apts.ac.uk
Academy for PhD Training in Statistics

Claire Miller, Tereza Neocleous
School of Mathematics & Statistics
The University of Glasgow

Flexible Regression

Lecture Notes

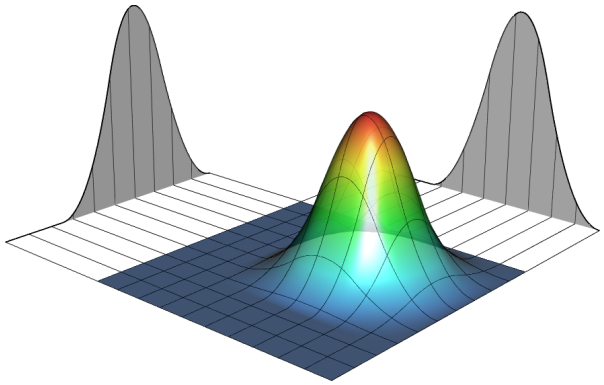


Table of Contents

1

Introduction

1.1 What this course is about?

This APTS course will cover a variety of methods which enable data to be modelled in a flexible manner. It will use and extend a variety of topics covered in earlier APTS courses, including

- linear models, including the Bayesian version;
- generalised linear models;
- R programming;
- matrix computations, Taylor series expansions and standard asymptotic methods;
- confidence intervals/hypothesis testing.

The main emphasis will be on regression settings, because of the widespread use and application of this kind of data structure.

As with any statistical topic, a rounded treatment involves a variety of approaches, including

- clear understanding of the underlying concepts;
- technical understanding of methods, with an exploration of their properties;
- appreciation of the practical computational issues;
- some knowledge of the tools available to fit relevant models in R;
- understanding of how these models can bring insight into datasets and applications.

The aim is to reflect all of these aspects in the course, but to varying degrees in different sections. There will not be time to cover all the material in the notes and some of the material is intended to provide pointers to topics which it might be of interest to explore in the context of your own research.

1.2 Broad concepts

The term 'flexible regression' refers to a wide range of methods which provide flexibility in the nature of the relationship being modelled. This APTS course will start with univariate smoothing and progress through standard forms of nonparametric regression to state-of-the art modelling tools (including quantile regression) which can be applied in a wide variety of settings. The models of interest in this course enable:

– flexibility in the mean;
– flexibility in the response quantile.

In particular, we will achieve this through smoothing (initially considering nonparametric regression) and quantile regression.

In parametric modelling (*e.g.* estimating the rate of a Poisson distribution using linear regression) we assume that we know the data generating process up to a finite number of parameters. In 'flexible' modelling, the term nonparametric is used to mean that the relationships or patterns of interest cannot be expressed in specific formulae which involve a fixed number of unknown parameters. We want to fit a function to data, without making such a strict parametric assumption. All we are willing to assume is typically that the function of interest is sufficiently smooth. More formally speaking, this corresponds to working with an infinite-dimensional parameter space. This takes us outside of the standard framework for parametric models. Additionally, in some circumstances regression models under standard assumptions will not adequately describe the distribution of the response. In such situations, relationships at different quantiles of the response distribution may be of interest. This course will discuss how we can introduce 'flexibility' into modelling the mean and nature of response modelled.

On a side note, the term nonparametric is sometimes used in the narrower setting of simple statistical methods based on the ranks of the data, rather than the original measurements. This is not the sense in which it will be used here.

1.3 Models of interest

1.3.1 Flexibility in the mean

In general, for a single explanatory variable with data $x_1, \ldots, x_n$, and response data $y_1, \ldots, y_n$ we can write a regression model as:

$$Y_i = f(x_i, \boldsymbol{\beta}) + \varepsilon_i.$$

The function $f(\mathbf{x}, \boldsymbol{\beta})$ describes the relationship between the response and the predictor variable, this might take the form of a straight line or some other function, which has parameters $\boldsymbol{\beta}$. The problem is to estimate this function f. Initially, in this course we will generally assume that,

$$\mathbb{E}(\varepsilon_i) = 0 \quad \text{and} \quad \text{Var}(\varepsilon_i) = \sigma^2$$

for all i, where σ^2 does not depend on any other unknown or on x_i, and x_i are assumed to be recorded without error. We also initially generally assume that $\varepsilon_i \sim \mathsf{N}(0, \sigma^2)$ and usually that ε_i and ε_j are uncorrelated for $i \neq j$, (i.e. independent identically distributed, i.i.d).

For Gaussian data we have a least squares loss function which we use to estimate the parameters $\boldsymbol{\beta}$ in our model, i.e. we choose $\boldsymbol{\beta}$ to minimise

$$\sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2.$$

Standard regression is sometimes referred to as mean regression, because the mean minimises the squared loss.

Previous APTS courses have considered linear and non-linear functions and the inclusion of both fixed and random effects in a regression model. In this course we will extend this by allowing $f()$ to be a data driven smooth function.

For a general smooth function $f()$ we refer to the approach as **nonparametric regression** (Chapter 2). This extends to **(generalised) additive models** (GAMs) for more than one smooth covariate (Chapter 4), and such models can include univariate, bivariate (or possibly higher order) terms and be extended to distributions other than the normal.

For initial references here see: Hastie and Tibshirani (1990), Bowman and Azzalini (1997), Ruppert *et al.* (2003), Wood (2017).

1.3.2 Flexibility in the response quantile

In some circumstances regression methods based on standard distributional assumptions will not capture all aspects of the distribution of the response variable of interest.

Usually regression models are based on a covariate-based model assumption for the mean only. However in some situations not just the mean, but also the spread and the shape of the distribution of the response depend on covariates. Therefore, additionally in this course we will consider **quantile regression** (Chapter 3) and combine this with

approaches which allow smooth functions for the covariates, to introduce **generalised additive quantile regression models** (Chapters 3 & 5).

When the quantity of interest is just one quantile it is easiest to fit a **quantile regression** model. Suppose we have data $\{(y_1, x_1), \ldots, (y_n, x_n)\}$ and a predictor function $f(\mathbf{x})$ which depends on parameters $\boldsymbol{\beta}$. Instead of using the least squares loss function above, if we were to use the absolute loss

$$\sum_{i=1}^{n} |y_i - f(\mathbf{x}_i)|$$

we would obtain median regression (also known as least absolute deviations regression).

Quantile regression is based on minimising,

$$\sum_{i=1}^{n} \rho_\tau(y_i - f(\mathbf{x}_i))$$

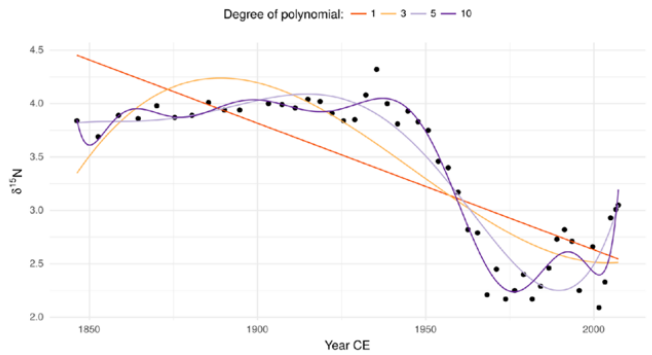
and results in an estimate of the τ-th quantile of the response distribution, where $\rho_\tau(\cdot)$ is the so-called check function,

$$\rho_\tau(u) = \begin{cases} \tau u & \text{if u } > 0 \\ (\tau - 1)u & \text{if u} \leq 0. \end{cases}$$

For initial references here see: Koenker (2005), Koenker *et al.* (2017).

1.4 Examples

Example 1.1. The following plot gives a very basic illustration of 'flexible regression'.

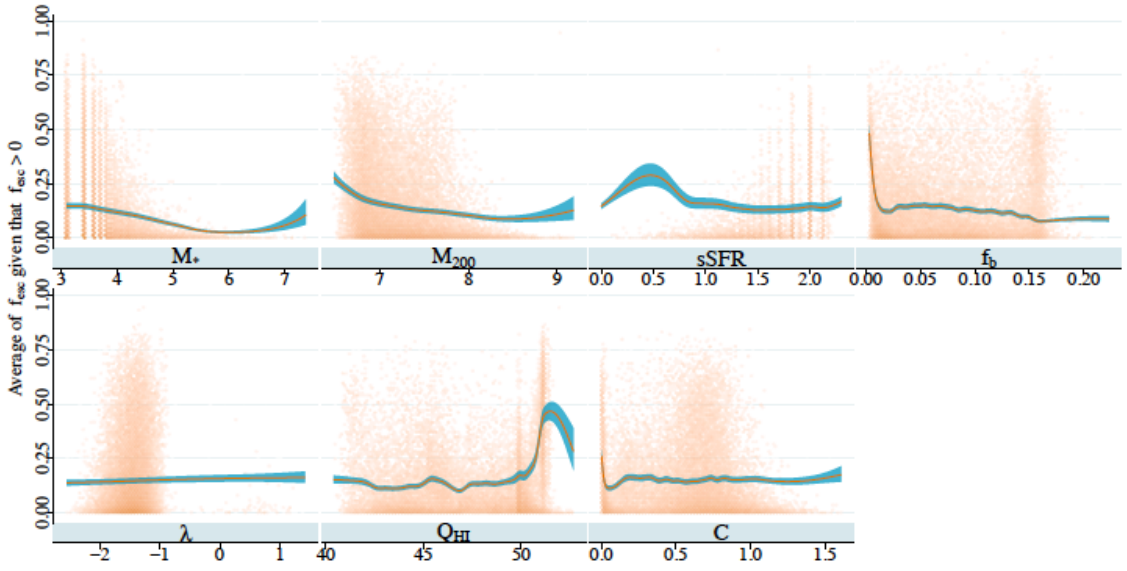


◁

The black dots refer to the data and the models fitted to try to explain the main patterns in the data, with different degrees of polynomial, are displayed. The higher the degree of polynomial in the model, the more flexible it can be[1].

Example 1.2. This next plot demonstrates a generalised additive model from the paper '*A case study of hurdle and generalized additive models in astronomy: the escape of ionizing radiation*' from May 2018 (Hattab *et al.*, 2018). In the paper, the authors use an array of different flexible models to understand the Epoch of Reionization (EoR), the first generation of stars residing in primeval galaxies to produce ultraviolet ionizing photons in a period when the cosmic gas changed from a neutral state to an ionized one. The authors state that a pivotal aspect to understaning the EoR is to account for non-linear relationships between the fraction of ionzing photons capable to escape dark haloes (also known as the escape fraction) and the physical properties of the galaxy... phew!

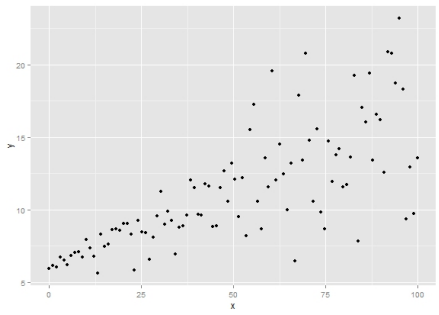
The authors estimate the mean curves relating the escape fraction response to various galaxy properties. The shaded areas with faint line depict 95% confidence intervals and the estimated mean (while varying only one galaxy property and holding the other properties fixed at their median values), and the cloud of partial residuals are laid out in the background.



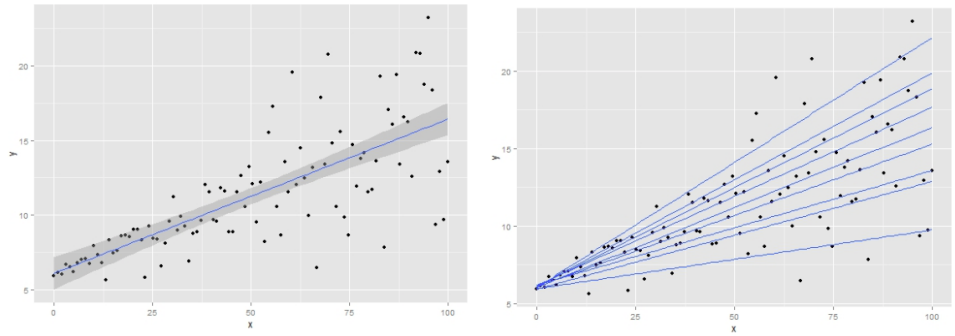
◁

[1] This example is taken from: `https://github.com/noamross/2017-11-14-noamross-gams-nyhackr`

Example 1.3. The following plots give some basic intuition around quantile regression. Let's imagine we have the data as follows.



Here, the cloud of data does not quite follow a linear increasing trend, the increasing variability is easy to see. As x gets bigger, y becomes more variable. This violates a key assumption in linear regression: normal errors with constant variance. As we can see in the plot in panel (a) below, with a linear regression fitted and 95% confidence intervals, linear regression provides a good estimate of y when x is close to 0. However, as x increases, the mean of y given x becomes less meaningful. As a predictor for y this model is not useful.

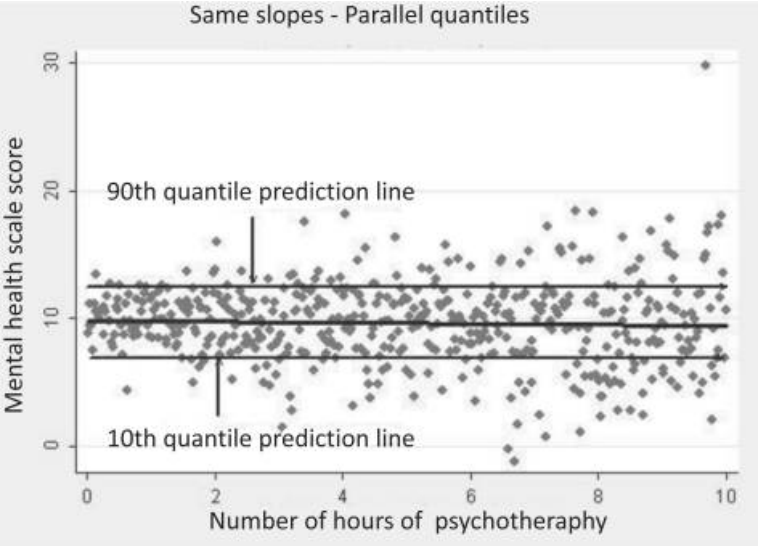


(a) linear regression (b) quantile regression

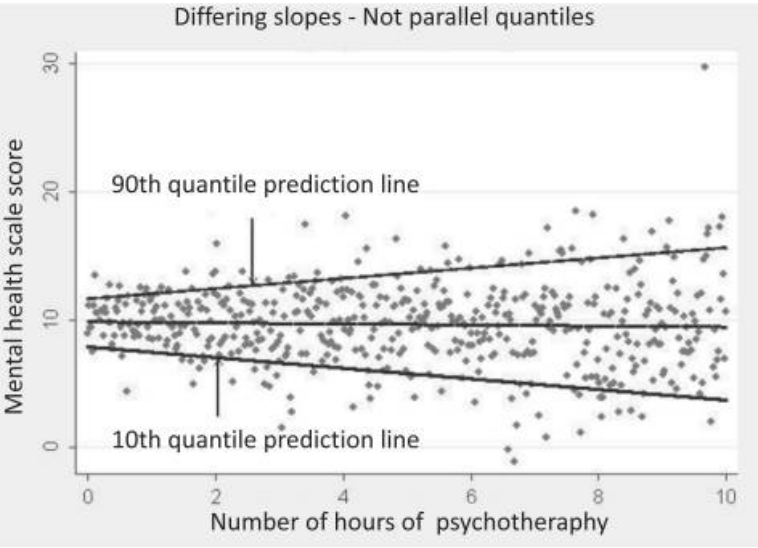
Therefore, we look at quantile regression in the plot in panel (b) above. The intercept estimate doesn't change much, but the slopes steadily increase through quantiles 0.1 through to 0.9 and we can take an average of these and provide confidence intervals. ◁

Example 1.4. Lastly, we can see the benefits of using quantile regression in comparison to ordinary least squares (OLS) regression with an example from the paper '*Thinking beyond the mean: a practical guide for using quantile regression methods for health services research*', February 2013, (Cook and Manning, 2013). In the paper, the authors fit the two models to describe the relationship between the number of hours attended of a hypothetical psychotherapy intervention (x-axis) and a fictitious scale of post-intervention

mental health (higher score indicates better mental health on the y-axis) for a group of 400 individuals. We look first at the OLS regression model.



The fitted line from OLS (shown above) is essentially flat, suggesting that there is no relationship between number of psychotherapy session-hours and mental health at follow up. In contrast, when quantile regression is used (shown below) this allows the slopes of the regression line to vary across quantiles of mental health scale and, although the median line is flat as before, the 90th quantile prediction line is significantly increasing whereas the 10th quantile prediction line is significantly decreasing. This suggests that the association between the hypothetical intervention and post-intervention mental health is positive for those with better post-intervention mental health but there is a negative association among those with poorer post-intervention mental health.



◁

2

Nonparametric regression

Regression is one of the most widely used modelling paradigms and this will be the main focus in the course. Here is an example which will be used to illustrate the initial discussion.

Example 2.1 (Great Barrier Reef data). A survey of the fauna on the sea bed lying between the coast of northern Queensland and the Great Barrier Reef was carried out. The sampling region covered a zone which was closed to commercial fishing, as well as neighbouring zones where fishing was permitted. The variables are:

`Zone`	an indicator for the closed (1) and open (0) zones
`Year`	an indicator of 1992 (0) or 1993 (1)
`Latitude`	latitude of the sampling position
`Longitude`	longitude of the sampling position
`Depth`	bottom depth
`Score1`	catch score 1
`Score2`	catch score 2

The details of the survey and an analysis of the data are provided by Poiner et al. (1997), *The effects of prawn trawling in the far northern section of the Great Barrier Reef*, CSIRO Division of Marine Research, Queensland Dept. of Primary Industries. ◁

The relationship between catch score (Score1) and longitude is of particular interest because, at this geographical location, the coast runs roughly north-south and so longitude is a proxy for distance offshore. We might therefore reasonably expect the abundance of marine life to change with longitude. The first of the three panels in Figure 2.1

shows that there is indeed a strong underlying negative relationship, with considerable variability also present. The middle panel summarises this in a simple linear regression which captures much of this relationship. However, if we allow our regression model to be more flexible in the mean then a more complex relationship is suggested in the right hand panel, with a broadly similar mean level for some distance offshore followed by a marked decline, possibly followed by some levelling off thereafter. This gives valuable informal and graphical insight into the data, but how can flexible regression models be constructed, and how can we use them to evaluate whether there is really evidence of non-linear behaviour in the data?

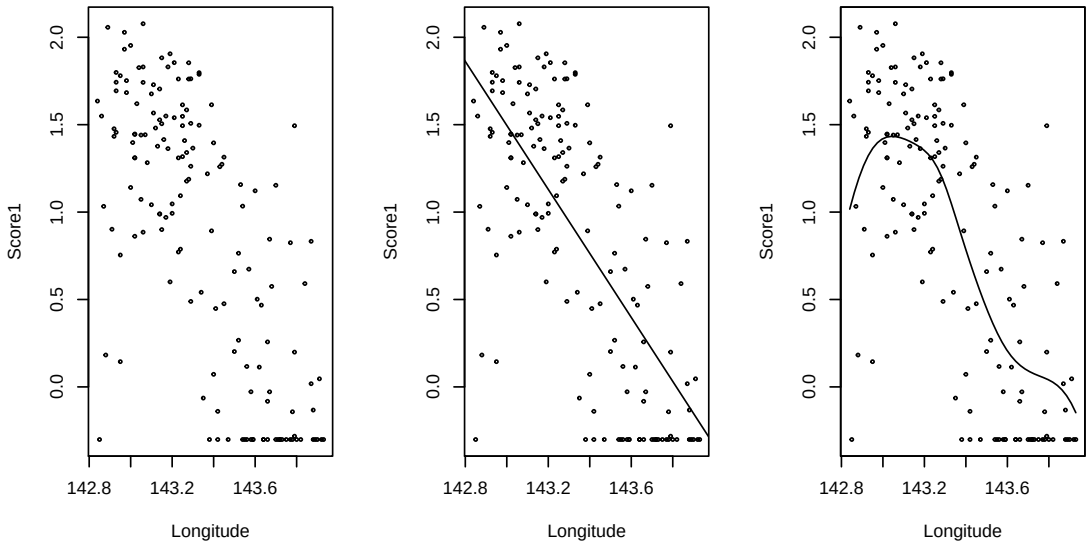


Figure 2.1. Reef fishing catch score against longitude (left) with fitted simple linear regression (middle) and fitted nonparametric regression (right).

2.1 Intro to nonparametric regression

A simple nonparametric model has the form

$$Y_i = f(x_i) + \varepsilon_i, \quad i = 1, \ldots, n$$

where the data (x_i, y_i) are described by a smooth curve f plus independent errors ε_i.

Smoothing techniques can be used to model the relationships between variables, i.e. to estimate $f()$, without specifying any particular form for the underlying regression function. Smoothers have two main uses:

Description - to aid 'visually' in the exploration of a relationship or pattern

Estimation - to estimate the dependence of the mean of Y on the predictor x.

The two key questions that arise regarding the definition of a smoother are:

– Which smoothing method should be used?
– What level of smoothing is appropriate?

We will start by considering the first question here to introduce appropriate smoothing methods and then focus more on the second question in Chapter 4.

In this course, we will briefly mention a local fitting approach to smoothing and then mainly focus on the application of spline based methods.

There is a variety of ways in which smooth curve estimates can be produced and it can sometimes reasonably be argued that the precise mechanism usually isn't too important and can be chosen for convenience/computational simplicity and efficiency.

2.2 A local fitting approach

One approach to fitting $f()$ is to take a model we know and fit it locally. For example, we can construct a *local linear regression*. This involves solving the least squares problem

$$\min_{\alpha,\beta} \sum_{i=1}^{n} \{y_i - \alpha - \beta(x_i - x)\}^2 \, w(x_i - x \, ; h)$$

and taking as the estimate at x the value of $\hat{\alpha}$, as this defines the position of the local regression line at the point x. The weight function, $w(x_i - x \, ; h)$, is a kernel function which we formulate in a similar way to that which we introduced for density functions in the preliminary material. This has an appealing simplicity and it can be generalised quite easily to other situations. This was the approach used to produce the nonparametric regression of the Reef data in the right hand panel of Figure 2.1.

An even simpler approach is to fit a local mean. Specifically, at any point of interest x, we choose our estimator of the curve there as the value of μ which minimises

$$\sum_{i=1}^{n} \{y_i - \mu\}^2 w(x_i - x; h)$$

and this is easily shown to produce the 'running mean'

$$\hat{f}(x) = \frac{\sum_{i=1}^{n} w(x_i - x; h) \, y_i}{\sum_{i=1}^{n} w(x_i - x; h)}.$$

If we do the algebra to minimise the sum-of-squares in the local linear approach, then an explicit formula for the local estimator can be derived as

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{\{s_2(x;h) - s_1(x;h)(x_i - x)\}w(x_i - x;h)y_i}{s_2(x;h)s_0(x;h) - s_1(x;h)^2},$$

where $s_r(x;h) = \{\sum(x_i - x)^r w(x_i - x;h)\}/n$.

In both the local mean and the local linear cases, the estimator is seen to be of the form $\sum_i \kappa_i y_i$, where the weights κ_i sum to 1. There is a broad sense then in which even the local linear method is 'locally averaging' the data. In fact, many other forms of nonparametric regression can also be formulated in a similar way.

2.2.1 Some simple properties

One question which immediately arises is whether it matters very much which form of nonparametric smoothing is used. Sometimes computational and other issues may constrain what choices are practical. However, if we take the simple local mean and local linear examples, what principles can we use to guide our choice? Deriving expressions which capture simple properties such as bias and variance is an obvious place to start.

We will start with the local mean estimator. The exploration will be a little informal, without the full technicality of formal proofs. The aim is to identify the properties of the estimator in conceptual form. If the numerator and denominator of the local mean estimator are both scaled by $1/n$, then the denominator has a familiar form, namely a kernel density estimator (which we saw in the preliminary material). Following from the preliminary material, this has expectation

$$\mathbb{E}\left\{\frac{1}{n}\sum_i w(x_i - x;h)\right\} = g(x) + \frac{h^2}{2}g''(x) + o(h^2),$$

(we will use $g()$ for our general function here to distinguish from the function $f(x)$ that we are estimating). As in the preliminary material, we assume for convenience that the kernel function can be rewritten as $\frac{1}{h}w((x_i - x)/h)$ and w is a symmetric probability density function around 0 with variance 1. Turning now to the numerator, we have

$$\begin{aligned}
\mathbb{E}\left\{\frac{1}{n}\sum_i w(x_i - x;h)\, y_i\right\} &= \frac{1}{n}\sum_i \frac{1}{h}w\left(\frac{x_i - x}{h}\right) f(x_i) \\
&\approx \int \frac{1}{h}w\left(\frac{z - x}{h}\right) f(z)g(z) \quad \text{[integral approximation]} \\
&= \int w(u)f(x + hu)g(x + hu)du \quad \text{[change of variable]}
\end{aligned}$$

Now apply a Taylor series expansion to the terms involving $x + hu$, to give

$$f(x+hu) = f(x) + hu\, f'(x) + \frac{(hu)^2}{2} f''(x) + o(h^2),$$
$$g(x+hu) = g(x) + hu\, g'(x) + \frac{(hu)^2}{2} g''(x) + o(h^2).$$

Substituting these in and integrating over u gives

$$\mathbb{E}\left\{ \frac{1}{n} \sum_i w(x_i - x; h)\, y_i \right\} \approx f(x)g(x) + h^2 \left\{ \frac{1}{2} g(x) f''(x) + f'(x) g'(x) + \frac{1}{2} g''(x) f(x) \right\} + o(h^2).$$

Dividing both numerator and demoninator by $g(x)$ gives

$$\text{numerator: } f(x) + h^2 \left\{ \frac{1}{2} f''(x) + f'(x) \frac{g'(x)}{g(x)} + \frac{1}{2} \frac{g''(x)}{g(x)} f(x) \right\} + o(h^2)$$
$$\text{denominator: } 1 + \frac{h^2}{2} \frac{g''(x)}{g(x)} + o(h^2)$$

The dominant term in the mean of the ratio of numerator and denominator is the ratio of the means. Applying the series expansion for $(1+x)^{-1}$ allows the reciprocal of the denominator to be written as

$$1 - \frac{h^2}{2} \frac{g''(x)}{g(x)} + o(h^2).$$

Multiplying the different terms out, we have

$$\begin{aligned} \mathbb{E}\left\{ \hat{f}(x) \right\} &\approx \left\{ f(x) + h^2 \left\{ \frac{1}{2} f''(x) + f'(x) \frac{g'(x)}{g(x)} + \frac{1}{2} \frac{g''(x)}{g(x)} f(x) \right\} + o(h^2) \right\} \\ &\quad \left\{ 1 - \frac{h^2}{2} \frac{g''(x)}{g(x)} + o(h^2) \right\} \\ &= f(x) + h^2 \left\{ \frac{1}{2} f''(x) + \frac{f'(x) g'(x)}{g(x)} \right\} + o(h^2). \end{aligned}$$

Phew!

A similar sequence of manipulations gives an asymptotic expression for the variance as

$$\text{Var}\left\{ \hat{f}(x) \right\} \approx \frac{1}{nh} \left\{ \int w(u)^2 du \right\} \sigma^2 \frac{1}{g(x)},$$

where σ^2 denotes the variance of the error terms ε_i.

In the local linear case, the estimator can be written as $\sum_i a_i y_i / \sum_i a_i$, where $a_i = \frac{1}{n} \frac{1}{h} w(\frac{x_i - x}{h}) \{ s_2 - (x_i - x) s_1 \}$. Consider first s_1, which can be written as

$$\begin{aligned} s_1 &= \frac{1}{n}\sum_j \frac{1}{h} w\left(\frac{x_i - x}{h}\right)(x_j - x) \\ &\approx \int \frac{1}{h} w\left(\frac{x - x}{h}\right) g(z)(z - x)dz \\ &= \int w(u)hu\{g(x) + hug'(x) + o(h)\}du \\ &= h^2 g'(x) + o(h^2). \end{aligned}$$

By a similar argument,

$$s_2 \approx h^2 g(x) + o(h^2).$$

The weights a_i can then be approximated by

$$a_i \approx \frac{1}{n}\frac{1}{h} w\left(\frac{x_i - x}{h}\right) h^2\{g(x) - (x_i - x)g'(x)\}.$$

The mean of the estimator is $\mathbb{E}\left\{\hat{f}(x)\right\} = \sum_i a_i f(x_i)/\sum_i a_i$. