Non parametric smoothing

(An introduction)

N. Torelli, G. Di Credico, V. Gioia

2023

University of Trieste

Nonlinear regression and scatterplot smoothing

Polynomial Regression

Step functions

Kernel Smoothing

Regression splines

Smoothing splines

Nonlinear regression and scatterplot smoothing

LINEAR MODELS (INCLUDING GLMS) RELY ON THE ASSUMPTION OF LINEARITY.
HOWEVER, IN PRACTICE:

1. LINEAR MODEL ARE OFTEN OVERSIMPLIFICATIONS OF COMPLEX RELATIONSHIPS AND THEY MAY FAIL TO CAPTURE MEANINGFUL PATTERNS IN DATA, ESPECIALLY IF THEY EXHIBIT NONLINEARITY.
2. NONLINEAR MODEL ALLOW FOR GREATER FLEXIBILITY BY RELAXING THE ASSUMPTION

=> A KEY TECHNIQUE IN NONLINEAR REGRESSION IS SCATTERPLOT SMOOTHING, WHERE THE AIM IS TO ESTIMATE A SMOOTH CURVE $f(x)$ THAT DESCRIBES THE RELATIONSHIP BETWEEN AN INDEPENDENT VARIABLE x AND A DEPENDENT VARIABLE y.

Intro: the limitations of linearity

- Models that are built upon the linear effects of predictors, such as **linear models** or **generalized linear models**, play a crucial and non-replaceable role in the applications of statistics.
- Linearity is always an approximation but in many cases it is simple, reasonable, and it leads to very sensible results.
- Yet, there are instances where linearity is too strong a limitation, preventing the development of realistic models.
- That's why **nonlinear models** are important in statistics.
- This set of slides is based on ch.7 of the book An Introduction to Statistical Learning by James, Wittem Hastie and Tibshirani (freely downloadable from *https://www.statlearning.com/*).

Classes of nonlinear models

- Whereas linear models (including also GLMs) are easy to characterize, nonlinear models may be of several different types.
- We **will not** consider here the case of models which are *nonlinear in the parameters*. They include **nonlinear regression models**, often based on some biological or physical model, but also **Neural networks** and their extension (such as the models used in *deep learning*). These models have their own peculiarities and would deserve a specific treatment.
- A case of great interest attains to models which are *nonlinear in the predictors*.
- They belong to the class of **semiparametric regression models** (often the term nonparametric is also used)

Semi-parametric regression models

- **Semi-parametric** regression modelling keeps the usual specification: $y = g(x_1, \ldots, x_{p-1}, \epsilon)$ but relaxes the assumption of linear combination of predictors, and replaces it with a much weaker assumption of a smooth g
- Pro's and con's
  - greater flexibility and potentially more accurate estimate of g
  - greater computation and sometimes more difficult-to-interpret results
- The more popular possible solutions are:
  - $\mapsto$ Polynomial Regression
  - $\mapsto$ Step functions
  - $\mapsto$ Kernel and local-polinomial smoothers
  - $\mapsto$ Regression and Smoothing splines
  - $\mapsto$ (Generalized) Additive models
  - $\mapsto$ Decision (regression) trees (and MARS)

Polynomial Regression

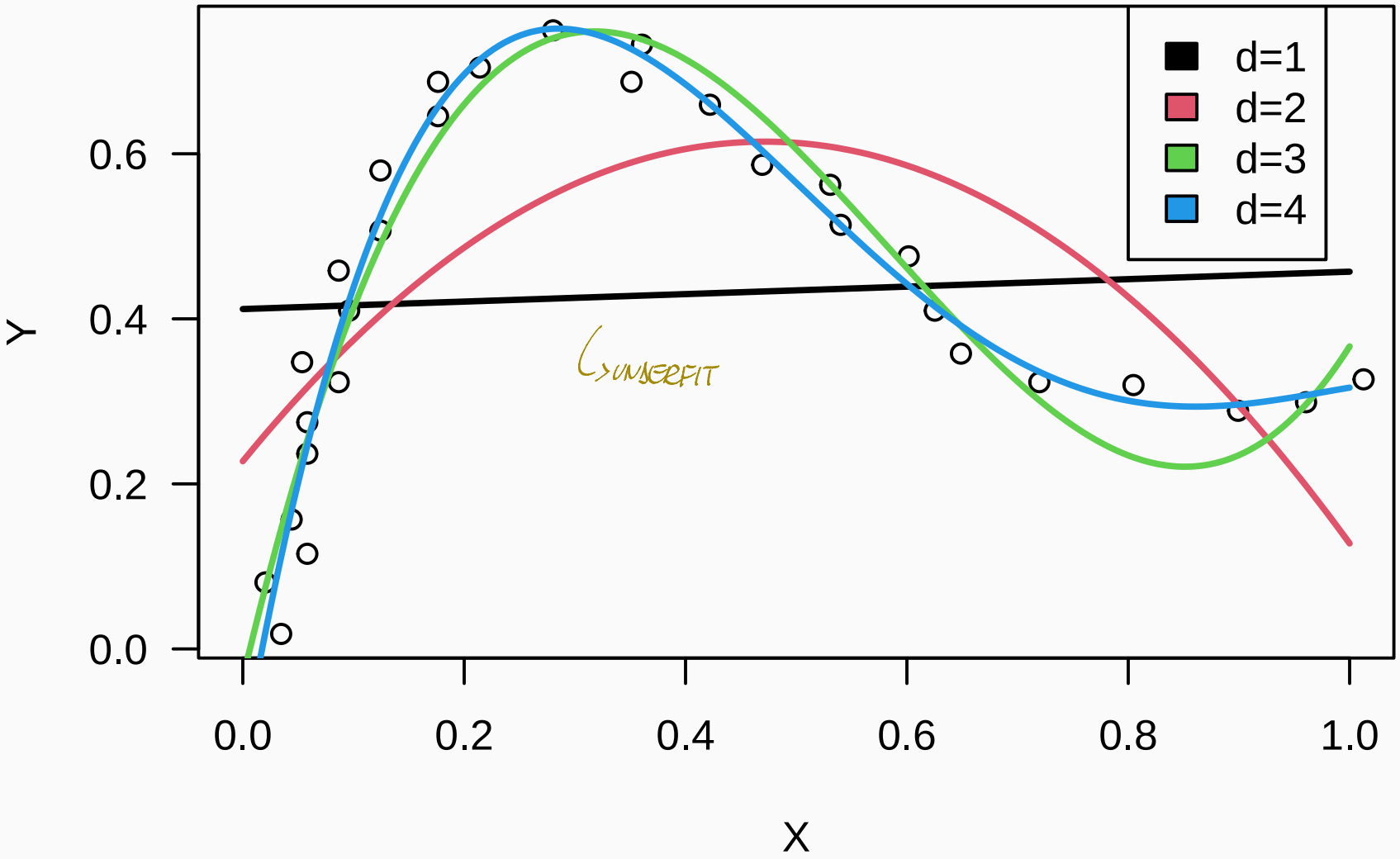
Polynomial Regression

- A simple, yet typical, way to extend linear regression for a single covariate (input variable) is to consider a polynomial of degree d within the classical linear model:

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

- For a modearately large d it fit to a scatterplot a curve which exibits evident non-linearities
- The model is linear in the coefficients $\beta_0, \beta_1, \ldots, \beta_d$ and they can be estimated by using ordinary least squares.
- It is actually a standard linear model with predictors $x_i, x_i^2, x_i^3, \ldots, x_i^d$.
- It is unusual to take d larger than 3 or 4, to avoid *overfitting* and possible shapes of the curve which are very strange within the support of x.

