# Capturing Nonlinearity using Splines

Data Mining

Prof. Dawn Woodard
School of ORIE
Cornell University

# Outline

# Reading

Assigned reading for this week:
JWHT Chapter 7 up through 7.4.

# Additivity Assumption

Multiple linear regression as we have described it so far has **ADDITIVITY** and **LINEARITY** assumptions: i.e., $E(Y \mid x_1, \ldots, x_p)$ is the sum of terms that are each linear in one predictor.

$$E(Y \mid x_1, \ldots, x_p) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p.$$

E.g., for $p = 2$ the prediction surface $E(Y \mid x_1, x_2)$ as a function of $x_1$ and $x_2$ is a plane. For $p > 2$ the surface is a hyperplane.

# Additivity Assumption

To weaken the additivity assumption one can include "interaction effects." For instance when $p = 2$ you can change the linear regression model to

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \epsilon$$
$$\epsilon \sim N(0, \sigma^2)$$

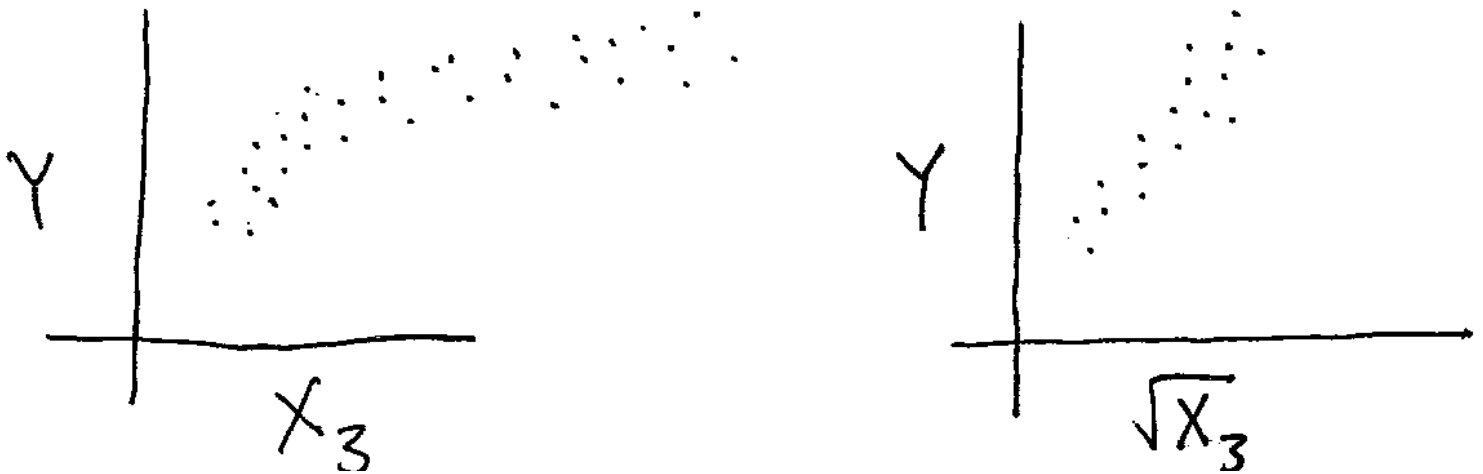
Then $E(Y \mid x_1, x_2)$ is no longer restricted to be a hyperplane.

# Additivity Assumption

Example:

# Linearity Assumption

Similarly, there are tricks to get away from the **linearity assumption**. We know one already: if the relationship between predictors and outcome is nonlinear, sometimes transforming one or both leads to a linear relationship.



# Linearity Assumption

But this doesn't always work. For instance:


Here there are no monotonic transformations $f$ and $g$ of $x_3$ and $Y$ that can make the relationship between $f(x_3)$ and $g(Y)$ linear.

# Linearity Assumption

One can handle a situation like this by including quadratic and/or higher-order **polynomial terms** in the regression model, e.g. when $p = 1$ take

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \beta_3 x_1^3 + \epsilon$$
$$\epsilon \sim N(0, \sigma^2)$$

Here we are assuming a cubic instead of linear relationship between $x_1$ and $Y$, which is much less restrictive!

