Splines

Patrick Breheny

September 27

Introduction

- Our next topic is nonparametric regression
- The regression problem involves modeling how the expected value (or some function of the expected value) of a response y changes in response to changes in an explanatory variable x:

$$\mathrm{E}(y|x) = f(x)$$

- Linear regression, as its name implies, assumes a linear relationship; namely, that $f(x) = \beta_0 + \beta_1 x$

Parametric vs. nonparametric approaches

- This is the nature of parametric statistics: to reduce an unknown and potentially complicated function down to a simple form with a small number of unknown parameters
- The nonparametric approach, in contrast, is to make as few assumptions about the regression function f as possible
- Instead, we will try to use the data as much as possible to learn about the potential shape of f, allowing f to be very flexible, yet smooth

Basis functions

- One approach for extending the linear model is to augment the linear component of x with additional, derived functions of x:

$$f(x) = \sum_{m=1}^{M} \beta_m h_m(x),$$

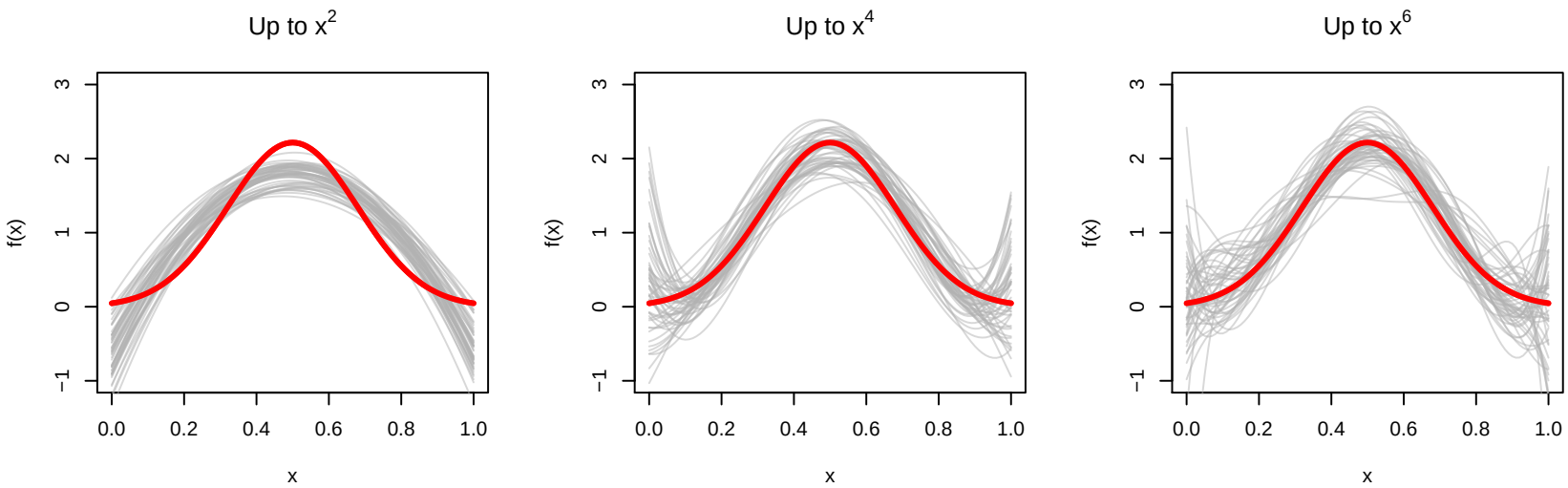
  where the $\{h_m\}$ are known functions called *basis functions*
- Because the basis functions $\{h_m\}$ are prespecified and the model is linear in these new variables, ordinary least squares approaches for model fitting and inference can be employed
- This idea is not new to you, as you have encountered transformations and the inclusion of polynomial terms in models in earlier courses

Problems with polynomial regression

- However, polynomial terms introduce undesirable side effects: each observation affects the entire curve, even for x values far from the observation
- Not only does this introduce bias, but it also results in extremely high variance near the edges of the range of x
- As our authors put it, "tweaking the coefficients to achieve a functional form in one region can cause the function to flap about madly in remote regions"

Problems with polynomial regression (cont'd)

To illustrate this, consider the following simulated example (gray lines are models fit to 100 observations arising from the true f, colored red):



