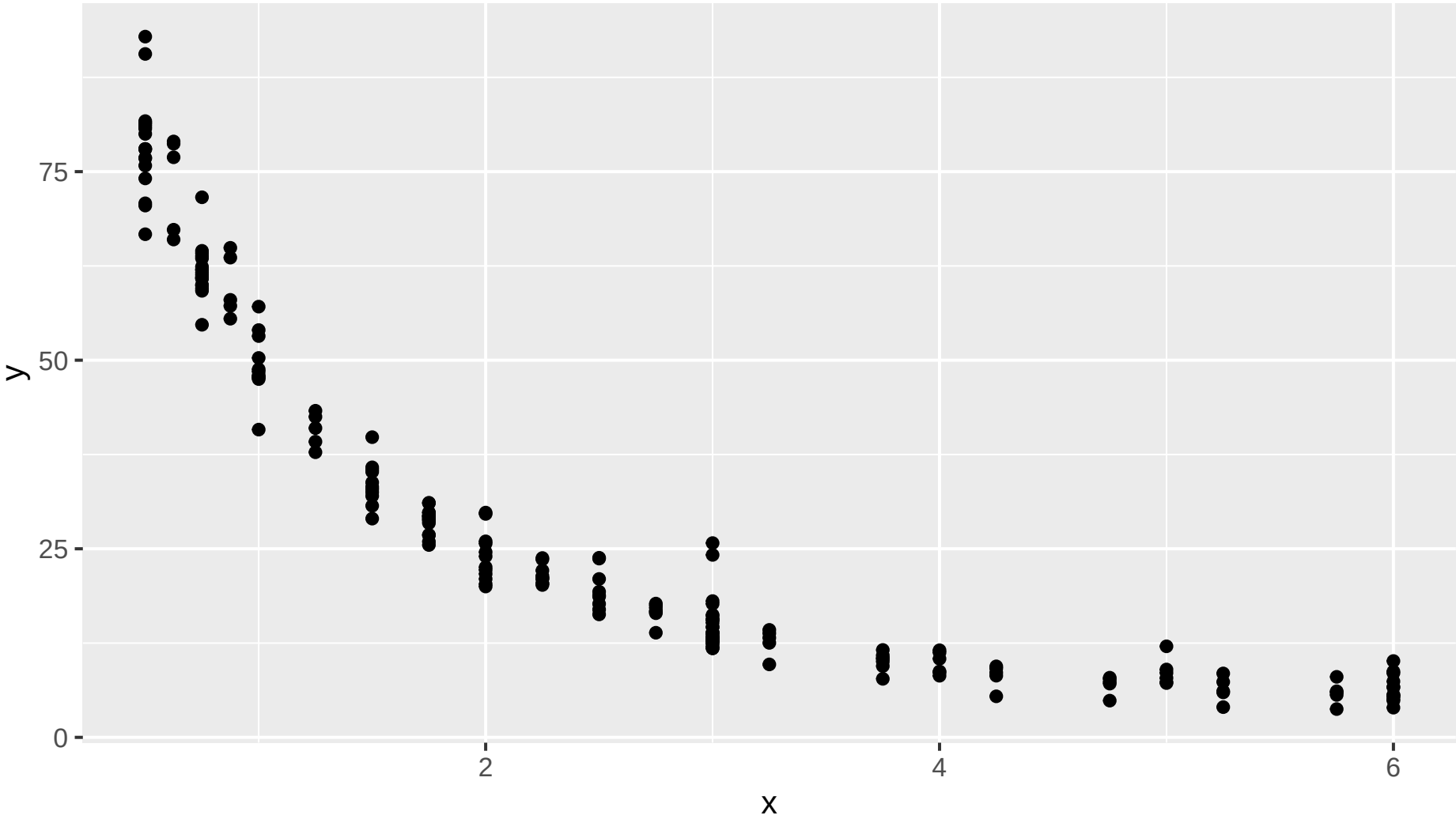
Regression Analysis
Chapter 11: Nonlinear Regression

