible and indexed by k. A polynomial is smooth when k is small (it is linear when $k = 1$), and it is able to represent complex nonlinearity when k is large. Finding a suitable k is always a challenging task.
- The polynomial regression function is linear in parameters and hence can be easily implemented using OLS. It is also easy to discuss the marginal effect of x on y.

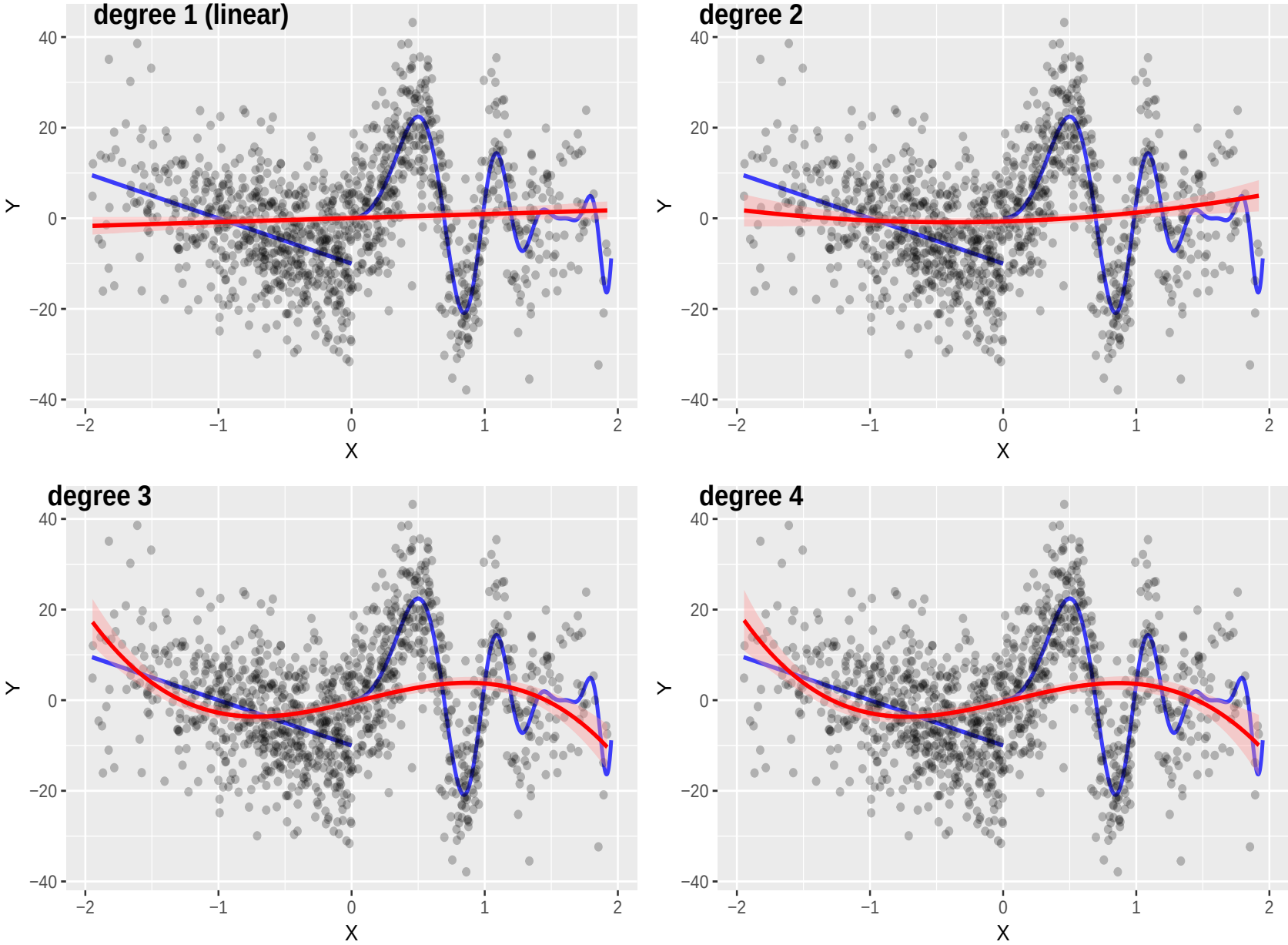
To examine the performance of polynomial regressions, we employ the DGP: $y_i = f(x_i) + u_i$, $i = 1, \ldots, 1200$, where

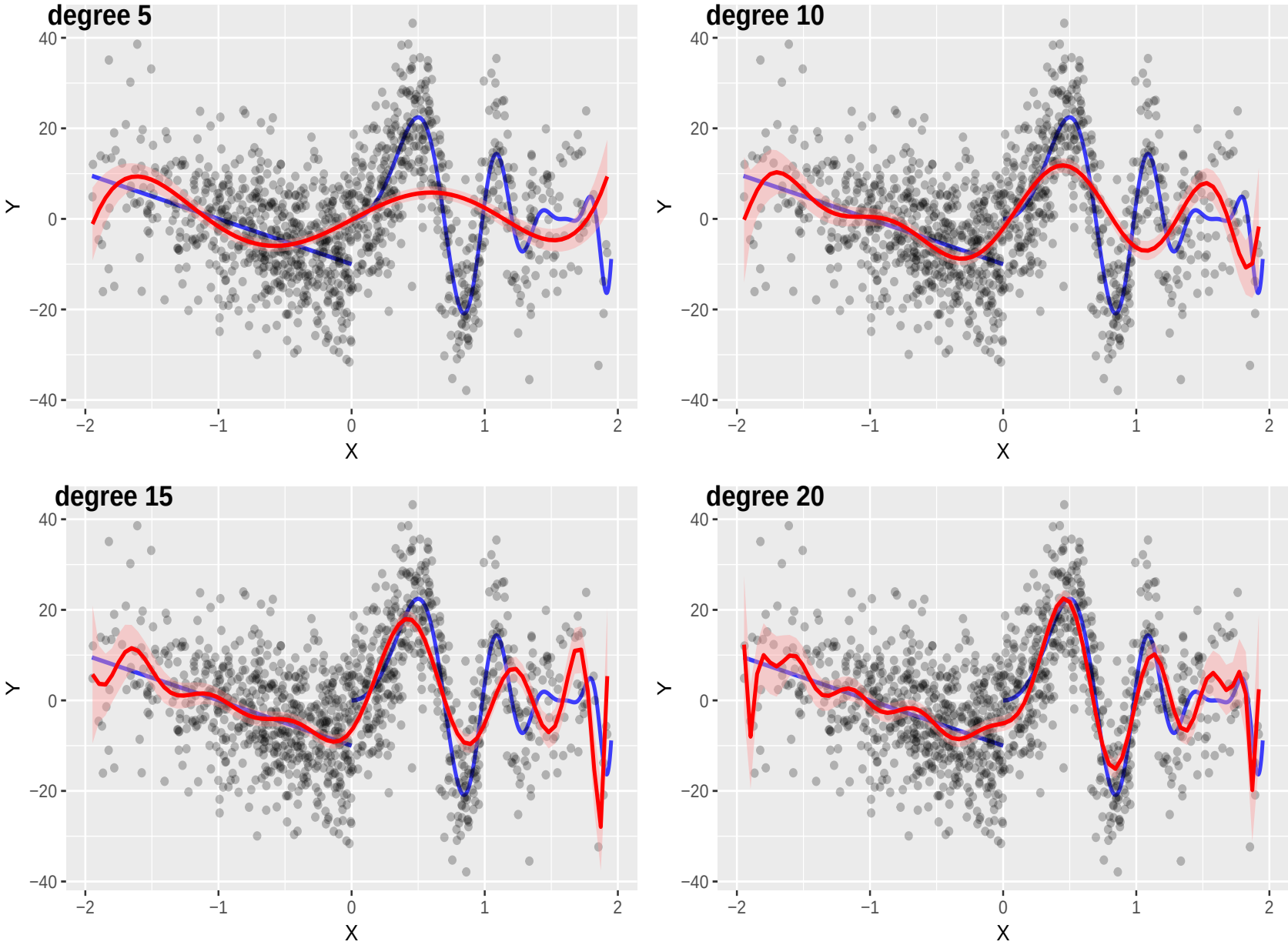
$$f(x) = \begin{cases} -10 - 5x, & -2 \leq x \leq 0, \\ (x+5)(x-1.6)^2(x+1)^3 \sin(6.5x) + 10, & 0 < x \leq 2, \end{cases}$$

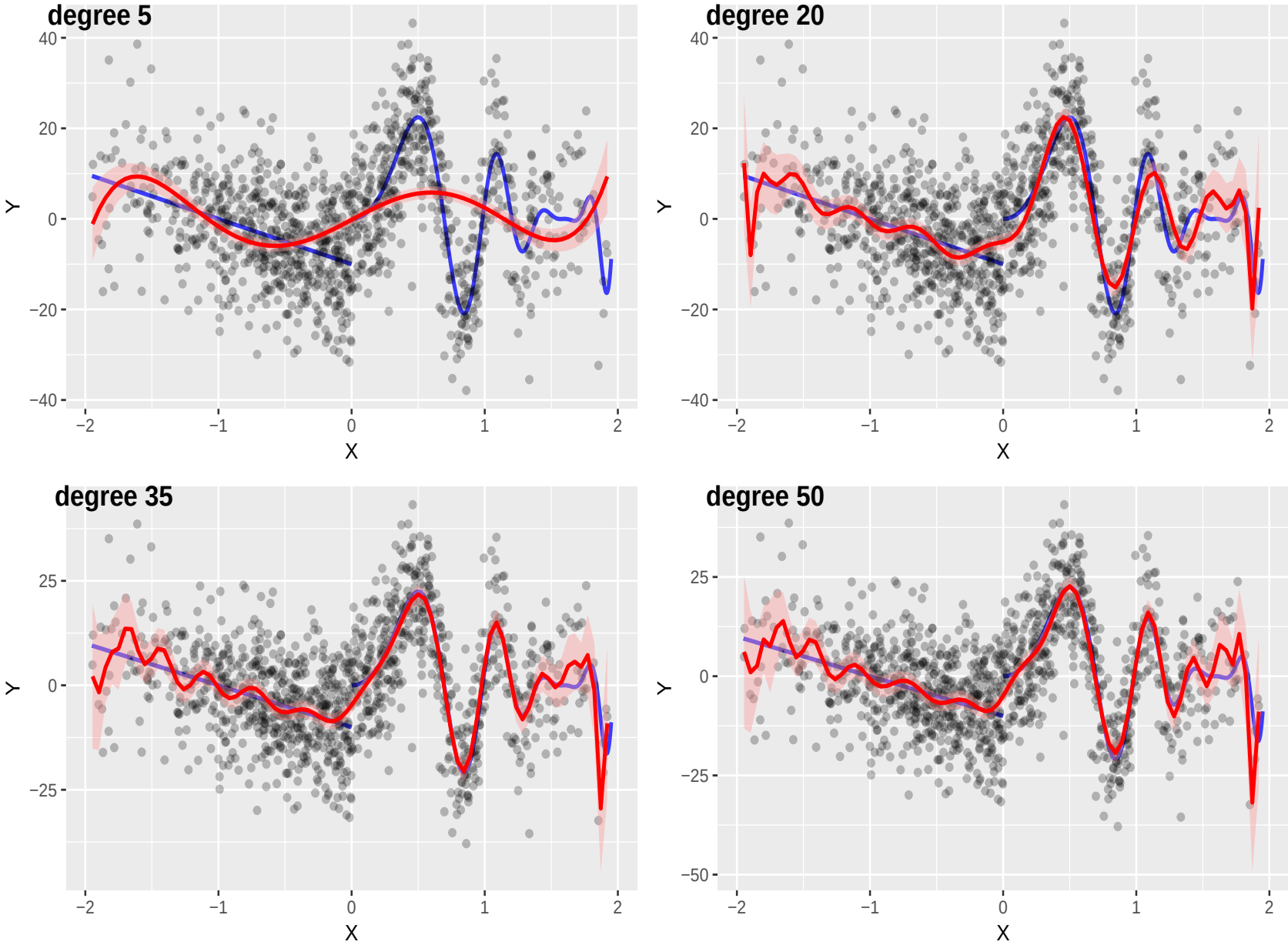
with x_i and u_i i.i.d. $\mathcal{N}(0, 0.81)$ and $\mathcal{N}(0, 100)$, respectively.



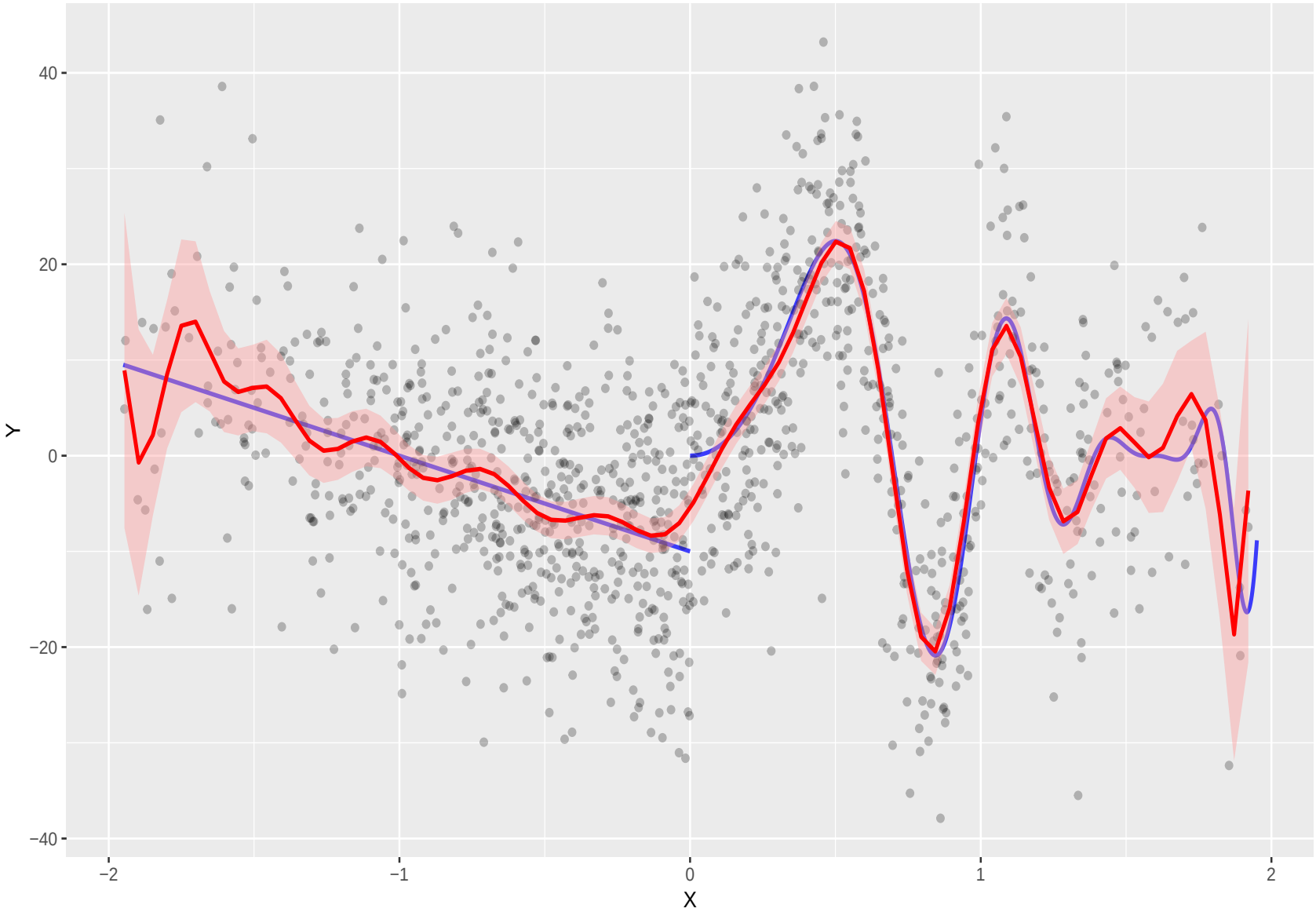
- We estimate the polynomial with degrees 1, 2, 3, 4, 5, 10, 15, 20, 35, and 50 using OLS. In the figures below, each red solid line is an estimated polynomial, with the red shaded area its 95% confidence interval.
- It can be seen that a polynomial with a low degree provides a good approximation to the linear part of f but under fits the other part of data (too simple for nonlinearity of f), while a polynomial with a high degree approximates the nonlinear part of f better but tends to over fits the data on the left (too complex for the linear function). These figures illustrate that a global approximation, such as polynomial regression, has limited applicability.
- In practice, we may determine the degree k by cross validation. This does not solve the problem, however; see the figure of polynomial with degree 31.


degree 1 (linear)
degree 2
degree 3
degree 4
Y
X
40
20
0
−20
−40
−2
−1
0
1
2


degree 5
degree 10
degree 15
degree 20
Y
X
40
20
0
−20
−40
−2
−1
0
1
2


degree 5
degree 20
degree 35
degree 50
Y
X

The degree of polynomial, 31, is chosen by 10-fold CV.