Example



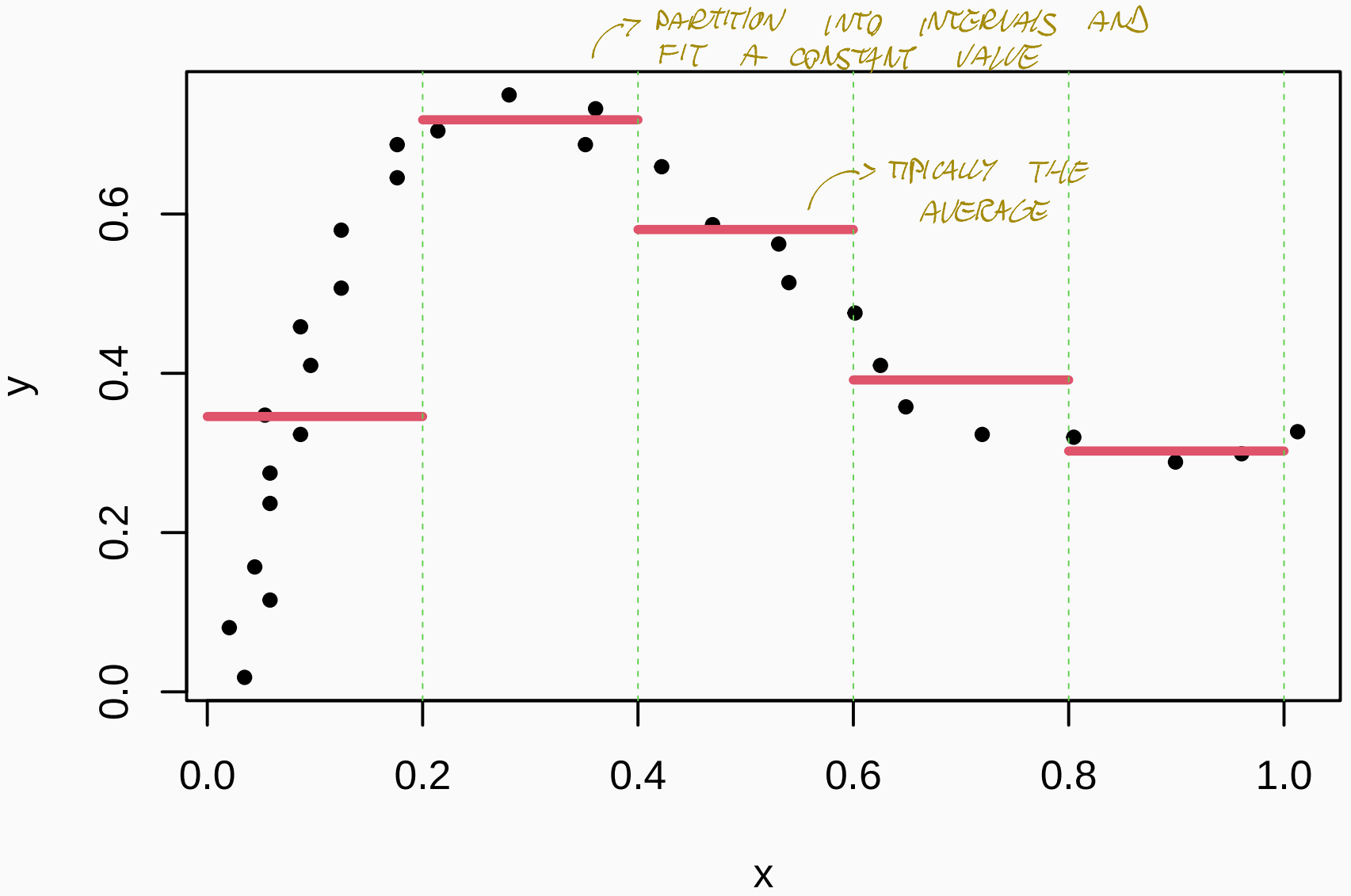
Step functions

Step functions

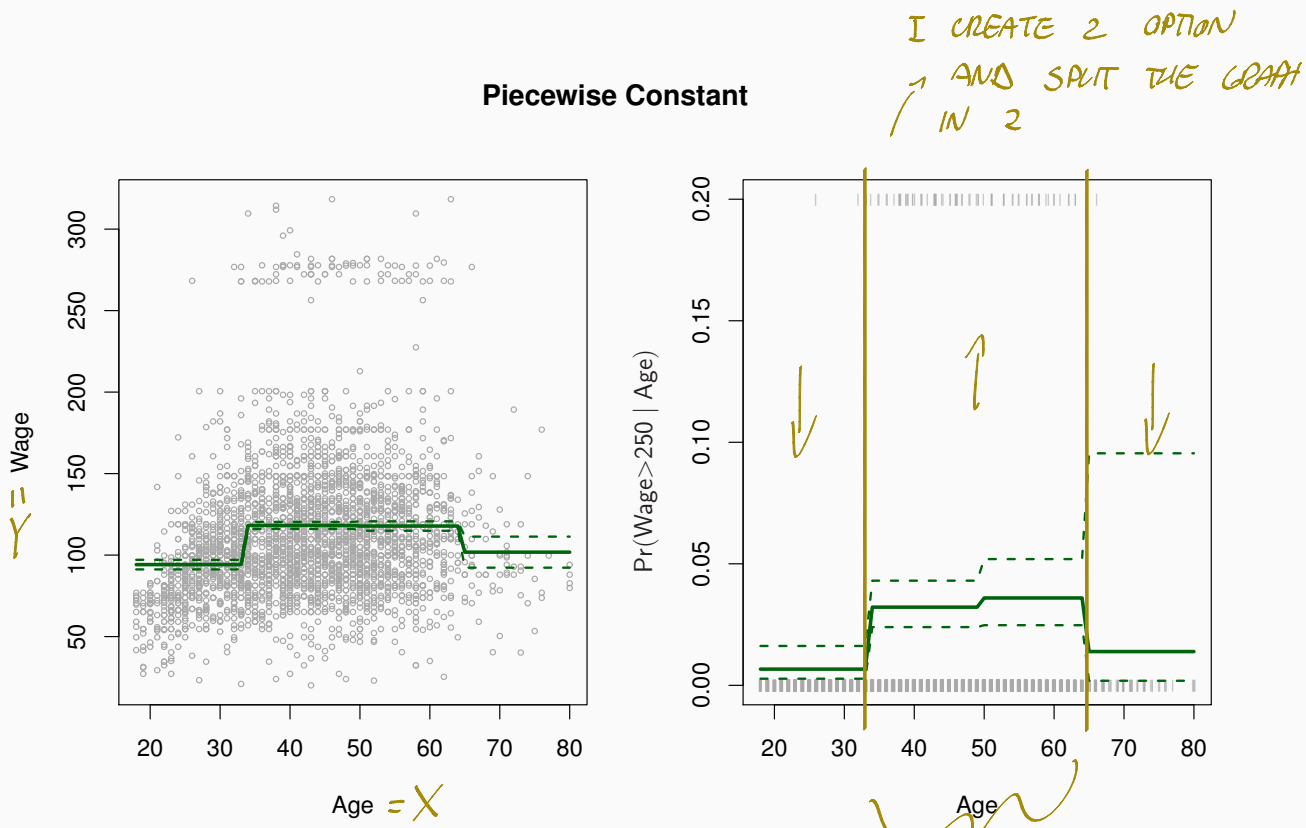
- USEFUL WHEN YOU SUSPECT THAT THE RELATIONSHIP BETWEEN THE PREDICTOR AND THE RESPONSE CHANGES ABRUPTLY AT CERTAIN POINTS, RATHER THEN CONTINUOS.

- A simple, yet effective, way to approximate a generic function $f(x)$ is to use a step function, that is, a piecewise constant function
- The values of the $X4$ variable is subdivided into K disjoint classes $[x_0 - x_1), [x_1 - x_2), \ldots [x_{K-1} - x_K]$
- In this case the value of the constant will be simply estimated by the average of the Y coordinates for those points within each interval (this is the least square solution)
- The level of the mean of the Y variable within each class k is then the predicted value for $X \in [x_{k-1} - k_k]$
- Fitting step functions is straightforward: it implies a linear regression model with a factor whose levels are the classes adopted to split the X variable.

Example


PARTITION INTO INTERVALS AND FIT A CONSTANT VALUE
TIPICALLY THE AVERAGE
y
0.0
0.2
0.4
0.6
x
0.0
0.2
0.4
0.6
0.8
1.0

Another example (with salary data)



The rigth panel represent a step function to estimate a model for a dicotomous variable.