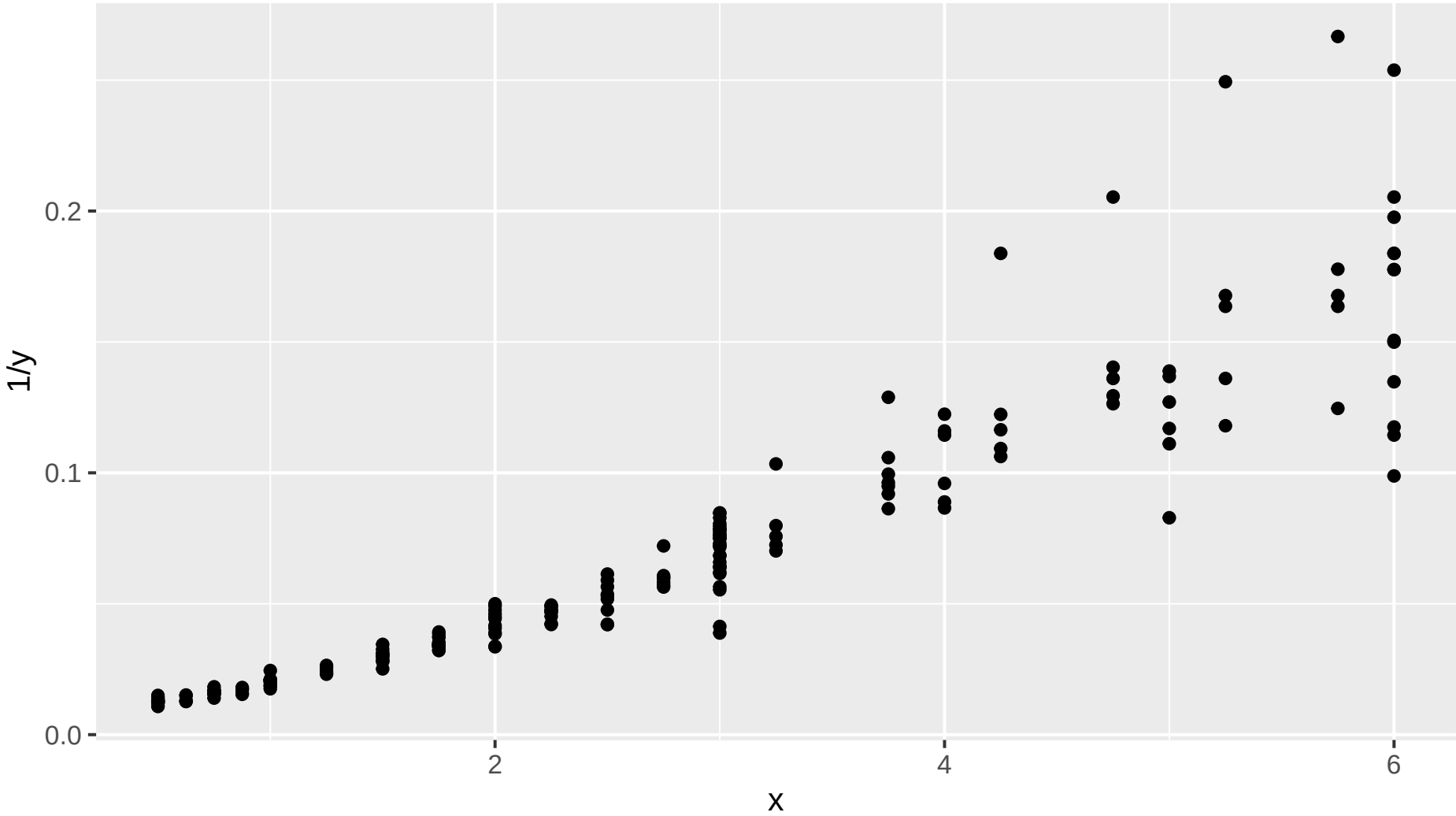
Shaobo Jin

Department of Mathematics

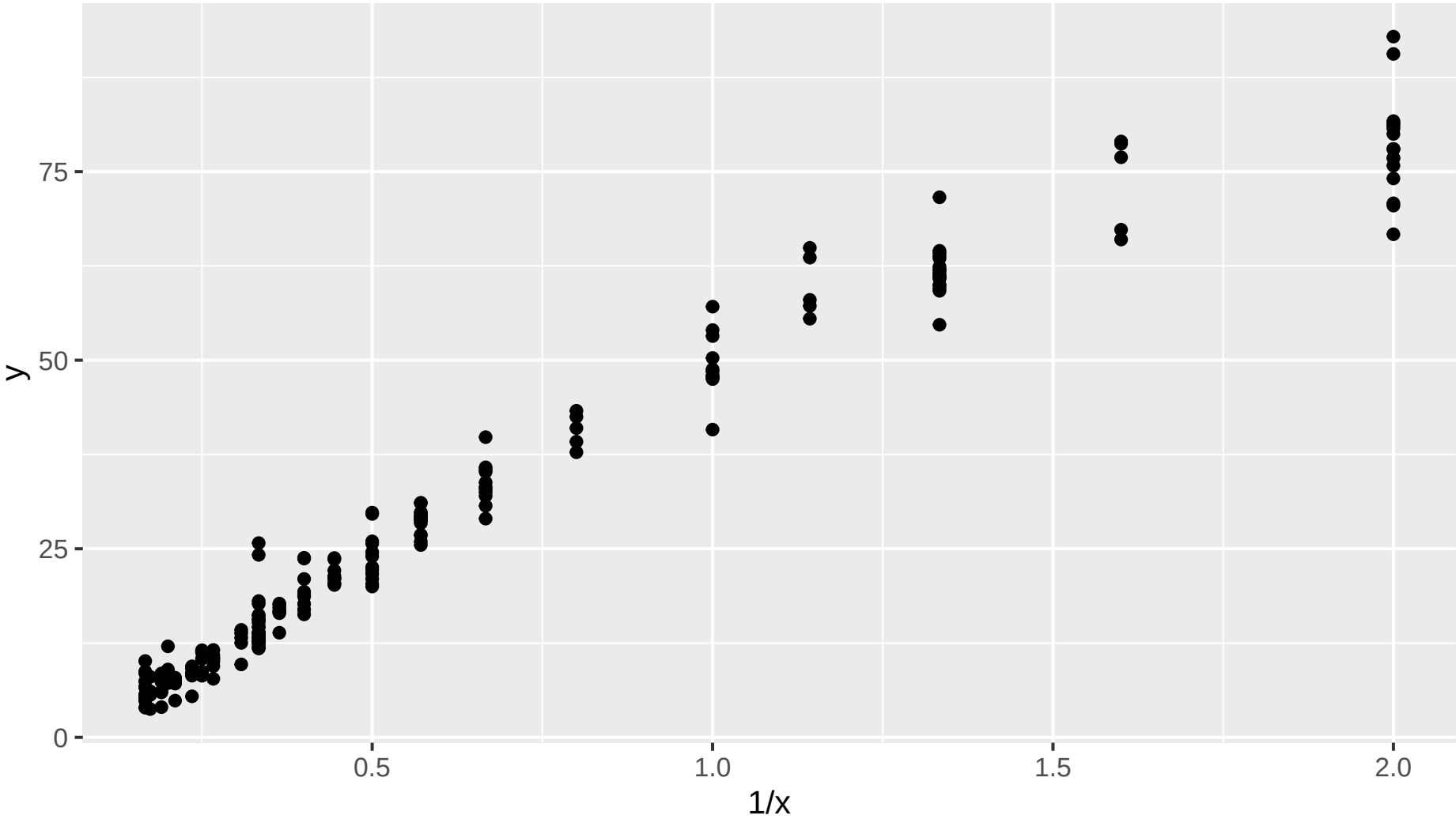
An Example: A Nonlinear Pattern



An Example: Transform y



An Example: Transform x



Nonlinear Mean Functions

In general, we consider

$$\mathrm{E}\left(Y \mid \boldsymbol{X} = \boldsymbol{x}\right) \quad = \quad m\left(\boldsymbol{x}, \boldsymbol{\theta}\right),$$

where $\boldsymbol{\theta}$ is a vector of parameters. The function form $m\left(\boldsymbol{x}, \boldsymbol{\theta}\right)$ can be

1. completely specified (e.g., parametric)
2. partially data-driven (e.g., semi-parametric)
3. data-driven (e.g., nonparametric)

Parametric Case

Suppose that we fully specify the function form of $m(\boldsymbol{x}, \boldsymbol{\theta})$, up to the values of $\boldsymbol{\theta}$.

- For example, for the data in the above example, we assume

$$\mathrm{E}(Y \mid X = x) = \frac{\exp(-ax)}{b + cx},$$

  where $\boldsymbol{\theta} = \begin{bmatrix} a & b & c \end{bmatrix}^T$ is the vector of parameters.
- The model is neither linear in the parameters nor in the regressors.

Estimation

Estimation of $\boldsymbol{\theta}$ can be done by maximum likelihood or least squares.

1. In maximum likelihood, we assume that $Y \mid \boldsymbol{x}$ follows some density $f\left(y \mid \boldsymbol{x}; \boldsymbol{\theta}\right)$ and maximize the likelihood function

$$\prod_{i=1}^{n} f\left(y_i \mid \boldsymbol{x}_i; \boldsymbol{\theta}\right).$$

2. In least squares, we minimize the residual sum of squares

$$\text{RSS} = \sum_{i=1}^{n} w_i \left[y_i - m\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right)\right]^2,$$

for some weights $\{w_i\}$ (can be used if e.g., $\text{Var}\left(y_i \mid \boldsymbol{x}_i\right) = \sigma^2 / w_i$).

However, the closed form solution is generally not available.

General Problem

- Consider a general problem that, for a scalar-valued function $h(\boldsymbol{\beta})$, we need to find the solution of

$$\mathbf{0} = \frac{\partial h(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}.$$

- The solution is approximately the solution of

$$\mathbf{0} = \frac{\partial h(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \approx \frac{\partial h\left(\boldsymbol{\theta}^{(t)}\right)}{\partial \boldsymbol{\theta}} + \frac{\partial^2 h\left(\boldsymbol{\theta}^{(t)}\right)}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T}\left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)}\right)$$

for some known $\boldsymbol{\theta}^{(t)}$, which yields

$$\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)} - \left[\frac{\partial^2 h\left(\boldsymbol{\theta}^{(t)}\right)}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T}\right]^{-1} \frac{\partial h\left(\boldsymbol{\theta}^{(t)}\right)}{\partial \boldsymbol{\theta}},$$

if the Hessian matrix is invertible.

Newton-Raphson Method or Newton's Method

We can name a first guess of $\boldsymbol{\theta}$, $\boldsymbol{\theta}^{(0)}$, and update the parameter estimates using

$$\begin{aligned}
\boldsymbol{\theta}^{(1)} &= \boldsymbol{\theta}^{(0)} - \left[\frac{\partial^2 h\left(\boldsymbol{\theta}^{(0)}\right)}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T}\right]^{-1} \frac{\partial h\left(\boldsymbol{\theta}^{(0)}\right)}{\partial\boldsymbol{\theta}}, \\
\boldsymbol{\theta}^{(2)} &= \boldsymbol{\theta}^{(1)} - \left[\frac{\partial^2 h\left(\boldsymbol{\theta}^{(1)}\right)}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T}\right]^{-1} \frac{\partial h\left(\boldsymbol{\theta}^{(1)}\right)}{\partial\boldsymbol{\theta}}, \\
&\vdots
\end{aligned}$$

until $\frac{\partial h\left(\boldsymbol{\theta}^{(t+1)}\right)}{\partial\boldsymbol{\theta}}$ is sufficiently close to $\mathbf{0}$ or $\boldsymbol{\theta}^{(t+1)}$ and $\boldsymbol{\theta}^{(t)}$ are sufficiently close.

Newton-Gauss Method for Nonlinear Least Squares

Suppose that

$$m\left(\boldsymbol{x}_i, \boldsymbol{\theta}\right) \approx m\left(\boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right) + \left[\frac{\partial m\left(\boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right)}{\partial \boldsymbol{\theta}}\right]^T \left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)}\right).$$

Then the residual sum of squares satisfies

$$\text{RSS} \approx \sum_{i=1}^{n} w_i \left\{ y_i - m\left(\boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right) - \left[\frac{\partial m\left(\boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right)}{\partial \boldsymbol{\theta}}\right]^T \left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)}\right) \right\}^2,$$

where $y_i - m\left(\boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right)$ is our new response and $\boldsymbol{\theta} - \boldsymbol{\theta}^{(t)}$ is our new parameter vector. Its minimizer has a closed form expression. We iterate until no more improvements can be made.

Inference

Under some regularity conditions, $\hat{\boldsymbol{\theta}} \mid \boldsymbol{X}$ is approximately

$$N\left(\boldsymbol{\theta},\ \sigma^2\left[\boldsymbol{U}^T\left(\boldsymbol{\theta}\right)\boldsymbol{W}\boldsymbol{U}\left(\boldsymbol{\theta}\right)\right]\right),$$

for large enough n.