Step Functions

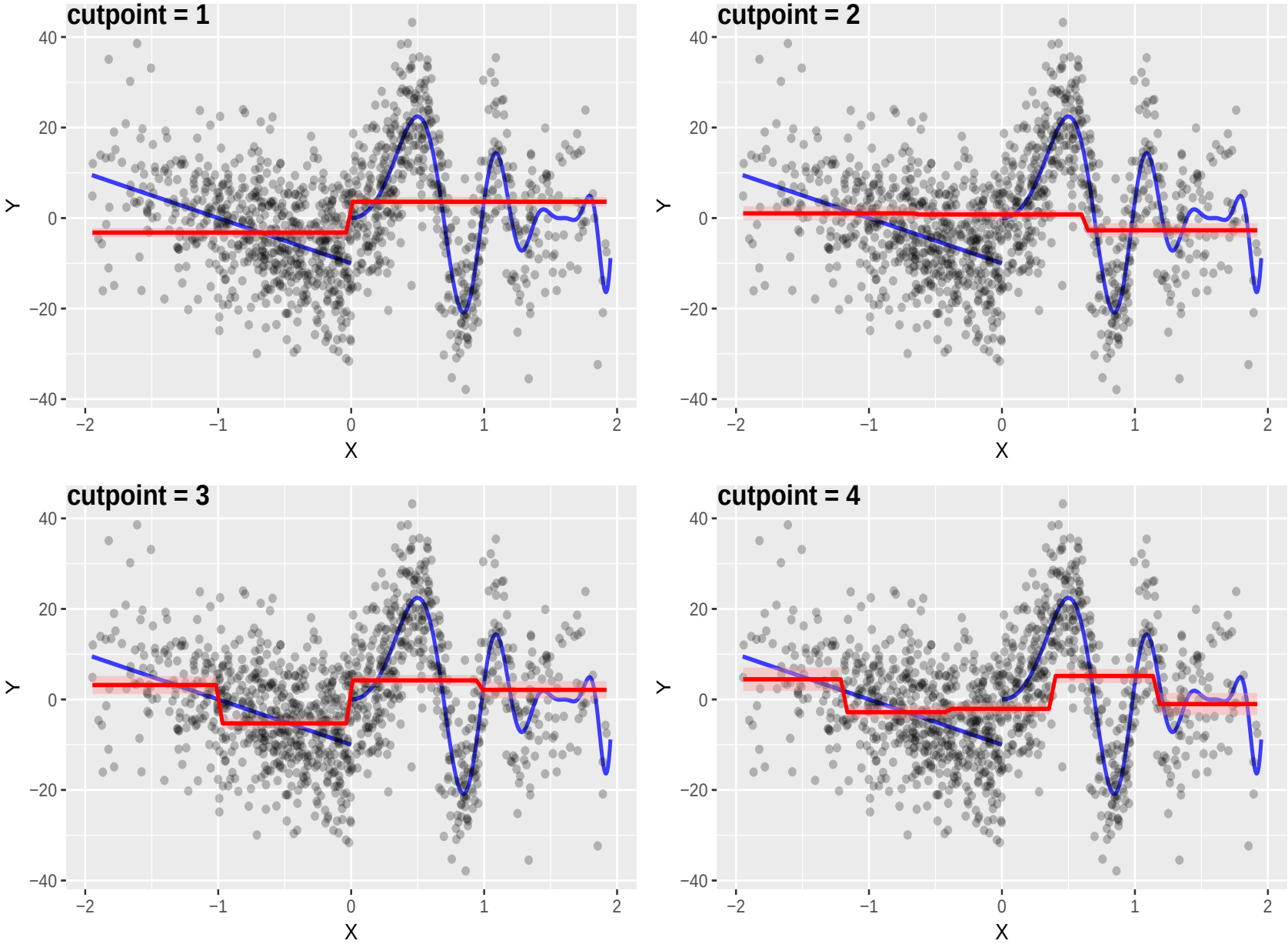
To avoid the global structure imposed by polynomial regressions, we may use step functions to locally approximate the true function. To this end, we choose the cutpoints $c_1, c_2, \ldots, c_k$ that partition the range of x into $k+1$ disjoint intervals, and fit the following piecewise constant function:

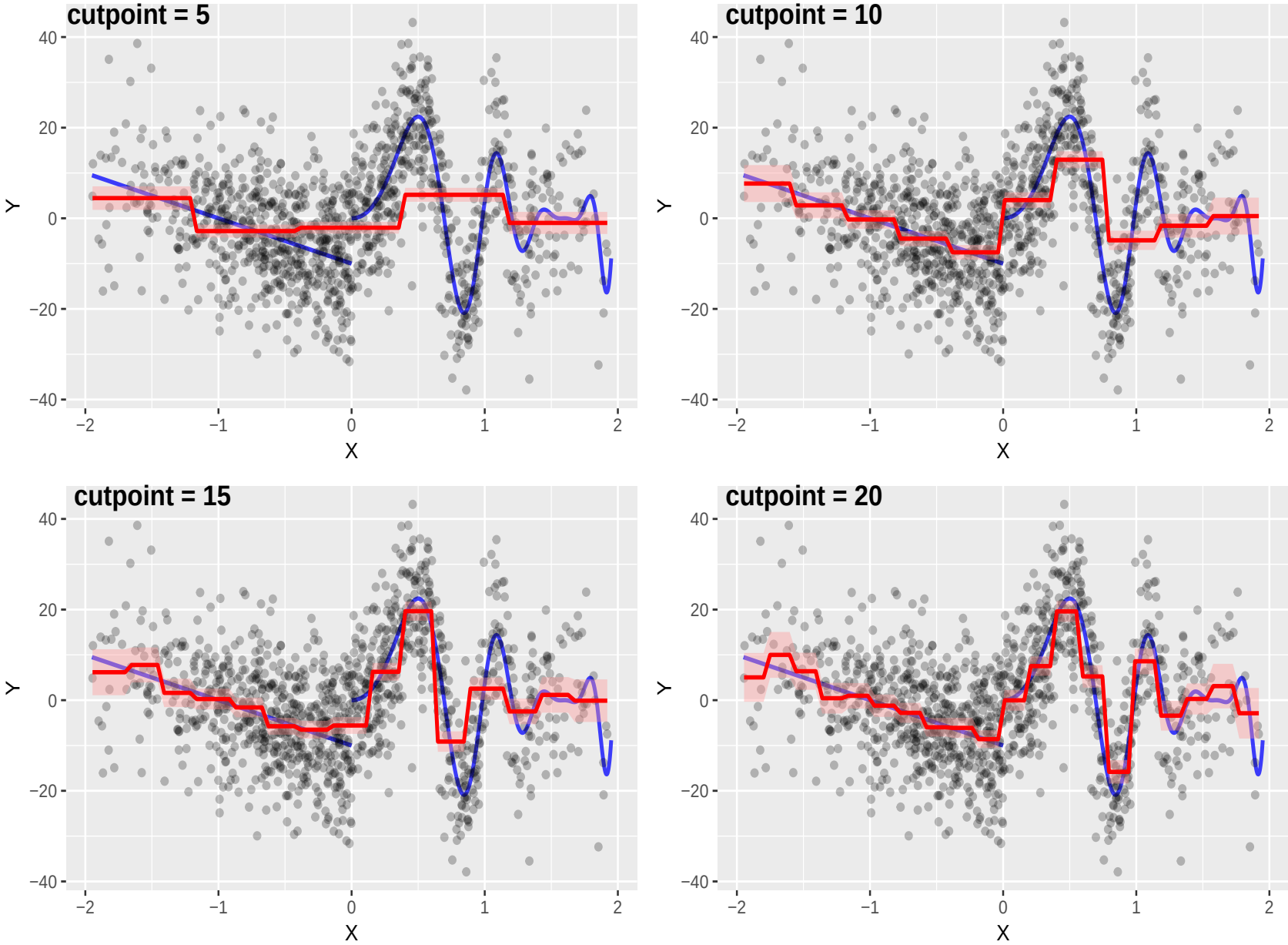
$$y_i = \beta_1 \mathbf{1}(x_i < c_1) + \beta_2 \mathbf{1}(c_1 \leq x_i < c_2) + \cdots + \beta_k \mathbf{1}(c_{k-1} \leq x_i < c_k) + \beta_{k+1} \mathbf{1}(c_k \leq x_i) + u_i,$$

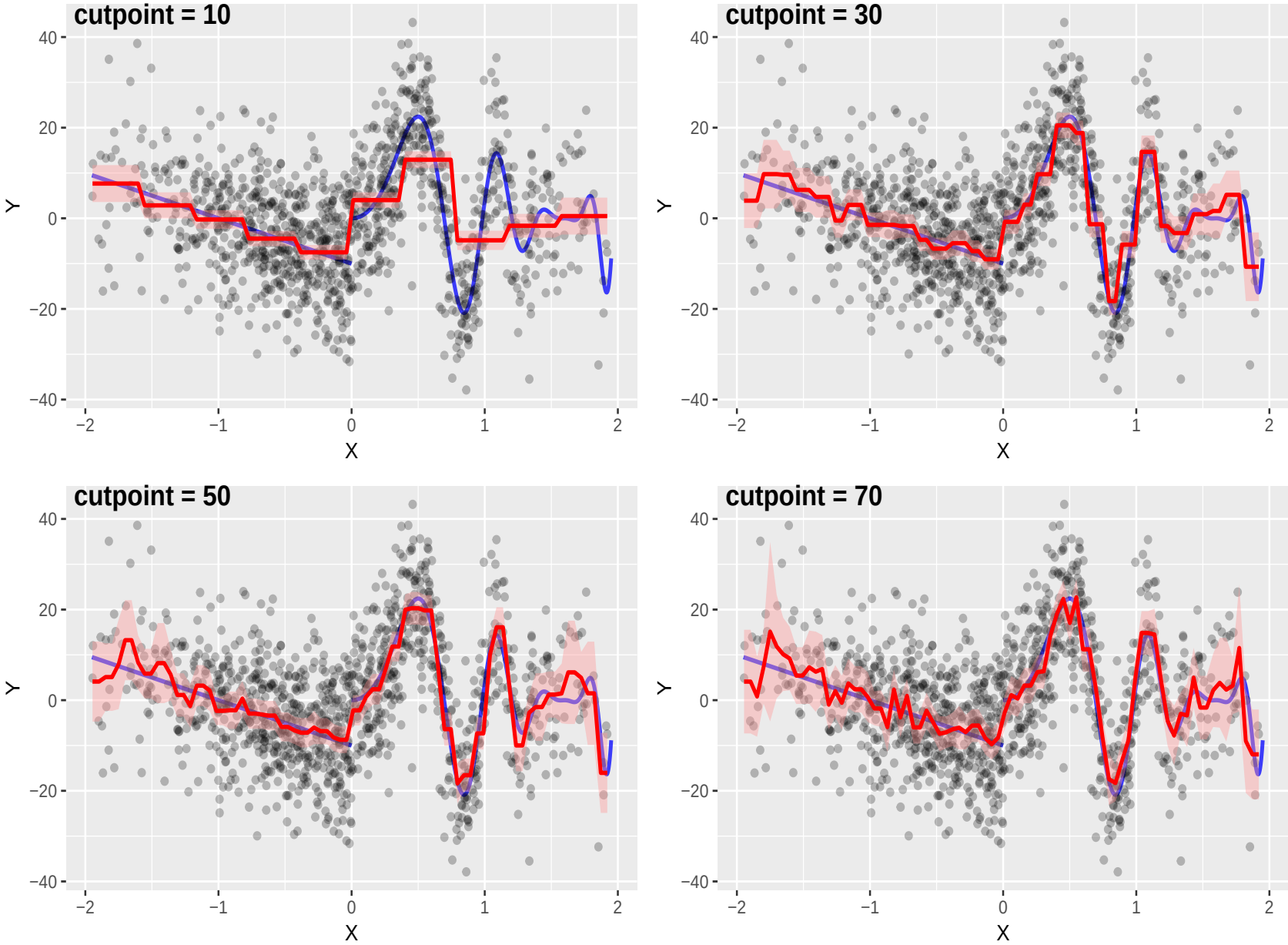
where $\mathbf{1}(\cdot)$ is the indicator function (a dummy variable).

- This function is linear in parameters and can be estimated using OLS.
- Each estimated coefficient is simply the average of y_i on the corresponding sub-interval.

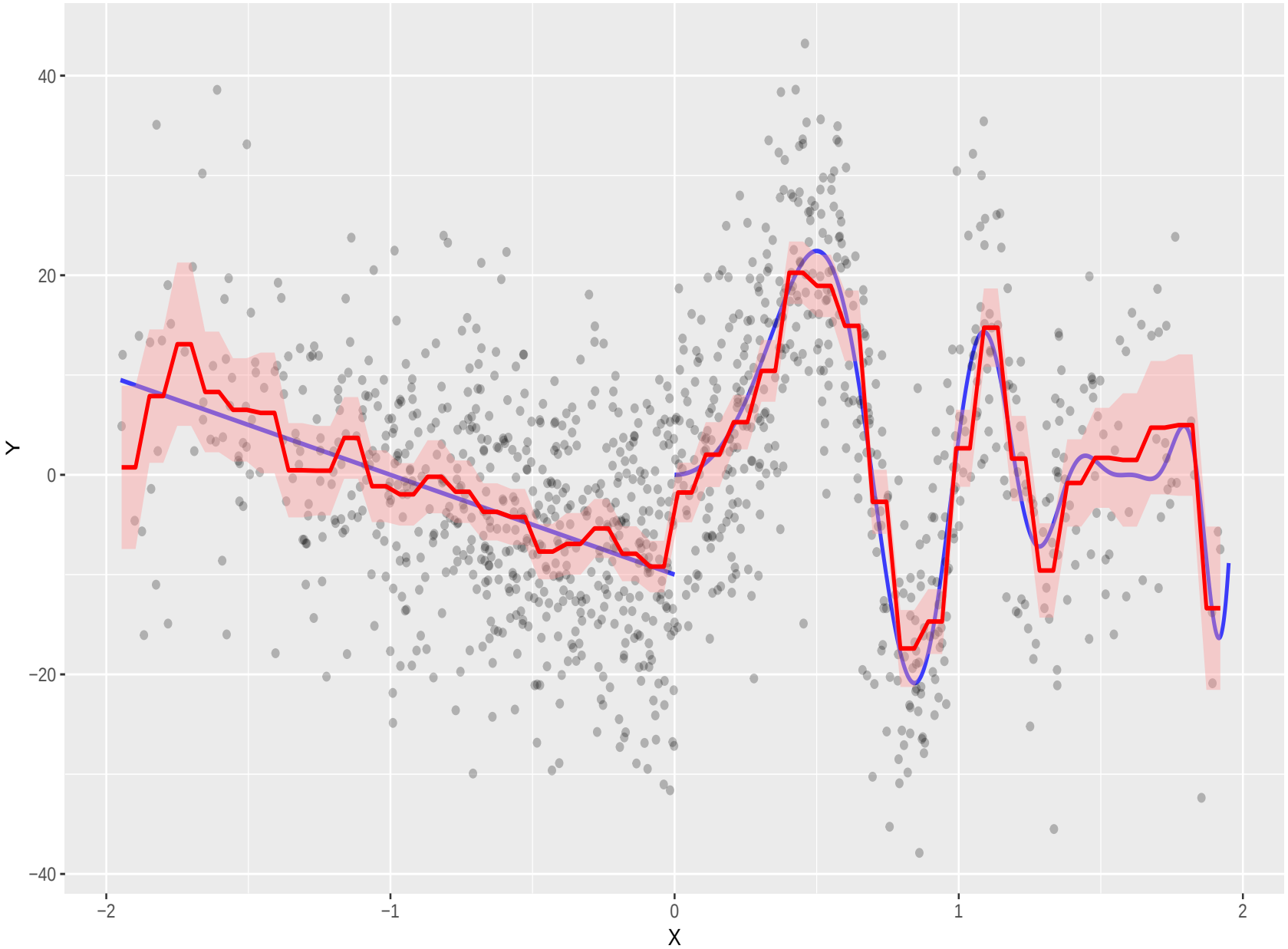
- The figures below illustrate the estimated step functions with different numbers of cutpoints, with the cutpoints evenly distributed on x.
- A large number of cutpoints may be helpful for approximating the nonlinear part while over fitting the linear part, but a small number of cutpoints may under fit the nonlinear part. In practice, one may choose k by cross validation.
- More generally, instead of using a piecewise constant function, one may consider piecewise linear (quadratic, cubic) function for local approximation.


cutpoint = 1
cutpoint = 2
cutpoint = 3
cutpoint = 4
Y
X
40
20
0
−20
−40
−2
−1
0
1
2


cutpoint = 5
cutpoint = 10
cutpoint = 15
cutpoint = 20
Y
X
40
20
0
−20
−40
−2
−1
0
1
2

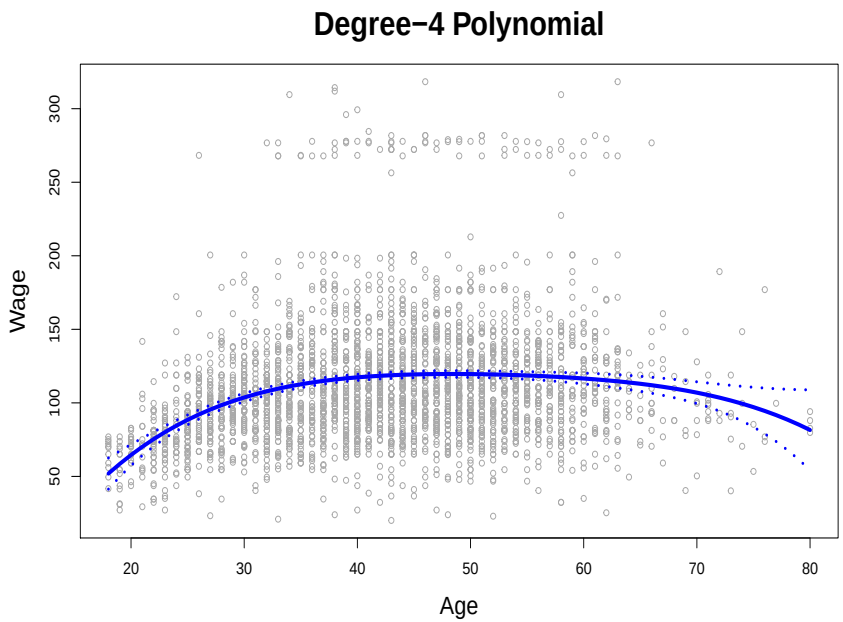

cutpoint = 10
cutpoint = 30
cutpoint = 50
cutpoint = 70
Y
X
40
20
0
−20
−40
−2
−1
0
1
2

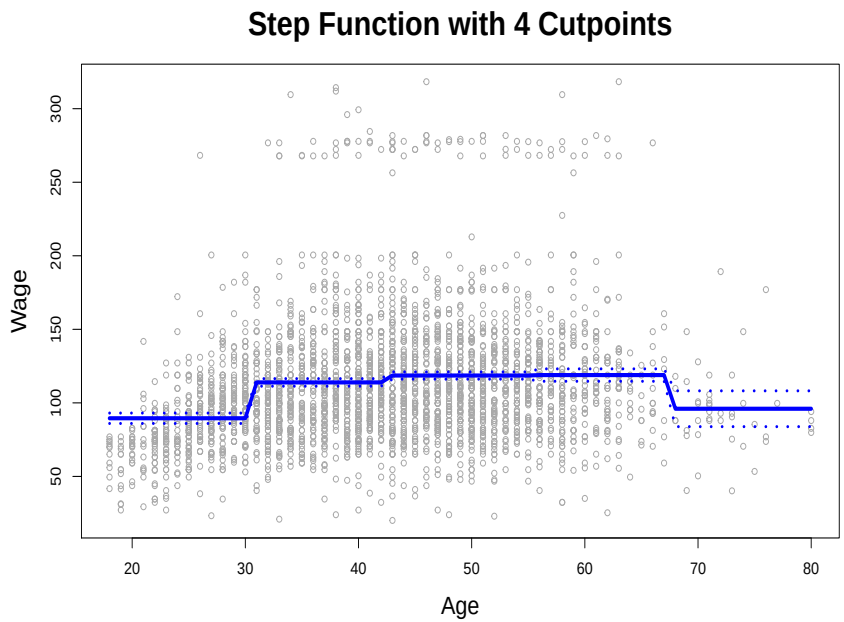
The number of cutpoints, 40, is chosen by 10-fold CV.



Example: Wage in the Mid-Atlantic region

The `Wage` dataset in `R` contains income and demographic information of $3,000$ male workers in the Mid-Atlantic region. For the variable of interest `wage` and the predictor `age`, the estimated polynomial with degree 4 and the estimated step functions with 4 cutpoints are plotted below; the dotted curves are the estimated 95% confidence interval.





Piecewise Polynomials

The polynomial regression and piecewise constant regression are special cases of the basis function approach. Let $b_1, b_2, \ldots, b_k$ be some basis functions of x. We may consider a model with $b_j(x)$ as predictors:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_k b_k(x_i) + u_i.$$

This model is linear in parameters and can be estimated using OLS.

- For polynomial regression, $b_j(x) = x^j$.
- For step function, $b_j(x) = \mathbf{1}(c_{j-1} \leq x < c_j)$.
- More generally, one may consider the piecewise polynomial which admit a polynomial, rather than a constant, on each sub-interval of x.