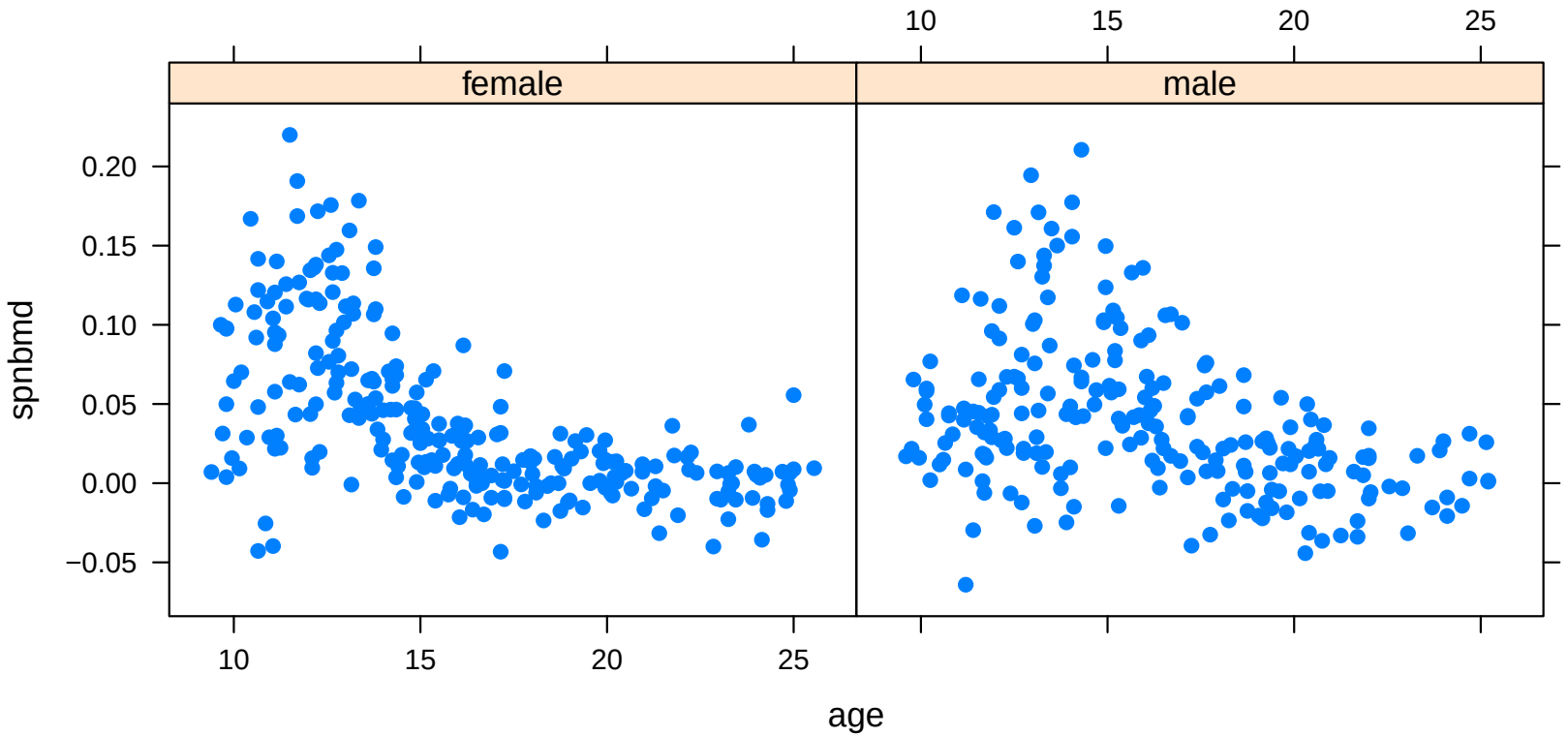
Global versus local bases

- Instead, let us consider *local* basis functions, thereby ensuring that a given observation affects only the nearby fit, not the fit of the entire line
- In this lecture, we will explore piecewise basis functions
- As we will see, *splines* are piecewise polynomials joined together to make a single smooth curve

Bone mineral density data

- As an example of a real data set with an interesting change in $\mathrm{E}(y|x)$ as a function of x, we will look at a study of changes in bone mineral density in adolescents
- The outcome is the difference in spinal bone mineral density, taken on two consecutive visits, divided by the average of the two measurements
- Age is the average age over the two visits
- A person's bone mineral density generally increases until the individual is done growing, then remains relatively constant until old age

Bone mineral density data (cont'd)

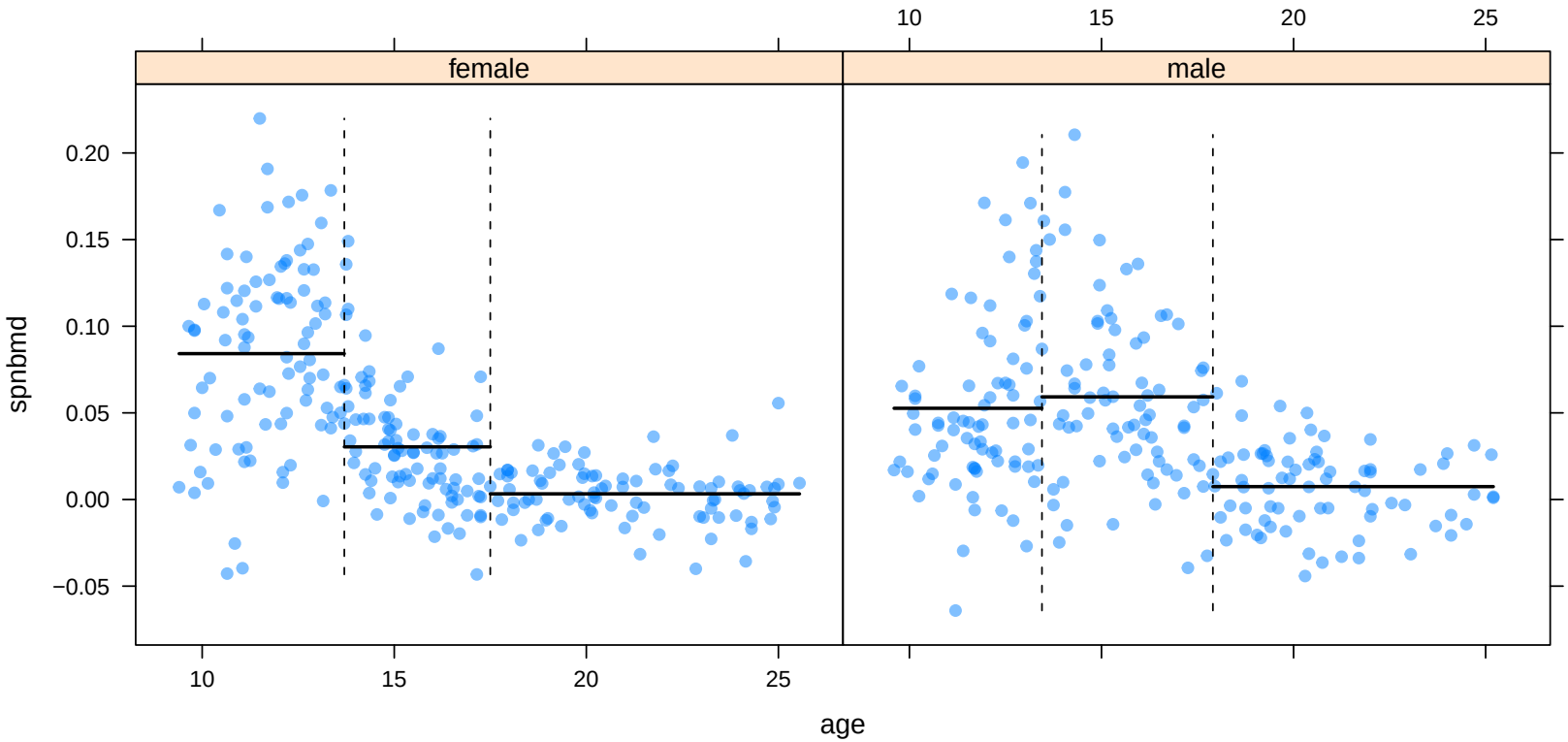


The piecewise constant model

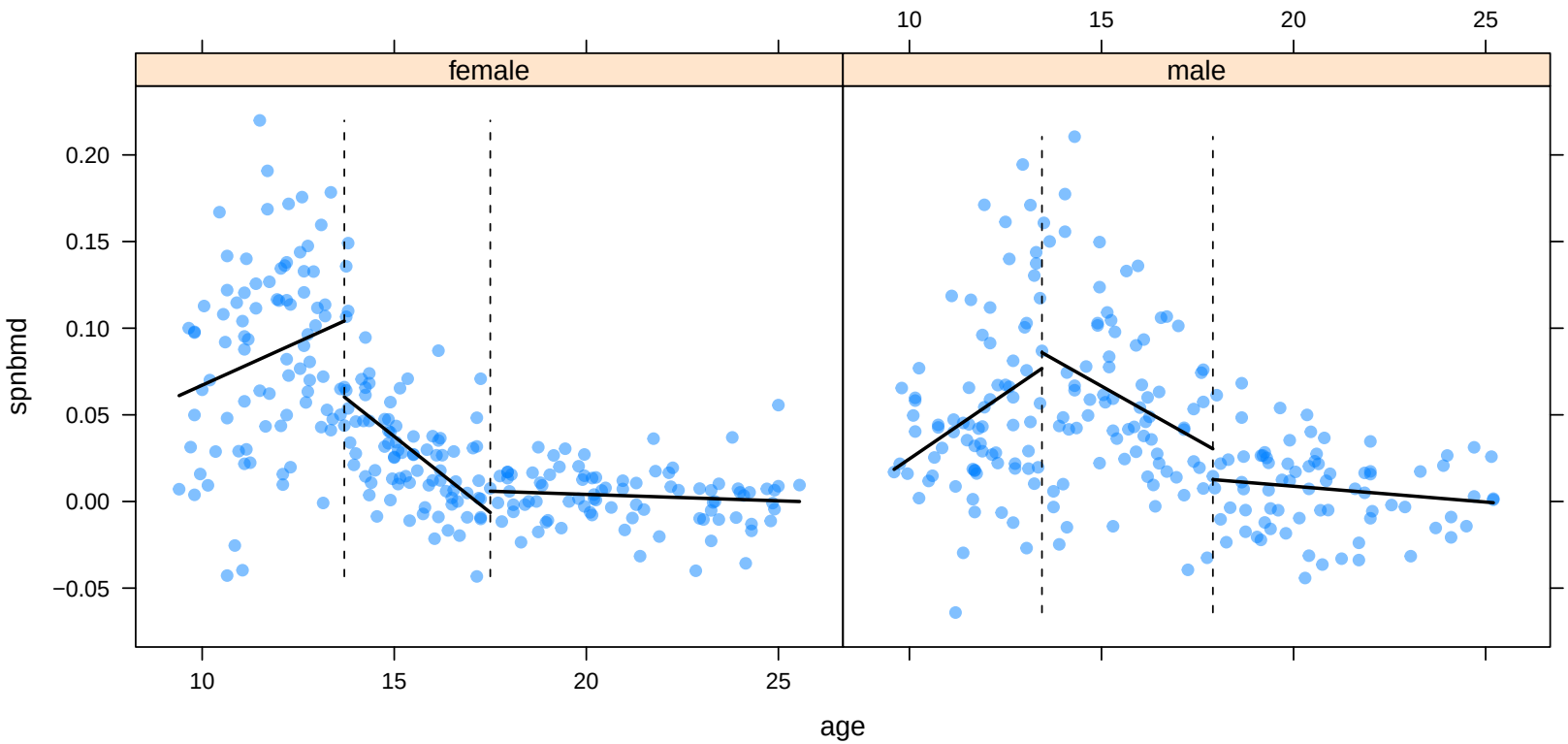
- To understand splines, we will gradually build up a piecewise model, starting at the simplest one: the piecewise constant model
- First, we partition the range of x into $K+1$ intervals by choosing K points $\{\xi_k\}_{k=1}^{K}$ called *knots*
- For our example involving bone mineral density, we will choose the tertiles of the observed ages, thereby giving three basis functions:

$$h_1(x) = I(x < \xi_1)$$
$$h_2(x) = I(\xi_1 \leq x < \xi_2)$$
$$h_3(x) = I(\xi_2 \leq x)$$

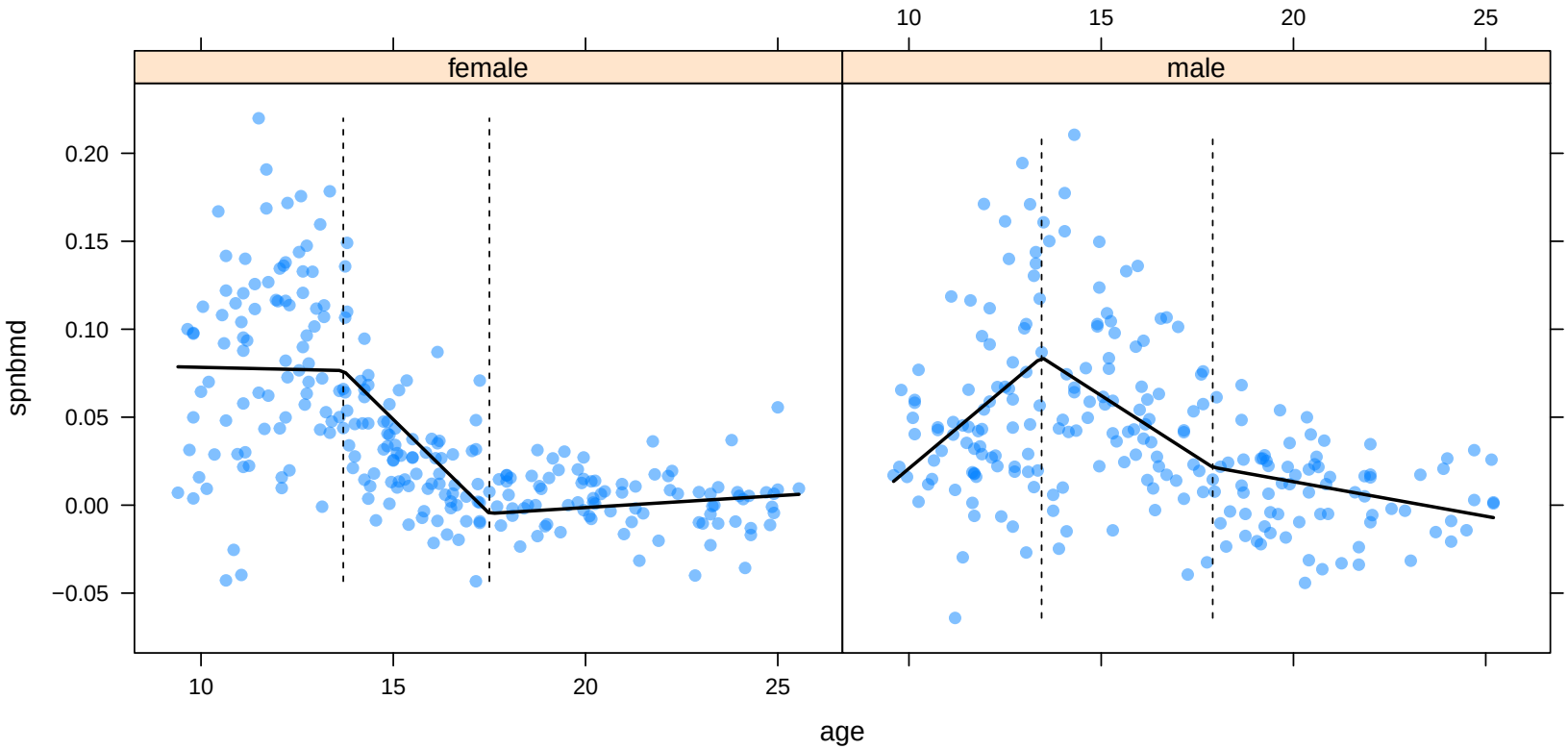
The piecewise constant model (cont'd)



The piecewise linear model



The continuous piecewise linear model



Basis functions for piecewise continuous models

These constraints can be incorporated directly into the basis functions:

$$h_1(x) = 1, \quad h_2(x) = x, \quad h_3(x) = (x - \xi_1)_+, \quad h_4(x) = (x - \xi_2)_+,$$

where $(\cdot)_+$ denotes the positive portion of its argument:

$$r_+ = \begin{cases} r & \text{if } r \geq 0 \\ 0 & \text{if } r < 0 \end{cases}$$

Basis functions for piecewise continuous models

- It can be easily checked that these basis functions lead to a composite function $f(x)$ that:
  - Is everywhere continuous
  - Is linear everywhere except the knots
  - Has a different slope for each region
- Also, note that the degrees of freedom add up: 3 regions $\times$ 2 degrees of freedom in each region - 2 constraints $=$ 4 basis functions

Splines

- The preceding is an example of a *spline*: a piecewise $m-1$ degree polynomial that is continuous up to its first $m-2$ derivatives
- By requiring continuous derivatives, we ensure that the resulting function is as smooth as possible
- We can obtain more flexible curves by increasing the degree of the spline and/or by adding knots
- However, there is a tradeoff:
  - Few knots/low degree: Resulting class of functions may be too restrictive (bias)
  - Many knots/high degree: We run the risk of overfitting (variance)

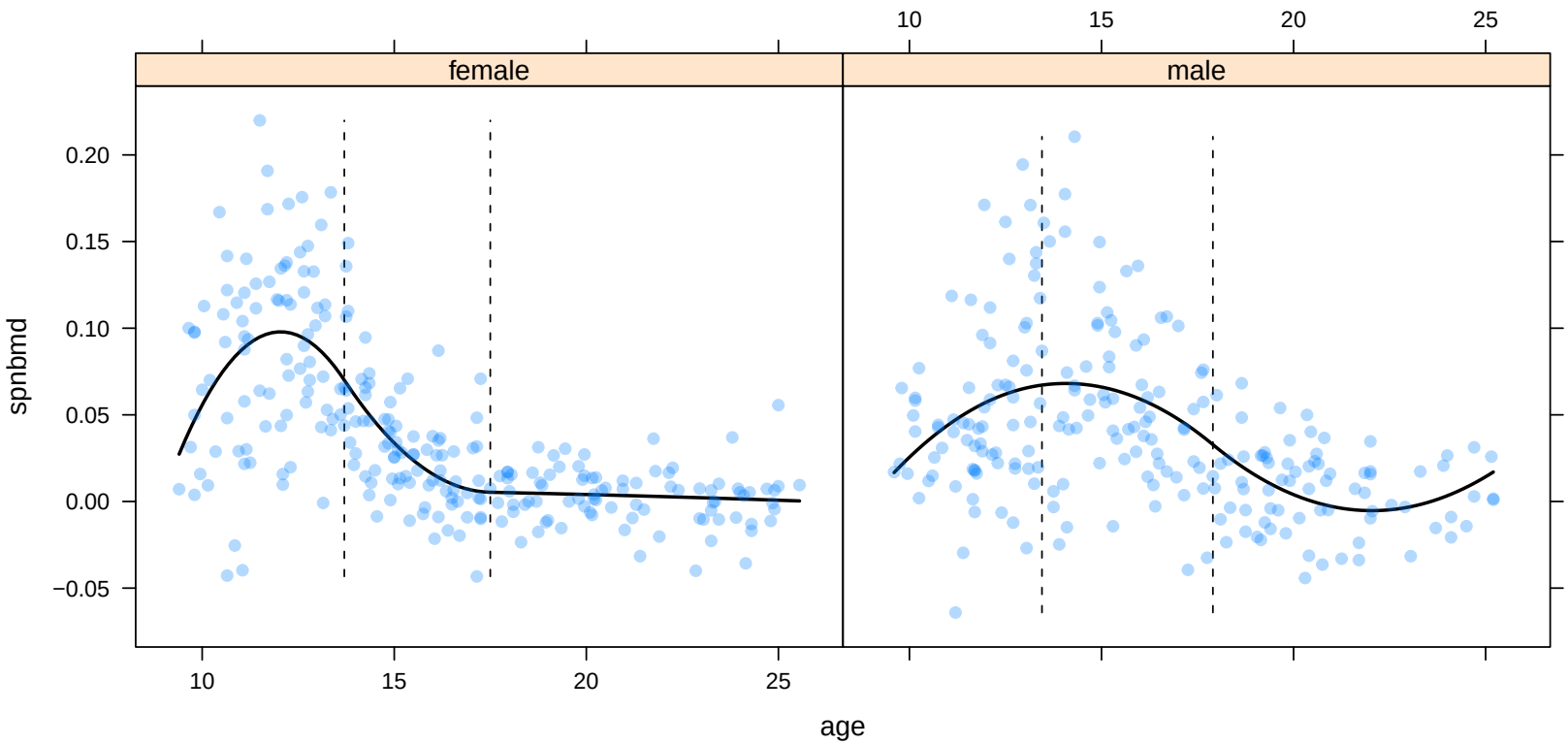
The truncated power basis

- The set of basis functions introduced earlier is an example of what is called the *truncated power basis*
- Its logic is easily extended to splines of order m:

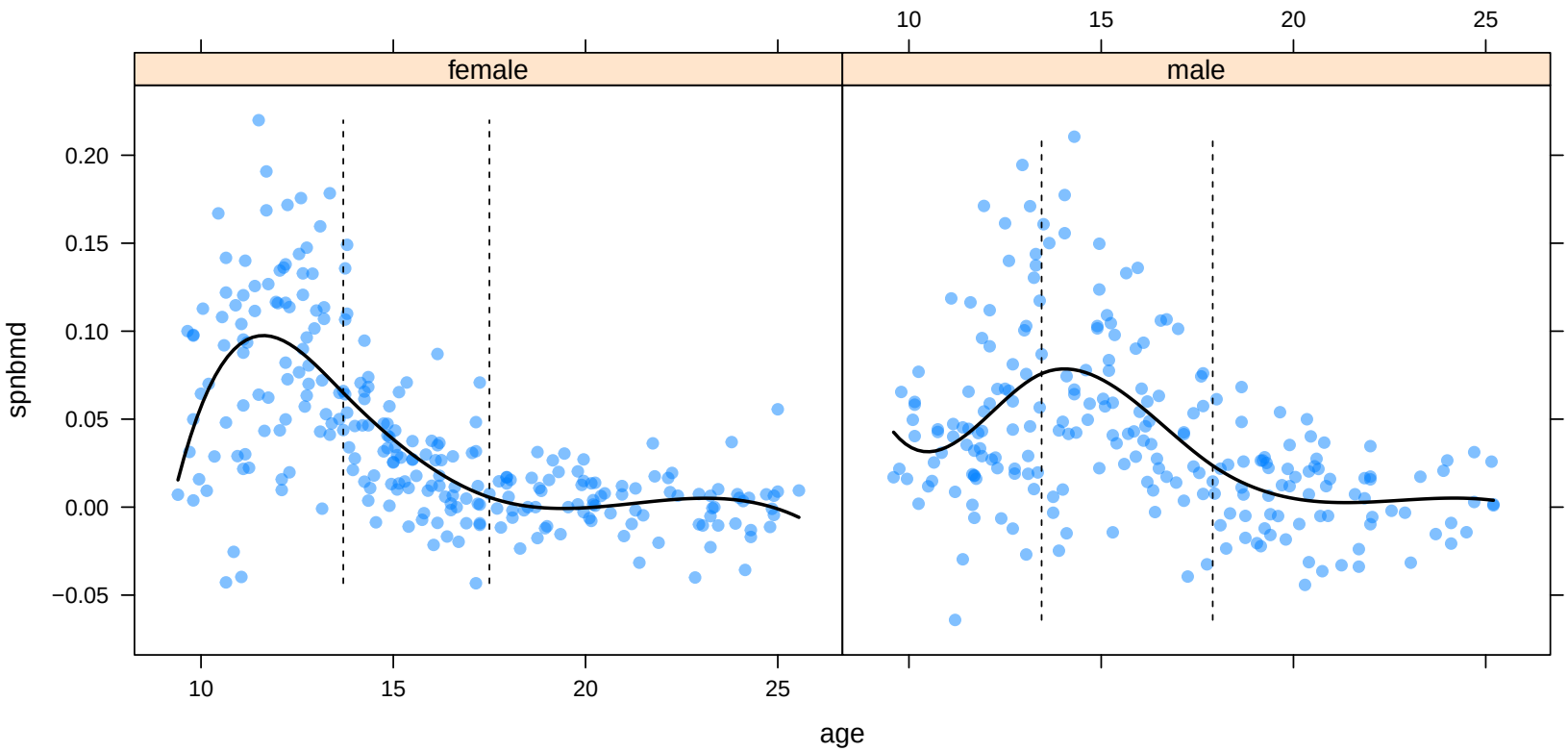
$$h_j(x) = x^{j-1} \qquad j = 1, \ldots, m$$
$$h_{m+k}(x) = (x - \xi_k)_+^{m-1} \qquad k = 1, \ldots, K$$

- Note that a spline has $m + K$ degrees of freedom

Quadratic splines



Cubic splines



Additional notes

- These types of fixed-knot models are referred to as *regression splines*
- Recall that cubic splines contain $4 + K$ degrees of freedom: $K + 1$ regions $\times$ 4 parameters per region - K knots $\times$ 3 constraints per knot
- It is claimed that cubic splines are the lowest order spline for which the discontinuity at the knots cannot be noticed by the human eye
- There is rarely any need to go beyond cubic splines, which are by far the most common type of splines in practice

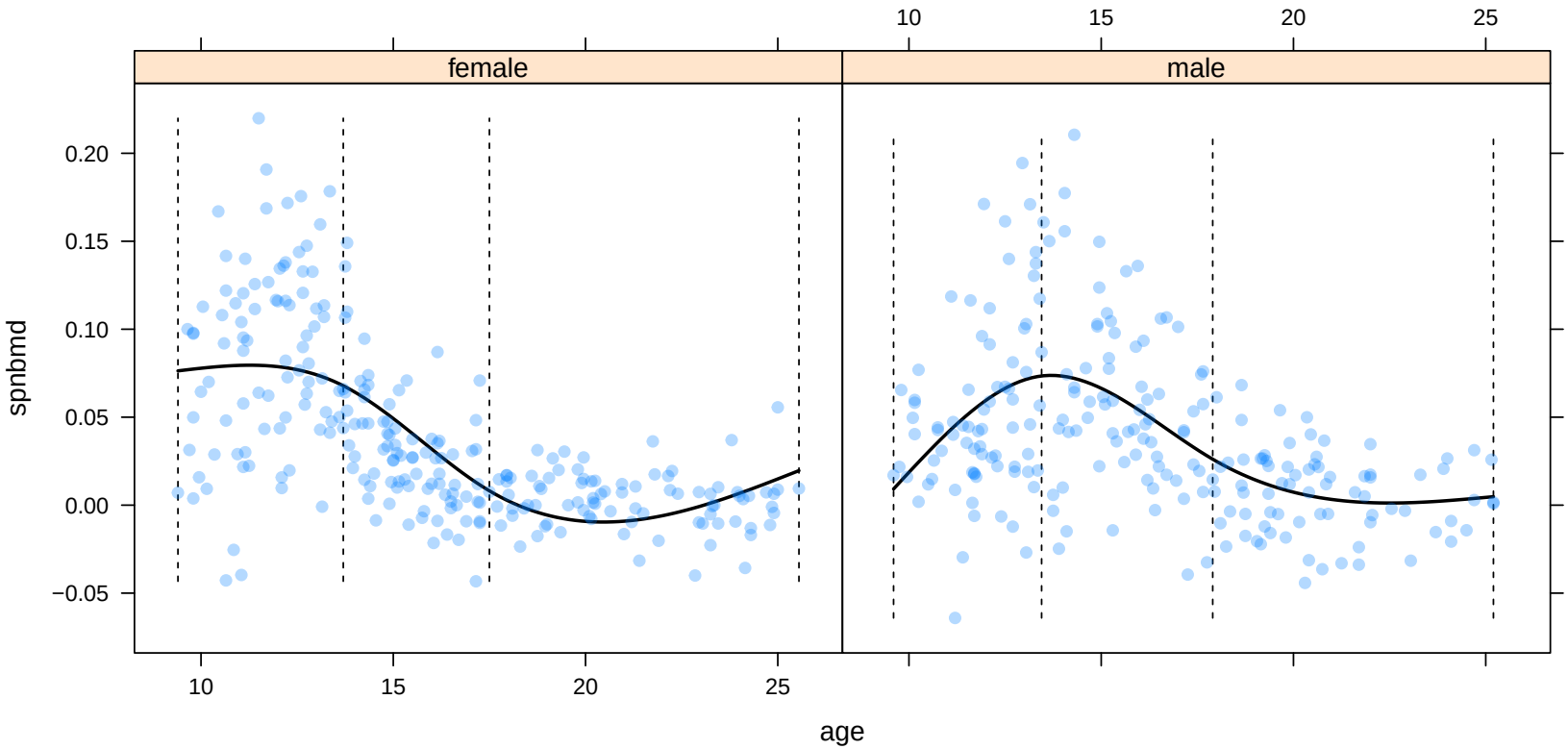
Implementing regression splines

- The truncated power basis has two principal virtues:
  - Conceptual simplicity
  - Simpler models are nested inside it, leading to straightforward tests of null hypotheses
- Unfortunately, it has a number of computational/numerical flaws – it's inefficient and can lead to overflow and nearly singular matrix problems
- The more complicated but numerically much more stable and efficient *B-spline* basis is often employed instead
- Fortunately, one can use B-splines without knowing the details behind their complicated construction

Natural cubic splines

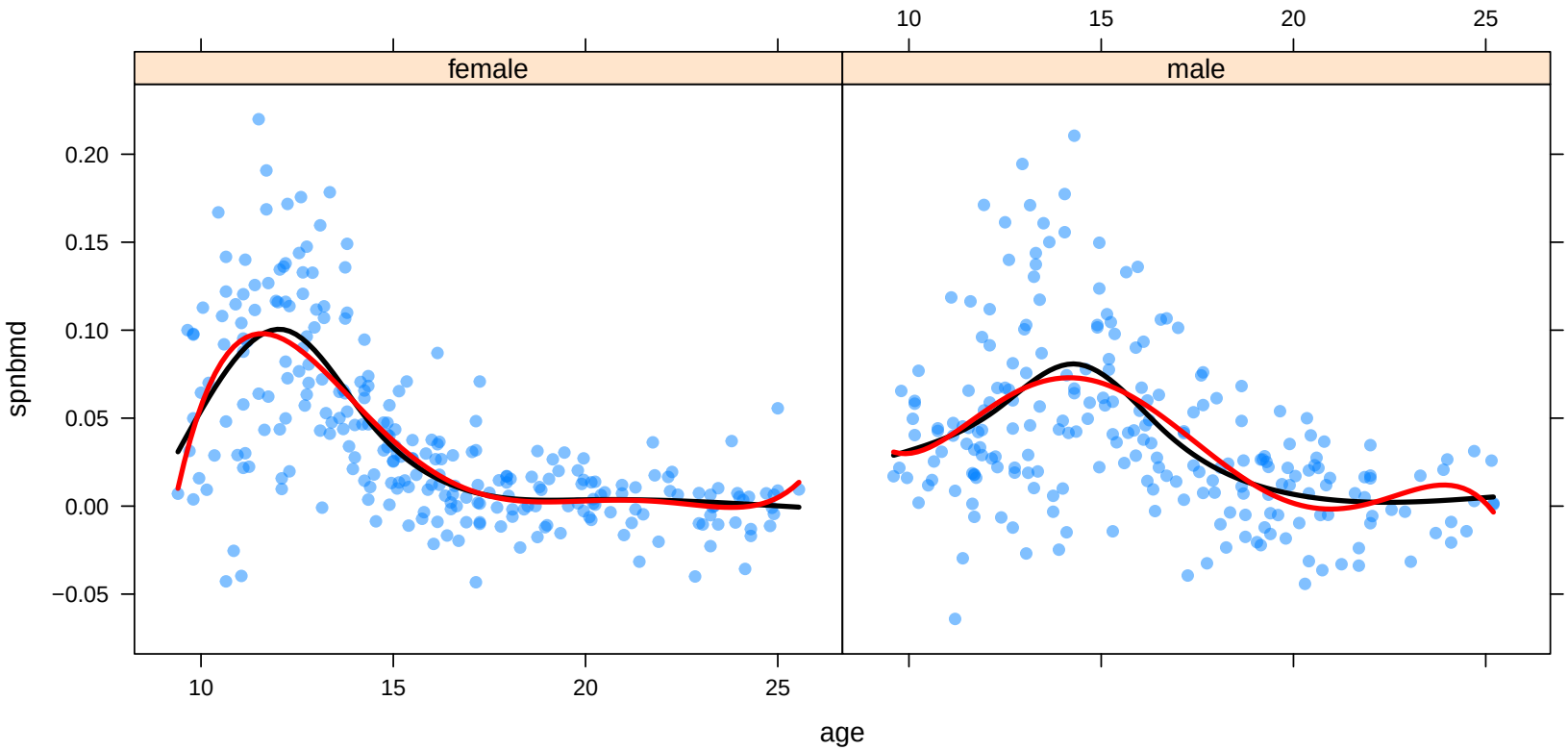
- Polynomial fits tend to be erratic at the boundaries of the data
- This is even worse for cubic splines
- *Natural cubic splines* ameliorate this problem by adding the additional (4) constraints that the function is linear beyond the boundaries of the data
- Note, then, that a natural cubic spline basis function with K knots has K degrees of freedom

Natural cubic splines (cont'd)

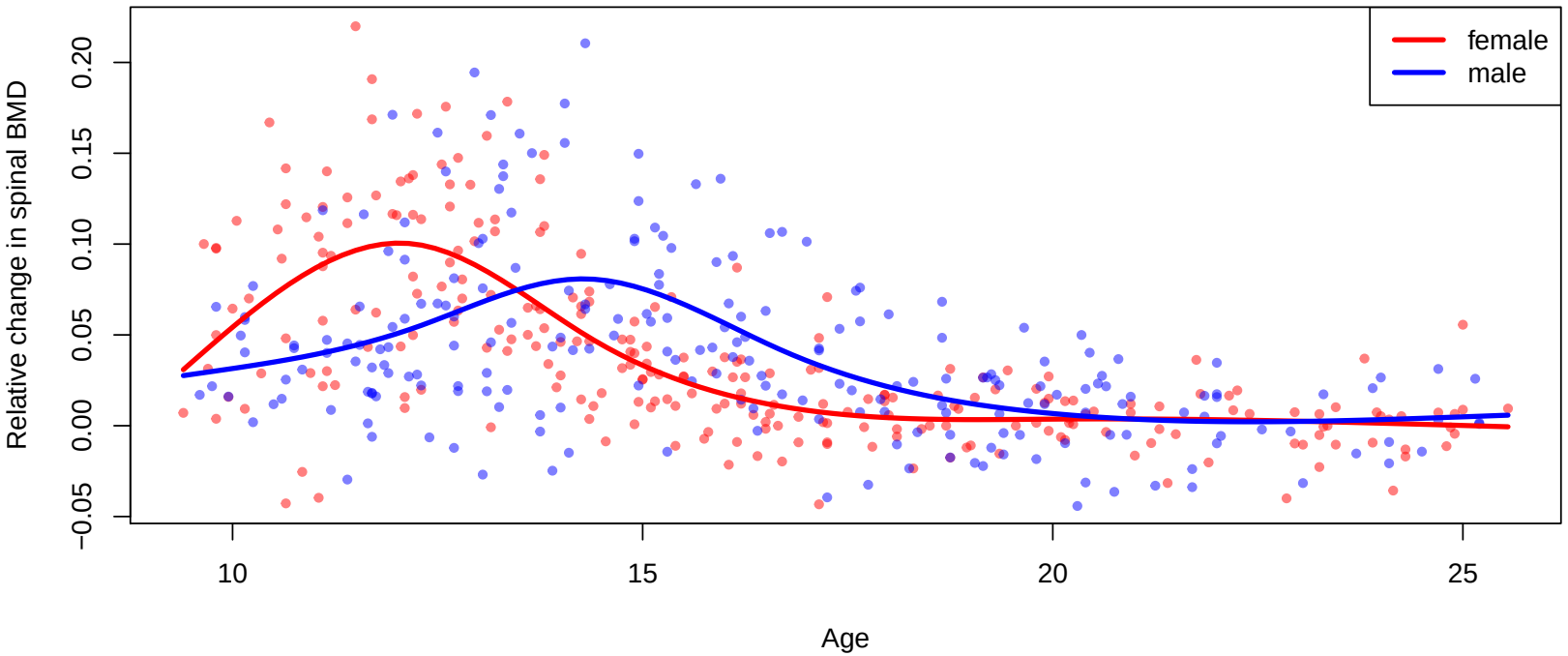


Natural cubic splines, 6 df

Black line: 6 df natural cubic spline; red line: 6 df polynomial



Natural cubic splines, 6 df (cont'd)



Mean and variance estimation

- Because the basis functions $\{h_m\}$ are fixed, model fitting and inference are straightforward extensions of ordinary least squares approaches:
  - The linear model is nested inside the spline representation, so F-tests are valid
  - Confidence intervals can be constructed based on

$$\mathrm{Var}(\boldsymbol{\Lambda}^T\widehat{\boldsymbol{\beta}}) = \sigma^2\boldsymbol{\Lambda}^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\Lambda}$$

- Furthermore, extensions to logistic regression, Cox proportional hazards regression, etc., are straightforward

The bias problem

- It is worth noting, however, that unless one makes the (rather unlikely) assumption that the true $f(x)$ is piecewise cubic with continuous first and second derivatives at exactly the knots you chose, bias will be present in its estimate
- In particular, letting $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ denote the projection matrix,

$$\mathrm{E}(\hat{\mathbf{f}}) = \mathbf{H}f(\mathbf{x});$$

  in words, the expected value of $\hat{\mathbf{f}}$ is not $f(\mathbf{x})$ itself, but the projection of $f(\mathbf{x})$ onto the space of functions spanned by the spline representation
- Practically speaking, however, it is difficult to implement this, as it depends on the unknown $f(\mathbf{x})$, and the issue of bias is typically ignored during inference

Splines in R

- R enables very low-level control over the construction of spline bases using the splines package, which is typically installed but not loaded with R
- Given a vector x, the function bs constructs splines using the B-spline basis alluded to earlier

```
X <- bs(x,knots=quantile(x,p=c(1/3,2/3)))
X <- bs(x,df=5)
X <- bs(x,degree=2,df=10)
X2 <- predict(X,newx=x2)
```

- For natural cubic splines, the ns function works in the same way

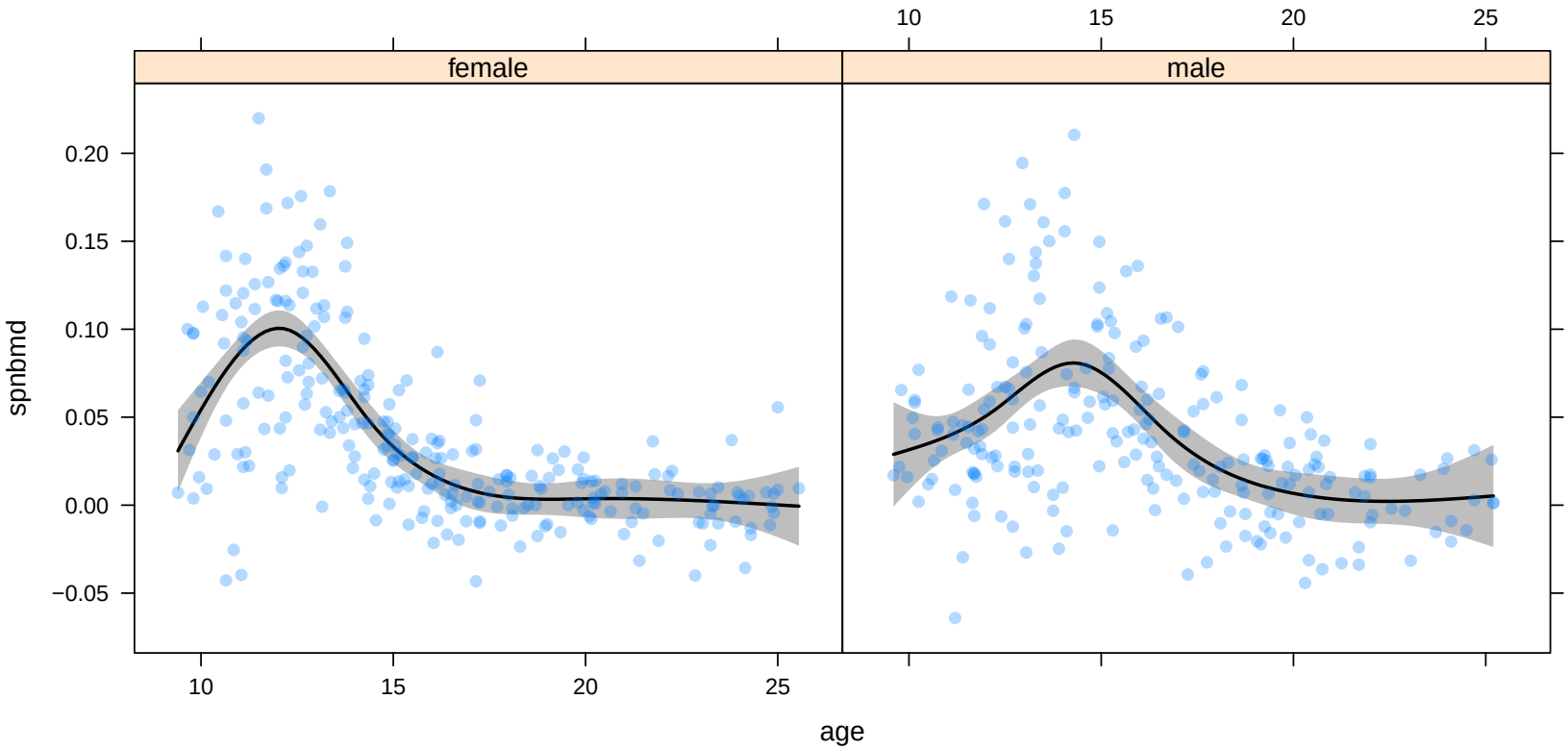
Splines in SAS

- A handful of procedures in SAS allow `EFFECT` statements, which enable you to construct a set of basis functions and handle them as a unit in specifying a model
- The most notable of these procedures is `PROC GLIMMIX`:

```
PROC GLIMMIX DATA=bmd;
  EFFECT AgeSpl = SPLINE(Age / KNOTMETHOD=PERCENTILES(3));
  CLASS Gender;
  MODEL Spnbmd = Gender|AgeSpl;
RUN;
```

- Note that `SPLINE` uses a B-spline basis; to get natural cubic splines, use `NATURALCUBIC`

Mean and variance estimation (cont'd)



Mean and variance estimation (cont'd)