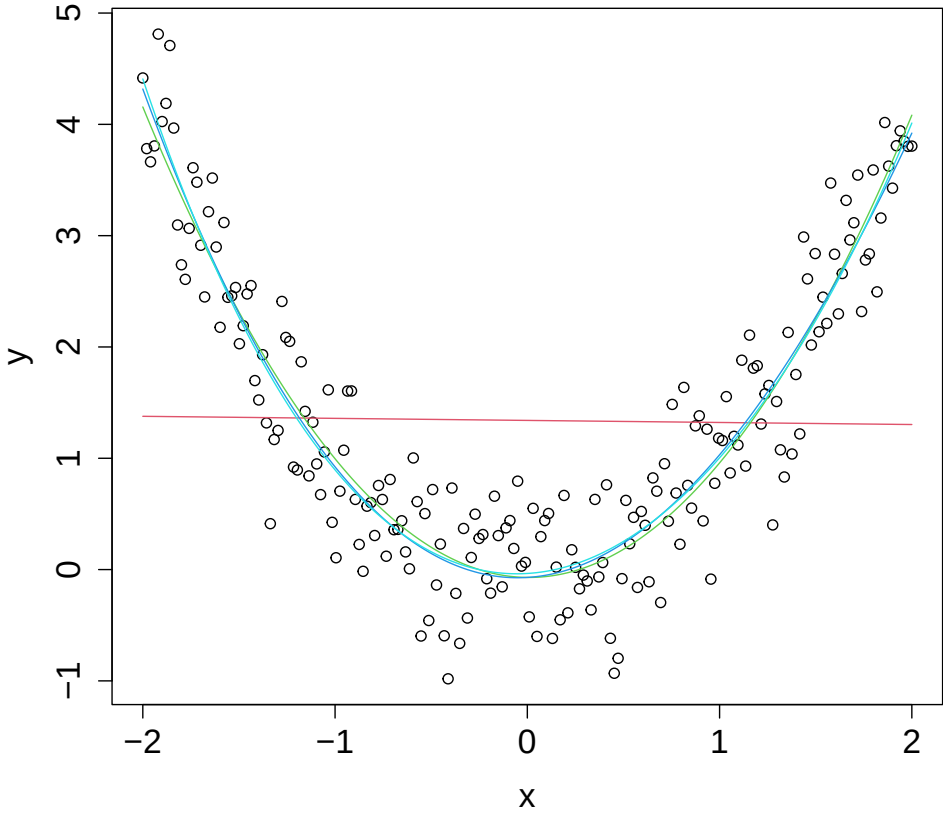
An estimator of σ^2 is

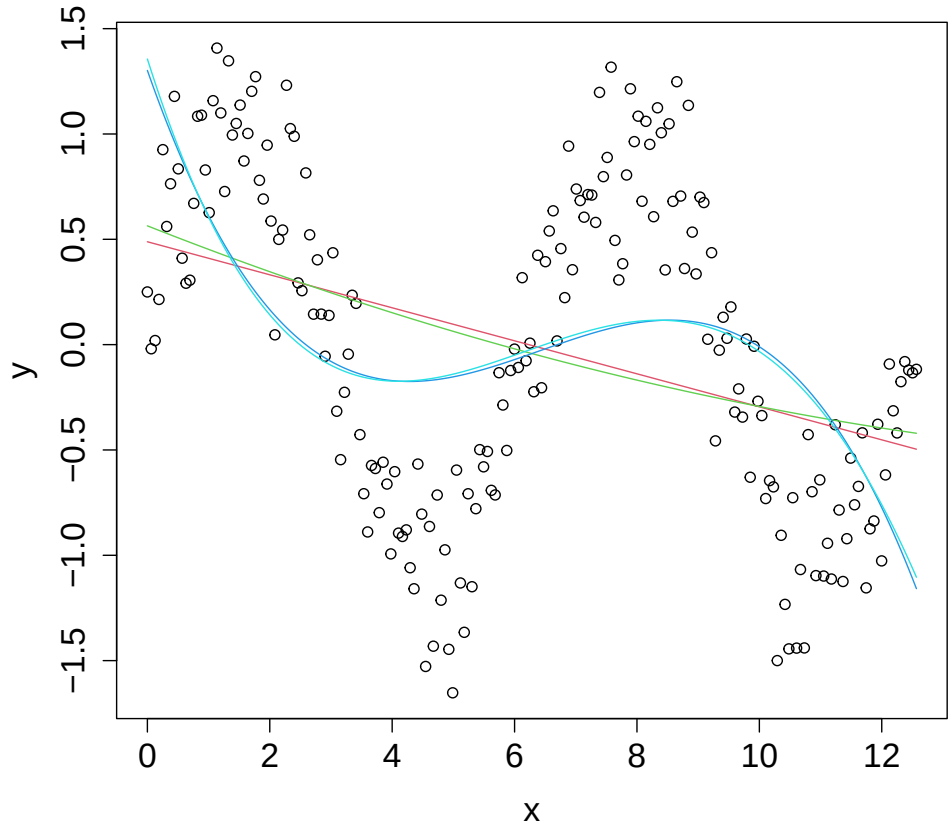
$$\hat{\sigma}^2 \quad = \quad \frac{\sum_{i=1}^{n} w_i\left[y_i - m\left(\boldsymbol{x}_i, \hat{\boldsymbol{\theta}}\right)\right]^2}{n-p},$$

where p is the number of parameters in the mean function.

Specify Mean Function

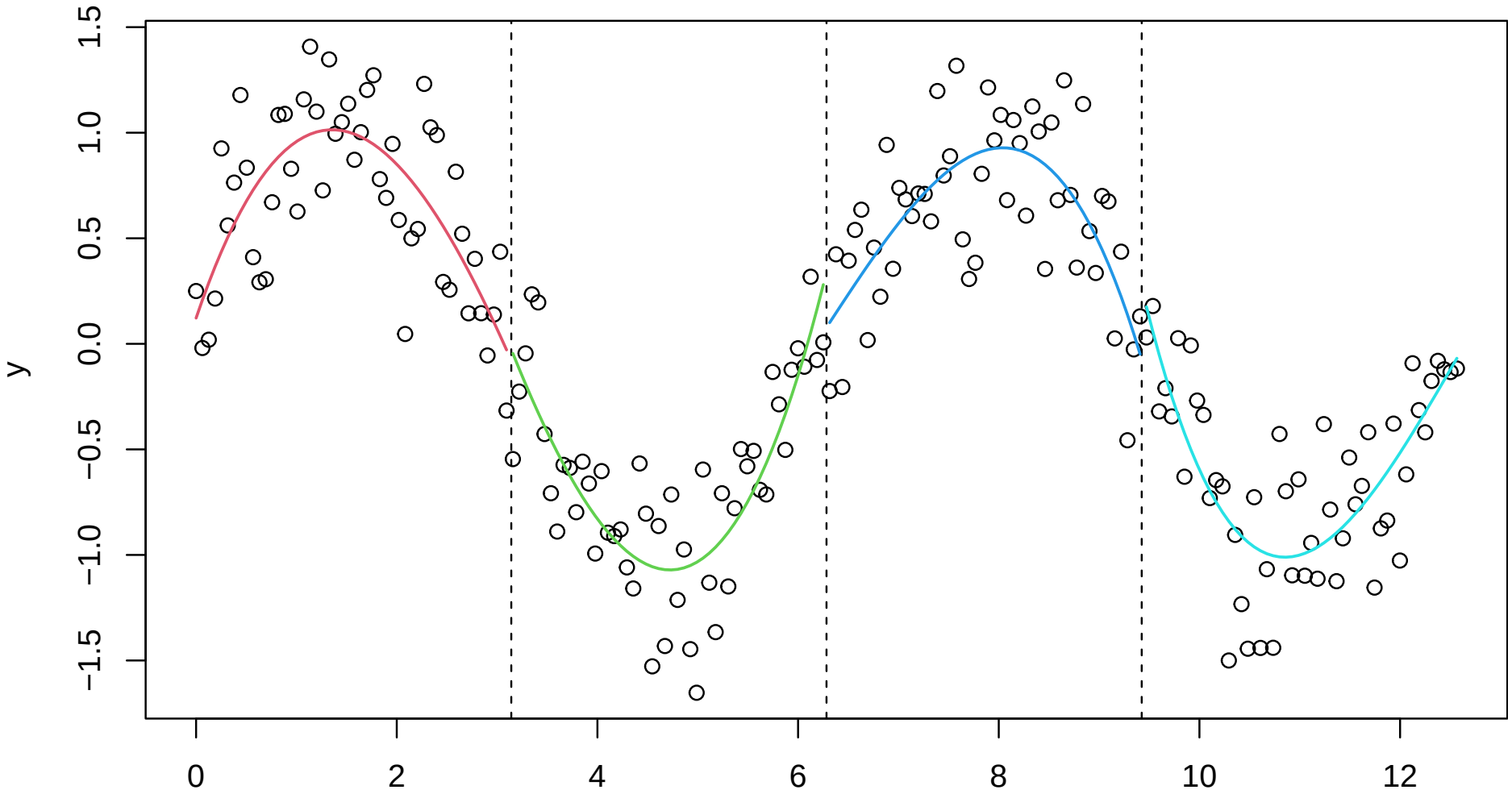
It is not always easy to specify the closed form expression of the mean function $m(\boldsymbol{x}, \boldsymbol{\theta})$.





Alternative: Piecewise Polynomial

We partition the data into several parts and fit polynomials to each part separately.



Piecewise polynomial

A piecewise polynomial is obtaind by

1. partitioning the range of x into contiguous intervals using the knots,
2. Between every two consecutive knots, fitting a polynomial model (in x) to the data points in the interval.

In practice, it is common to use the cubic polynomials (with degree 3 and order 4).

Example: Piecewise Linear

Consider two knots ξ_1 and ξ_2. We fit three linear models

$$\begin{aligned}
\text{for } x < \xi_1: &\quad \mathrm{E}_1(Y \mid X = x) = \beta_0^{(1)} + \beta_1^{(1)} x, \\
\text{for } \xi_1 \le x < \xi_2: &\quad \mathrm{E}_2(Y \mid X = x) = \beta_0^{(2)} + \beta_1^{(2)} x, \\
\text{for } x \ge \xi_2: &\quad \mathrm{E}_3(Y \mid X = x) = \beta_0^{(3)} + \beta_1^{(3)} x.
\end{aligned}$$

It is the same as

$$\begin{aligned}
\mathrm{E}(Y \mid X = x) = & \; 1(x < \xi_1)\left(\beta_0^{(1)} + \beta_1^{(1)} x\right) \\
& + 1(\xi_1 \le x < \xi_2)\left(\beta_0^{(2)} + \beta_1^{(2)} x\right) \\
& + 1(x \ge \xi_2)\left(\beta_0^{(3)} + \beta_1^{(3)} x\right).
\end{aligned}$$

However, the model fitted by piecewise polynomials are discontinuous at knots. We want our fitted model to be continuous.

Example: Piecewise Linear

In order to achieve continuity, we need to impose restrictions:

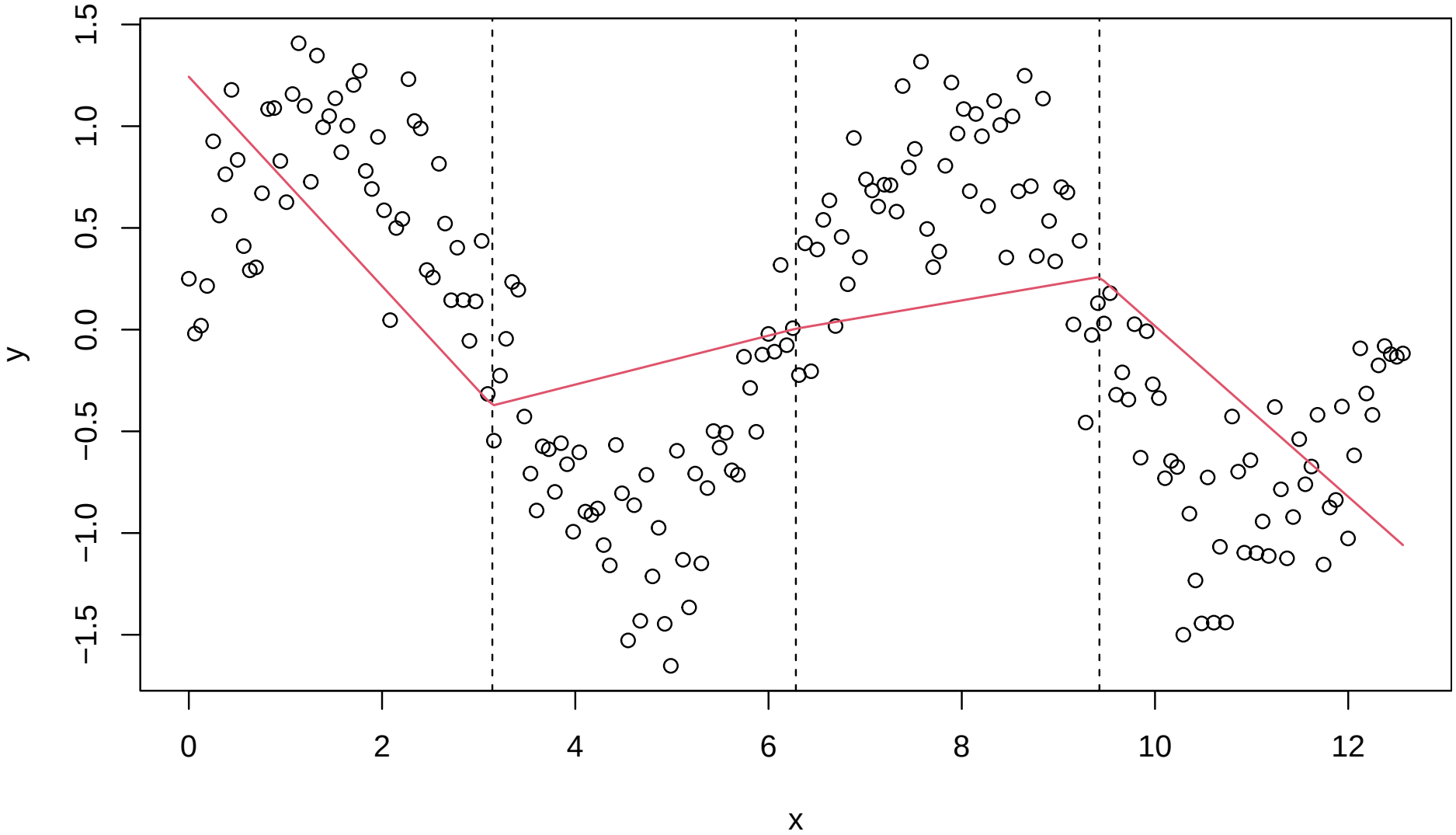
$$\begin{aligned} \beta_0^{(1)} + \beta_1^{(1)}\xi_1 &= \beta_0^{(2)} + \beta_1^{(2)}\xi_1, \\ \beta_0^{(2)} + \beta_1^{(2)}\xi_2 &= \beta_0^{(3)} + \beta_1^{(3)}\xi_2. \end{aligned}$$

Consequently, $\beta_0^{(1)}$ and $\beta_0^{(2)}$ are not free parameters and our model becomes

$$\begin{aligned} \mathrm{E}\left(Y \mid X = x\right) &= \left(\beta_0^{(3)} + \beta_1^{(3)}\xi_2 - \beta_1^{(2)}\xi_2 + \beta_1^{(2)}\xi_1 - \beta_1^{(1)}\xi_1\right) \\ &\quad + \beta_1^{(1)}x + \left(\beta_1^{(2)} - \beta_1^{(1)}\right)\max\left(0, x - \xi_1\right) \\ &\quad + \left(\beta_1^{(3)} - \beta_1^{(2)}\right)\max\left(0, x - \xi_2\right). \end{aligned}$$

It is the same as regress Y on the intercept, x, $\max\left(0, x - \xi_1\right)$ and $\max\left(0, x - \xi_2\right)$.

Example: Piecewise Linear



Example: Piecewise Cubic Polynomial

Consider two knots ξ_1 and ξ_2. We fit three cubic polynomials

$$\begin{aligned}
\text{for } x < \xi_1: &\quad \mathrm{E}_1(Y \mid X = x) = \beta_0^{(1)} + \beta_1^{(1)} x + \beta_2^{(1)} x^2 + \beta_3^{(1)} x^3, \\
\text{for } \xi_1 \le x < \xi_2: &\quad \mathrm{E}_2(Y \mid X = x) = \beta_0^{(2)} + \beta_1^{(2)} x + \beta_2^{(2)} x^2 + \beta_3^{(2)} x^3, \\
\text{for } x \ge \xi_2: &\quad \mathrm{E}_3(Y \mid X = x) = \beta_0^{(3)} + \beta_1^{(3)} x + \beta_2^{(3)} x^2 + \beta_3^{(3)} x^3.
\end{aligned}$$

It is the same as

$$\begin{aligned}
\mathrm{E}(Y \mid X = x) = &\ 1(x < \xi_1)\left(\beta_0^{(1)} + \beta_1^{(1)} x + \beta_2^{(1)} x^2 + \beta_3^{(1)} x^3\right) \\
&+1(\xi_1 \le x < \xi_2)\left(\beta_0^{(2)} + \beta_1^{(2)} x + \beta_2^{(2)} x^2 + \beta_3^{(2)} x^3\right) \\
&+1(x \ge \xi_2)\left(\beta_0^{(3)} + \beta_1^{(3)} x + \beta_2^{(3)} x^2 + \beta_3^{(3)} x^3\right).
\end{aligned}$$

However, we still cannot guarantee continuity. We want our fitted model to be continuous and smooth (sufficiently many continuous derivatives).

Cubic Spline

In order to produce a continuous and smooth fitted curve, we will impose the following constraints.

1. The fitted curve must be continuous everywhere, including the knots.
2. The fitted curve has continuous first and second order derivatives.

If piecewise cubic polynomials are used, then we have a cubic spline.

Cubic Spline: Restrictions

In order to achieve continuity, we need

$$\begin{aligned} \mathrm{E}_1\left(Y \mid X=\xi_1\right) &= \mathrm{E}_2\left(Y \mid X=\xi_1\right) \\ \mathrm{E}_2\left(Y \mid X=\xi_2\right) &= \mathrm{E}_3\left(Y \mid X=\xi_2\right). \end{aligned}$$

In order to achieve smoothness, we need,

$$\begin{aligned} \left.\frac{d\mathrm{E}_1\left(Y \mid X=x\right)}{dx}\right|_{x=\xi_1} &= \left.\frac{d\mathrm{E}_2\left(Y \mid X=x\right)}{dx}\right|_{x=\xi_1} \\ \left.\frac{d\mathrm{E}_2\left(Y \mid X=x\right)}{dx}\right|_{x=\xi_2} &= \left.\frac{d\mathrm{E}_3\left(Y \mid X=x\right)}{dx}\right|_{x=\xi_2} \\ \left.\frac{d^2\mathrm{E}_1\left(Y \mid X=x\right)}{dx^2}\right|_{x=\xi_1} &= \left.\frac{d^2\mathrm{E}_2\left(Y \mid X=x\right)}{dx^2}\right|_{x=\xi_1} \\ \left.\frac{d^2\mathrm{E}_2\left(Y \mid X=x\right)}{dx^2}\right|_{x=\xi_2} &= \left.\frac{d^2\mathrm{E}_3\left(Y \mid X=x\right)}{dx^2}\right|_{x=\xi_2}. \end{aligned}$$

Example: Cubic Spline

Consider two knots ξ_1 and ξ_2. With the continuity and smoothness requirements, we get

$$\begin{aligned}
\mathrm{E}\left(Y \mid X = x\right) &= \beta_0^{(3)} + \left(\beta_1^{(2)} - \beta_1^{(1)}\right)\xi_1^3 + \left(\beta_1^{(3)} - \beta_1^{(2)}\right)\xi_2^3 \\
&\quad + \beta_1^{(1)} x + \beta_2^{(1)} x^2 + \beta_3^{(1)} x^3 \\
&\quad + \left(\beta_3^{(2)} - \beta_3^{(1)}\right)\left[\max\left(0, x - \xi_1\right)\right]^3 \\
&\quad + \left(\beta_3^{(3)} - \beta_3^{(2)}\right)\left[\max\left(0, x - \xi_2\right)\right]^3 .
\end{aligned}$$

It is the same as regress Y on the intercept, x, x^2, x^3, $\left[\max\left(0, x - \xi_1\right)\right]^3$, and $\left[\max\left(0, x - \xi_2\right)\right]^3$.

Cubic Spline

Suppose that we have K knots (excluding the lower and upper limits of the range). Then, there are

$$4(K+1) - K - K - K \quad = \quad K+4$$

free parameters to be estimated in cubic spline. That is, a cubic spline with K knots has $K+4$ degrees of freedom. That is, the cubic spline is equivalent to

$$\mathrm{E}(Y \mid X = x) \quad = \quad \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{k=1}^{K} \beta_{k+3} \left[\max(0, x - \xi_k)\right]^3 .$$

In general, spline is a function defined by piecewise polynomials with continuity and smoothness conditions.

Still Not Necessarily Enough

- The fit of a cubic spline is often poor for very small or very large x values, due to the lack of information and large variation.
- We need to impose additional boundary constraints, i.e. the curve is linear in the region where X is smaller than or larger than the observed values. Then we have a natural spline.
- If we impose the boundary constraints to a cubic spline, we have a natural cubic spline.

More General Approach: Basis Expansion

From the above examples, we have

$$\mathrm{E}\left(Y \mid X = x\right) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{k=1}^{K} \beta_{k+3} \left[\max\left(0, x - \xi_k\right)\right]^3 .$$

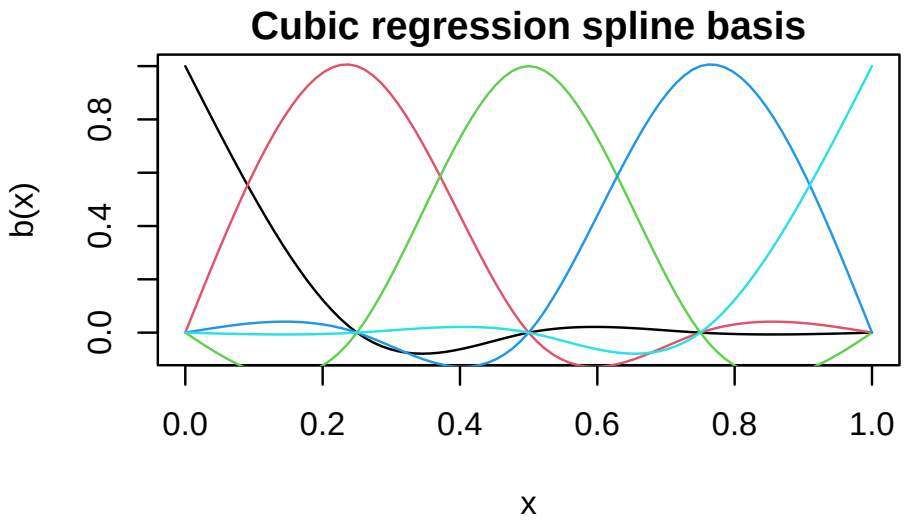
It is equivalent to using 1, x, x^2, x^3, and $\left[\max\left(0, x - \xi_k\right)\right]^3$ as basis functions and performing a "global" regression.

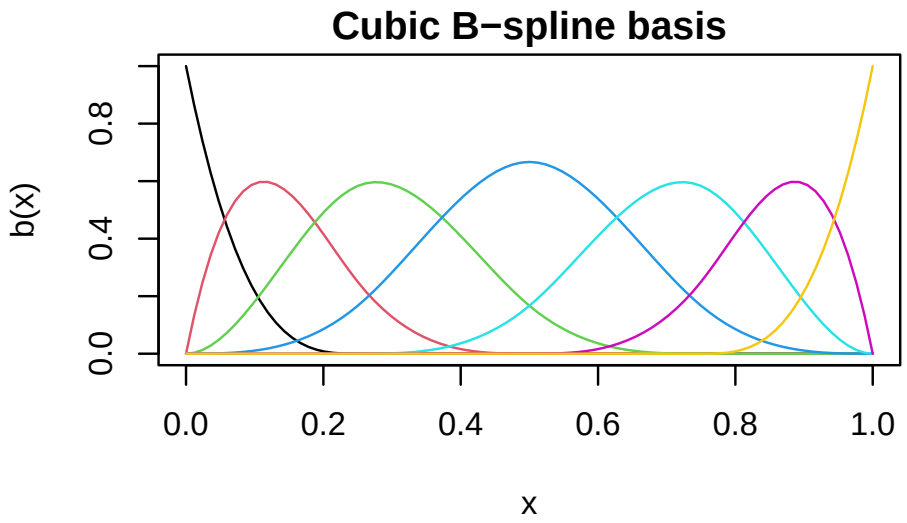
- We choose a series of functions $\{b_k(x)\}$ and use global data to fit

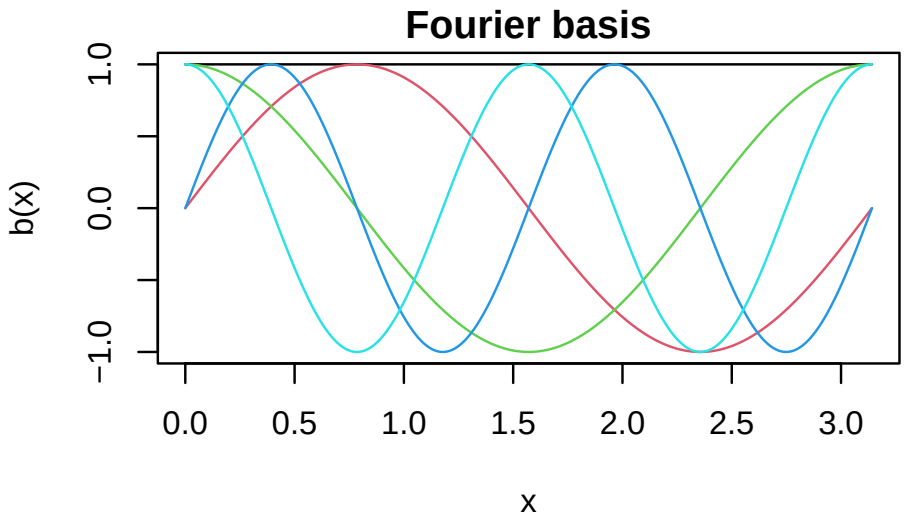
$$\mathrm{E}\left(Y \mid X = x\right) = \sum_{k} \beta_k b_k(x) .$$

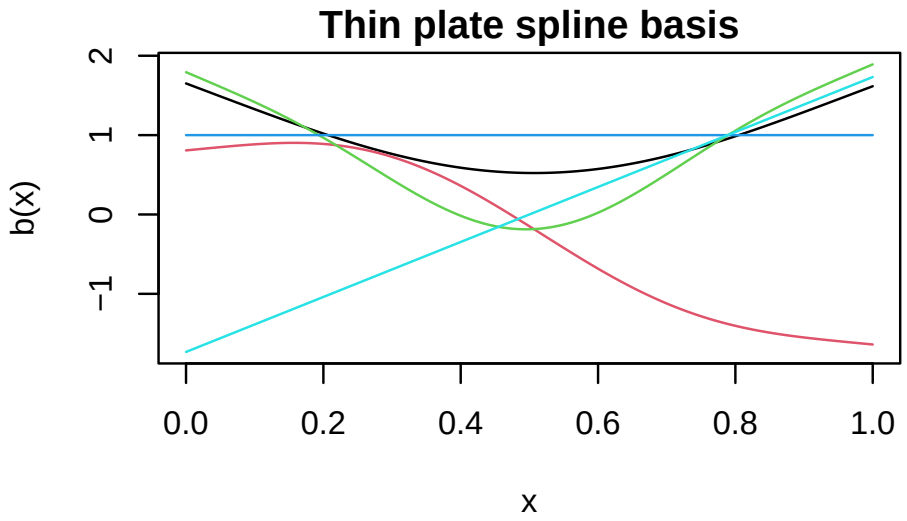
- $b_k(x)$'s are the basis functions and $\sum_k \beta_k b_k(x)$ is the basis expansion.
- The basis functions are fixed ahead of time (known forms) and the shape of $\mathrm{E}\left(Y \mid X = x\right)$ is controlled by varying β_k's.

Choice of Basis Functions

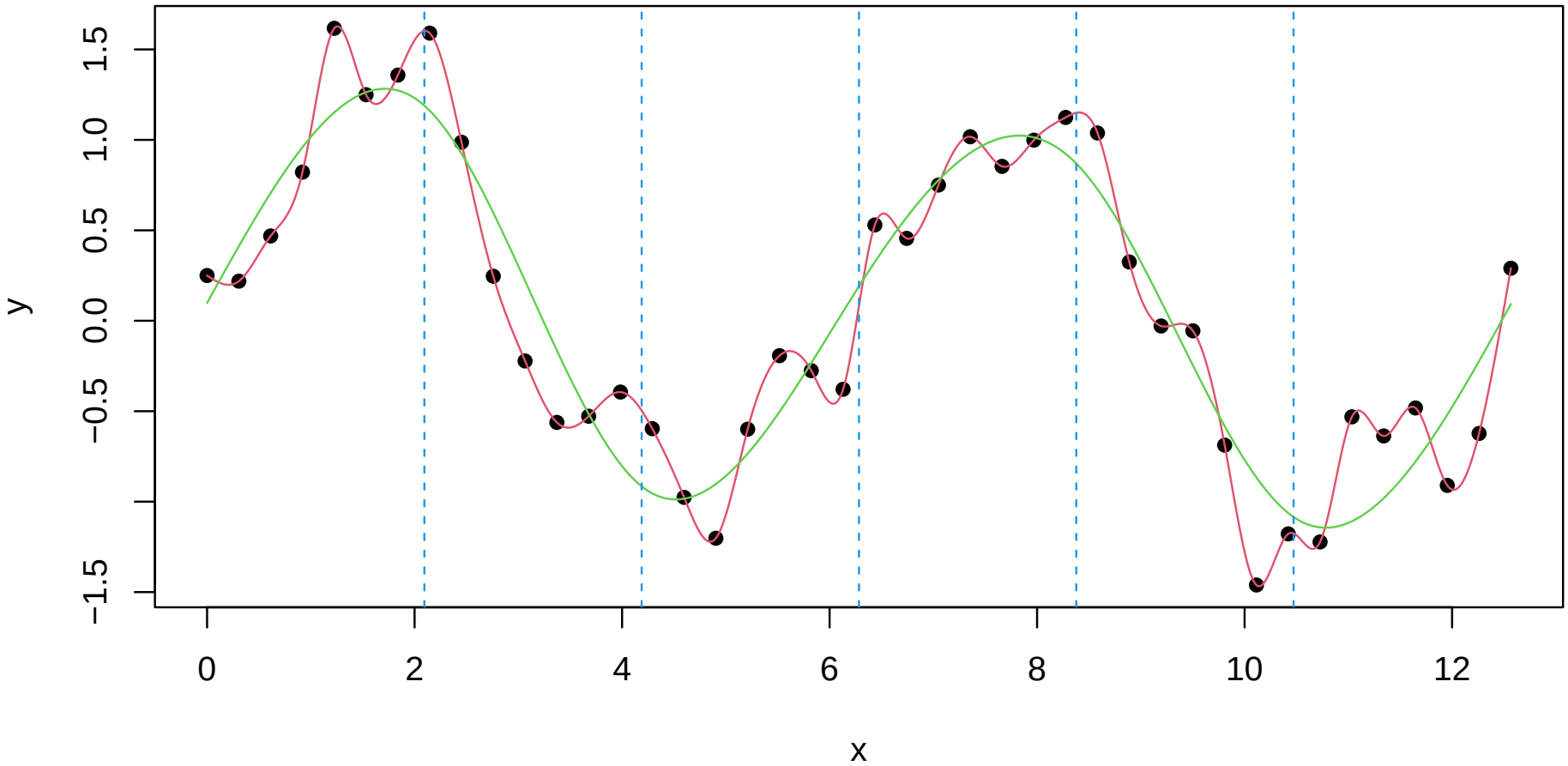








Overfitting



Wiggliness

- In practice, we often want to find a function $m(x)$ that minimizes

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

  to avoid overfitting, where the smoothness is controlled by

$$\int [m''(t)]^2 \, dt.$$

- In general, we want to minimize

$$\text{loss function} + \text{penalty for wiggliness}.$$

- Penalties other than the derivatives are also possible, such as

$$\sum (\beta_j - \beta_{j-1})^2 .$$

Ridge Regression Perspective

Suppose that

$$m(x) = \sum_k \beta_k b_k(x) = \boldsymbol{b}^T(x)\boldsymbol{\beta}.$$

Then

$$\sum_{i=1}^n [y_i - m(x_i)]^2 + \lambda \int [m''(t)]^2 dt$$
$$= \sum_{i=1}^n \left[y_i - \boldsymbol{b}^T(x)\boldsymbol{\beta}\right]^2 + \lambda \boldsymbol{\beta}^T \left[\int \frac{d^2\boldsymbol{b}(t)}{dt^2}\frac{d^2\boldsymbol{b}^T(t)}{dt^2} dt\right]\boldsymbol{\beta},$$

which is simply a ridge regression with regression coefficients $\boldsymbol{\beta}$.

Additive Model

- To account for non-linearity, we can assume

$$\mathrm{E}\left(Y_i \mid \boldsymbol{x}_i\right) \quad = \quad m\left(x_{1i}, x_{2i}, \cdots, x_{pi}\right),$$

  where the function form $m\left(\right)$ is estimated from the data.
- However, this formulation suffers from curse of dimensionality.
- In practice, we often consider the generalized additive model (GAM), such as

$$\begin{aligned} m\left(x_{1i}, x_{2i}, \cdots, x_{pi}\right) =& f_1\left(x_{1i}\right) + f_2\left(x_{2i}\right) + f_2\left(x_{2i}\right) + \cdots + f_p\left(x_{pi}\right), \\ m\left(x_{1i}, x_{2i}, \cdots, x_{pi}\right) =& f_1\left(x_{1i}\right) + f_{2,3}\left(x_{2i}, x_{3i}\right) + \cdots + f_p\left(x_{pi}\right), \end{aligned}$$

  where all $f\left(\right)$ have unknown forms and are estimated.
- Roughly speaking, GAM uses basis expansions to approximate unknown functions forms, and uses some penalty terms to control the wiggliness.

Kernel Regression

Suppose that we want to model $\mathrm{E}(Y \mid X = x)$.

- The observed $\{x_i\}$ that are close to x should carry more information about Y than $\{x_i\}$ that are far away.
- More informative $\{x_i\}$ should be given higher weights.

The kernel regression estimator is

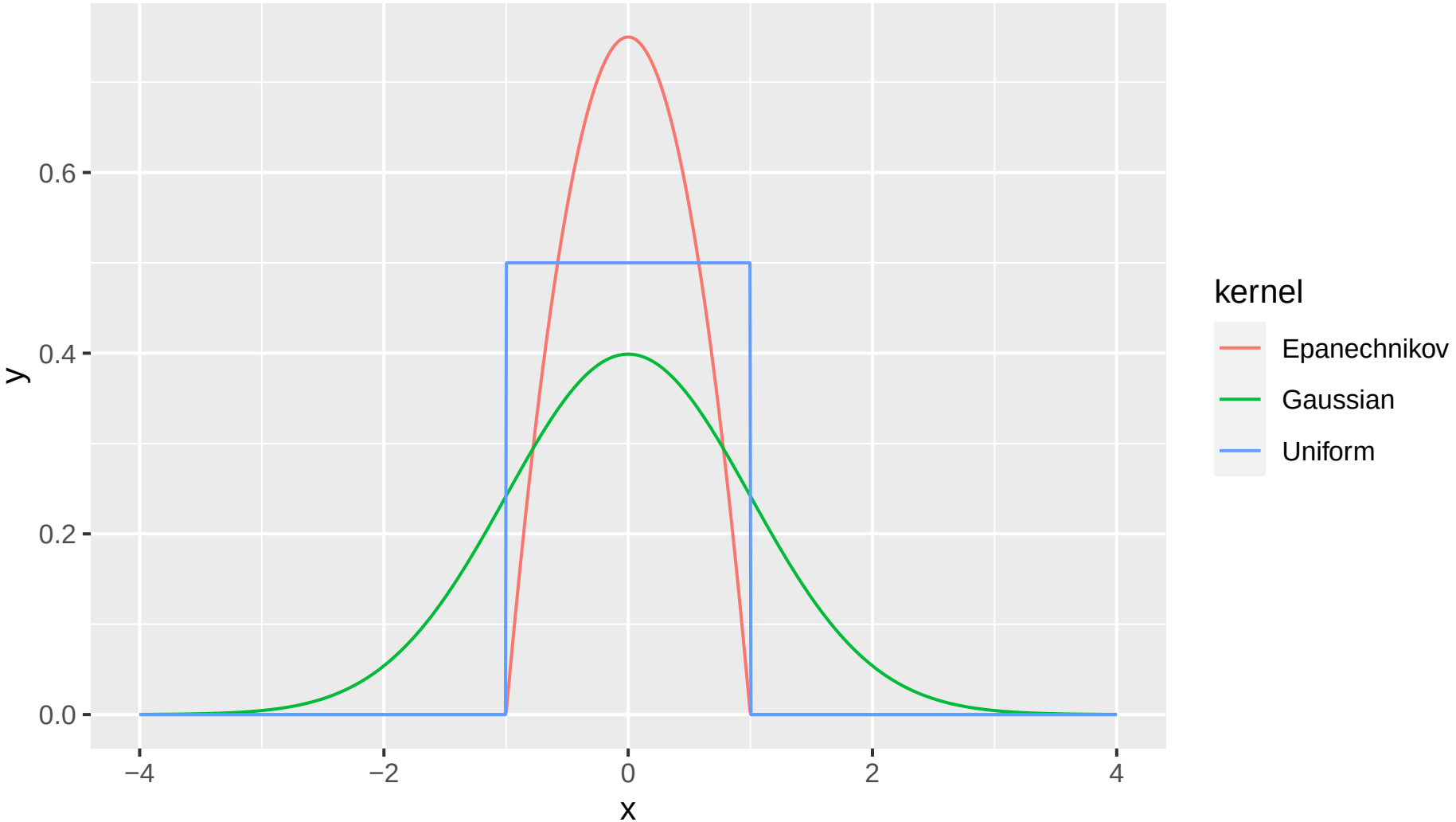
$$\hat{m}(x) = \frac{\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right) y_i}{\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right)},$$

where $K(\cdot)$ is a kernel, and h is a bandwidth. This estimator is called the Nadaraya-Watson estimator.

- A large h typically means a large bias and a small variance.
- A small h typically means a small bias and a high variance.

Kernel Function for Kernel Regression

A non-negative function $K(x)$ is a kernel function in kernel regression if $\int K(x)\,dx = 1$ and $K(x) = K(-x)$.



Local Regression

The local regression also weigh the "close points" more than the "remote points", but the regression function in a local neighborhood of x is approximated by a polynomial with estimated coefficients.

- Say, within a neighborhood of x_0,

$$m(x) \approx m(x_0) + m'(x_0)(x - x_0) + \frac{1}{2}m''(x_0)(x - x_0)^2.$$

- For fixed x, the local regression estimator minimizes

$$\sum_{i=1}^{n} K\left(\frac{x_i - x}{h}\right)\left[y_i - \beta_0 - \beta_1(x_i - x) - \beta_2(x_i - x)^2\right]^2.$$

- The estimated model is $\hat{m}(x) = \hat{\beta}_0$.

Limitation

A limitation of classic parametric regression is that we need to specify the form of the covariates $\boldsymbol{x}$ in $\boldsymbol{x}^T\boldsymbol{\beta}$ ourselves.

The gradient of the log-likelihood function is

$$\frac{\partial \text{RSS}}{\partial \beta_k} = \sum_{i=1}^{n} \left(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta}\right) x_{ik}.$$

If the gradient descend algorithm is used to update $\boldsymbol{\beta}$, the initial guess $\boldsymbol{\beta}^{(0)} = \mathbf{0}$ of $\boldsymbol{\beta}$ is improved by

$$\begin{aligned}
\boldsymbol{\beta}^{(1)} &= \boldsymbol{\beta}^{(0)} - \sum_{i=1}^{n} y_i\boldsymbol{x}_i = -\sum_{i=1}^{n} y_i\boldsymbol{x}_i, \\
\boldsymbol{\beta}^{(2)} &= \boldsymbol{\beta}^{(1)} - \sum_{i=1}^{n} \left(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta}^{(1)}\right) \boldsymbol{x}_i,
\end{aligned}$$

until no updates can be made.

Euclidean Inner Product

- Hence, we can expect the final estimate of $\boldsymbol{\beta}$ to be a linear combination of $\boldsymbol{x}_i$, say

$$\hat{\boldsymbol{\beta}} = \sum_{i=1}^{n} d_i \boldsymbol{x}_i,$$

  for some coefficients $\{d_i\}$.
- For a new $\boldsymbol{x}_0$, the predicted value of Y is

$$\boldsymbol{x}_0^T \hat{\boldsymbol{\beta}} = \sum_{i=1}^{n} d_i \boldsymbol{x}_0^T \boldsymbol{x}_i = \sum_{i=1}^{n} d_i \langle \boldsymbol{x}_i, \boldsymbol{x}_0 \rangle,$$

  where $\langle \boldsymbol{x}_i, \boldsymbol{x}_0 \rangle$ denotes the Euclidean inner product.

New Features

If we are not satisfied with the results using $\boldsymbol{x}$, we can create a vector of new features $\boldsymbol{\delta}(\boldsymbol{x}_i)$ and fit a new model

$$\mathrm{E}(Y_i \mid \boldsymbol{x}_i) = \boldsymbol{\gamma}^T \boldsymbol{\delta}(\boldsymbol{x}_i).$$

For a new $\boldsymbol{x}_0$, the predicted value of Y is

$$\begin{aligned} \boldsymbol{\delta}^T(\boldsymbol{x}_0)\hat{\boldsymbol{\gamma}} &= \sum_{i=1}^{n} d_i^* \boldsymbol{\delta}^T(\boldsymbol{x}_0)\boldsymbol{\delta}(\boldsymbol{x}_i) \\ &= \sum_{i=1}^{n} d_i^* \langle \boldsymbol{\delta}(\boldsymbol{x}_0), \boldsymbol{\delta}(\boldsymbol{x}_i) \rangle, \end{aligned}$$

for some coefficients $\{d_i^*\}$.

Kernel Function and Kernel Matrix

A function $\kappa(\boldsymbol{x}, \boldsymbol{z})$ is a kernel function if

1. it is symmetric, $\kappa(\boldsymbol{x}, \boldsymbol{z}) = \kappa(\boldsymbol{z}, \boldsymbol{x})$,
2. the kernel matrix $\boldsymbol{K}$ with (i, j)th entry $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is positive semi-definite for all $\boldsymbol{x}_1$, ..., $\boldsymbol{x}_n$.

If $\kappa(\boldsymbol{x}, \boldsymbol{z})$ is a kernel function, then you must be able to find a function $\boldsymbol{\delta}()$ such that

$$\kappa(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{\delta}^T(\boldsymbol{x})\boldsymbol{\delta}(\boldsymbol{z}).$$

As a kernel function, we will have an eigen-decomposition

$$\kappa(\boldsymbol{x}, \boldsymbol{z}) = \sum_{m=1}^{\infty} \rho_m e_m(\boldsymbol{x}) e_m(\boldsymbol{z}),$$

for some eigenvalues ρ_k and eigenfunctions $e_m(\boldsymbol{x})$.

Kernel Trick

Kernel trick is a commonly used trick to create new features from your original observed features.

- If the new features are created using $\boldsymbol{\delta}(\boldsymbol{x})$, then the prediction is

$$\boldsymbol{\delta}^T(\boldsymbol{x}_0)\hat{\boldsymbol{\gamma}} = \sum_{i=1}^{n} d_i^* \langle \boldsymbol{\delta}(\boldsymbol{x}_0), \boldsymbol{\delta}(\boldsymbol{x}_i) \rangle .$$

- If κ is the corresponding kernel function, the prediction is equivalent to

$$\boldsymbol{\delta}^T(\boldsymbol{x}_0)\hat{\boldsymbol{\gamma}} = \sum_{i=1}^{n} d_i^* \kappa(\boldsymbol{x}_0, \boldsymbol{x}_i) .$$

- This means that we only need to choose the kernel function κ, if we need to create new features from the raw covariates $\boldsymbol{x}$.

Benefits of Kernel Trick

The eigen-decomposition implies that

$$\begin{aligned}\sum_{j=1}^{n} d_j^* \kappa(\boldsymbol{x}_0, \boldsymbol{x}_j) &= \sum_{j=1}^{n} d_j^* \sum_{m=1}^{\infty} \rho_m e_m(\boldsymbol{x}_0) e_m(\boldsymbol{x}_j) \\ &= \sum_{m=1}^{\infty} \underbrace{\left(\sum_{j=1}^{n} d_j^* \sqrt{\rho_m} e_m(\boldsymbol{x}_j) \right)}_{\text{our new coefficients } d_j^*} \underbrace{\sqrt{\rho_m} e_m(\boldsymbol{x}_0)}_{\text{our new feature } \phi_m(\boldsymbol{x})}\end{aligned}$$

That is, we are using the vector of new features $\boldsymbol{\delta}(\boldsymbol{x})$, possibly of infinite dimension, to model ϕ_j.

Reproducing Kernel Hilbert Space (RKHS) Regression

- The kernel trick means that we first choose a kernel function, compute the kernel matrix $\boldsymbol{K}$ using the data matrix $\boldsymbol{X}$, and use $\boldsymbol{K}$ as the data matrix to fit our logistic regression model.
- However, when the data matrix $\boldsymbol{X}$ is of dimension $n \times p$, then the kernel matrix $\boldsymbol{K}$ is $n \times n$ and we have n regression coefficients to estimate. We often cannot obtain a unique estimator.
- Hence, we often use penalized least squares: the coefficients can be estimated by minimizing

$$(\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\gamma})^T (\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\gamma}) + \lambda \boldsymbol{\gamma}^T \boldsymbol{K} \boldsymbol{\gamma},$$

where λ is the tuning parameter. It is often chosen by cross validation.

Multivariate Gaussian Distribution

The univariate normal distribution with mean μ and variance σ^2 has the probability density function

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}, \quad -\infty < x < \infty.$$

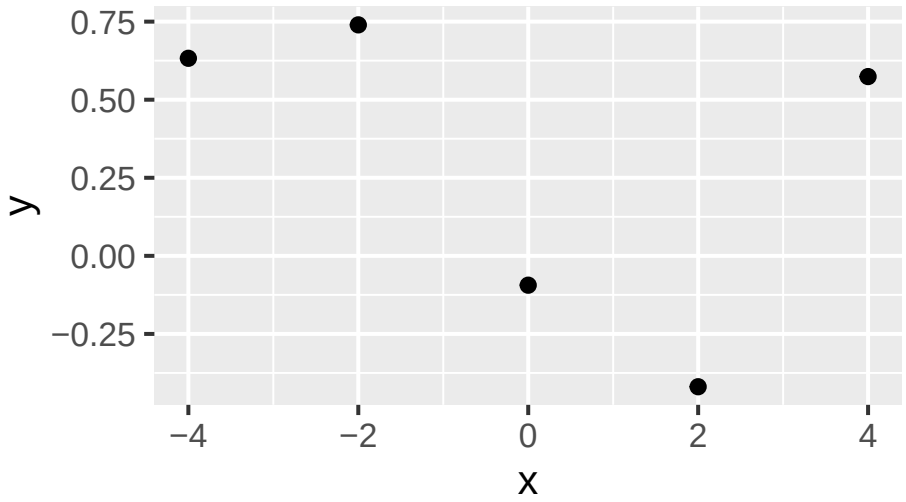
We often denote it by $X \sim N(\mu, \sigma^2)$. If $Z \sim N(0,1)$, then $X = \sigma Z + \mu \sim N(\mu, \sigma^2)$.

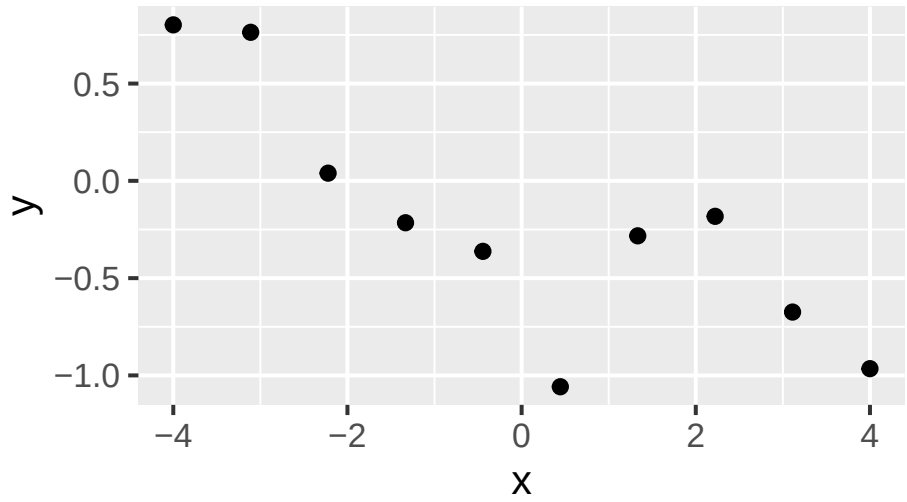
Let $\boldsymbol{Z} = \begin{bmatrix} Z_1 & Z_2 & \cdots & Z_p \end{bmatrix}^T$ be a random vector, each $Z_j \sim N(0,1)$, and Z_j is independent of Z_k for any $j \neq k$. Let $\boldsymbol{A}$ be a constant matrix and $\boldsymbol{\mu}$ be a constant vector. Then,

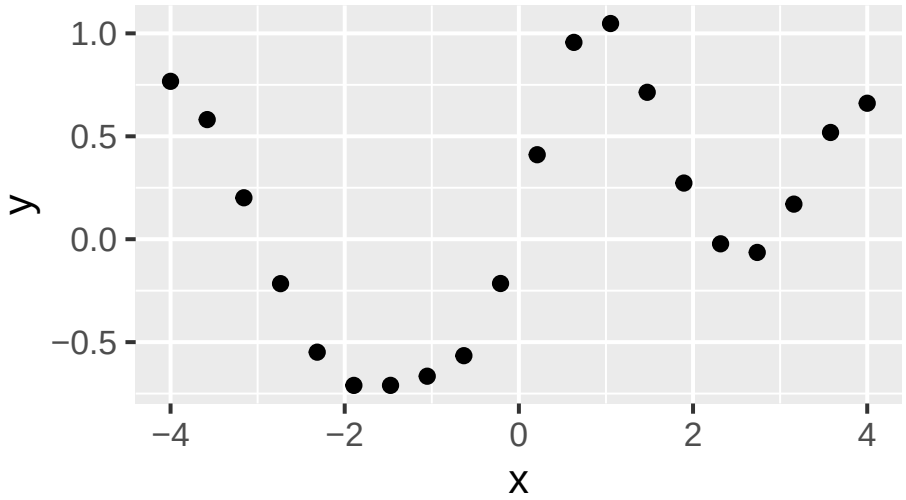
$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{Z} + \boldsymbol{\mu}$$

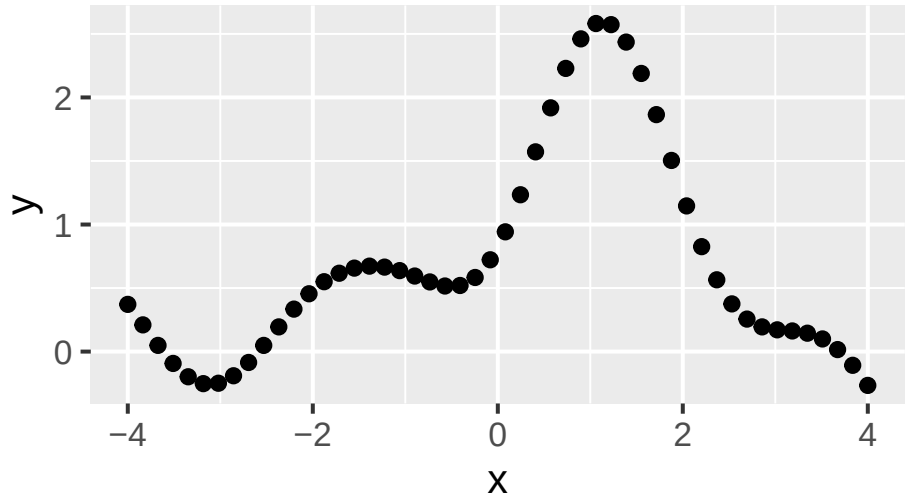
follows a $p-$dimensional multivariate normal distribution. It is denoted by $\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Gaussian Random Numbers









Conditional Distribution

Let $\begin{bmatrix} Y_0 \\ \boldsymbol{y} \end{bmatrix}$ be distributed as $N_n\left(\begin{bmatrix} \mu_0 \\ \boldsymbol{\mu} \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}_{11} & \boldsymbol{K}_{12} \\ \boldsymbol{K}_{21} & \boldsymbol{K}_{22} \end{bmatrix}\right)$. Then the conditional distribution of Y_0 given that $\boldsymbol{y}$, is

$$Y_0 \mid \boldsymbol{y} \quad \sim \quad N\left\{\boldsymbol{\mu}_0 + \boldsymbol{K}_{12}\boldsymbol{K}_{22}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}),\ \boldsymbol{K}_{11} - \boldsymbol{K}_{12}\boldsymbol{K}_{22}^{-1}\boldsymbol{K}_{21}\right\},$$

provided that $\boldsymbol{K}_{22}$ is invertible.

Gaussian Process Regression

Suppose that we have observed $(\boldsymbol{y}, \boldsymbol{X})$ and that we want to predict the value of Y_0 at $\boldsymbol{x}_0$. For simplicity, we assume that $\boldsymbol{\mu} = \mathbf{0}$ and $\boldsymbol{\mu}_0 = \mathbf{0}$. Then, $Y_0 \mid \boldsymbol{y}$ is Gaussian with

$$\begin{aligned} \mathrm{E}\left[Y_0 \mid \boldsymbol{y}\right] &= \tau^2 \boldsymbol{K}_{12} \left(\tau^2 \boldsymbol{K}_{22} + \sigma^2 \boldsymbol{I}\right)^{-1} \boldsymbol{y}, \\ \mathrm{Var}\left[Y_0 \mid \boldsymbol{y}\right] &= \tau^2 \boldsymbol{K}_{11} - \left(\tau^2 \boldsymbol{K}_{12}\right) \left(\tau^2 \boldsymbol{K}_{22} + \sigma^2 \boldsymbol{I}\right)^{-1} \left(\tau^2 \boldsymbol{K}_{21}\right), \end{aligned}$$

where τ^2 is a tuning parameter, and σ^2 is the error variance (also a tuning parameter) in $Y = \mathrm{GP} + \mathrm{error}$.

1. The predicted value is $\hat{Y}_0 = \tau^2 \boldsymbol{K}_{12} \left(\tau^2 \boldsymbol{K}_{22} + \sigma^2 \boldsymbol{I}\right)^{-1} \boldsymbol{y}$.
2. The confidence interval is constructed by

$$\hat{Y}_0 \pm 1.96 \sqrt{K_{11} - \boldsymbol{K}_{12} \left(\boldsymbol{K}_{22} + \sigma^2 \boldsymbol{I}\right)^{-1} \boldsymbol{K}_{21}}.$$