Kernel Smoothing

Goals of smoothing

- The goal is again estimation of a regression function:

$$f(x) = E(y|x)$$

  through a model $y = f(x) + \epsilon$
- data on y and x are available $(x_i, y_i), \quad i = 1, 2, \ldots, n$
- a straightforward simple solution would be the following:
  - to predict y at $x = x_0$ gather all the pairs (x_i, y_i) having $x_i = x_0$, then
  - estimate $f(x_0)$ as the mean of the y_i values:

$$\hat{f}(x_0) = \text{Average}(y|x = x_0)$$

  - typically available data do not include observations having exactly $x_i = x_0$

THIS ALLOW THE ESTIMATE $\hat{f}(x)$ TO ADAPT LOCALLY

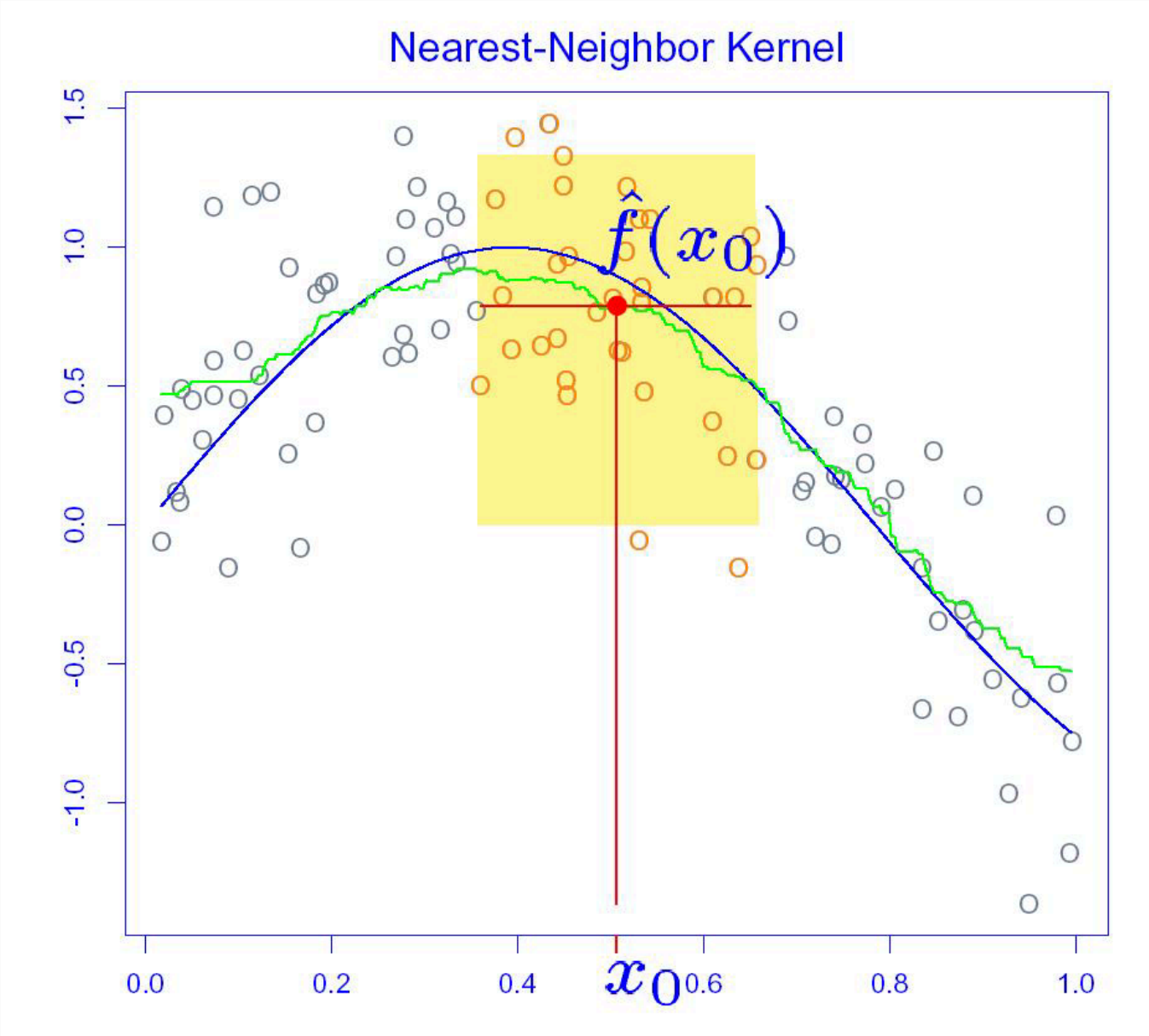
Nearest Neighbour Averaging

- an altrenative solution could be to estimate $E(y|x = x_0)$ by averaging those y_i whose x_i are in a neighbourhood of $x = x_0$
- e.g. define the neighbourhood to be the set of k observations having values x_i closest to x_0 in euclidean distance $||x_i - x_0||$
- (in the univariate case this is simply the absolute value $|x_i - x_0|$).

This method is called **nearest neighbour**

THE ESTIMATE BECOMES: $\hat{f}(x_0) = \frac{1}{K} \sum_{i \in \text{NEIGHBOUR}} y_i$

Nearest neigbour



Choosing k

- Small k implies that we use only the k points which are closer to target x (low bias), but averages when based on a small sample have high variance.
- Large k includes points far from x (high bias), but they have smaller variance.
- selecting a "good" value for k depends on how smooth the true function $f(x)$ is, and how noisy y is.
- One could try different values of k on a validation dataset, and pick the one with the best prediction performance.
- cross-validation can be also used.
- An alternative is to penalize the fitting criterion (in this case is least square).

- SMALL K => LOW BIAS BUT HIGH VARIANCE
- LARGE K => HIGH BIAS BUT LOW VARIANCE

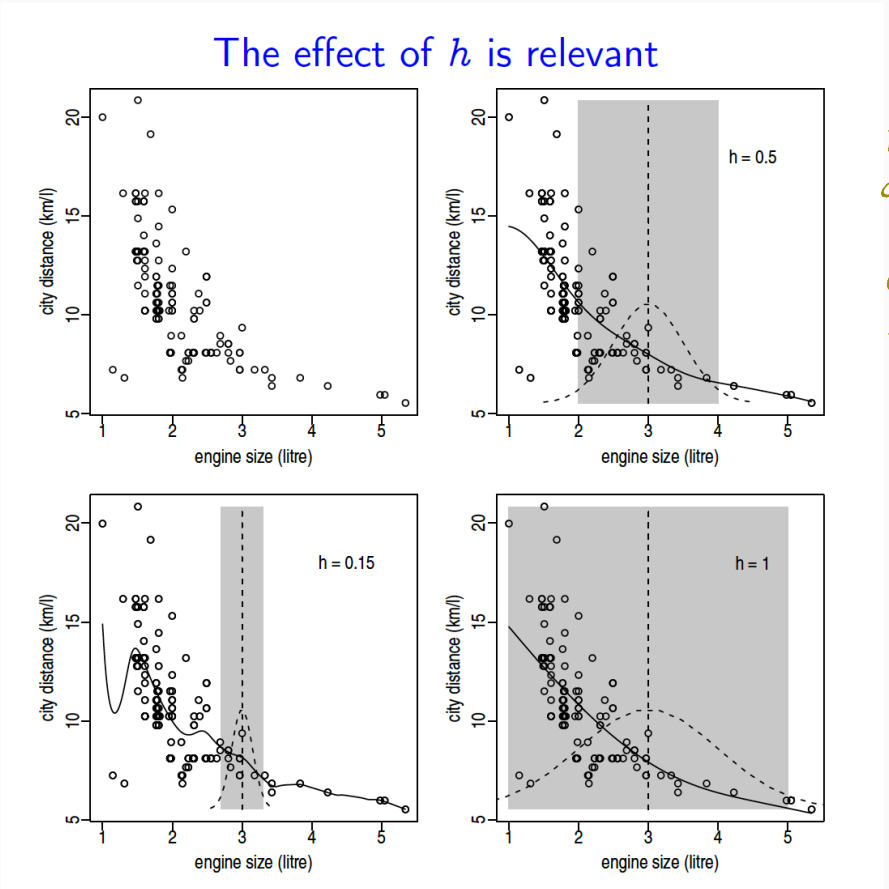
Local regression (kernel smoothers)

- A different solution could be to consider the weighted least squares by weighting observations x_i with their distance from x_0:

$$\min_{\alpha,\beta} \sum_{i=1}^{n} [y_i - \beta_0 - \beta_1(x_i - x_0)]^2 w_h(x_i - x_0)$$

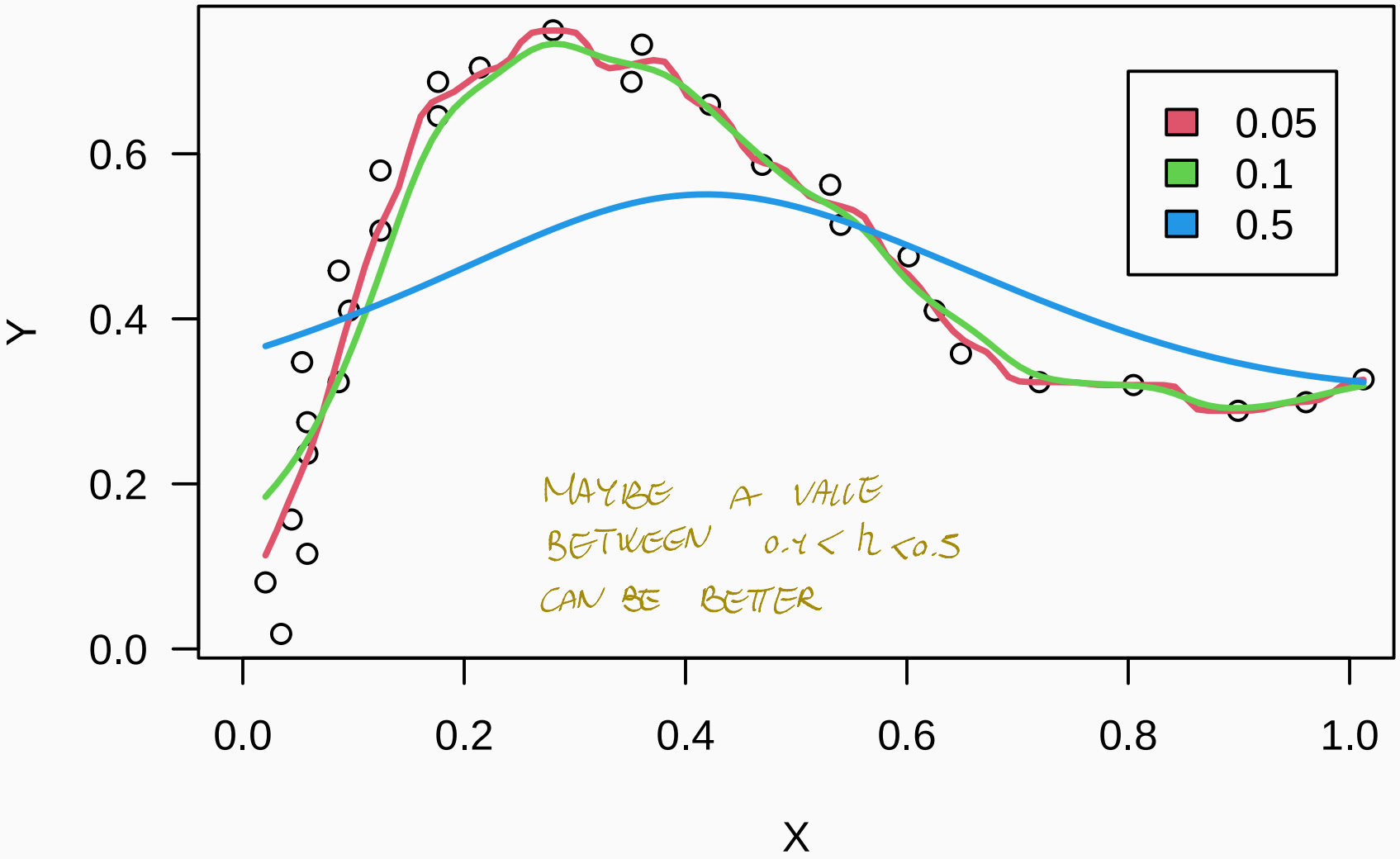
- $h > 0$ is a scale factor, called bandwidth or smoothing parameter, and
  - $w_h(\cdot)$ is a symmetric density function around 0, said **kernel** whose variance depends on h.
  - By varying x_0, we obtain a whole estimated curve $\hat{f}(x)$.
  - The most important ingredient is the **smoothing parameter** h, which regulates the smoothness of the curve, while the choice of the kernel w is less relevant.
  - w is often taken to be the density of the normal distribution $\mathcal{N}(0, h^2)$
- the `R` function `ksmooth` can be used to obatain a simple solution using local averages

The effect of h



FOR SMALL h THE ESTIMATE CURVE $\hat{f}(x)$ WILL CLOSELY FOLLOW THE DATA POINTS, CAPTURING SMALL-SCALE FLUCTUATIONS BUT POTENTIALLY OVERFITTING OTHERWISE, HIGH h MEANS THE CURVE WILL BE MORE SMOOTHER, IGNORING LOCAL VARIATIONS AND POTENTIALLY UNDERFITTING

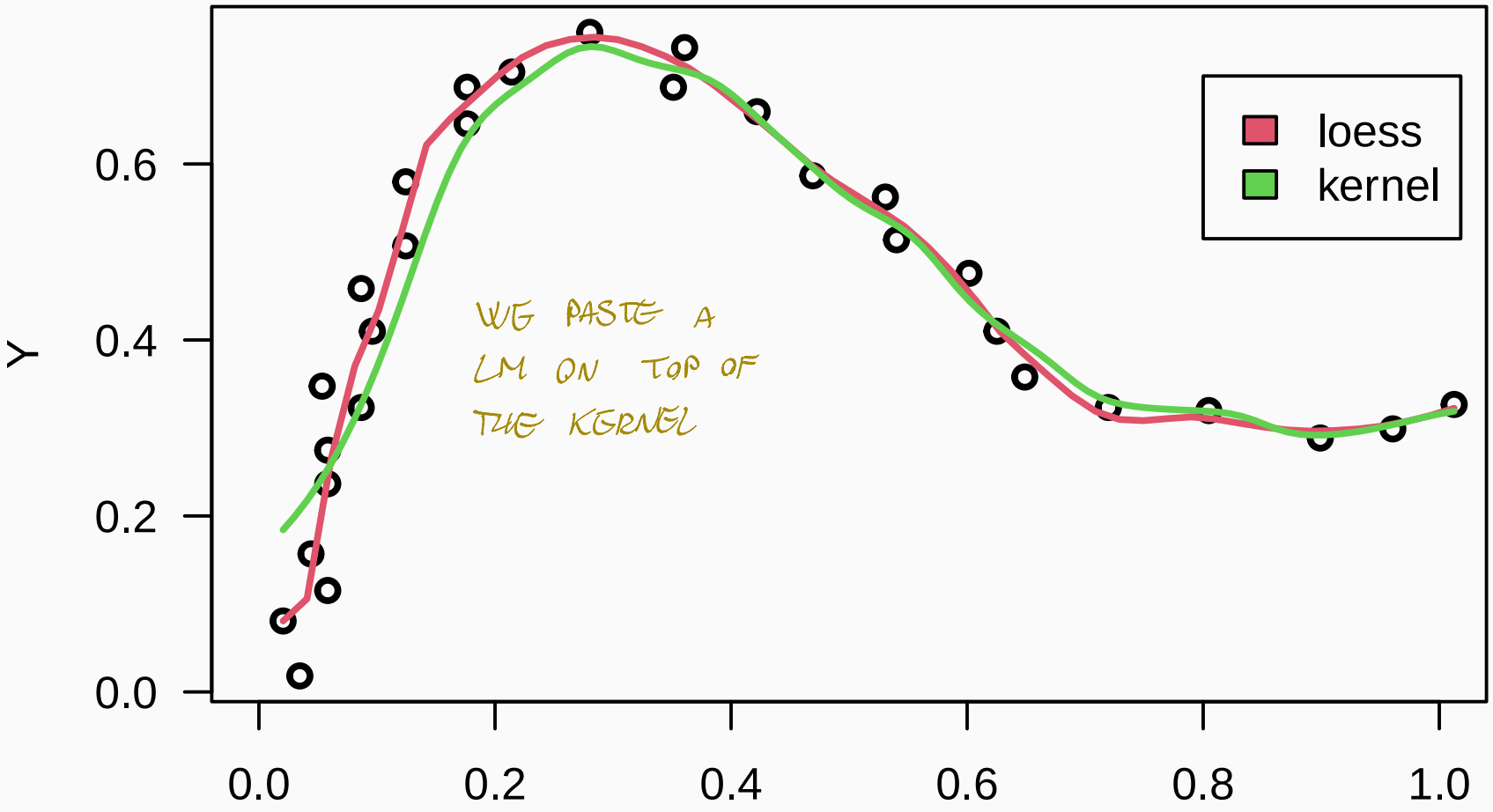
Some kernel smoothers



Variable bandwidth and the `loess`

- In many cases, there is an advantage in using a nonconstant bandwidth along the x-axis, according it to the level of sparseness of observed points
- variable bandwidth: it is reasonable to use larger values of h when x_i are more scattered
- Good idea! but how do we modify h?
- a popular solution is given by **loess**: it expresses the smoothing parameter by defining the fraction of effective observations for estimating $f(x)$ at a certain point x_0 on the x-axis;
- this fraction is kept constant
- this imply automatically a setting of the bandwidth related to the sparsity of data
- in addition, this idea is combined with the use of robust estimation
- `loess` is a very popular technique for smoothing a scatterplot (see the `R` function `loess`)

Loess vs kernel smoothinng



IT COMBINES THE FLEXIBILITY OF KERNEL SMOOTHING WITH THE ABILITY TO FIT MORE COMPLEX STRUCTURES

Regression splines

Regression splines extend polynomial regression by introducing knots that divide the data into intervals. Within each interval, a low-degree polynomial is fitted, and constraints are imposed at the knots to ensure smoothness. This approach balances flexibility (to capture nonlinearity) and simplicity (to avoid overfitting).

Basis function

- Polynomial and piecewise-constant regression models are in fact special cases of a **basis function** —> TRANSFORMATION OF X
- The idea is to have at hand a family of functions or transformations that can be applied to a variable X, $b_1(X), b_2(X), \ldots, b_K(X)$. Instead of a linear model in X let us fit the model:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \ldots + \beta_K b_K(x_i) + \epsilon_i,$$

- basis functions $b_1(\cdot), b_2(\cdot), \ldots, b_K(\cdot)$ are fixed and known.

E.G.
  - In polinomial regression $b_j(x_i) = x_i^j$
  - with step function $b_j(x_i) = I(c_j \leq x_i < c_{j+1})$.

- Least squares can be used in both cases to estimate the β parameters.

Piecewise regression

- Instead of fitting a high-degree polynomial over the entire range of X, **piecewise polynomial regression** involves fitting separate low-degree polynomials over disjoint regions of X separated by points called **knots**.
- For example, a piecewise cubic polynomial with a single knot at c has the form:

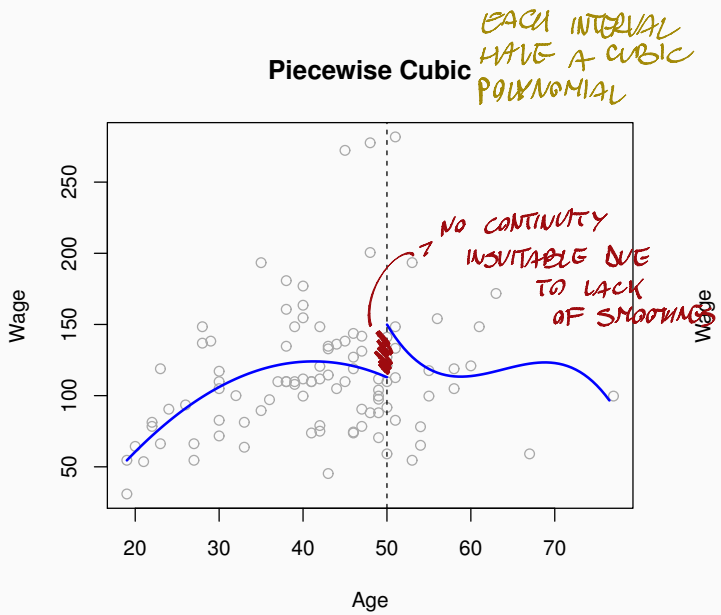
$$y_i = \begin{cases} \beta_{01} + \beta_{11} x_i + \beta_{21} x_i^2 + \beta_{31} x_i^3 + \epsilon_i & \text{if } x_i < c; \\ \beta_{02} + \beta_{12} x_i + \beta_{22} x_i^2 + \beta_{32} x_i^3 + \epsilon_i & \text{if } x_i \geq c. \end{cases}$$

- Note that 8 degrees of freedom are needed in fitting this model

Splines

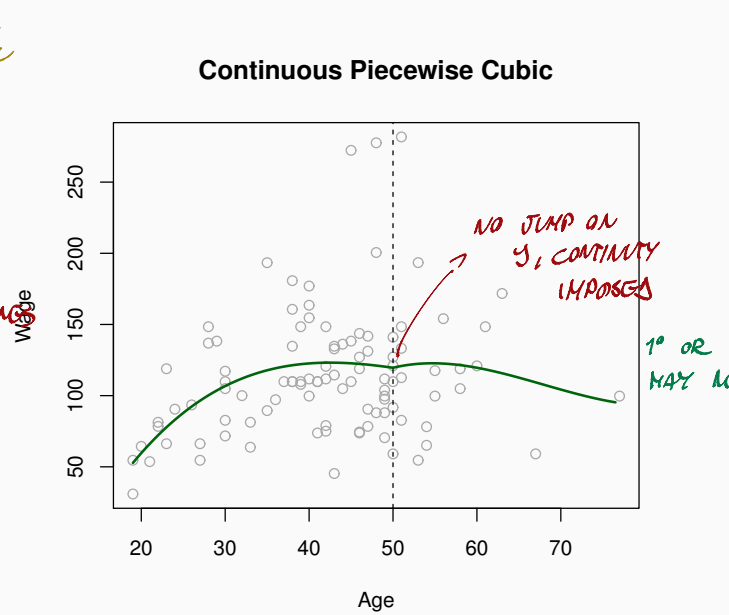
- Unlike polynomials, where flexibility is gained by using higher powers, splines introduce flexibility by increasing the number of knots keeping low the degree of the polynomials: this second option seems to be more feasible.
- With K knots, $K + 1$ distinct polynomials are defined (over each interval defined by the sequence of knots).
- Using polynomial with lower degree is possible, but one concern is continuity of the function in the knots
- Some contraints, such as continuity of the function in the knots (possibly also of the first and second derivatives) can be imposed.
- This leads to reduction of complexity of the curve and then frees up degrees of freedom.
- The cubic with one knot in the example seen before will need 5 degrees of freedom (continuity up to the second derivatives frees 3 degrees of freedom). A cubics pline with K knots needs $4 + K$ degrees of freedom
- Continuity can be also imposed when piecewise linear function are used.

Credit dataset



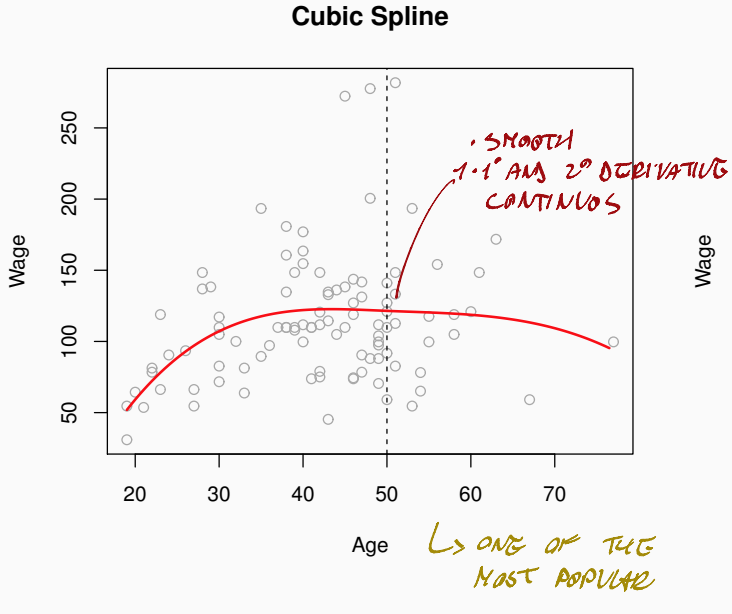
EACH INTERVAL HAVE A CUBIC POLYNOMIAL

NO CONTINUITY INSUITABLE DUE TO LACK OF SMOOTHNESS



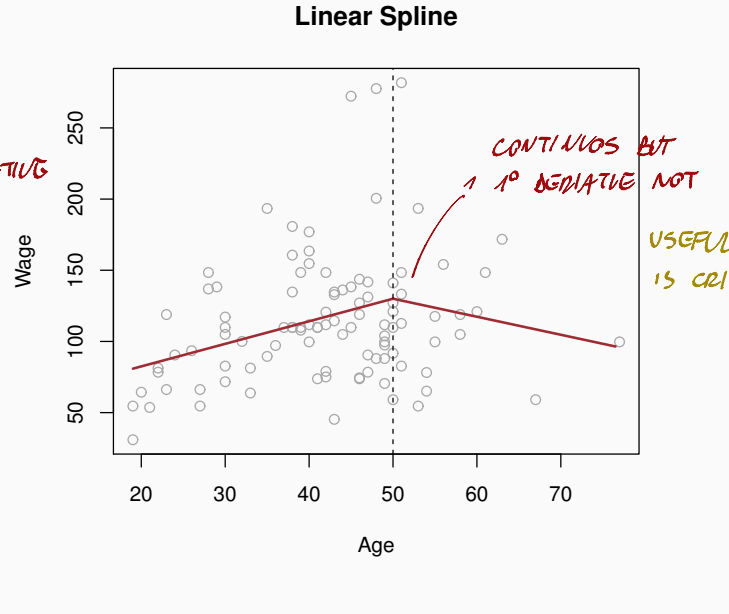
NO JUMP ON y, CONTINUITY IMPOSED

1° OR 2° DERIVATIVE MAY NOT BE CONTINUOS



SMOOTH 1-1° AND 2° DERIVATIVE CONTINUOS

↳ ONE OF THE MOST POPULAR



CONTINUOS BUT 1° DERIVATIVE NOT

USEFUL WHEN SIMPLICITY IS CRITICAL

Linear splines trough basis functions

a linear spline with knots in ξ_k, $k = 1, 2, \ldots, K$ is a piecewise linear function continuos in the knots

It can be represented as

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \ldots + \beta_{K+1} b_{K+1}(x_i) + \epsilon_i,$$

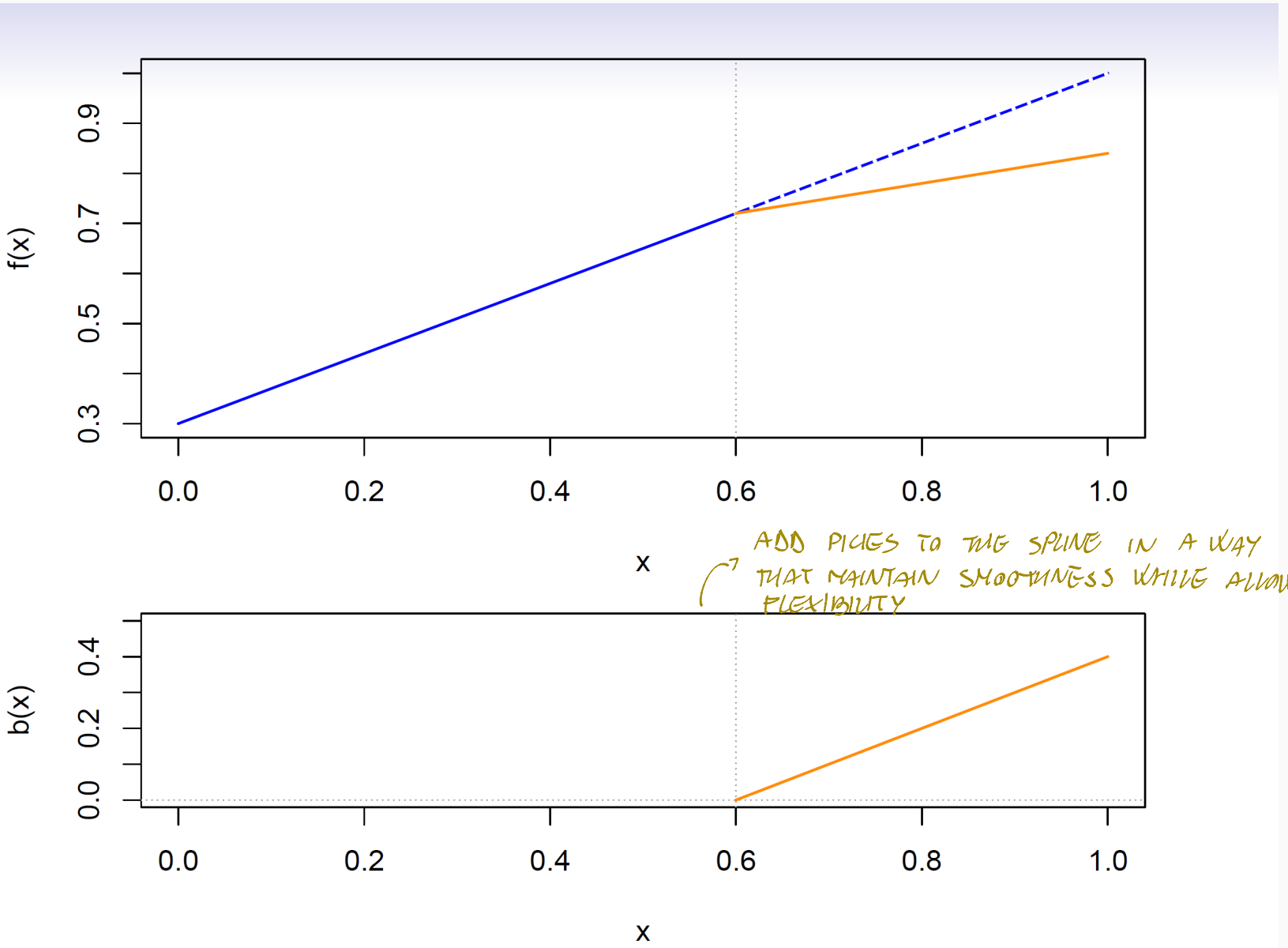
where b_k are *basis functions*:

$$\begin{aligned} b_1(x_i) &= x_i \\ b_{k+1}(x_i) &= (x_i - \xi_k)_+, \quad k = 1, \ldots, K. \end{aligned}$$

The symbol $(\cdot)_+$ denotes the *positive part*:

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

The positive part



Cubic splines through basis splines

A cubic spline with K knots in ξ_k, $k = 1, 2, \ldots, K$ is a piecewise cubic polinomial with continuos derivatives up to order two

It turns out that we can use the basis functions representation for a spline. A cubic spline with K knots can be modeled as:

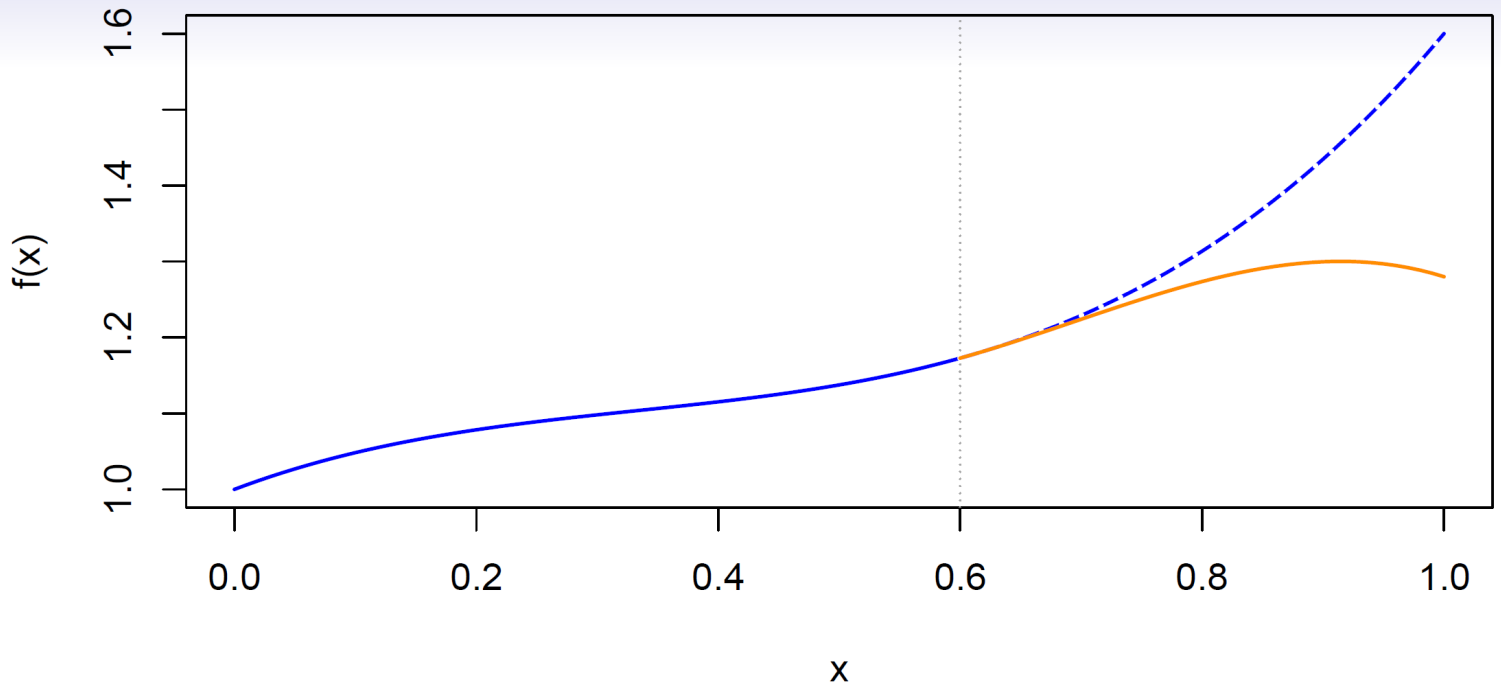
$$\begin{aligned} y_i = \ & \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \ldots + \beta_{K+3} b_{K+3}(x_i) + \epsilon_i, \\ b_1(x_i) &= x_i \\ b_2(x_i) &= x_i^2 \\ b_3(x_i) &= x_i^3 \\ b_{k+3}(x_i) &= (x_i - \xi_k)_+^3, \quad k = 1, \ldots, K, \end{aligned}$$

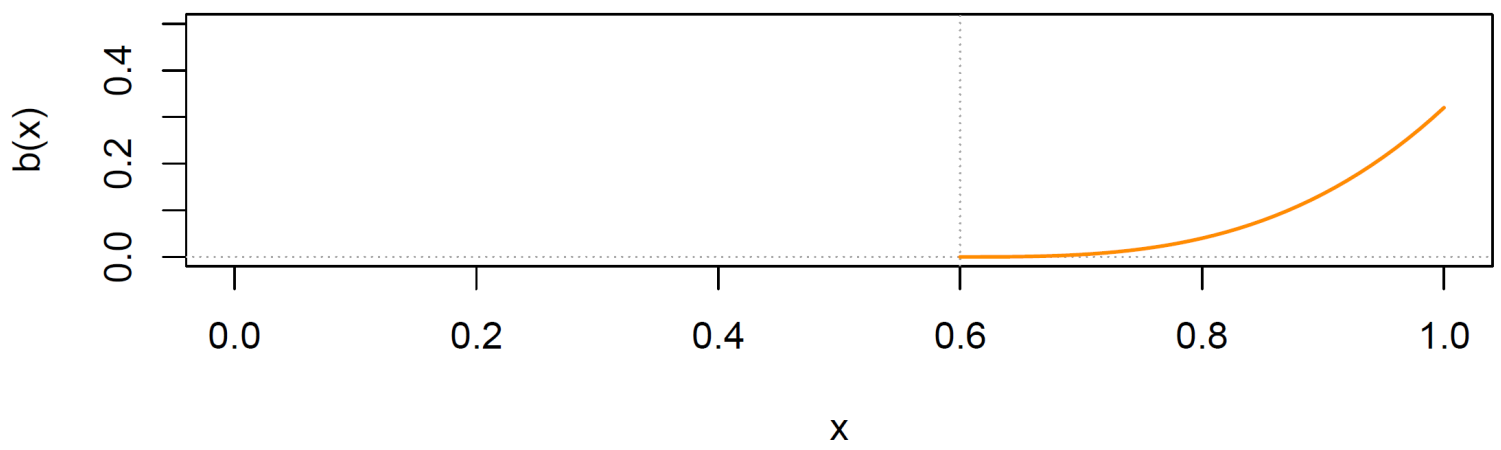
where a trucated power basis function is added for each knot. A truncated power basis function is defined as:

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

- To fit a cubic spline with K knots to a data set we can use least squares regression with an intercept and 3+K predictors, of the form $X, X^2, X^3, h(X, \xi_1), h(X, \xi_2), ..., h(X, \xi_K)$, where $\xi_1, \ldots \xi_K$ are the knots. This amounts to estimating a total of $K + 4$ regression coefficients.

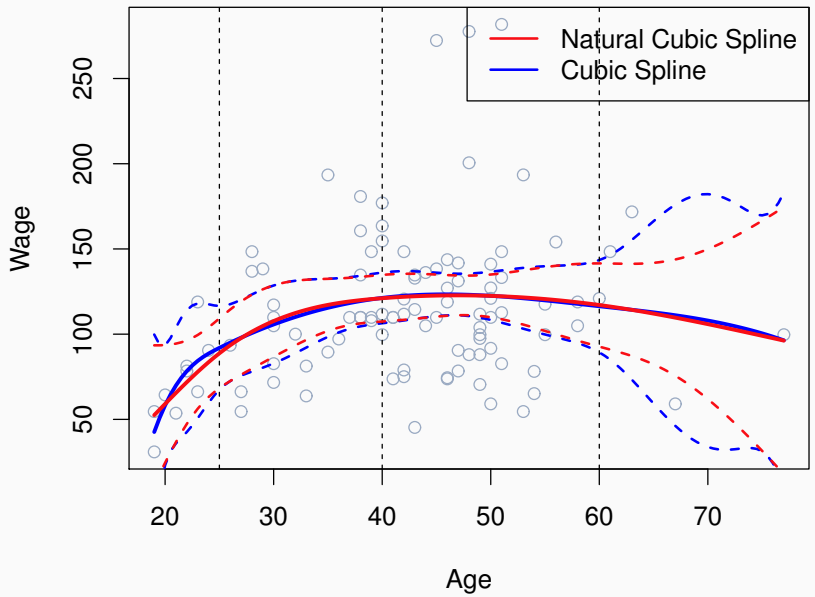
The positive part (cubic spline basis)





Natural cubic splines

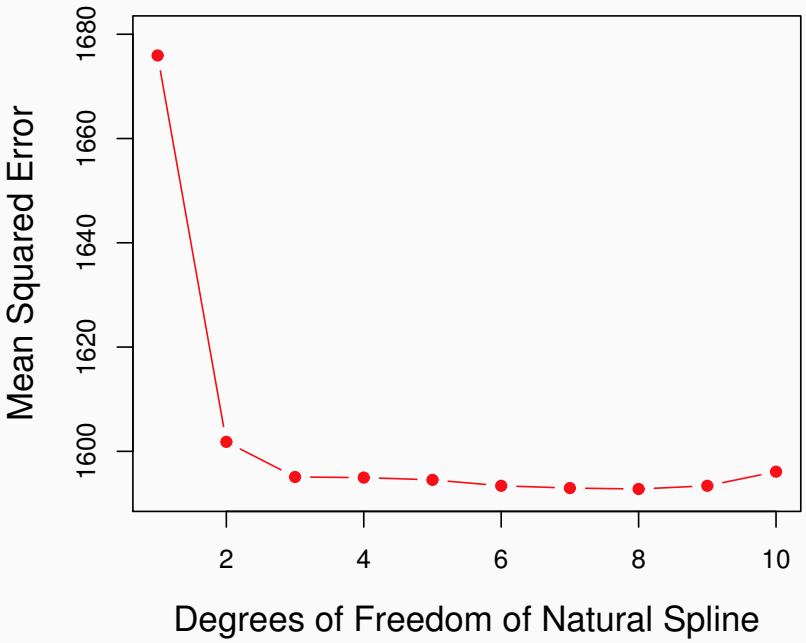
A *natural cubic spline* A natural spline is a regression spline with additional boundary constraints: the natural function is required to be linear at the boundary (in the region where X is smaller than the smallest knot, or larger than the largest knot).

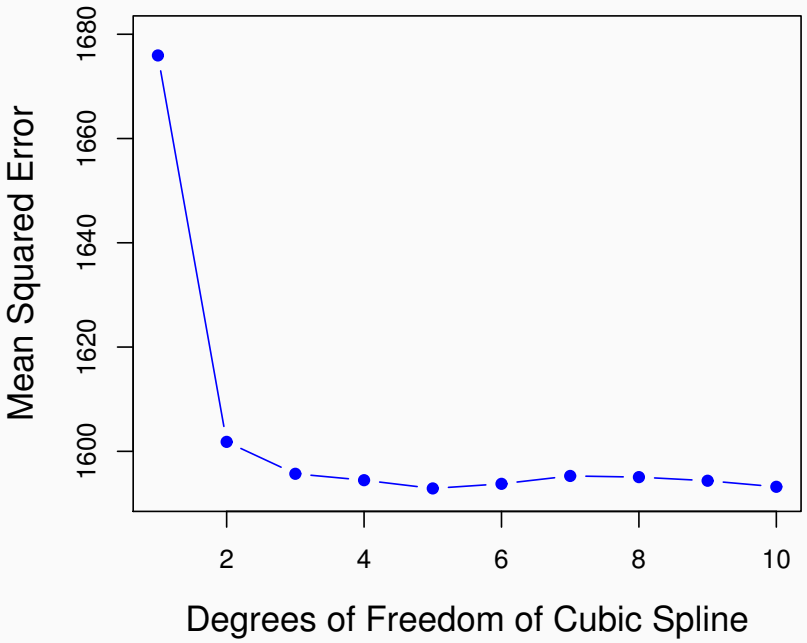


Number and Locations of the Knots

- Once the degree of the polynomial is decided, where should we place the knots? And how many knots should we consider?
- The number of knots can be fixed according to the desired degrees of freedom and the desired flexibility of the curve.
- For a given number of knots a possible choice is to position them uniformely over the range of X.
- A more sensible choice could be to set more knots where the function changes more rapidly.
- Another strategy is to put the knots in a sequence of quantiles of X.
- Cross-validation can be also used:
  - use a portion of the data (say 10 %) and fit a spline with a certain number of knots to the remaining data, and then use the spline to make predictions for the held-out portion
  - repeat this process multiple times and then compute the overall cross-validated RSS.
  - repeated for different numbers of knots K. Then the value of K corresponding to the smallest RSS is chosen.

Choosing K with CV





Smoothing splines

Smoothing splines

- The splines introduced so far are called **regression splines**. Their use implies:
  - choosing knots (number and position)
  - consider appropriate basis functions
  - use least squares to estimate coefficients
- When detecting a function $g(\cdot)$ to represent the relationship between a input variable X and a output Y two conflicting goals are: minimizing an appropriate loss measure, such as RSS, and obtaining a simple, smooth, curve.
- According to this we can find *smooth* function $g(x)$ which minimizes:

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

Smoothing splines

- $$\min_{g \in \mathcal{S}} \sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$$
- The first term is the Residuals Sum of Squares RSS
- the second term is a **penalty term** that penalizes the variability in $g(\cdot)$ (roughness penalty) whose relevance depends on λ. In other words, $\int g''(t)^2 dt$ is simply a measure of the total change in the function $g'(t)$, over its entire range.
- $\lambda \geq 0$ is a **tuning parameter** : the larger the value of λ, the smoother the function will be.
  - When $\lambda = 0$, the penalty term has no effect, and the function $g(\cdot)$ will exactly interpolate the observations.
  - When $\lambda \to \infty$, the function $g(\cdot)$ is linear.
- For an intermediate value of λ the function can lead to a reasonable fit of the data but will be somewhat smooth. λ controls the bias-variance trade-off of the smoothing spline.

Smoothing splines

$\lambda \int g''(t)^2 dt$

- The function g(x) that minimizes the penalized least squares, can be shown to have some special properties:
  - it is a piecewise cubic polynomial with knots at the unique values of the observed values of X and continuous first and second derivatives at each knot.
  - it is linear in the region outside of the extreme knots (natural cubic splines).
- However, it is not the same (natural) cubic spline that one would get if one applied the basis function approach described for regression splines: it is a regolarized version of such a (natural) cubic spline, where the value of the tuning parameter controls the level of regularization.

Choosing the Smoothing Parameter

- It might seem that a smoothing spline will have too many degrees of freedom, since a knot at each data point allows excessive flexibility.
- the tuning parameter λ controls the roughness of the smoothing spline, and hence the effective degrees of freedom.
- It is possible to show that asλ increases from 0 to ∞, the effective degrees of freedom, df_λ, decrease from n to 2. Actually, df_λ can be specified instead of λ.
- For instance, in R: `smooth.spline(age, wage)`.
- In fitting a smoothing spline, we do not need to select the number or location of the knots. Instead, we have another problem: we need to choose the value of λ.

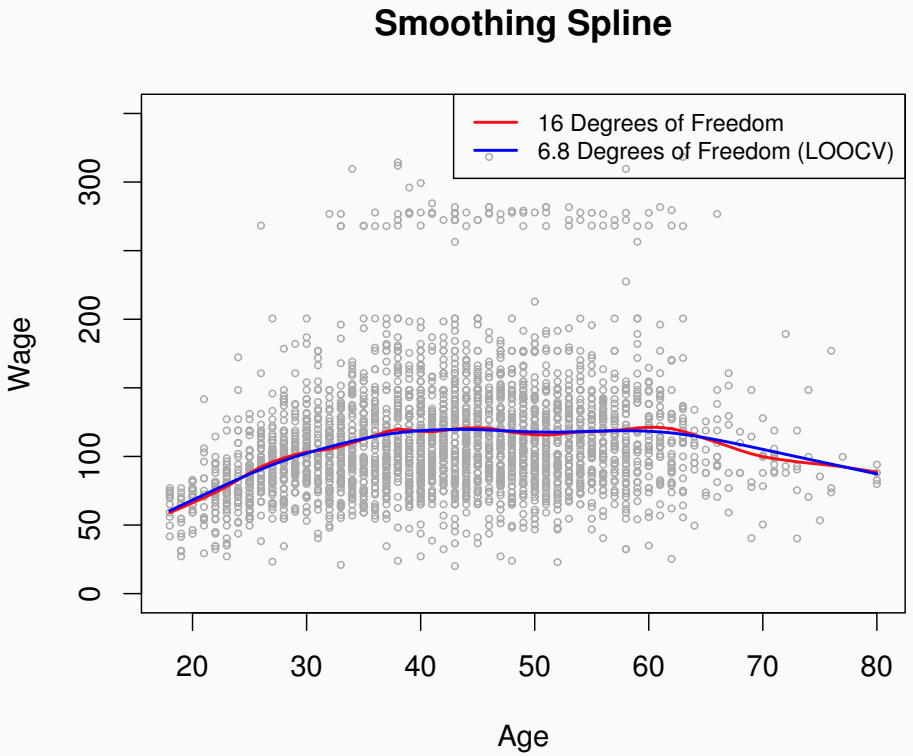
Using cross-validation

- We can find the value of λ that makes the cross-validated RSS as small as possible
- The leave-one-out cross-validation error (LOOCV) can be computed very efficiently for smoothing splines,

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

  - where $\hat{g}_\lambda^{(-i)}(x_i)$ is the fitted value evaluated at x_i, using all the data except the i-th, (x_i, y_i)
  - $\mathbf{S}_\lambda$ (whose formal definition is not detailed here) can be thought to be equivalent of the H matrix in linear models (such that $\hat{\mathbf{g}}_\lambda = \mathbf{S}_\lambda \mathbf{y}$ is the solution of penalized least squares for a given λ.
  - the effective degrees of freedom corresponds to the trace of the matrix $\mathbf{S}_\lambda$.

Smoothing splines for the credit dataset



- 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.