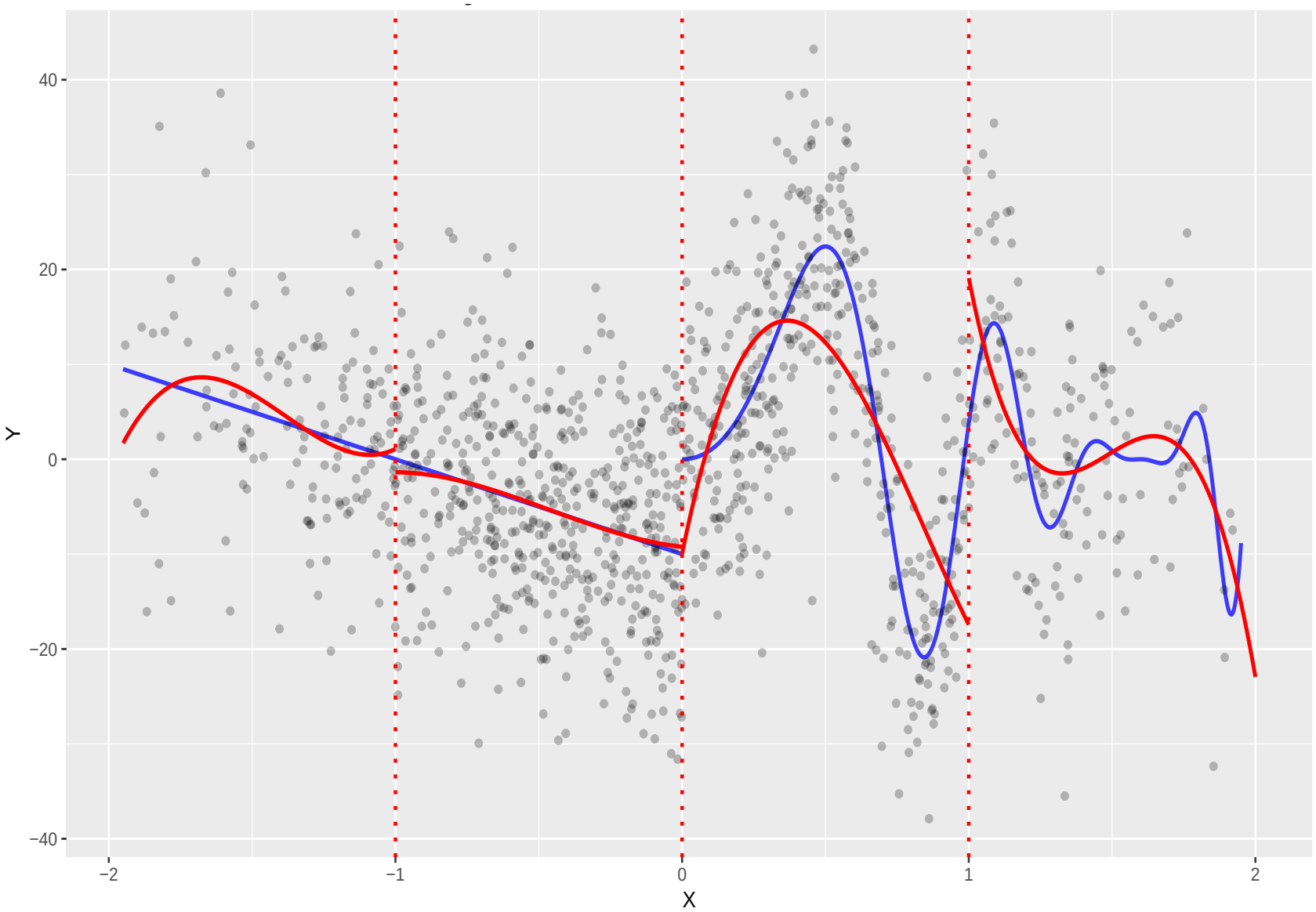
The function below is a piecewise cubic polynomial with 3 uniformly distributed cutpoints:

$$f(x) = \begin{cases} \beta_{00} + \beta_{01} x_i + \beta_{02} x_i^2 + \beta_{03} x_i^3, & -2 < x \leq -1, \\ \beta_{10} + \beta_{11} x_i + \beta_{12} x_i^2 + \beta_{13} x_i^3, & -1 < x \leq 0, \\ \beta_{20} + \beta_{21} x_i + \beta_{22} x_i^2 + \beta_{23} x_i^3, & 0 < x \leq 1, \\ \beta_{30} + \beta_{31} x_i + \beta_{32} x_i^2 + \beta_{33} x_i^3, & 1 < x \leq 2, \end{cases}$$

which involves 4 cubic polynomials on 4 sub-intervals of x. Letting $b_{ij}(x) = x^j \mathbf{1}(c_i < x \leq c_{i+1})$, $i, j = 0, 1, 2, 3$, with $c_0 = -2$, $c_1 = -1$, $c_2 = 0$, $c_3 = 1$ and $c_4 = 2$, we can write

$$y_i = \sum_{i=0}^{3} \sum_{j=0}^{3} \beta_{ij} b_{ij}(x) + u_i.$$

A piecewise cubic polynomial with 3 uniformly distributed knots



Continuous Piecewise Polynomials

A drawback of the piecewise polynomial is that it is discontinuous at each cutpoint. To avoid discontinuity, we need a constraint that makes the piecewise polynomial continuous. To this end, consider the following function with a given ξ:

$$h(x,\xi) = (x-\xi)_+ = \begin{cases} 0, & x \le \xi, \\ (x-\xi), & x > \xi. \end{cases}$$

Then, $f(x) = \beta_0 + \beta_1 x + \beta_2 h(x,\xi)$ leads to the piecewise linear function:

$$f(x) = \begin{cases} \beta_0 + \beta_1 x, & x \le \xi, \\ (\beta_0 - \beta_2\xi) + (\beta_1 + \beta_2)x, & x > \xi, \end{cases}$$

where two line segments are connected at the cutpoint (knot) $x = \xi$.

Similarly, for

$$h^2(x,\xi) = (x-\xi)_+^2 = \begin{cases} 0, & x \le \xi, \\ (x-\xi)^2, & x > \xi, \end{cases}$$

the quadratic polynomial with $h^2(x,\xi)$ is

$$\begin{aligned} f(x) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 h^2(x,\xi) \\ &= \begin{cases} \beta_0 + \beta_1 x + \beta_2 x^2, & x \le \xi. \\ (\beta_0 + \beta_3 \xi^2) + (\beta_1 - 2\beta_3 \xi) x + (\beta_2 + \beta_3) x^2, & x > \xi. \end{cases} \end{aligned}$$

These two quadratic curves are connected at the knot: $x = \xi$, so that this $f(x)$ is a continuous, piecewise quadratic function. Moreover, the first derivative, $f'(x)$, is also continuous, because the derivatives of these two curves, $\beta_1 + 2\beta_2 x$ and $(\beta_1 - 2\beta_3\xi) + 2(\beta_2 + \beta_3)x$, are connected at $x = \xi$.

Cubic Splines

The functions $h(x,\xi)$, $h^2(x,\xi)$, and $h^3(x,\xi)$ below are referred to as the truncated power basis function at ξ:

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

which connect two cubic curves at $x = \xi$. The cubic spline with k knots $\xi_1, \ldots, \xi_k$ is the cubic polynomial with k truncated power basis functions:

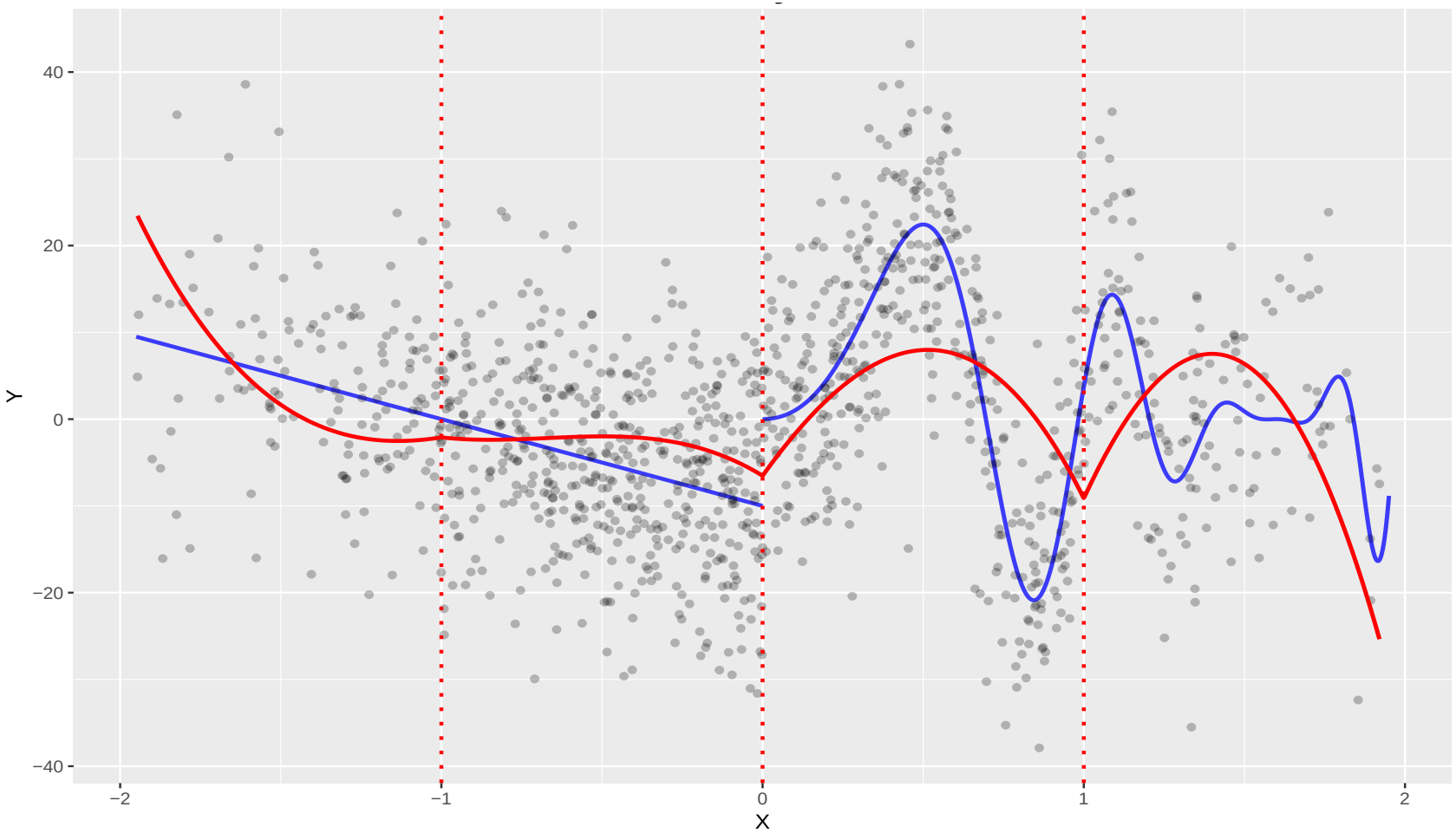
$$f(x) = \beta_0 + \beta_1 x + \beta_2 x_2 + \beta_3 x^3 + \beta_4 h^3(x,\xi_1) + \cdots + \beta_{3+k} h^3(x,\xi_k).$$

This is a continuous function, with continuous first and second derivatives. More generally, a degree-d spline has continuous derivatives up to order $d-1$.

Below is an estimated piecewise cubic polynomial with $h(x;\xi)$ at 3 knots:

$$\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 h(x_i, -1) + \beta_5 h(x_i, 0) + \beta_6 h(x_i, 1).$$

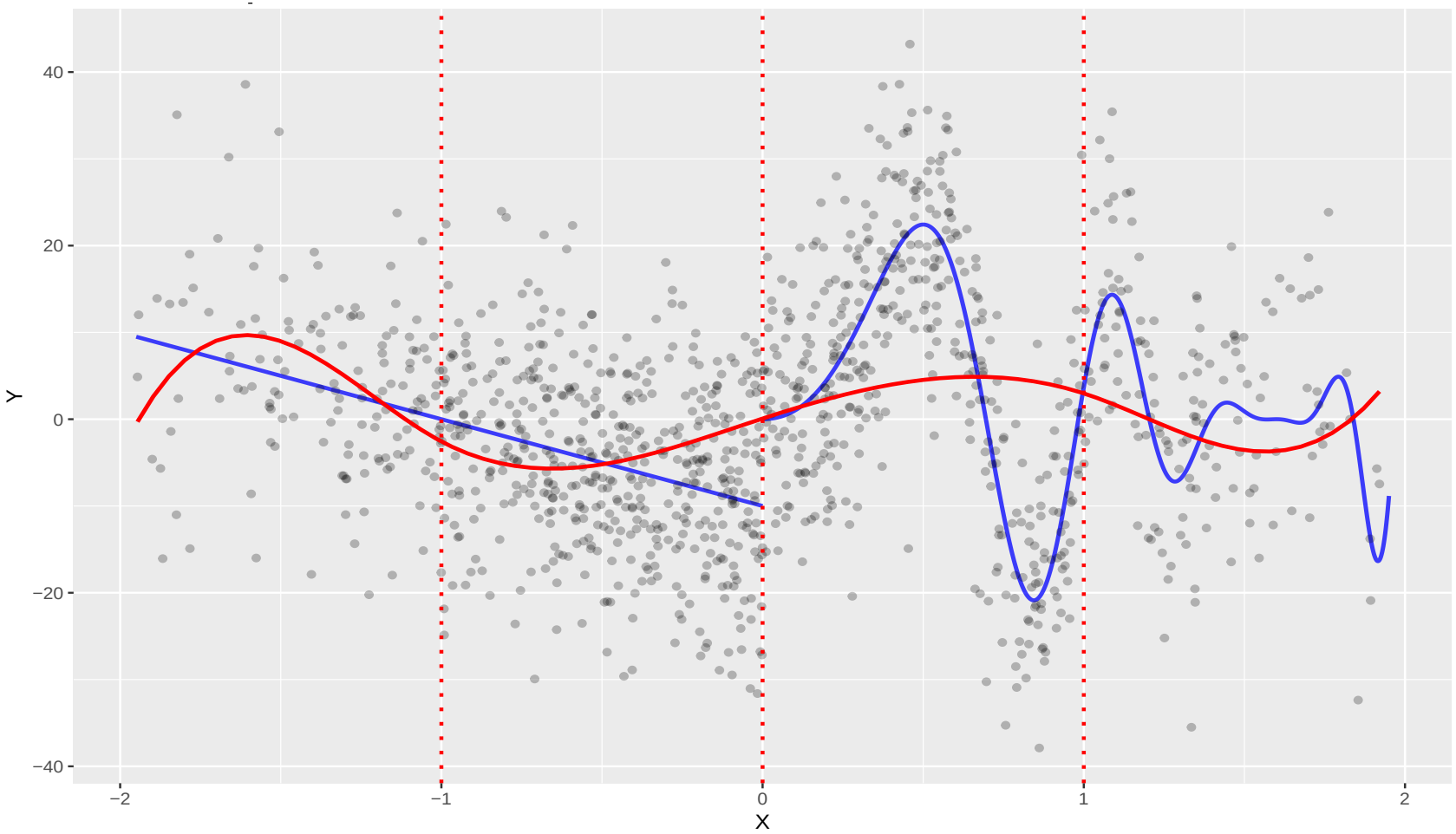
This curve is continuous but with sharp transition at each knot.



Below is an estimated cubic spline with $h^3(x, \xi)$ at 3 knots:

$$\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 h^3(x_i, -1) + \beta_5 h^3(x_i, 0) + \beta_6 h^3(x_i, 1).$$

Note that this curve is very smooth at each knot.

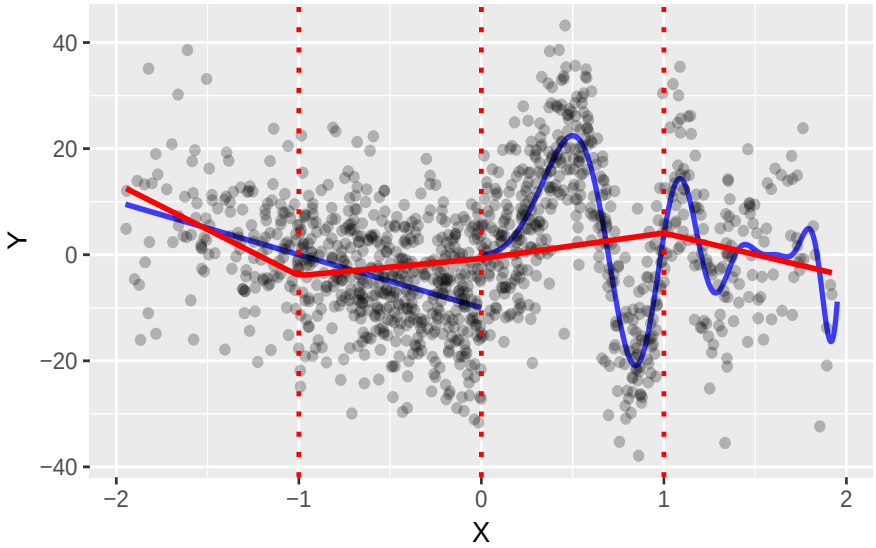


Why Cubic Splines?

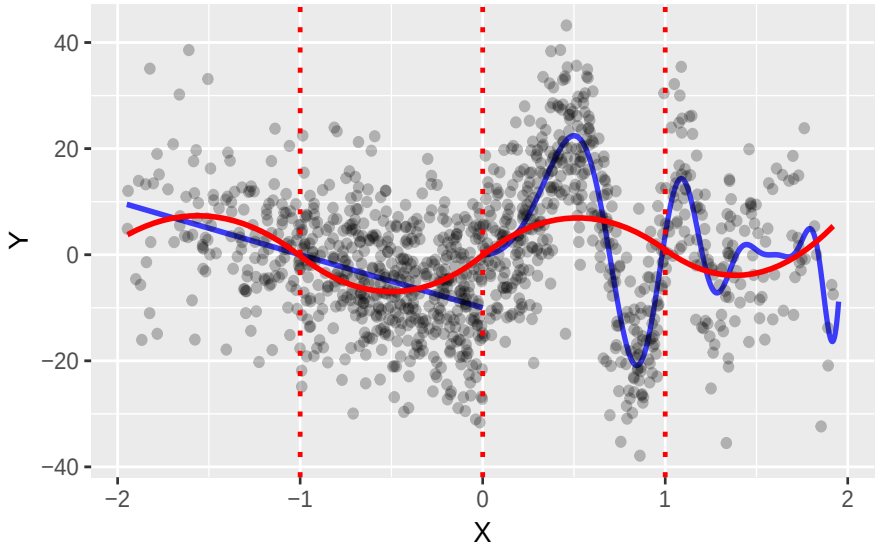
Cubic splines are a common choice in applications. Recall that a cubic spline is continuous and has continuous first and second derivatives. Thus, a cubic spline is very smooth at each knot, and 2 segments have the same curvature (characterized by their second derivatives) at the knot they meet. This is in contrast with quadratic splines whose curvature may change at each knot. Beyond curvature, it is hard to visualize the difference between high-order derivatives of 2 segments.

We plot the linear, quadratic, cubic and degree-4 splines with 3, 10, and 20 knots in the following pages. As we can see, it is not easy to tell the differences between the smoothness of quadratic, cubic and degree-4 splines.

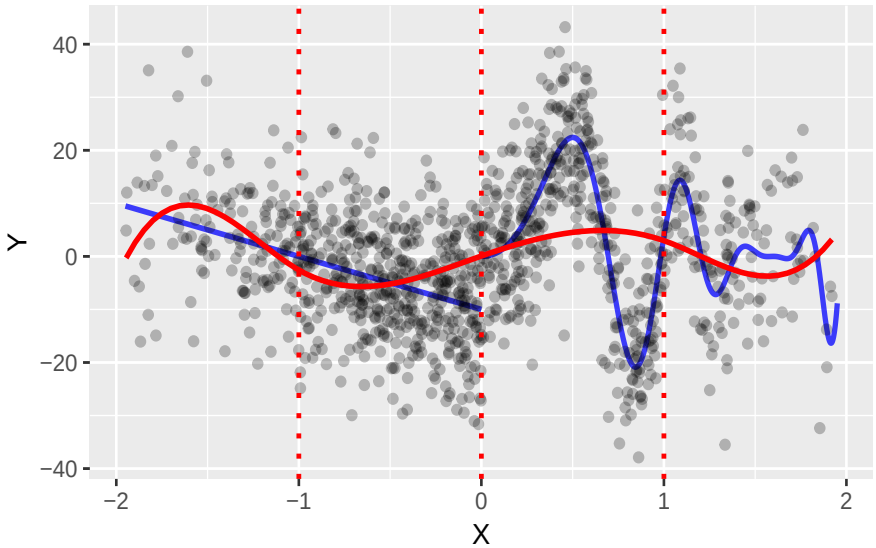
Degree 1 (linear) spline with 3 knots



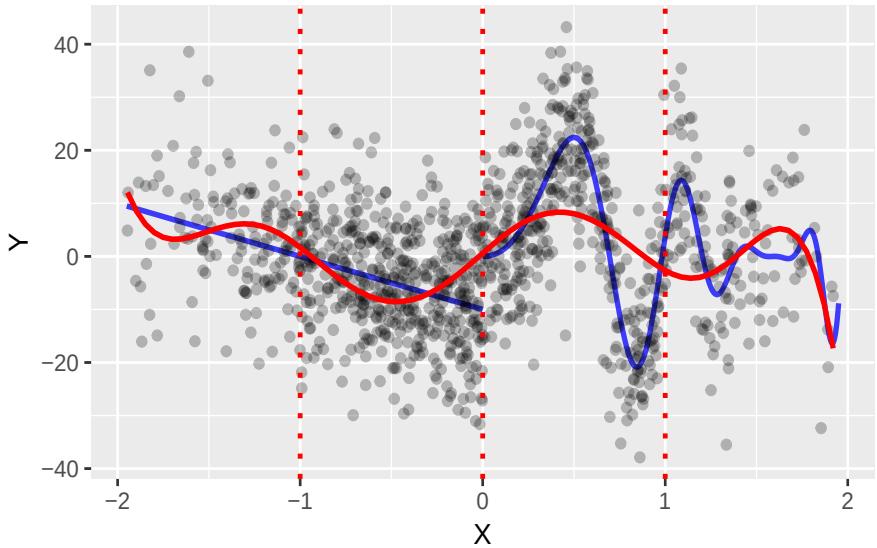
Degree 2 (quadratic) spline with 3 knots



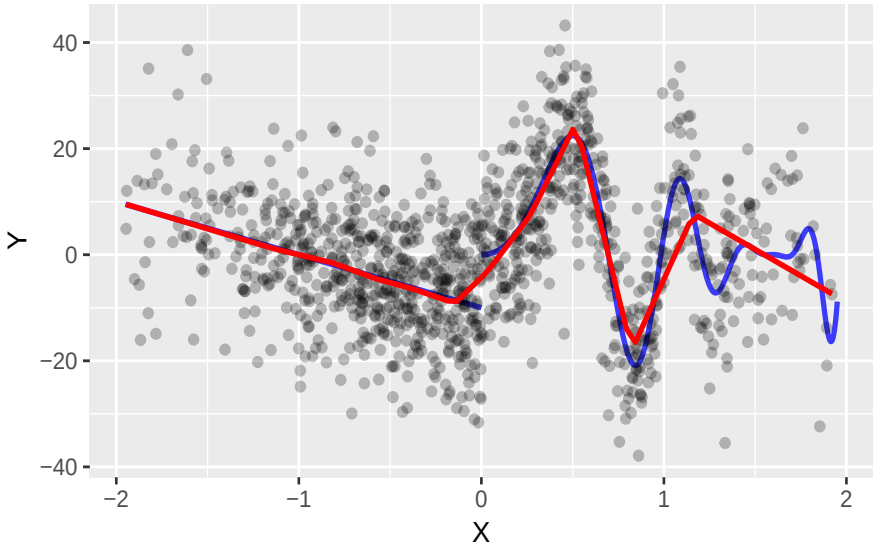
Degree 3 (cubic) spline with 3 knots



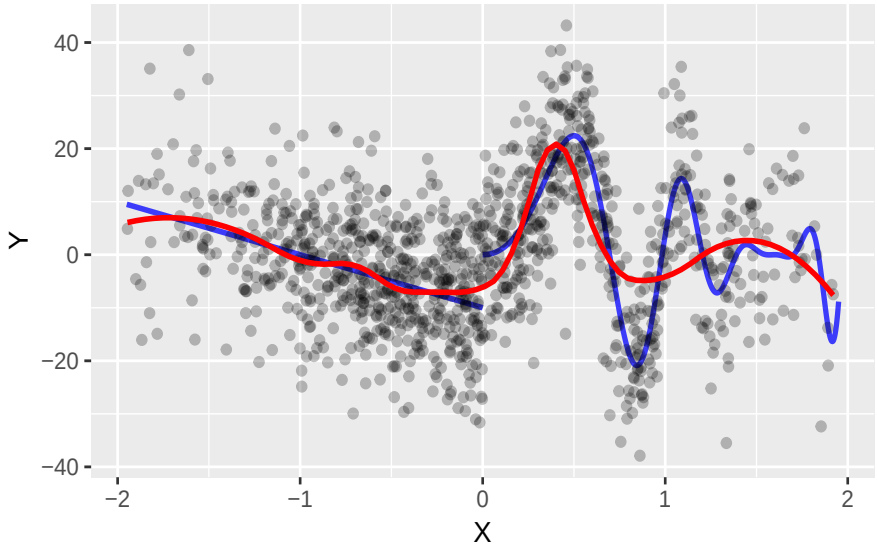
Degree 4 spline with 3 knots



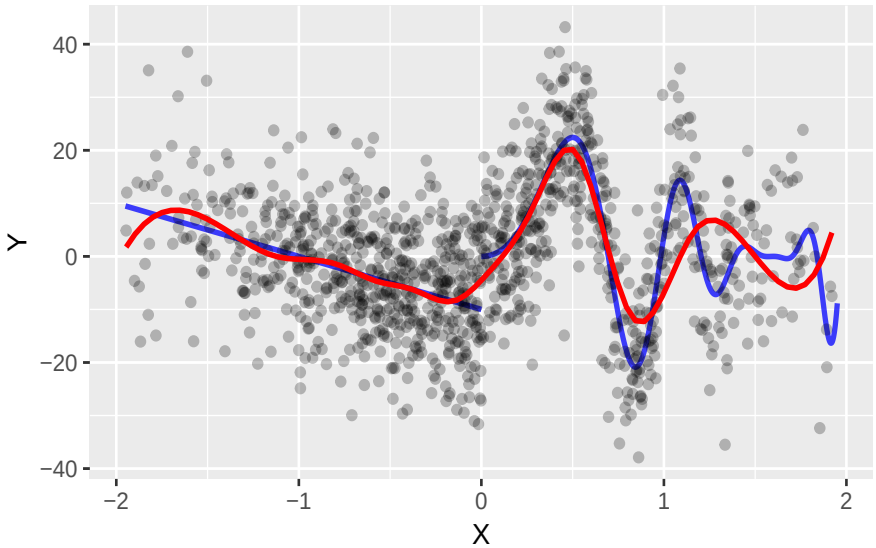
Degree 1 (linear) spline with 10 knots



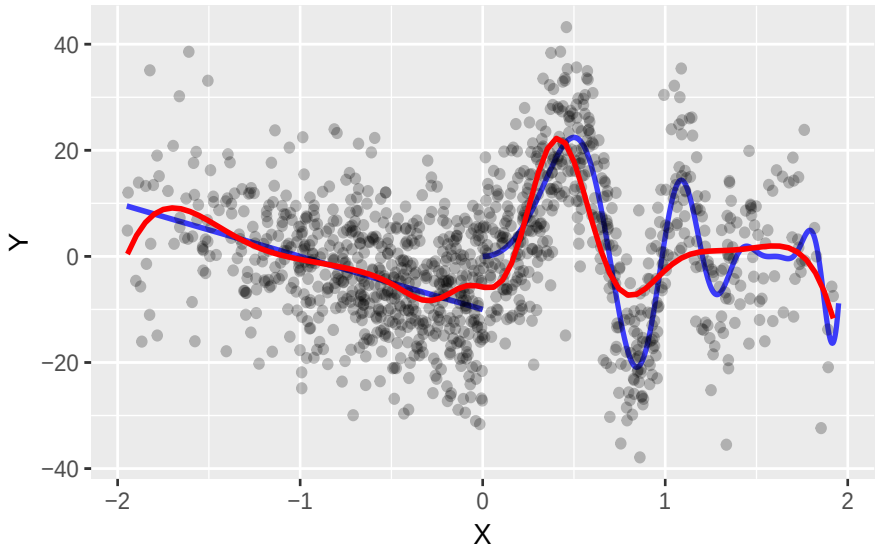
Degree 2 (quadratic) spline with 10 knots



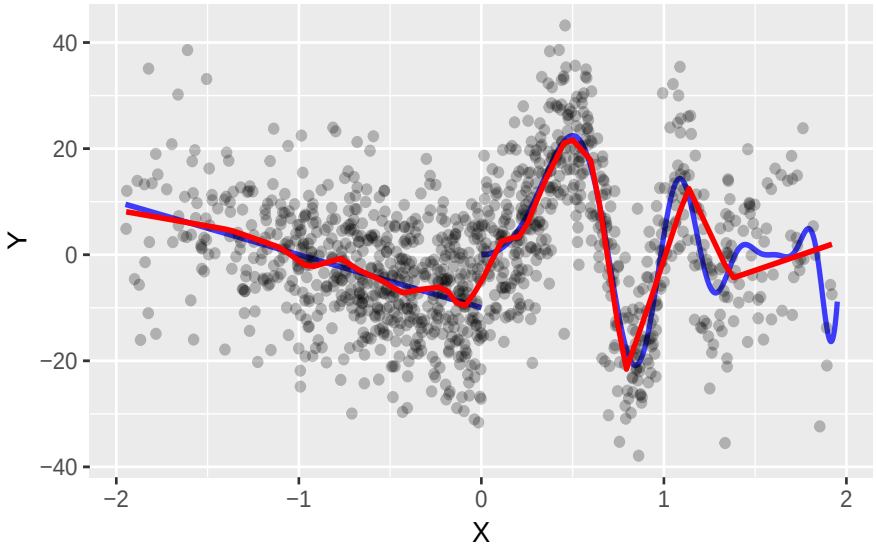
Degree 3 (cubic) spline with 10 knots



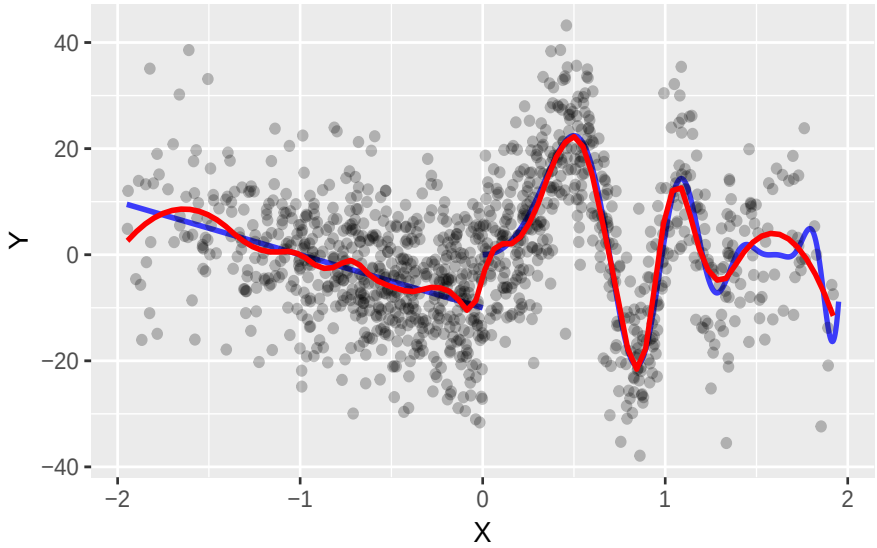
Degree 4 spline with 10 knots



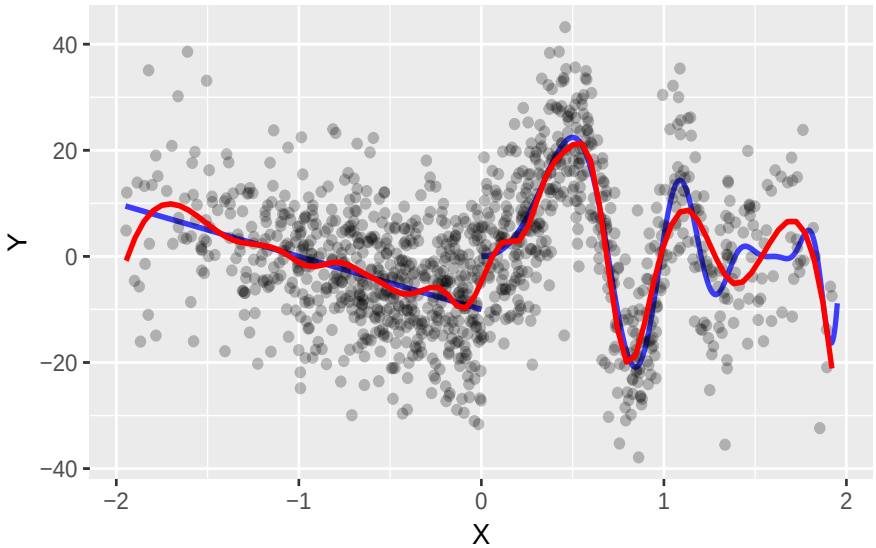
Degree 1 (linear) spline with 20 knots



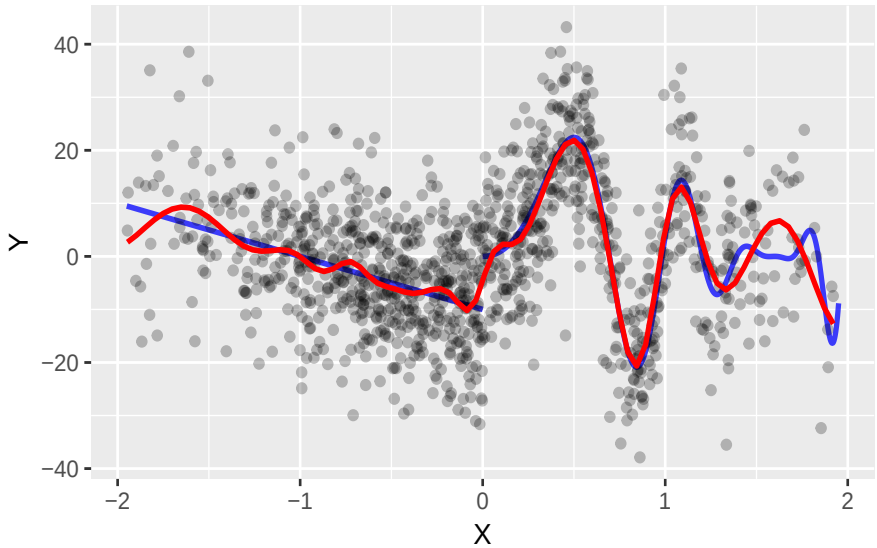
Degree 2 (quadratic) spline with 20 knots



Degree 3 (cubic) spline with 20 knots



Degree 4 spline with 20 knots



Natural Splines

- A drawback of the cubic spline is that, because the data are usually scarce near the boundaries of the predictor x, it has large variances and may exhibit erratic patterns at two ends.
- To improve the boundary behaviors of a cubic spline, one may add additional boundary constraints such that the cubic spline is linear beyond the boundary knots of x. This leads to the natural cubic spline (or simply the natural spline). Due to the linearity constraints, a natural spline typically has smaller variance near the boundaries of x than does a cubic spline.
- When fitting a natural spline, one must first determine the (interior) knots $\xi_1, \ldots, \xi_k$, as in fitting a cubic spline.

Boundary Constraints

Consider the cubic spline with k knots:

$$\beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 h^3(x, \xi_1) + \cdots + \beta_{3+k} h^3(x, \xi_k).$$

For $x < \xi_1$, the linearity requirement suggests $\beta_2 = \beta_3 = 0$. For $x > \xi_k$,

$$\beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x - \xi_1)^3 + \cdots + \beta_{3+k} (x - \xi_k)^3,$$

where $(x - \xi_j)^3 = x^3 - 3\xi_j x^2 + 3\xi^2 x - \xi_j^3$. The linearity requirement then suggests $\beta_2 - 3\sum_{j=1}^k \beta_{3+j}\xi_j = 0$ and $\beta_3 + \sum_{j=1}^k \beta_{3+j} = 0$, so that the following constraints must be satisfied:

$$\sum_{j=1}^k \beta_{3+j}\xi_j = \sum_{j=1}^k \beta_{3+j} = 0.$$

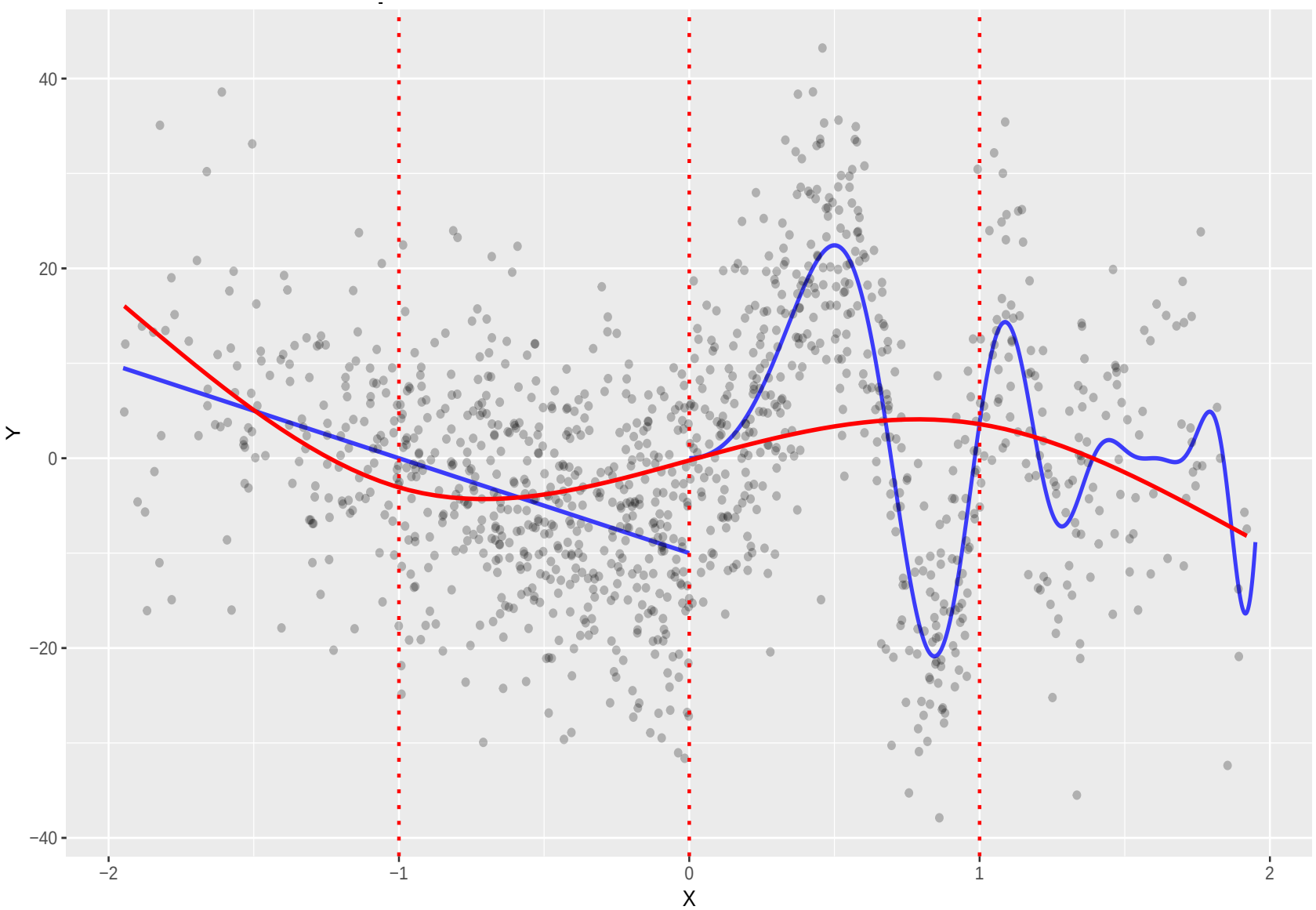
With these four constraints, it can be shown that a natural cubic spline with k knots can be represented using $k+2$ basis functions:

$$\sum_{j=1}^{k+2} N_j(x)\theta_j,$$

where N_j are the basis functions with $N_1(x) = 1$ and $N_2(x) = x$; for other basis functions, see pp. 145–146 of HTF (2009). Hence, a natural cubic spline can also be estimated using OLS.

Remark: To estimate a natural cubic spline with k (interior) knots, `ns` in `R` adds two boundary knots at the minimum and maximum values of x as its default; one may also specify the locations of the boundary knots. The fitted natural spline in `R` is linear outside the boundary knots of x.

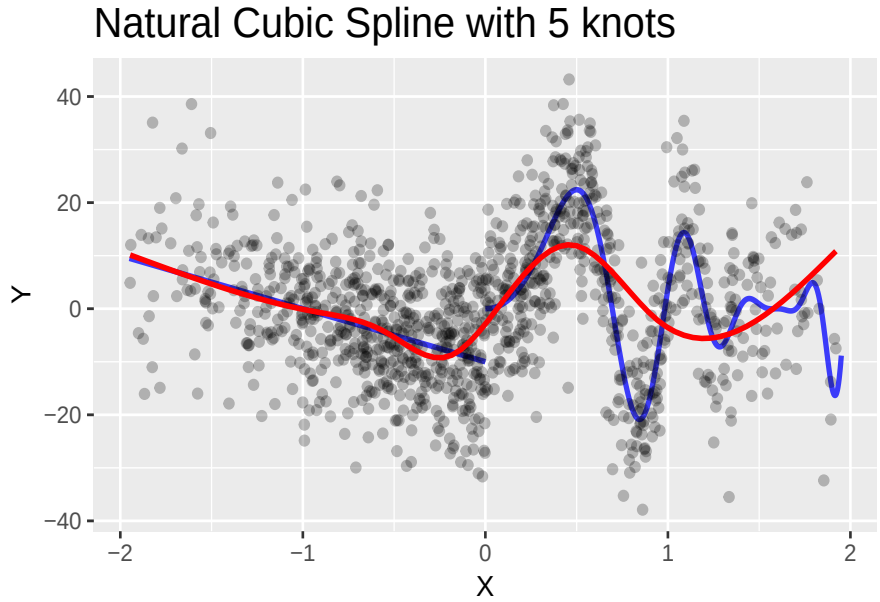
Compared with the cubic spline in p. 26, the natural cubic spline behaves very differently at two ends.

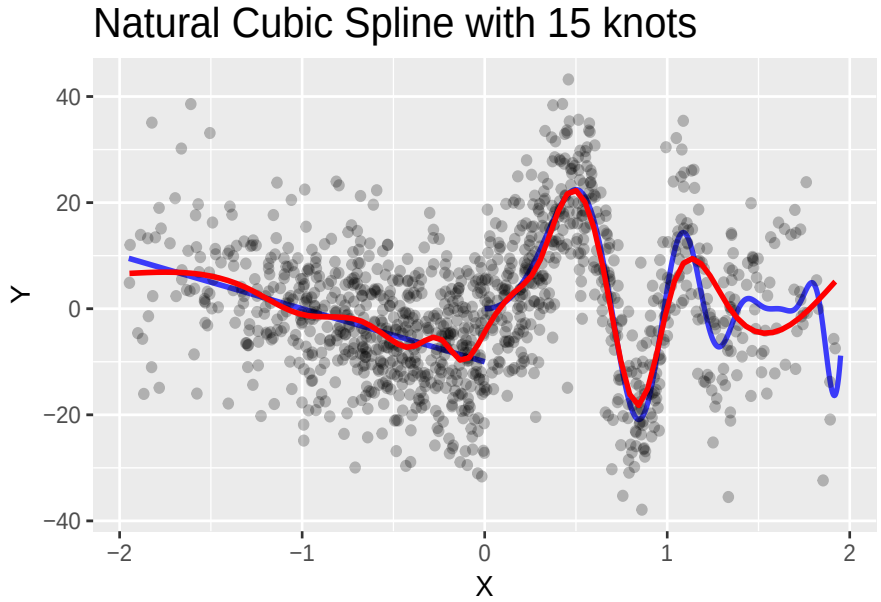


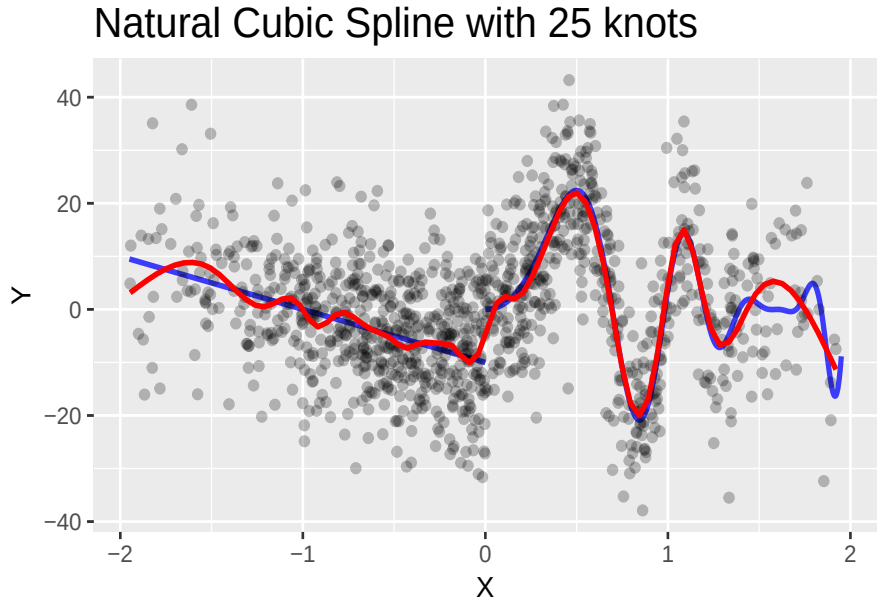
Number and Locations of Knots

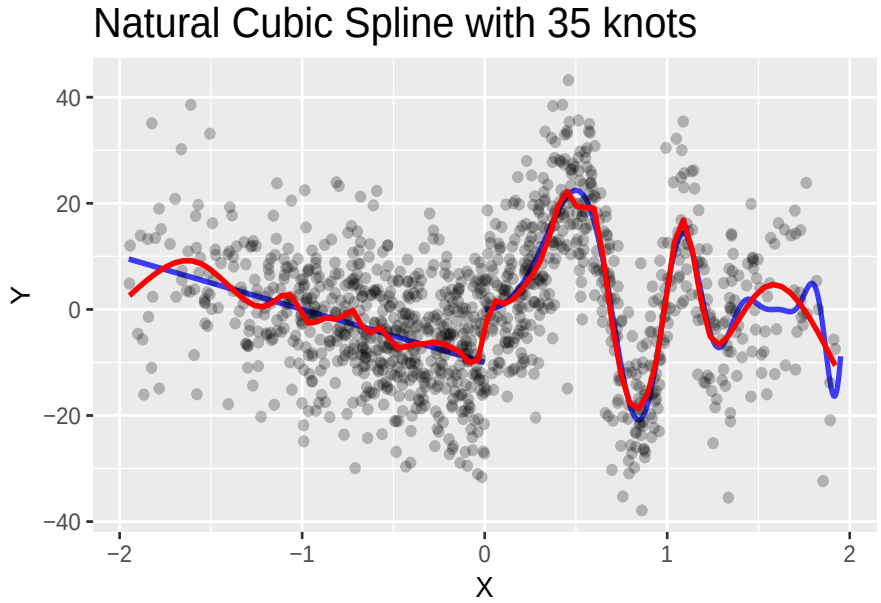
To capture complex nonlinearity, a cubic spline can be made more flexible by increasing the number of knots and/or placing more (less) knots in the regions where the data appear to vary rapidly (be stable).

- We plot in the next page the natural cubic spline with 5, 15, 25, and 35 knots, all uniformly distributed on data quantiles. It can be observed that the curve with 5 knots misses much variation of the true function on the right. When there are more knots, the resulting spline captures more nonlinearity and fits the data on the right better. Yet, they also result in unnecessary fluctuation for the linear function part.
- We also compare the splines with knots evenly distributed on x and those uniformly distributed on certain quantiles; for the latter, there are more (less) knots in the region where the data are dense (sparse).

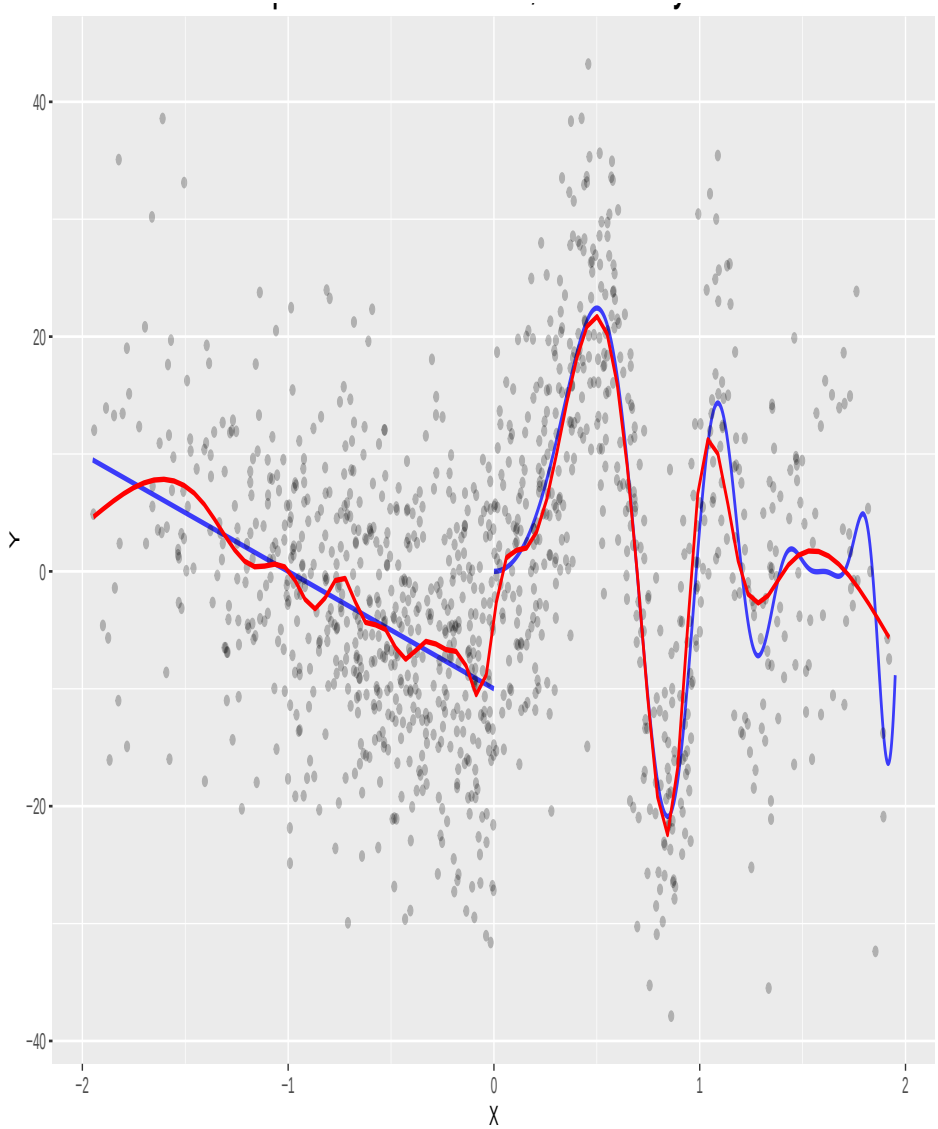

Natural Cubic Spline with 5 knots
40
20
0
−20
−40
Y
−2
−1
0
1
2
X

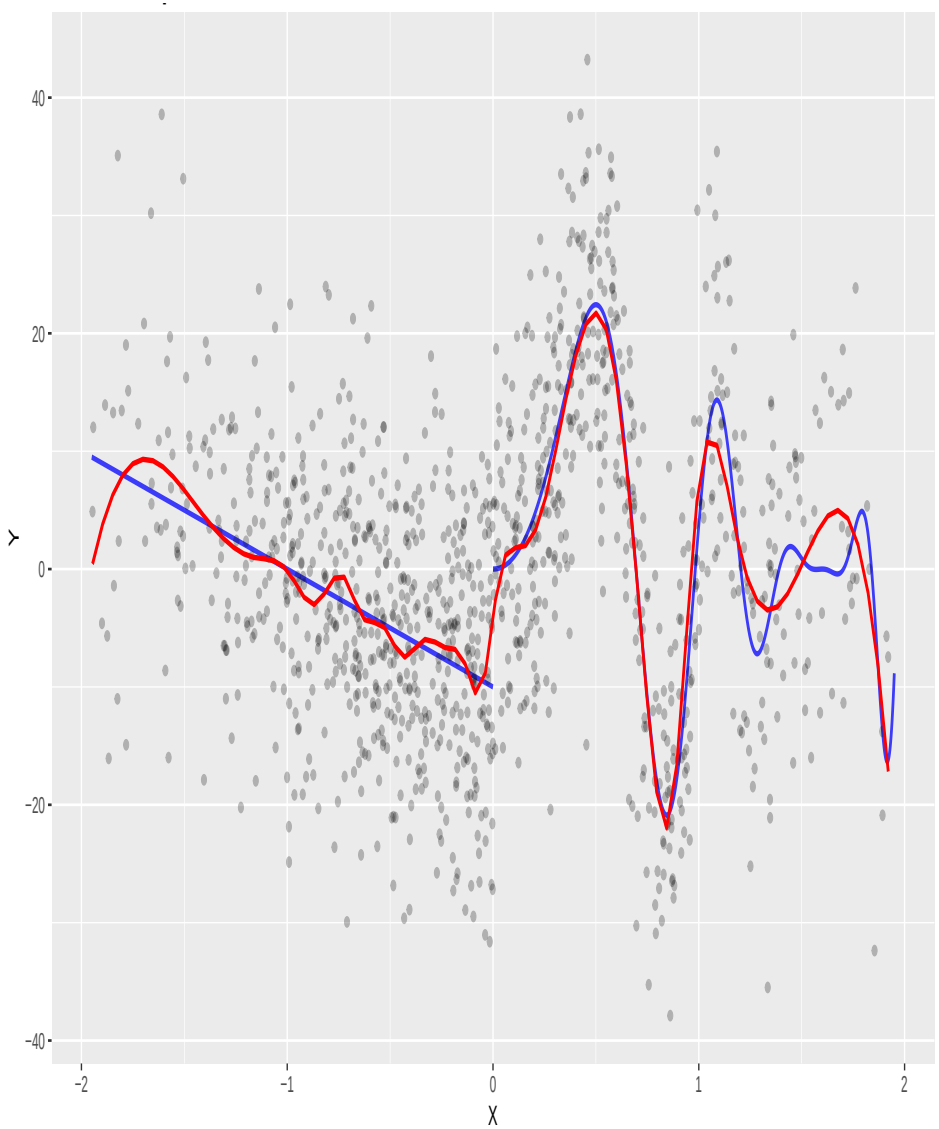

Natural Cubic Spline with 15 knots
40
20
0
−20
−40
Y
−2
−1
0
1
2
X


Natural Cubic Spline with 25 knots
40
20
0
−20
−40
Y
−2
−1
0
1
2
X

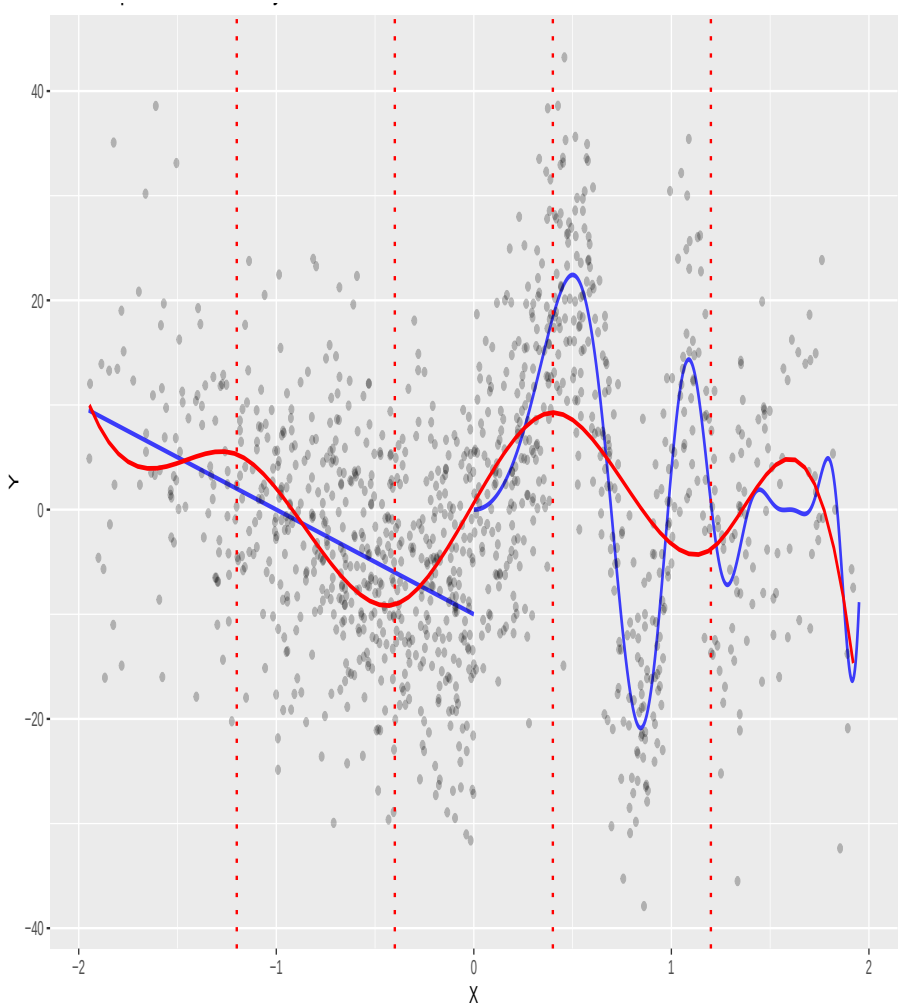

Natural Cubic Spline with 35 knots
40
20
0
−20
−40
Y
−2
−1
0
1
2
X

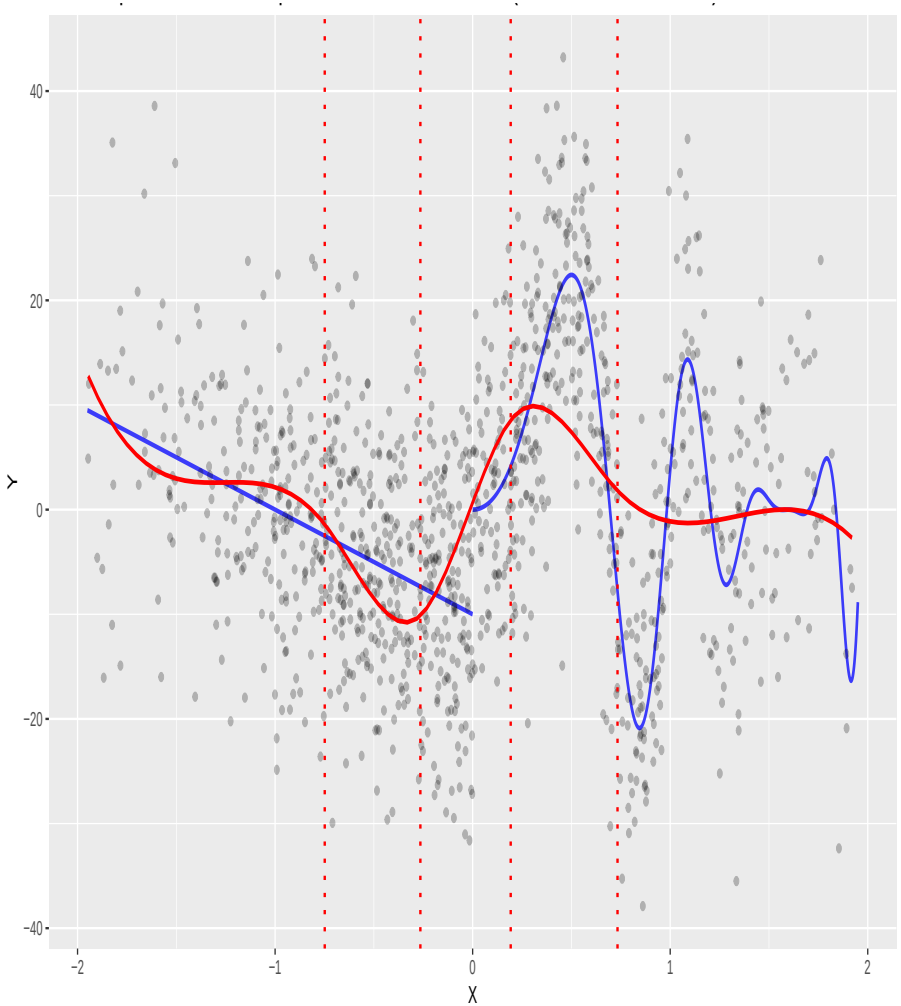
Natural cubic spline with 22 knots determined by 10-fold CV (left) and the cubic spline with 22 knots (right).