# Implementation in R

It is extremely easy to include interaction and polynomial terms in the regression model in R. Notice that a model like

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_2^2 + \beta_4 x_1 x_2 + \epsilon$$

can be viewed as a standard multiple linear regression model, where there are 4 predictors: $\tilde{x}_1 = x_1$, $\tilde{x}_2 = x_2$, $\tilde{x}_3 = x_2^2$, and $\tilde{x}_4 = x_1 x_2$.

$$Y = \beta_0 + \beta_1 \tilde{x}_1 + \beta_2 \tilde{x}_2 + \beta_3 \tilde{x}_3 + \beta_4 \tilde{x}_4 + \epsilon$$

# Implementation in R

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_2^2 + \beta_4 x_1 x_2 + \epsilon$$

To fit this model in R:

# Implementation in R

It's tempting to include a lot of interactions and high-order polynomial terms in the model, to make the model less restrictive. But beware of the risk of overfitting!!!

A popular approach is to include some of these terms, but then do model selection to cut back on the number of terms.

# Splines

**Sometimes the relationship between a continuous predictor $x$ and the expected value of the outcome $Y$ can't be captured effectively using a low-degree polynomial.**

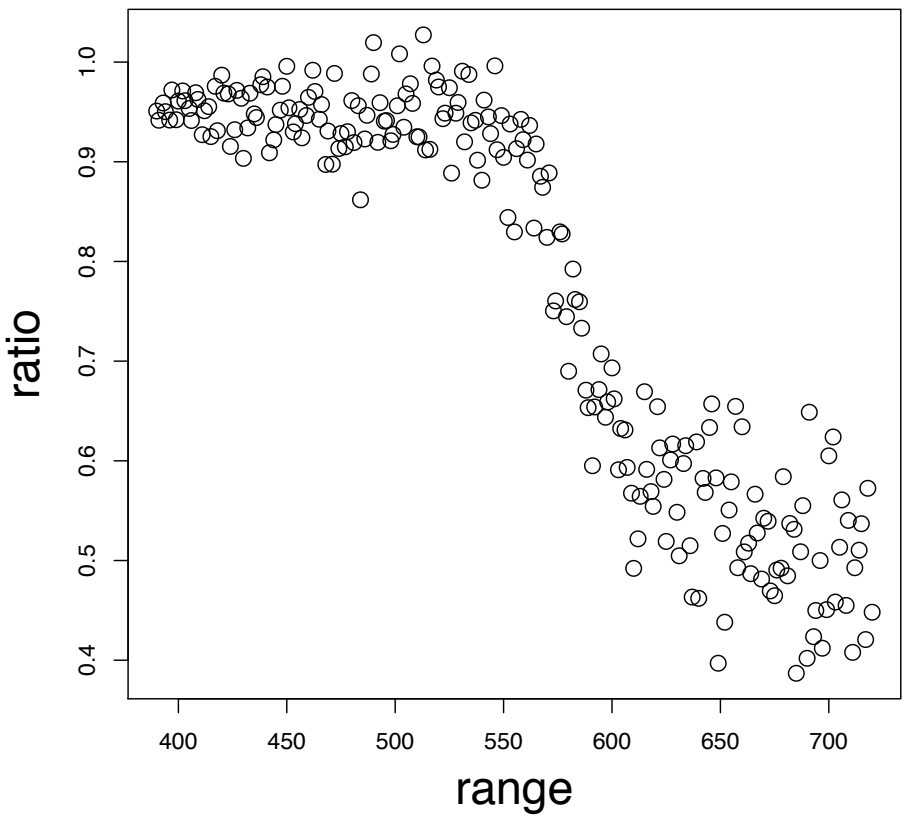
For now let's focus on the case of a single predictor, although this can easily be generalized to more than one.

Note: Don't use splines unless simpler approaches don't work AND a scatterplot of $y$ vs. $x$ shows nonlinear, nonpolynomial behavior

The first dataset today regards the LIDAR (light detection and ranging) technique, which uses the reflection of laser-emitted light to detect chemical comporids in the atmosphere. This is useful for **monitoring air pollution**.

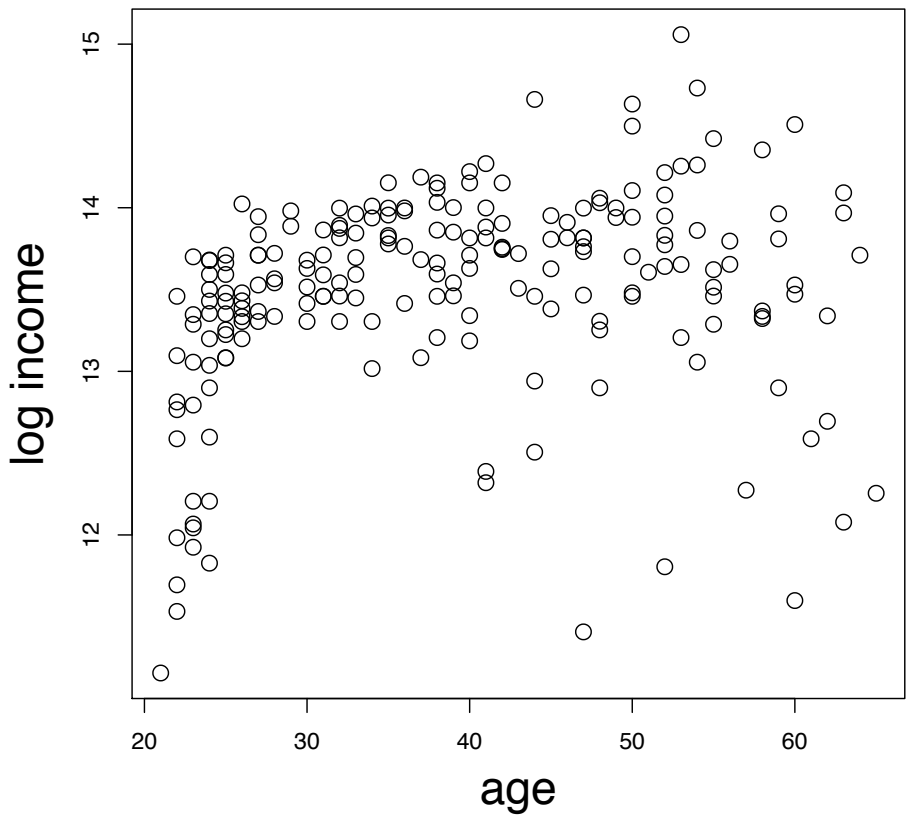
# Splines

Range: distance traveled before the light is reflected back to its source
Ratio: ratio of received light from two laser sources



# Splines

**Another example: predicting income based on age**



# Splines

One approach to handle nonlinearity would be to divide the domain of $x$ into intervals, and fit a different line within each interval:

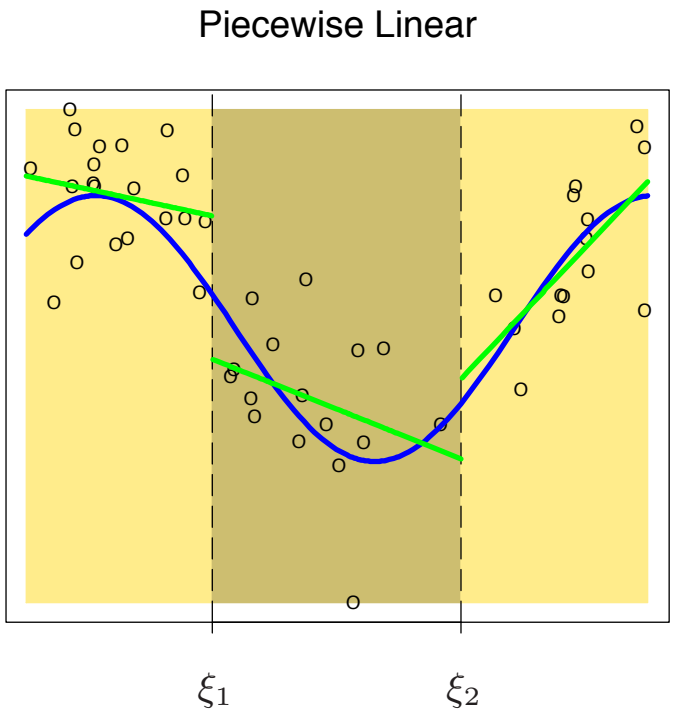



Figure from Hastie, Tibshirani, and Friedman (2009). This is simulated data, where the true expectation of $Y$ is shown in blue.

But we generally would prefer that the prediction were continuous

# Splines

One would like to fit a line in each interval, but restrict the prediction to be continuous:

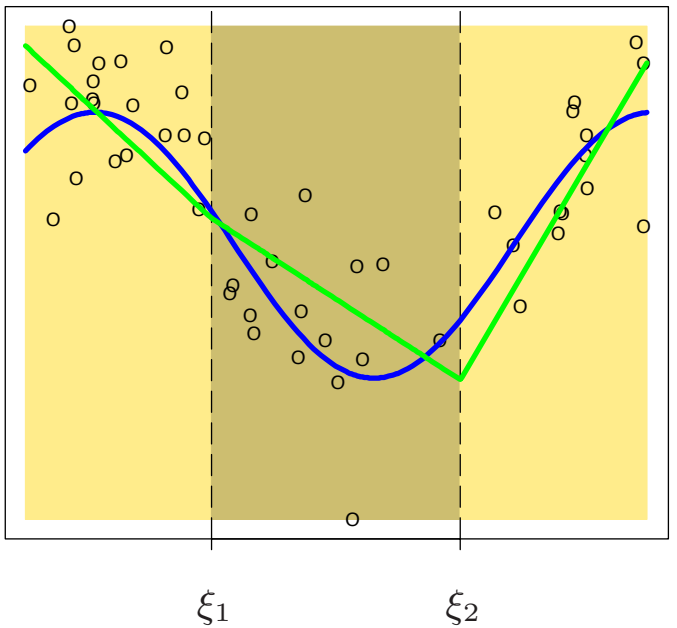



Figure from Hastie, Tibshirani, and Friedman (2009).

# Splines

Functions that are linear in the intervals $(-\infty, \xi_1)$, $(\xi_1, \xi_2)$, and $(\xi_2, \infty)$ and are continuous at $\xi_1$ and $\xi_2$ can be written in the form

$$\beta_0 + \beta_1 x + \beta_2 (x - \xi_1)_+ + \beta_3 (x - \xi_2)_+$$

where $t_+$ denotes the positive part. Here's a plot of $(x - \xi_1)_+$:

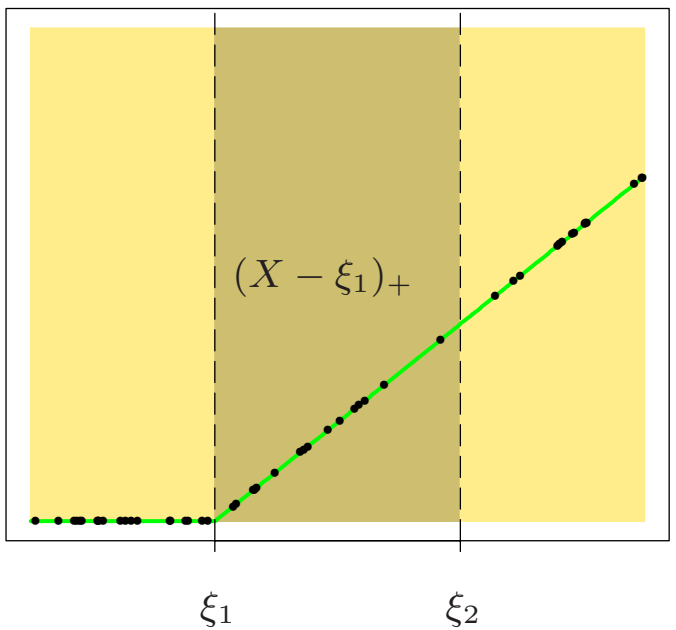



Figure from HTF text

# Splines

# Splines

So we can fit the model:

$$Y = \beta_0 + \beta_1 x + \beta_2 (x - \xi_1)_+ + \beta_3 (x - \xi_2)_+ + \epsilon$$
$$\epsilon \sim N(0, \sigma^2)$$

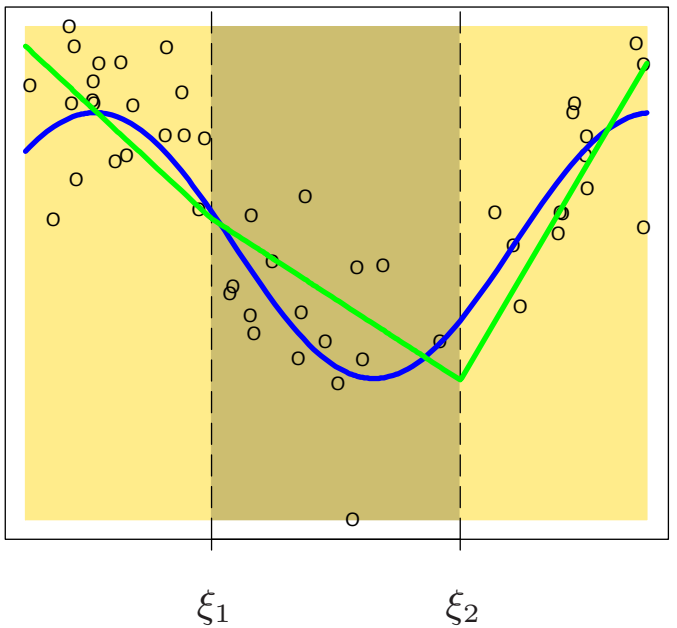
How?

# Splines

How to choose the knots?

# Splines

Perhaps we would like a smoother function to predict $Y$. Our function has a discontinuous first derivative at the knots, and so is sensitive to our choice of the knots.



# Splines

If we want the function to be continuous and have continuous first derivatives, we can fit a quadratic function in each interval, and restrict the function to be smooth in these ways. With 2 knots this corresponds to fitting a function of the form

$$\beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 (x - \xi_1)_+^2 + \beta_4 (x - \xi_2)_+^2$$

Why?

# Splines

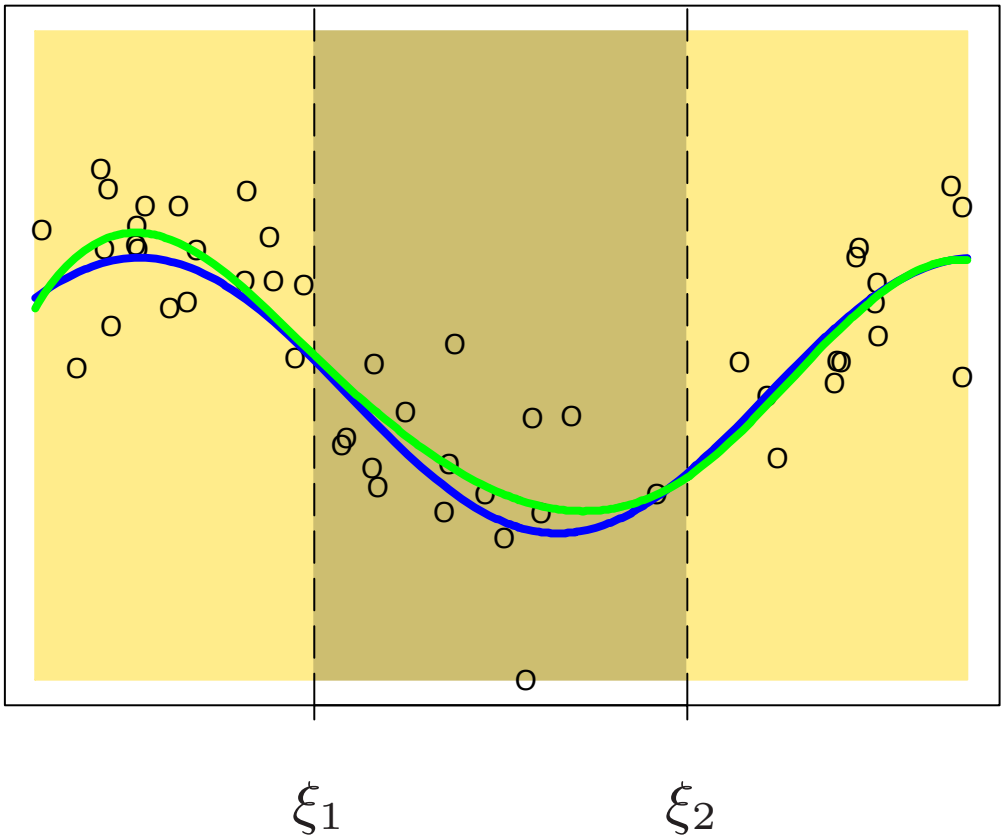
These “quadratic splines” are a common choice. It’s also common to use “cubic splines” to get an even smoother function; these ensure that the function and its first and second derivatives are continuous. With 2 knots this corresponds to fitting a function of the form

$$\beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x - \xi_1)_+^3 + \beta_5 (x - \xi_2)_+^3.$$

So our statistical model is

# Splines

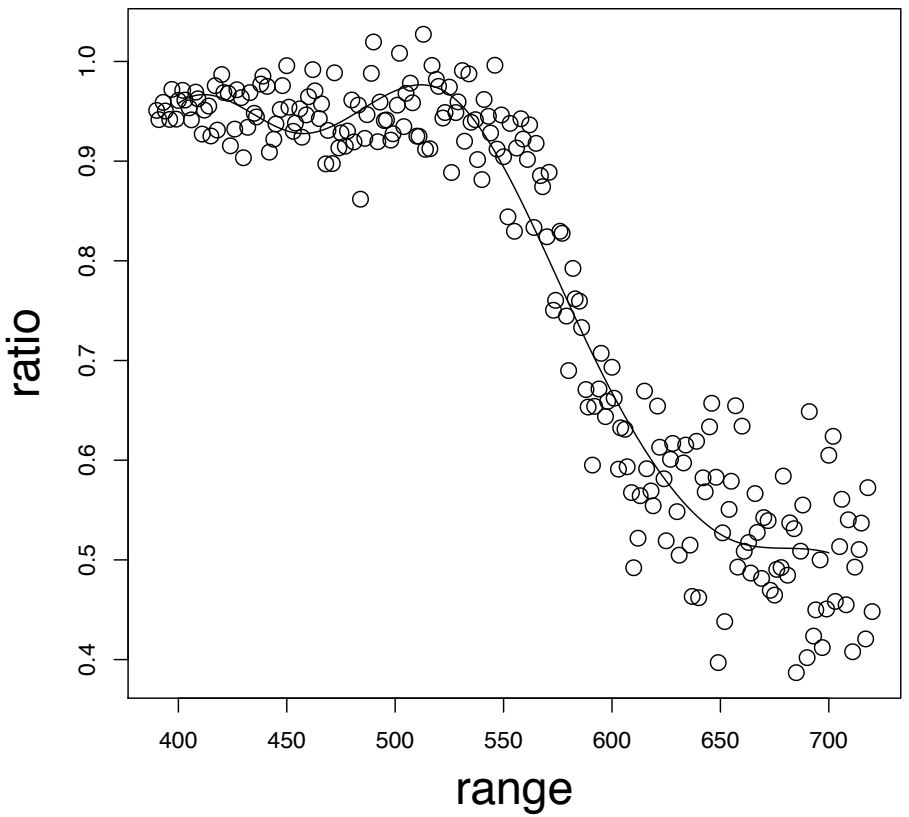
Here's the cubic spline fit for the simulated data example:




The estimated curve is very close to the true curve!

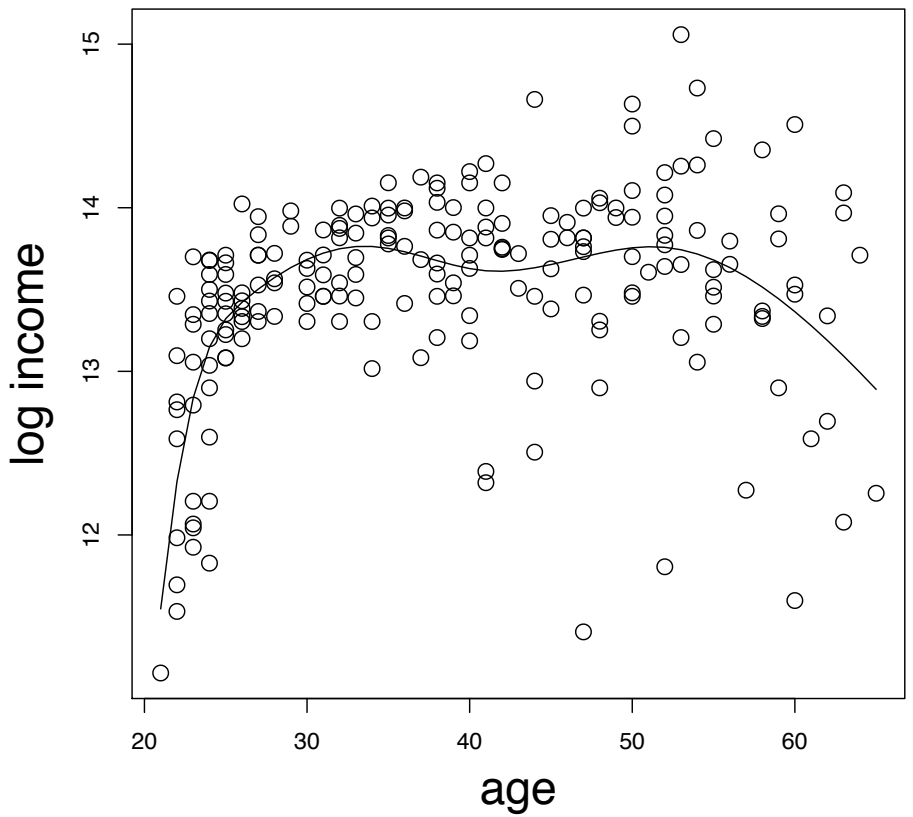
# Splines

Here's the cubic spline prediction function for the LIDAR data:



# Splines

Here it is for the income / age data:



# Splines

R code for cubic spline fit, 4 knots: