Introduction to Splines in Linear Regression

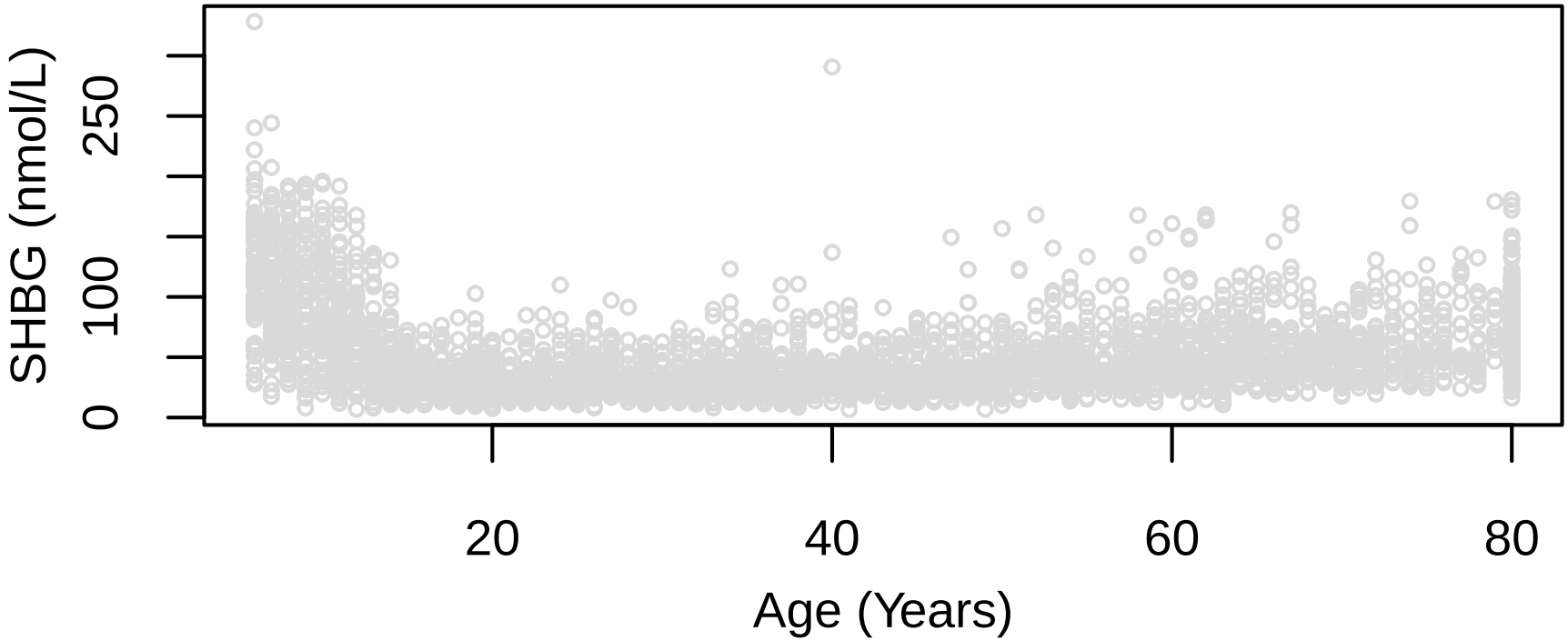
BIOS 6618

CU Anschutz

Motivating Example

Motivation

We are interested in the association between sex hormone binding globulin (SHBG) and age in years for males 6-80 years old from the NHANES 2015-2016 cycle ($n = 3390$ with SHBG)*. The scatterplot suggests there is some sort of non-linear trend:

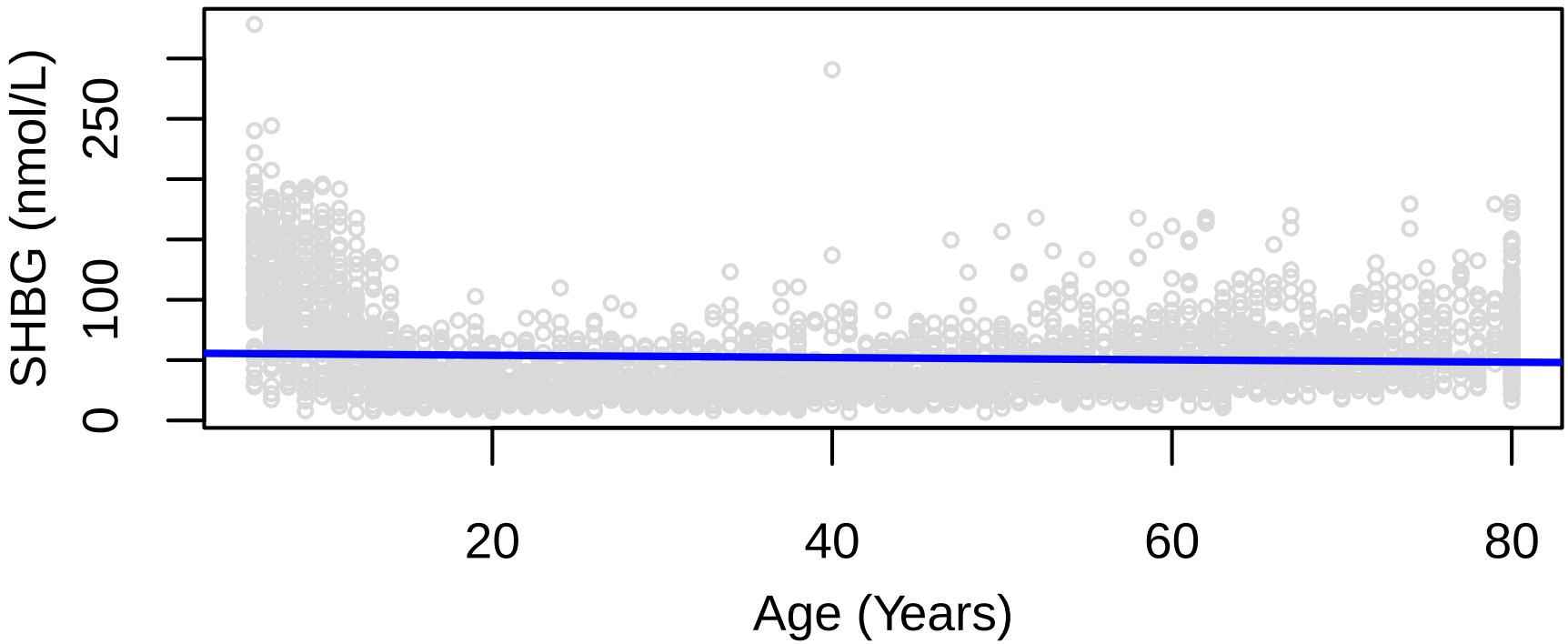


*Note: NHANES uses a complex, multistage, probability sampling design to mirror the US population, but will ignore this for our example. Also, all individuals 80+ are given an age of 80.

Linear Regression

One modeling strategy from class would be simple linear regression:

```
mod_lm <- lm(SHBG~RIDAGEYR, data=dat)
```

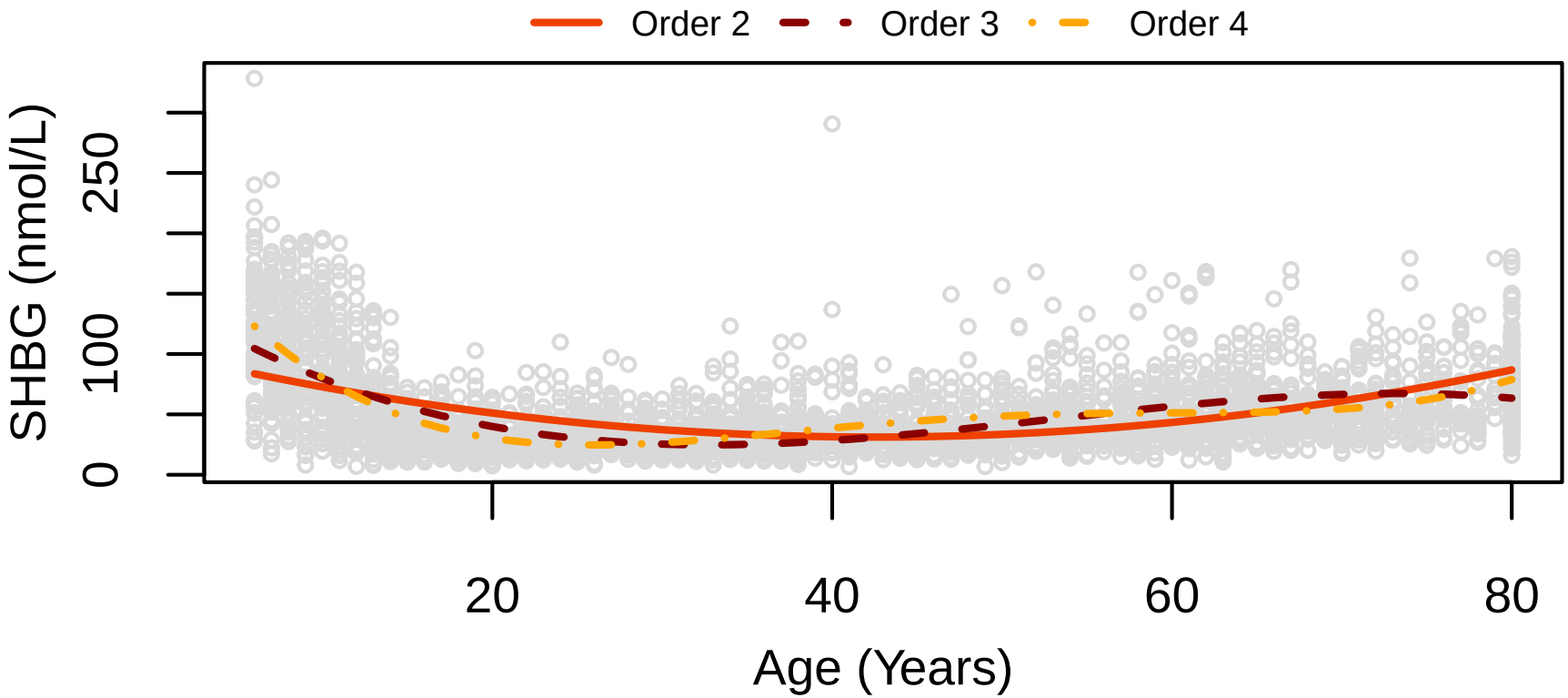


We can see that our SLR does not fit the data very well.

Polynomial Regression

Another modeling strategy may be polynomial regression (i.e., including higher order terms of X):

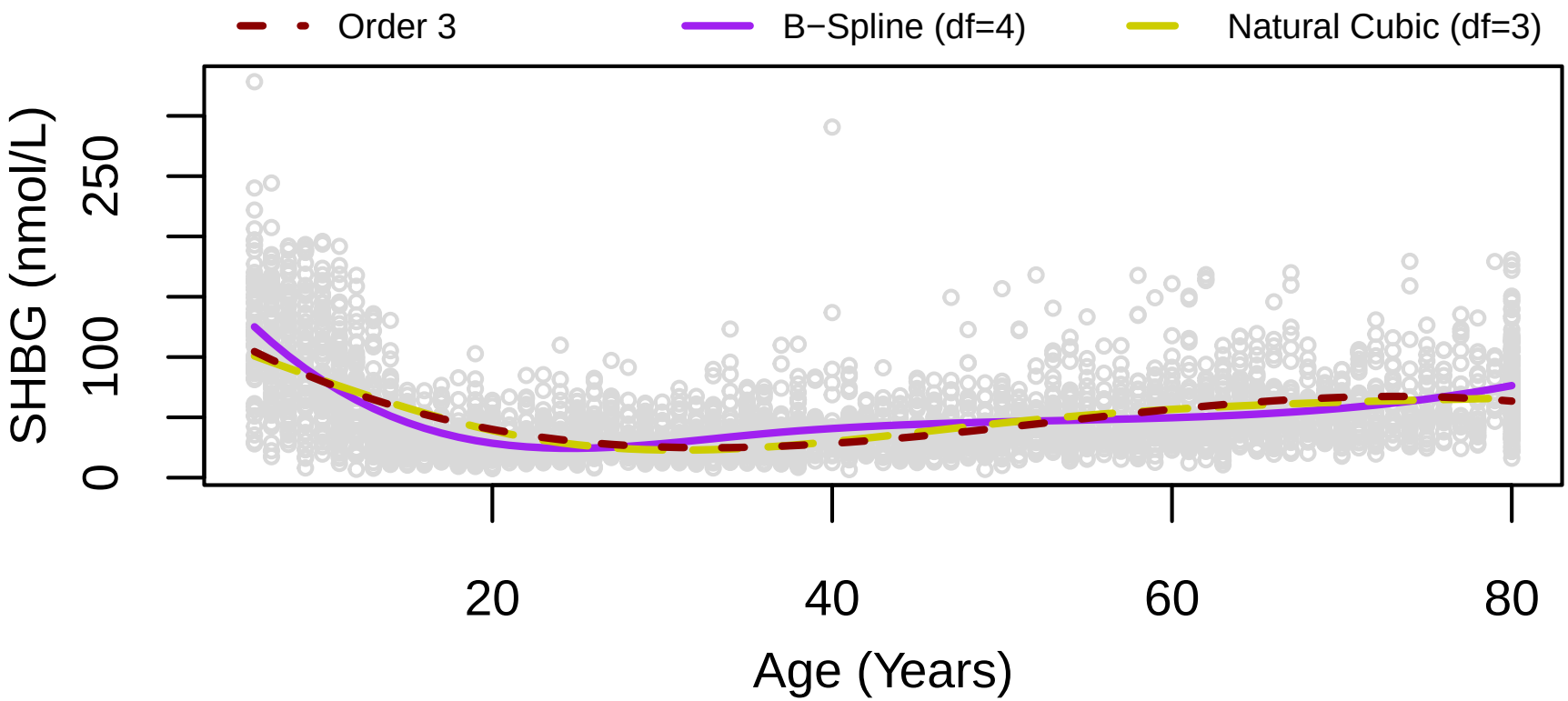
```
mod_poly2 <- lm(SHBG~poly(RIDAGEYR,2, raw=T), data=dat) # Order 2
```



We see the polynomial models are better fits than the SLR, but there may be concerns with poor fit (Order 2) or overfitting (Order 4).

Polynomial Splines

A more flexible alternative to polynomial regression that we will explore in this lecture is the use of *polynomial splines* to flexibly model the association of age with SHBG:



Introduction to Splines

Splines

Splines allow us to mathematically reproducible flexible shapes.[1]

In linear regression, splines provide flexibility to model continuous predictors that have non-linear trends. The concept is that *knots* are placed at several places in the data range of X with adjacent functional pieces joined together.

There are many types of splines, but in this lecture we will focus on B-splines and natural cubic splines. The type of polynomial and number/placement of knots is what then defines the type of spline.

[1] Many of our details come from: Perperoglou, A., Sauerbrei, W., Abrahamowicz, M., & Schmid, M. (2019). A review of spline function procedures in R. *BMC medical research methodology*, 19(1), 1-16.

Building from Polynomial Regression

Our polynomial regression model with order 3 is:

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \beta_3 X_i^3 + \epsilon_i$$

We can think of this also as:

$$Y_i = \beta_0 + \beta_1 u_i + \beta_2 u_i^2 + \beta_3 u_i^3 + \epsilon_i$$

where u is a function of X called a *basis* function. In the case of polynomial regression, it can be represented by a matrix as:

$$U = \begin{bmatrix} 1 & X_1 & X_1^2 & X_1^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & X_n & X_n^2 & X_n^3 \end{bmatrix}$$

Building from Polynomial Regression

$$Y_i = \beta_0 + \beta_1 u_i + \beta_2 u_i^2 + \beta_3 u_i^3 + \epsilon_i$$

can also be represented as

$$Y_i = f(X_i) + \epsilon$$

where $f()$ is some function/transformation of the predictor.

One limitation of polynomial regression is non-locality, where the fitted regression line at any arbitrary point X_0 depends on the data across the entire range. Therefore, changes to observed values near the boundary (e.g., min or max of X) can lead to changes in the fitted function far from that value.

Building from Polynomial Regression

As an alternative to fitting a global polynomial over the entire range of X, we can instead partition X into smaller intervals based on an arbitrary number and position of points τ (i.e., *knots*) and fit localized polynomials. This is the general concept behind splines.

If we define a set of knots, $\tau_1 < \cdots < \tau_K$, over the range of X, our spline $f(X)$ will be a smooth function of polynomial of degree d.

We can obtain more flexible curves by either increasing the number of knots and/or the degree of the polynomial. However, we have to be balance the bias-variance trade-off:

- If we increase the number of knots and/or d too greatly, we may overfit the data and increase our variance.
- If we choose too few knots and/or a low polynomial degree, we may poorly fit the data and increase our bias.

B-Splines (Technical)

The B-spline basis is a common spline basis and can be fit in `R` using the `splines::bs()` function.

The B-spline basis is based on the knot sequence:

$$\xi_1 \leq \cdots \leq \xi_d \leq \xi_{d+1} < \xi_{d+2} < \cdots \xi_{d+K+1} < \xi_{d+K+2} \leq \xi_{d+K+3} \leq \cdots \leq \xi_{2d+K+2}$$

The "inner knots" are represented by $\xi_{d+2} = \tau_1, \cdots, \xi_{d+K+1} = \tau_K$.

The "boundary knots" are defined as $\xi_{d+1} = a$ and $\xi_{d+K+2} = b$.

The choice of additional knots $\xi_1, \cdots, \xi_d$ and $\xi_{d+K+3}, \cdots, \xi_{2d+K+2}$ is somewhat arbitrary, with a common strategy being to set them equal to the boundary knots.[2]

[2] Perperoglou, A., Sauerbrei, W., Abrahamowicz, M., & Schmid, M. (2019). A review of spline function procedures in R. *BMC medical research methodology*, 19(1), Page 5.

B-Splines (Technical)

From Perperoglou, et al.,[3], for $d > 0$, B-spline basis functions of degree d (denoted by $B_k^d(x)$) are defined by the recursive formula:

$$B_k^d(x) = \frac{x - \xi_k}{\xi_{k+d} - \xi_k} B_k^{d-1}(x) - \frac{\xi_{k+d+1} - x}{\xi_{k+d+1} - \xi_{k+1}} B_{k+1}^{d-1}(x)$$

for $k = 1, ..., d + K + 1$ where

$$B_k^0(x) = \begin{cases} 1 & \xi_k \leq x < \xi_{k+1} \\ 0 & \text{else} \end{cases}$$

and $B_k^0(x) \equiv 0$ if $\xi_k = \xi_{k+1}$.

[3] Perperoglou, A., Sauerbrei, W., Abrahamowicz, M., & Schmid, M. (2019). A review of spline function procedures in R. *BMC medical research methodology*, 19(1), Page 1-16.

Natural Cubic Splines (Technical)

Polynomial splines like the B-spline can be erratic on the boundaries of the data.

To address this potential limitation, natural splines represent cubic splines with additional constraints that they are linear in the tails of the boundary knots: $(-\infty, a], [b, +\infty)$.

This constraint is achieved by requiring the spline function f to satisfy that both the second and third derivative equal 0: $f'' = f''' = 0$.

This requirement leads to four additional constraints that a natural spline basis on K knots has $K + 1$ degrees of freedom.[4]

[4] Perperoglou, A., Sauerbrei, W., Abrahamowicz, M., & Schmid, M. (2019). A review of spline function procedures in R. *BMC medical research methodology*, 19(1), Page 1-16.

Spline Interpretations

One challenge of splines is interpretation of regression coefficients related to the terms included in the spline. This is a similar challenge to polynomial regression.

For a primary explanatory variable (PEV) of interest, it may be most advantageous to focus on graphical summaries of the data and overall tests of the contribution of the PEV modeled by the spline.

In multiple linear regression models, you may be using splines to adjust for other variables beyond the PEV. In this case, we can interpret our PEV in the same way we do for a multiple linear regression (i.e., for a 1-unit increase in X_{PEV} there is an average change of $\hat{\beta}_{PEV}$ in our outcome, *after adjusting for all other variables in the model*).

Example in R

R Code

Let's revisit our NHANES example and see some properties of the B-splines and natural cubic splines with code.

We will examine:

- General code to fit/evaluate
- Changing the degrees of freedom (i.e., number of knots)
- Changing the polynomial degree
- Plotting with confidence intervals

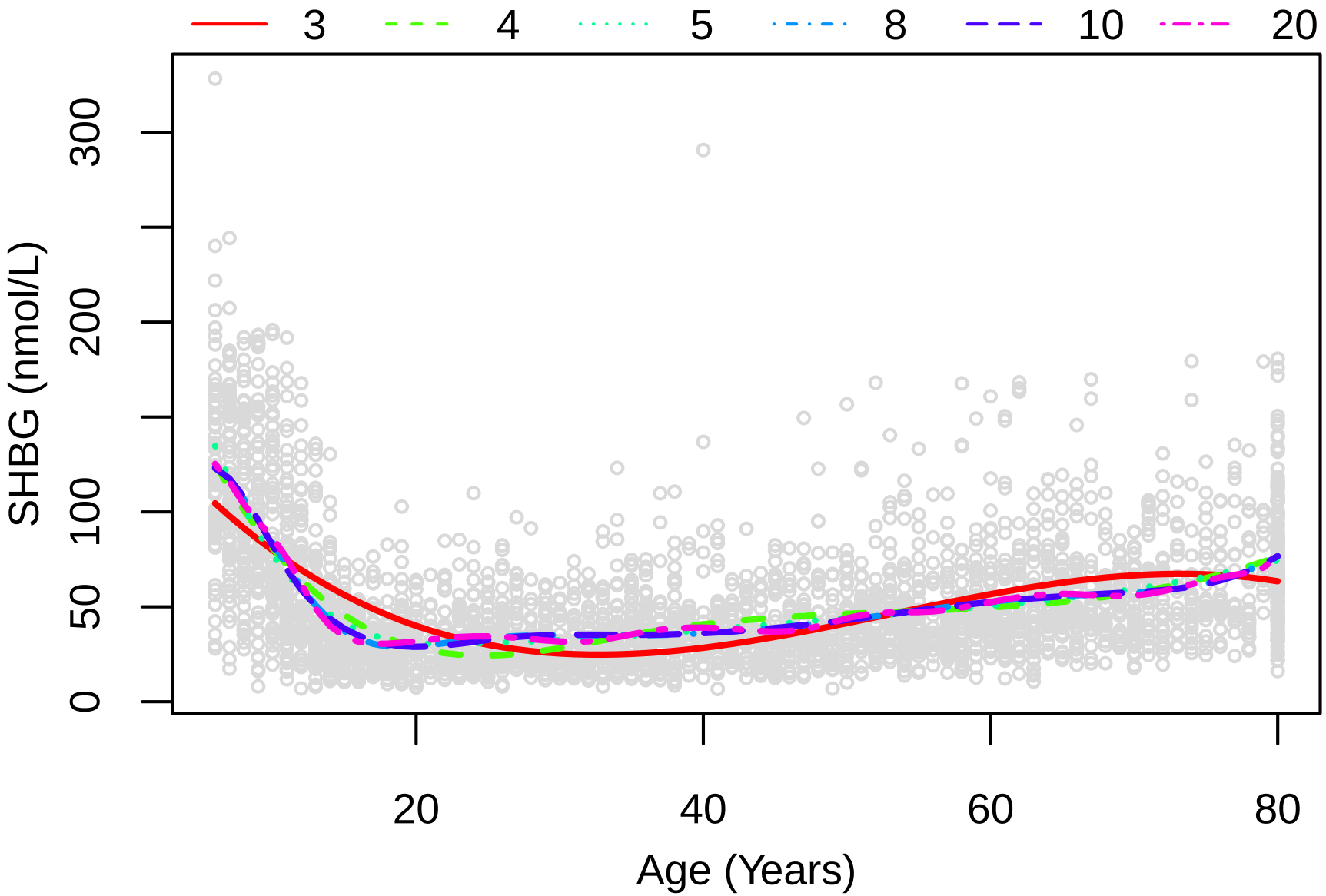
First we will load the `splines` package included in base `R`:

```
library(splines)
```

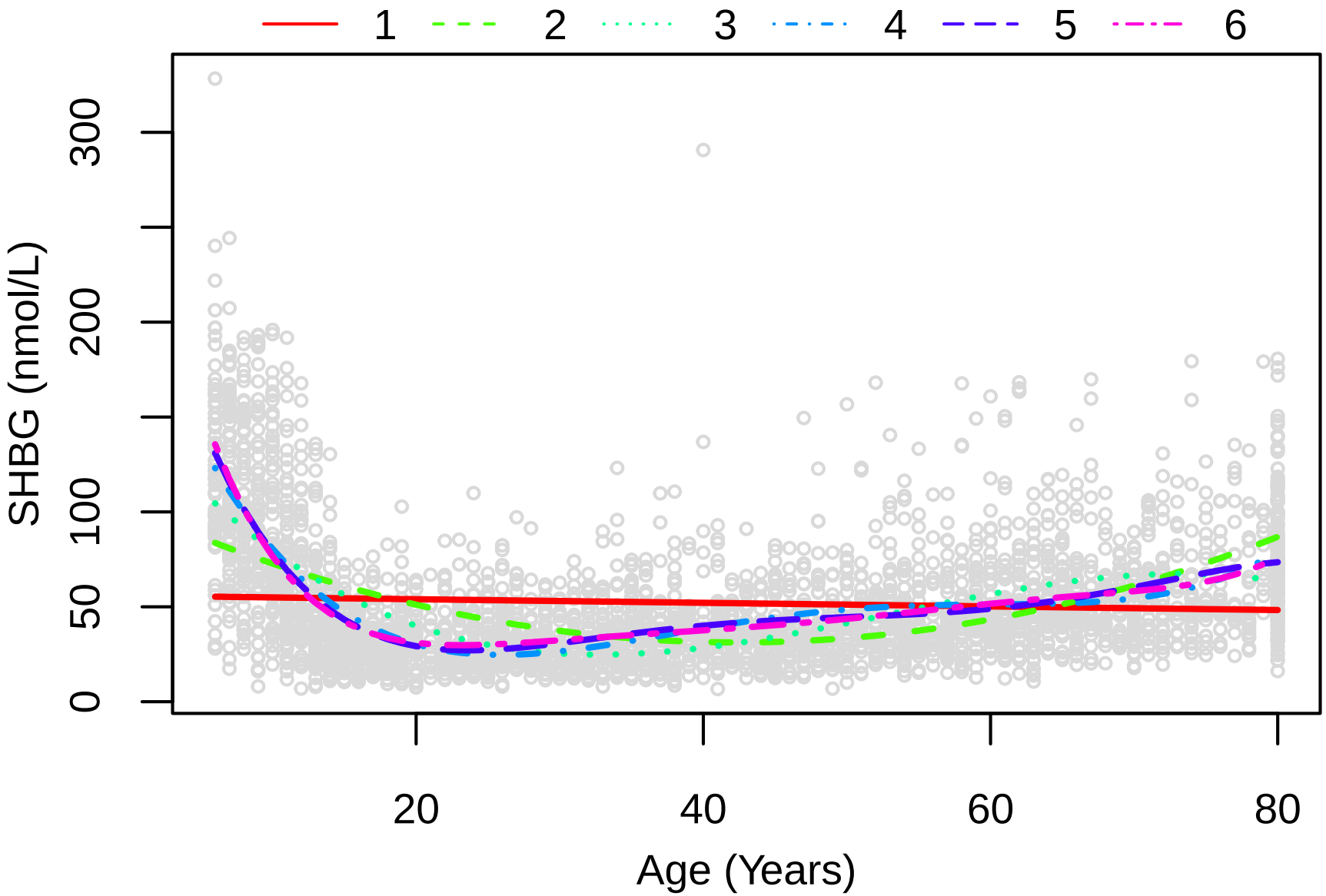
bs and ns Splines Code

```r
# Model with defined by degrees of freedom
lm(SHBG ~ bs(RIDAGEYR, df=4), data=dat)
lm(SHBG ~ ns(RIDAGEYR, df=4), data=dat)

# Model with B-spline defining degree of polynomial (default is 3)
lm(SHBG ~ bs(RIDAGEYR, df=4, degree=6), data=dat)
# ns assumes cubic polynomial terms, no degree argument

## Predicting values for fitted regression
mod <- lm(SHBG ~ bs(RIDAGEYR), data=dat) # fit model
newdat <- data.frame(RIDAGEYR = 6:80) # create DF for use in predict
y_pred <- predict(mod, newdata=newdat) # predict Y-hat
y_pred_withCI <- predict(mod, newdata=newdat,
                         interval='confidence') # include 95% CI
```

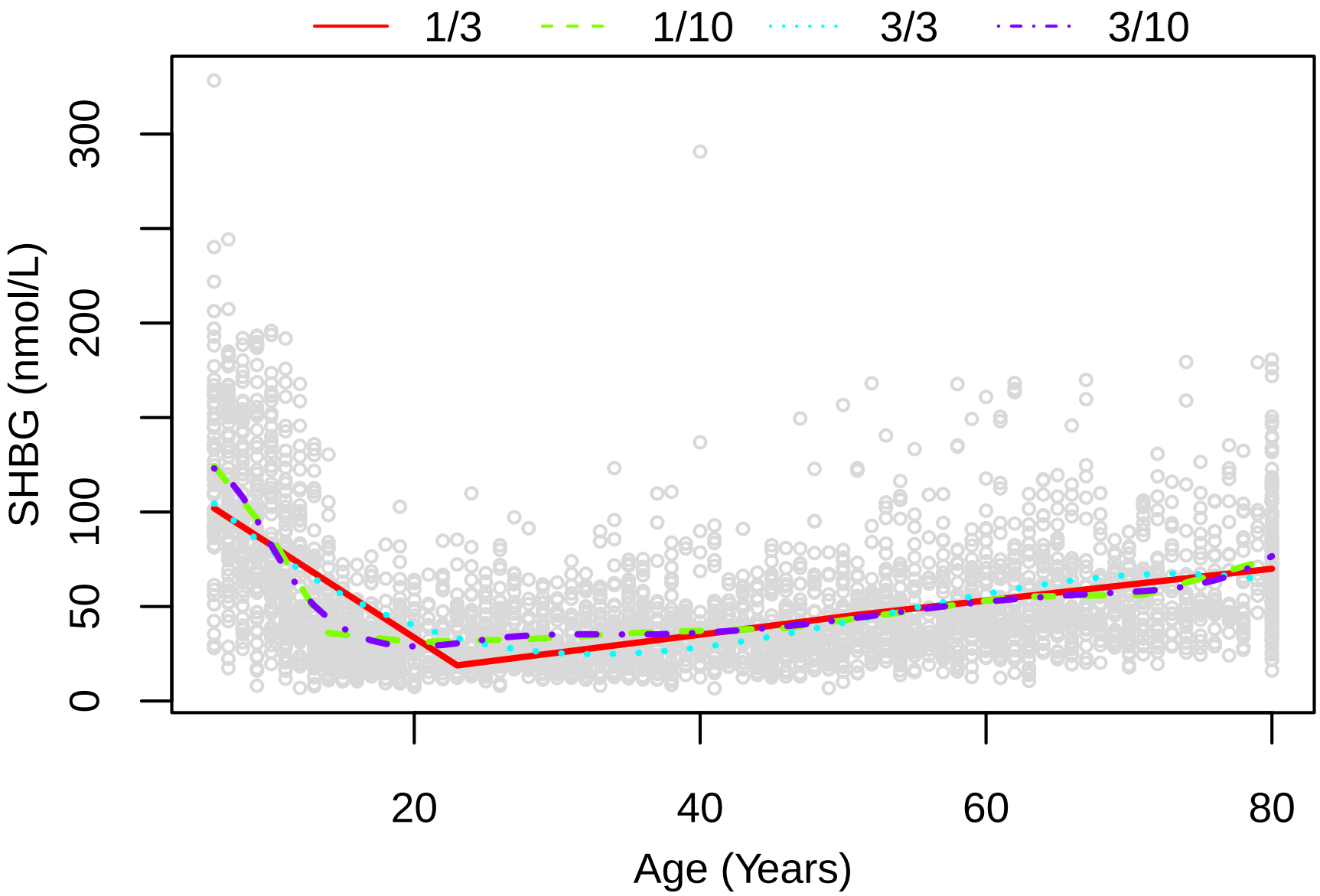
B-spline Examples: Degrees of Freedom



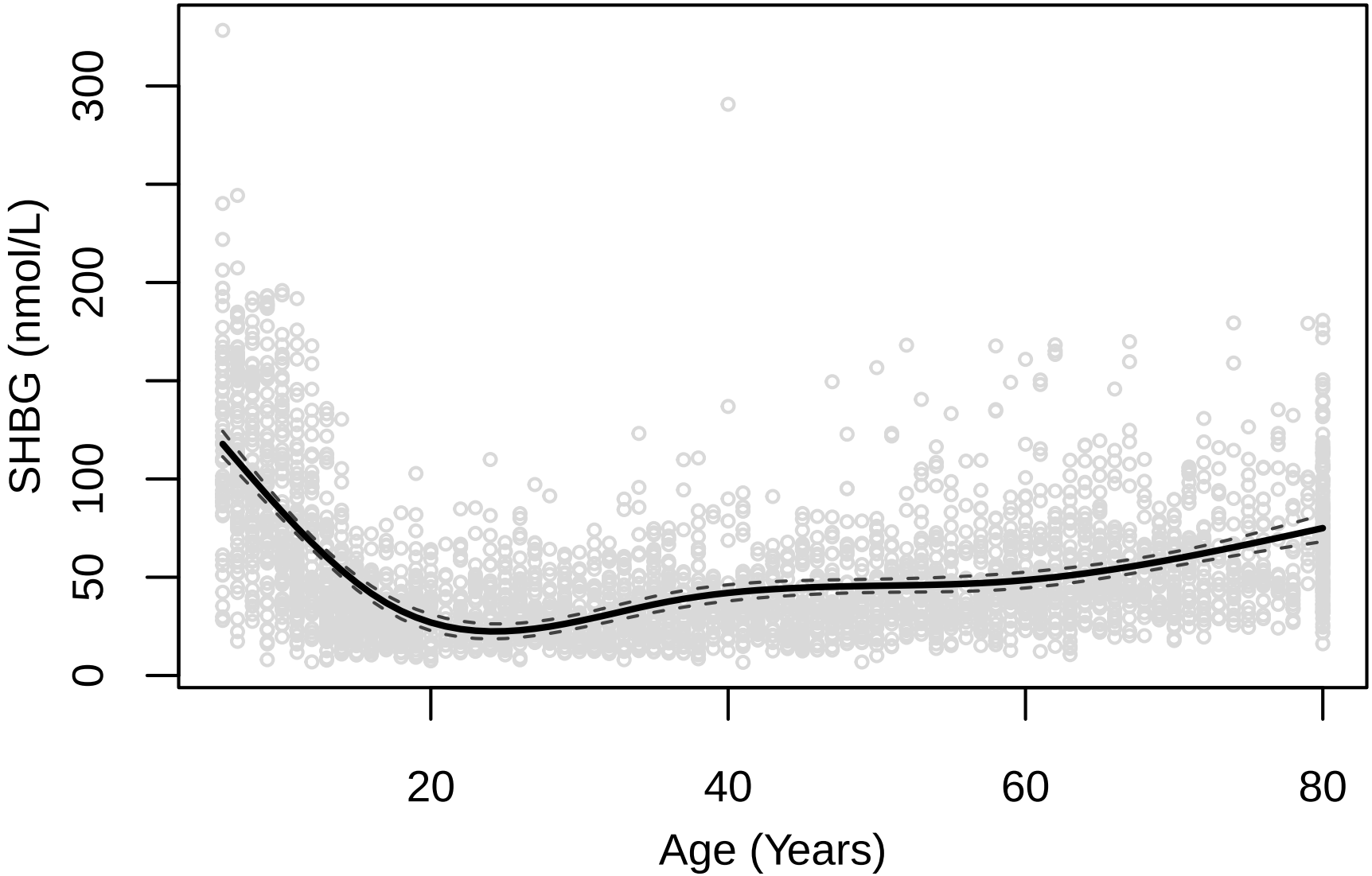
B-spline Examples: Polynomial Degree



B-spline Examples: Polynomial Degree/DF



Natural Cubic Spline Example: 99.99% CI



Given large n, using 99.99% CI to show interval on plot for `ns(RIDAGER, df=4)` example.

Choice of "Optimal" Parameter for Smoothing

We will not delve too deeply into example of how to select the optimal parameter for smoothing, but various approaches exist (some of which are covered in our class more generally):

- Use of visual evaluation (e.g., seeing that the data does not appear to be overfit)
- Model selection criterion (e.g., AIC, AICc, BIC, etc.) to compare models with different parameters (minimized AIC is preferred)
- Cross validation (CV) by dividing the N data points into K groups/folds for train/test set evaluation

Splines Summary

Splines provide a flexible approach to modeling non-linear predictors that may appropriately account for their association with an outcome.

We saw that different parameters led to different fits, some of which may be over- or under-fitting the data.

There are numerous types/approaches to splines that go well beyond the introduction here. Feel free to explore smoothing splines, penalized splines, etc. to consider their abilities to handle non-linear data in different settings.

There are also alternatives to using splines that we discuss in other lectures, including polynomial regression or segmented (change point) regression models.