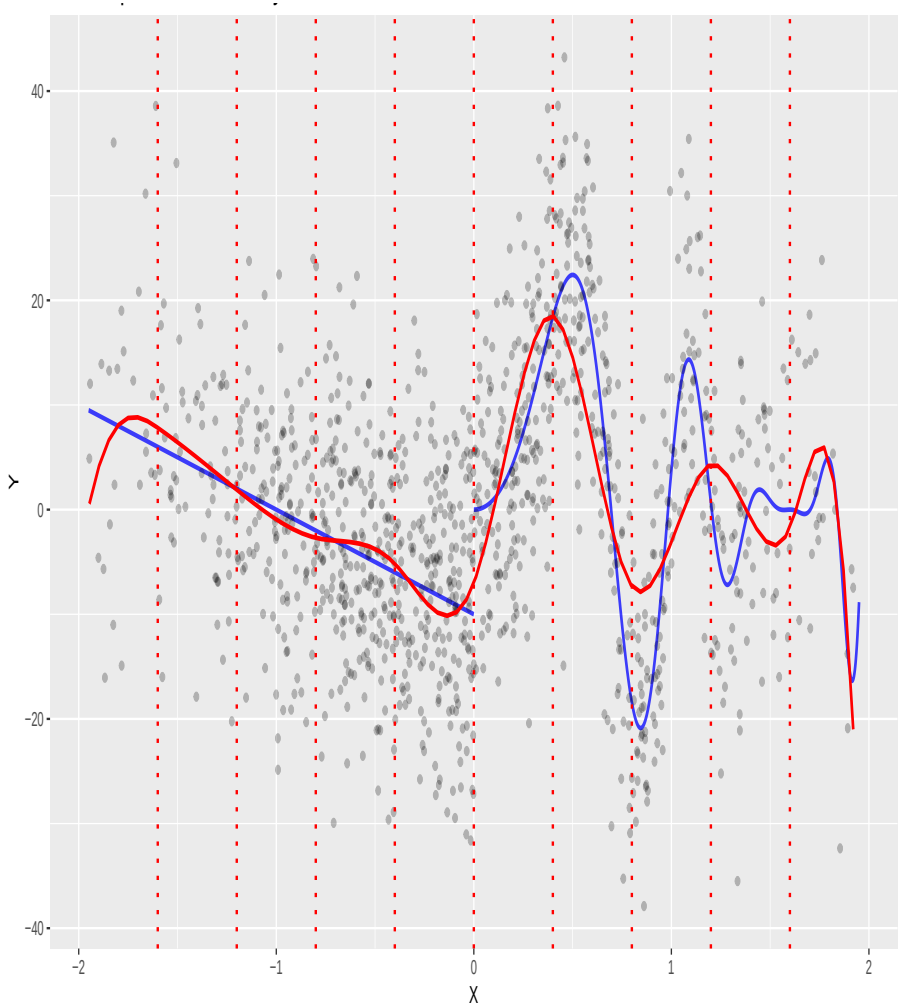


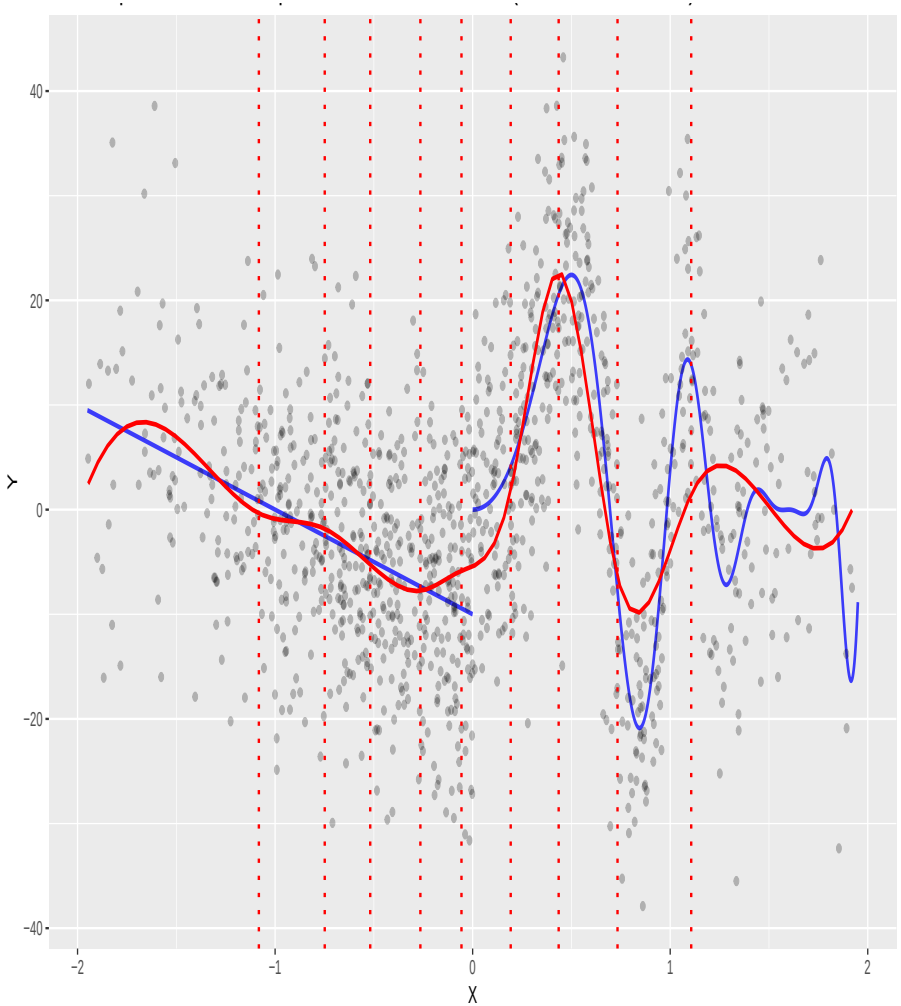
Below are the cubic spline with 4 knots evenly distributed on x (left) and the one with 4 knots on data quintiles (right).





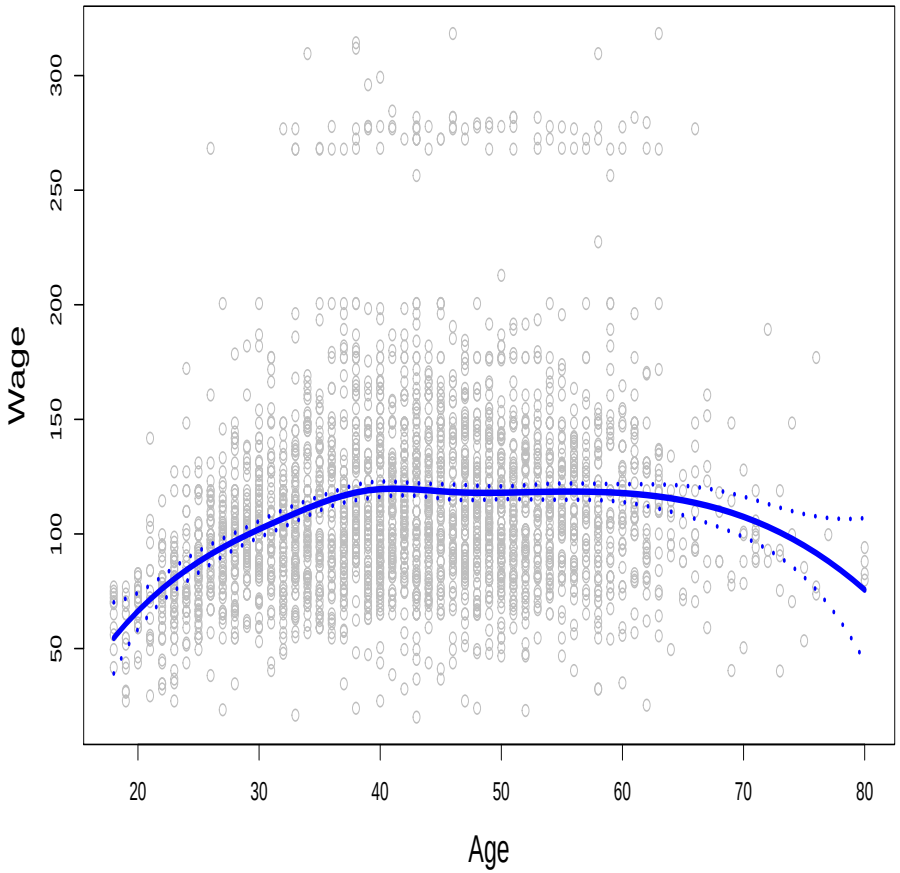
Below are the cubic spline with 9 knots evenly distributed on x (left), and the one with 9 knots on data deciles (right).

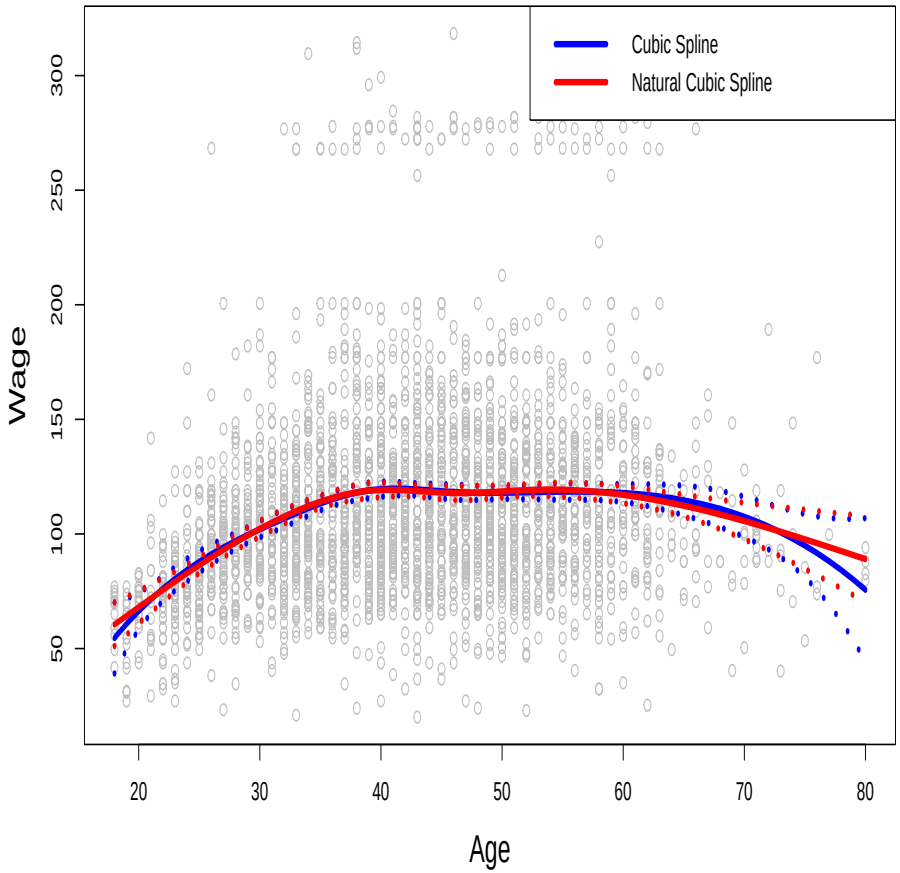




Example: Wage in the Mid-Atlantic Region

We fit a cubic spline and a natural cubic spline, with age the predictor and 4 knots determined by data quintiles. Note that the variance of the natural cubic spline (red) is smaller near the boundary of age.





Smoothing Splines

To fit a function g to a sample of n observations, it is natural to consider minimizing $\sum_{i=1}^{n}[y_i - g(x_i)]^2$. Without any constraint, this leads to the g function that interpolates all y_i so that the error sum of squares is zero. Such a function clearly over fits the data and is not a desirable solution. Instead, we may want to find a function that is sufficiently smooth. To this end, we minimize:

$$\sum_{i=1}^{n}[y_i - g(x_i)]^2 + \lambda \int g''(t)^2 \, \mathrm{d}t,$$

where λ is a non-negative tuning parameter, and g'' is the second-order derivative of g. The resulting solution is known as a smoothing spline.

- $g''(t)$ measures the change of the slope $g'(t)$. If g is smooth, $g'(t)$ would not be changing much, so that the total change of the slope, $\int g''(t)^2 \, \mathrm{d}t$, would be small; otherwise, $g'(t)$ would change a lot, resulting a large integral value. The penalty term $\lambda \int g''(t)^2 \, \mathrm{d}t$ thus controls the smoothness (or roughness) of the solution.
- When $\lambda = 0$, there is no control of the smoothness of the solution. When λ increases, a rough function will be penalized more, so that the solution will tend to be smoother; when $\lambda \to \infty$, the solution will have to be perfectly smooth (a straight line), and the minimization problem is simply a (nonlinear) LS problem.
- It turns out that the resulting solution is a natural cubic spline with (interior and boundary) knots at the distinct values of $x_1, \ldots, x_n$. That is, the solution has continuous first and second derivatives and is linear outside the range of x_i.

Remark: The smoothing spline is not the same as the natural spline with the fixed knots at $x_1, \ldots, x_n$, but it is a shrunken version of such spline, where the level of shrinkage is determined by the smoothing parameter λ. To see this, we note that a natural cubic spline can be represented using the basis functions N_j and hence can write the smoothing spline as: $g(x) = \sum_{j=1}^{n} N_j(x)\theta_j$. Let $\boldsymbol{N}$ be the $n \times n$ matrix with $N_{ij} = N_j(x_i)$. Then, the objective function for the smoothing spline is

$$(\boldsymbol{y} - \boldsymbol{N\theta})'(\boldsymbol{y} - \boldsymbol{N\theta}) + \lambda \boldsymbol{\theta}' \boldsymbol{\Omega} \boldsymbol{\theta},$$

where $\boldsymbol{\Omega}$ is the $n \times n$ matrix with $\Omega_{jk} = \int N_j''(t) N_k''(t)\, \mathrm{d}t$. It is readily seen that the minimizer is

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{N}'\boldsymbol{N} + \lambda \boldsymbol{\Omega})^{-1} \boldsymbol{N}' \boldsymbol{y},$$

as in the Ridge regression.

It follows that the vector of the fitted smoothing spline is

$$\hat{\boldsymbol{g}}_\lambda = \boldsymbol{N}(\boldsymbol{N}'\boldsymbol{N} + \lambda\boldsymbol{\Omega})^{-1}\boldsymbol{N}'\boldsymbol{y} =: \boldsymbol{S}_\lambda \boldsymbol{y}.$$

In the context of linear regression with k parameters, recall that the vector of the OLS fitted values is $\hat{\boldsymbol{y}} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$, where the orthogonal projection matrix $\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$ is $n \times k$ with rank k, which is also its trace (the sum of the diagonal elements):

$$\text{trace}(\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}') = \text{trace}(\boldsymbol{X}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}) = \text{trace}(\boldsymbol{I}_k).$$

In analogy, we may define the effective degrees of freedom of a smoothing spline as the trace of $\boldsymbol{S}_\lambda$, which depends on λ. The value of λ determines the coefficients for the basis functions that will be shrunken towards zero and hence the number of basis functions that virtually matter.

Choosing the Smoothing Parameter

Although there is no need to determine the number of knots in fitting a smoothing spline, we need to choose the smoothing parameter λ. We may determine λ by cross validation, i.e., the λ that yields the smallest error sum of squares. In the light of leave-one-out cross validation (LOOCV) discussed in Lecture 6, the LOOCV error sum of squares for each λ can be computed using only the smoothing spline fits to all data. As such, we find the λ value that minimizes

$$\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i-\hat{g}_\lambda(x_i)}{1-s_{\lambda,i}},\right)^2.$$

where $s_{\lambda,i}$ is the i^{th} diagonal element of $\boldsymbol{S}_\lambda$.

Local Regressions

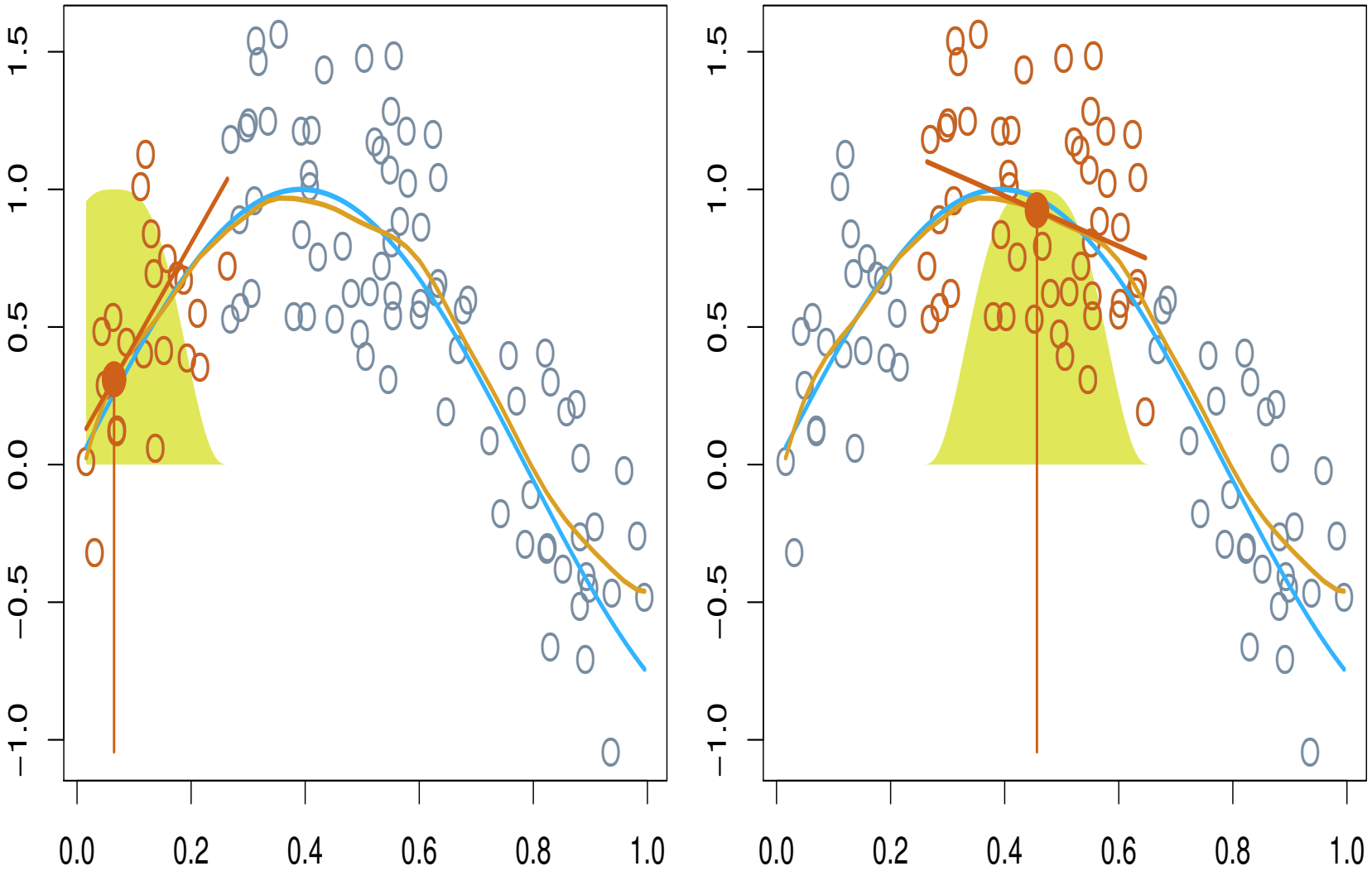
Local Regression is a weighted least squares regression that computes the fitted value at each point x_o using only those data in a neighborhood of x_o. The procedure below is repeated for every x_o in the sample.

1. Select a fraction (span) of the data, say $s\%$, that are closest to x_o.
2. Assign weights $K_{io} = K(x_i, x_o)$ to the data x_i in the selected span of x_o, where a point closer to x_o is assigned a larger weight. The data outside the span receive zero weight.
3. Find a linear, quadratic, or high-order function of x that minimizes:

$$\sum_{i=1}^{n} K_{io}[y_i - f(x_i)]^2.$$

The fitted value of x_o is given by $\hat{f}(x_o)$.

Local Regression



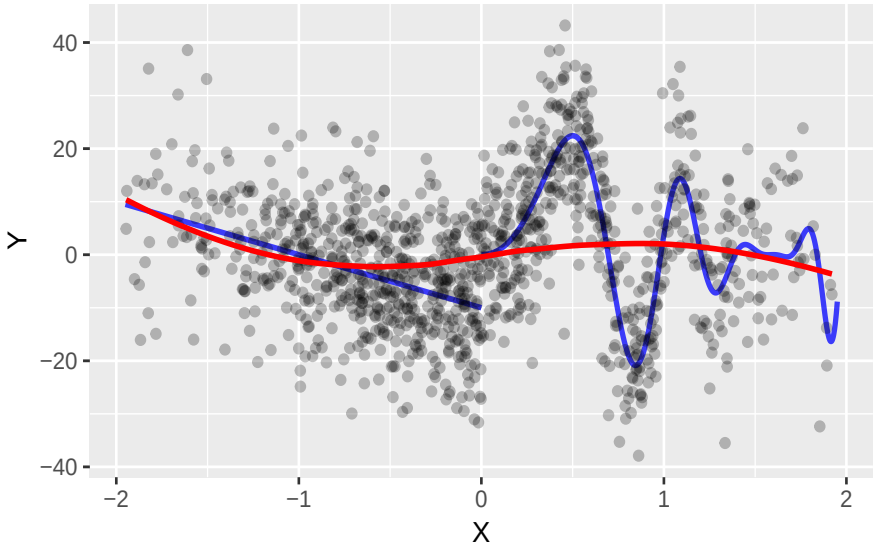
Source: Figure 7.9 of JWHT (2013)

- A local regression must first determine the regression function f (linear, quadratic, or cubic). In addition, for each x_o, we need to choose:
  - Span s: A tuning parameter that controls the flexibility of the regression fit.
  - Weight function K_{io}: `loess` (locally estimated scatterplot smoothing) of R uses the "tricubic" weights:

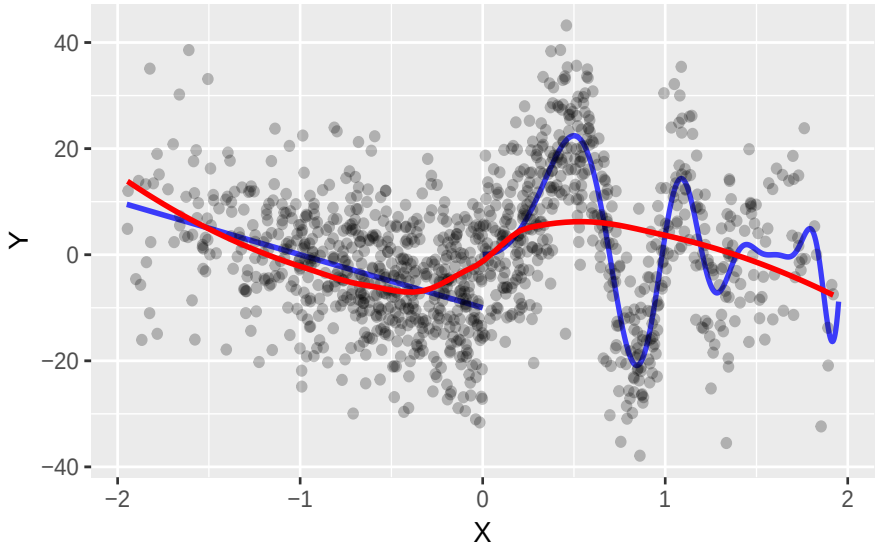
$$[1 - (\text{dist}/\text{max-dist})^3]^3,$$

    where "dist" is the distance between x_i in the span and x_o, and "max-dist" is the largest "dist" from x_o. Clearly, a point whose distance to x_o close to the max-dist receives a smaller weight; otherwise, it receives a larger weight.
- In local regressions, only part of the sample is used for a local fit. Hence, a large sample is needed for better fitness. fit.
- Below are the estimated local quadratic regressions with tricubic weights. We observe that a smaller s yields a more local and rough fit and that a large s leads to a less local but smoother fit.

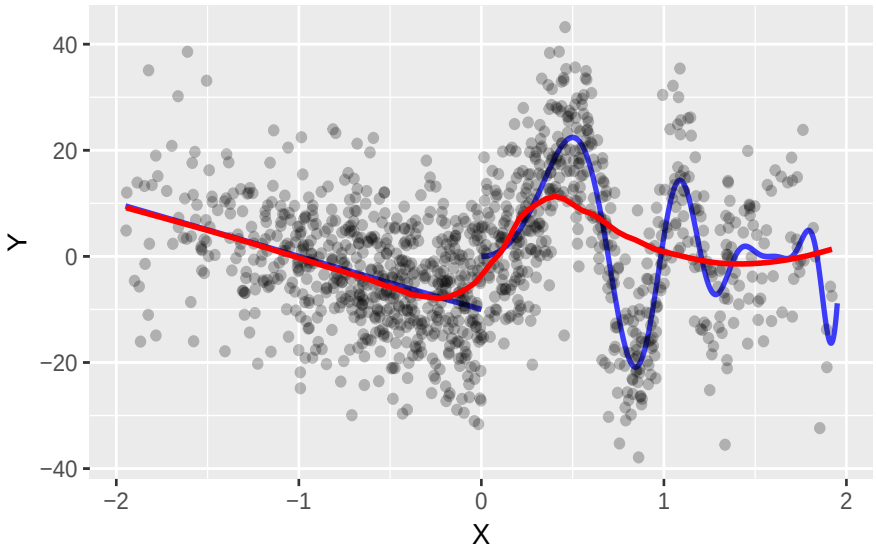
Local Regression with span = 100%



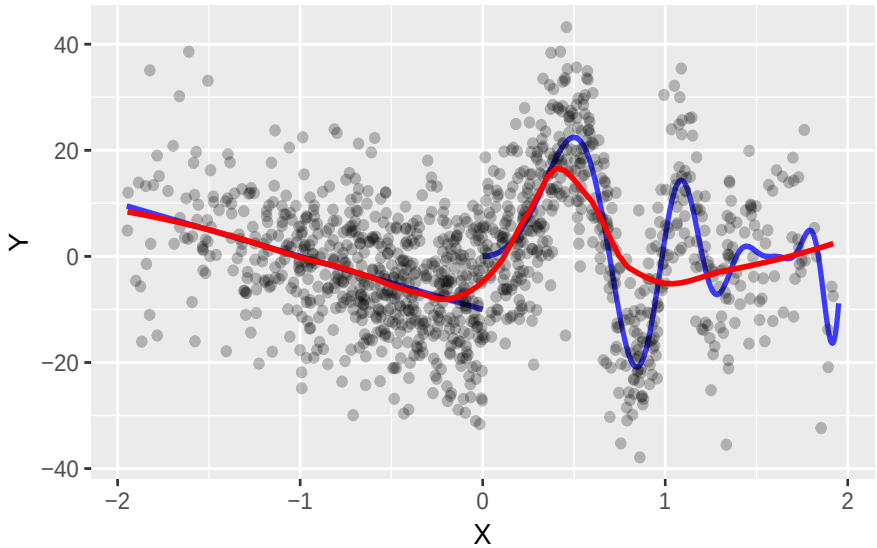
Local Regression with span = 80%

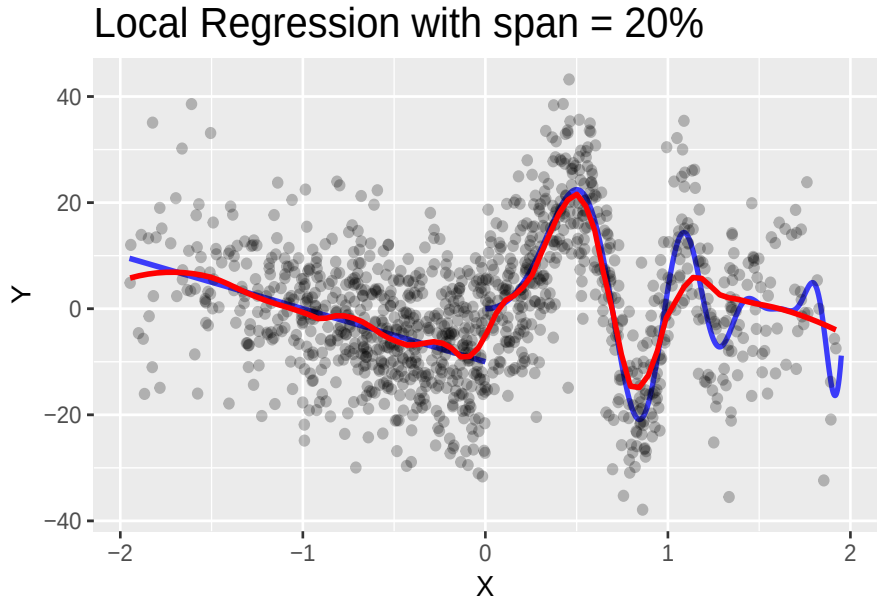


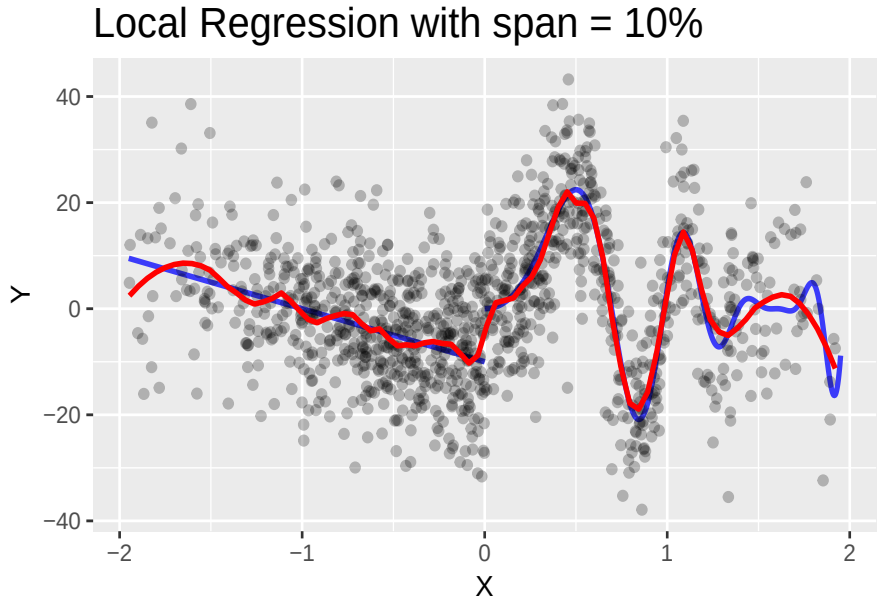
Local Regression with span = 60%

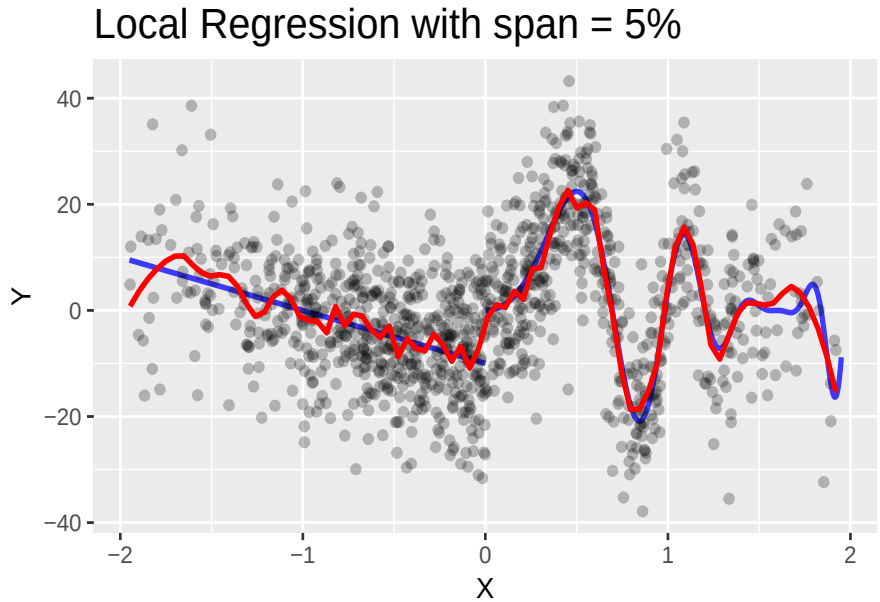


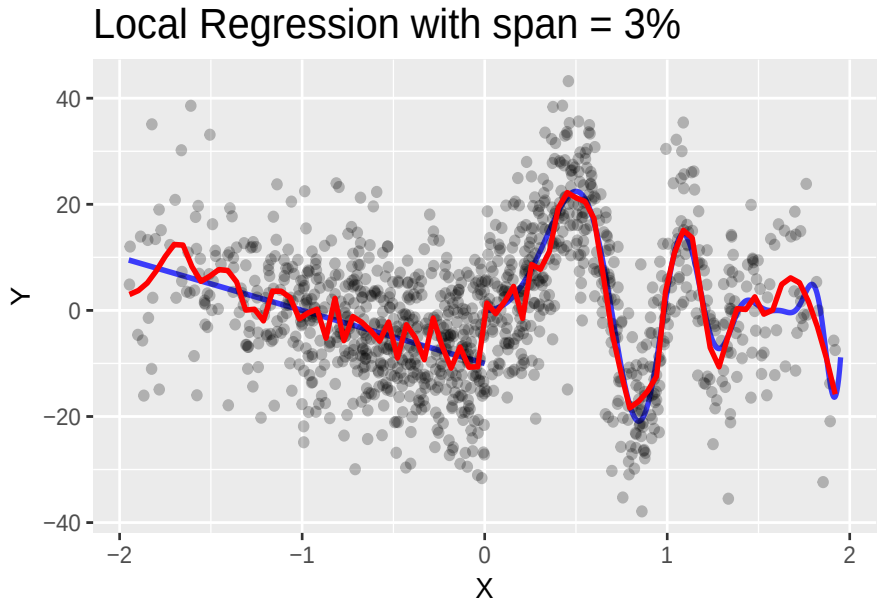
Local Regression with span = 40%




Local Regression with span = 20%
40
20
0
−20
−40
Y
−2
−1
0
1
2
X

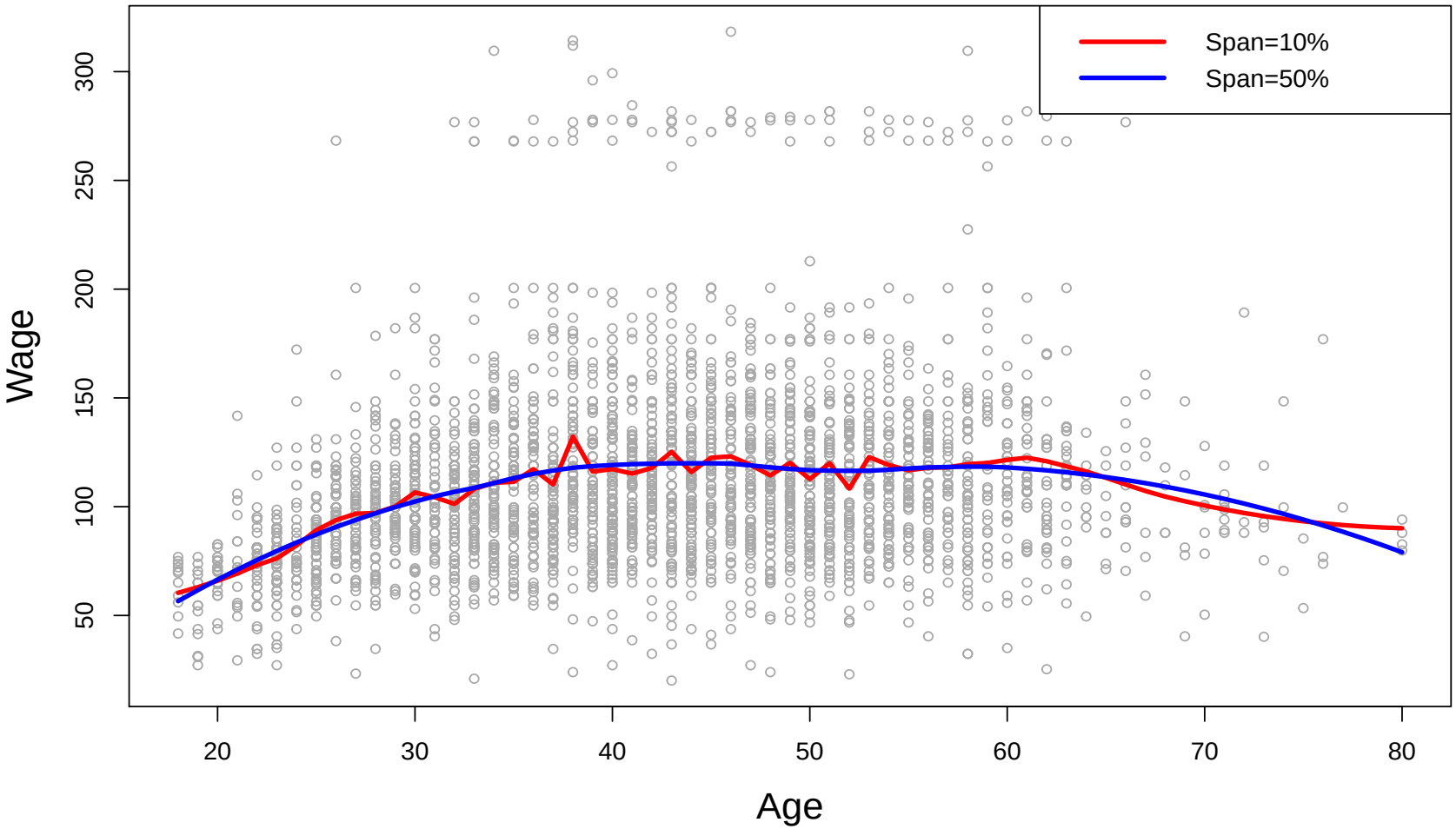

Local Regression with span = 10%
40
20
0
−20
−40
Y
−2
−1
0
1
2
X


Local Regression with span = 5%
40
20
0
−20
−40
Y
−2
−1
0
1
2
X


Local Regression with span = 3%
40
20
0
−20
−40
Y
−2
−1
0
1
2
X

Example: Wage in the Mid-Atlantic region

We estimate local quadratic regressions of wage on age, with the span $s = 10\%$ and 50%. Note the curve with $s = 10\%$ is very rough.



Generalized Additive Models

We now extend the previous analysis to models with multiple predictors $x_1, x_2, \ldots, x_p$. The Generalized Additive (GA) model can be expressed as:

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

where f_j may be a polynomial, cubic spline, or local regression.

- Instead of jointly modelling all predictors $x_1, x_2, \ldots, x_p$, this model admits a nonlinear function for each predictor and adds them up. Alternatively, we may also consider a joint model for a small number of predictors, e.g., $f(x_1, x_2)$.
- The additivity form excludes more complex interactions between predictors, but it permits study of the marginal effect of one predictor while holding other predictors fixed.

For example, a GA model with 2 predictors x_1 and x_2 such that f_1 and f_2 are a cubic spline with respective k knots: $\xi_1 \ldots, \xi_k$ and $\zeta_1, \ldots, \zeta_k$:

$$\begin{aligned} y_i &= \beta_0 + f(x_{i1}) + f(x_{i2}) + u_i \\ &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i1}^2 + \beta_3 x_{i1}^3 \\ &\quad + \beta_4 h^3(x_{i1}, \xi_1) + \cdots + \beta_{3+k} h^3(x_{i1}, \xi_k) \\ &\quad + \gamma_1 x_{i2} + \gamma_2 x_{i2}^2 + \gamma_3 x_{i2}^3 \\ &\quad + \gamma_4 h^3(x_{i2}, \zeta_1) + \cdots + \gamma_{3+k} h^3(x_{ik}, \zeta_k) + u_i. \end{aligned}$$

More generally, a GA model may employ different functions for x_1 and x_2, say, a cubic spline for x_1 and a local regression for x_2.

Note: The GA model is linear in parameters in general and can be easily estimated using OLS. Yet, OLS is not readily applied when a "smoothing spline" is involved; see JWHT (2013, pp. 284–285).

Example: Wage in the Mid-Atlantic Region

In addition to the variables `wage` and `age` studied earlier, we now also consider the following variables:

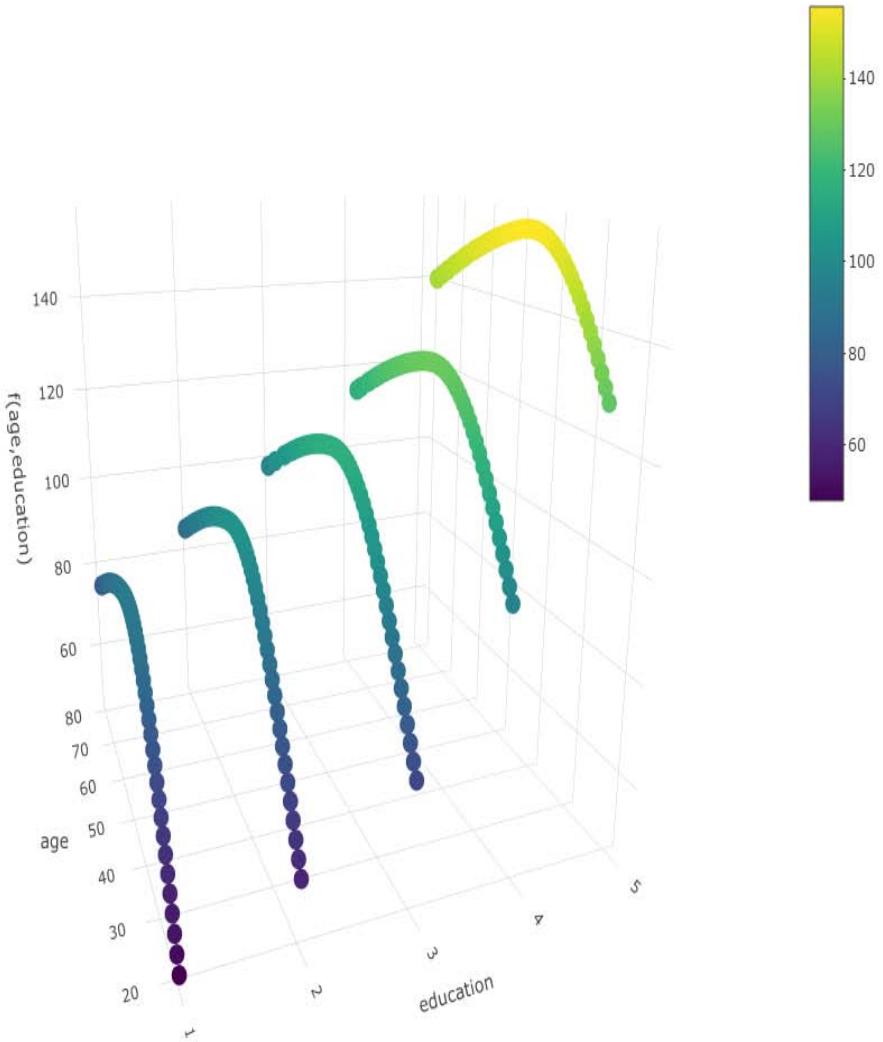
- `year`: Year that wages were recorded (from 2003 to 2009).
- `education`: 5 education levels of the worker: (1) $<$ HS Grad, (2) HS Grad, (3) Some College, (4) College Grad, (5) Advanced Degree.

We first fit the following GA model:

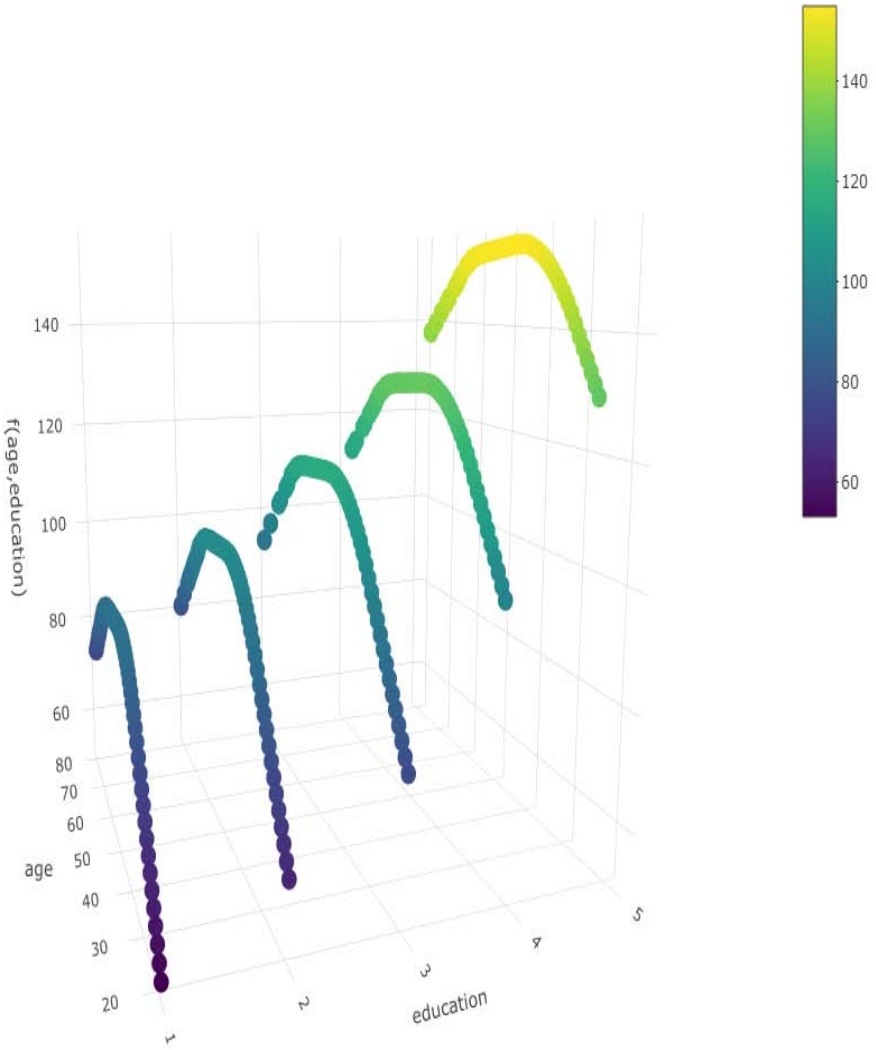
$$\texttt{wage} = \beta_0 + f_1(\texttt{age}) + f_2(\texttt{education}) + u.$$

We consider two cases: (1) f_1 is a natural cubic spline with 3 knots, and f_2 includes dummies for education levels; (2) f_1 is a local regression with span 50%, and f_2 includes education dummies.

Case 1: Natural cubic spline on age (3 knots) and education dummies



Case 2: Local regression on age (span 50%) and education dummies



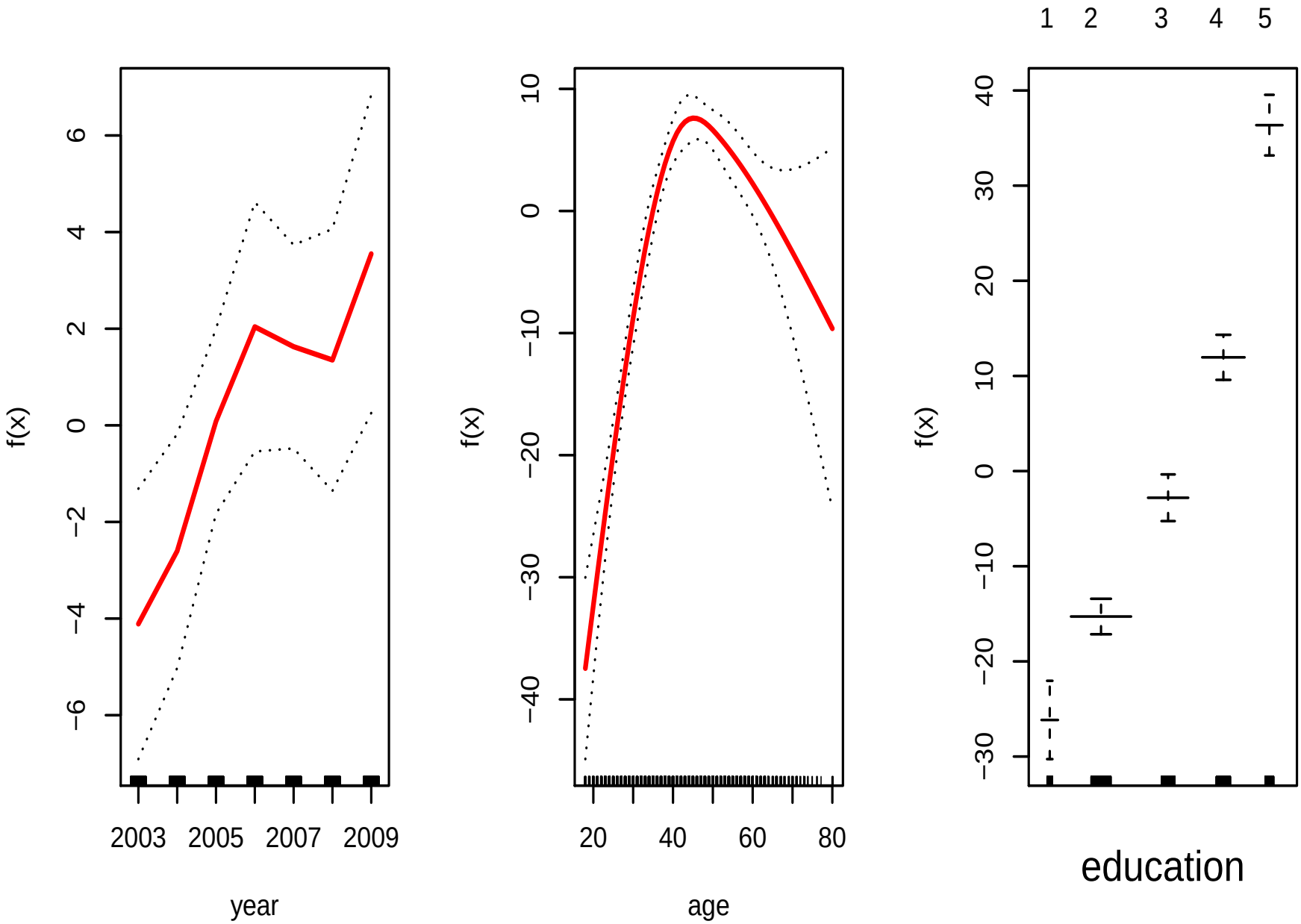
We then fit the following GA model with 3 predictors:

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

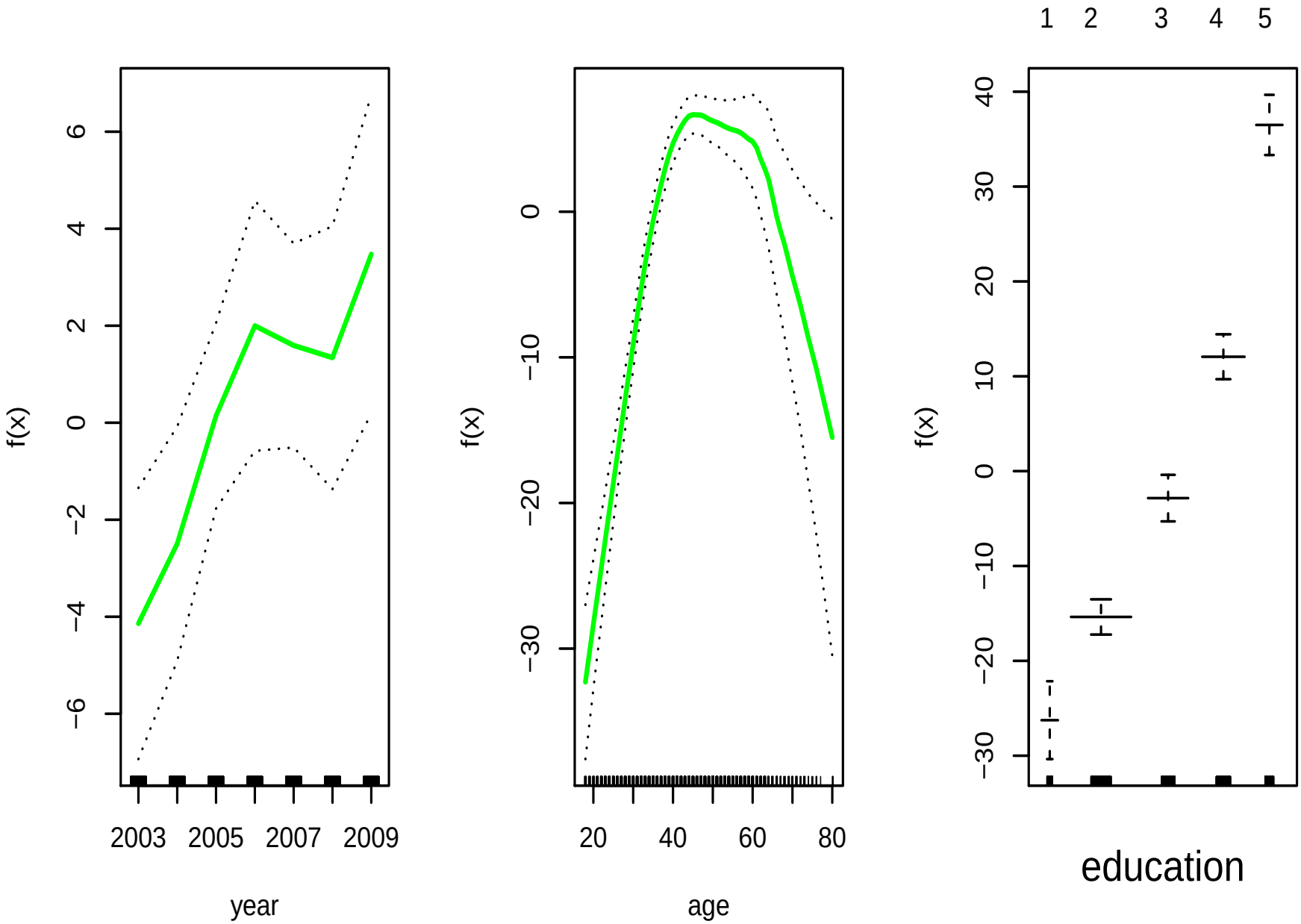
We consider two cases: (1) f_1 and f_3 are natural cubic splines with 3 knots, and f_2 education dummies; (2) f_1 is a local regression with span 50%, and f_2 and f_3 are the same as in (1). The results in the next two slides are summarized below.

- Left panel: Holding `age` and `education` fixed, `wage` tends to increase with `year`, with a drop during the 2008 financial crisis.
- Middle panel: Holding `year` and `education` fixed, `wage` is higher for the middle-age workers.
- Right panel: Holding `year` and `age` fixed, `wage` increases with `education`.

Case1: Natural cubic spline: age, year; dummies: education



Case 2: Local regression: age; natural cubic spline: year; dummies: edu



References and Acknowledgement

References

1. James, G., D. Witten, T. Hastie, and R. Tibshirani (2013). *An Introduction to Statistical Learning, with Applications in R*, New York: Springer. (JWHT (2013))
2. Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning*, Second Edition (12th printing 2017), New York: Springer. (HTF (2009))

Some of the figures in this presentation are taken from JWHT (2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani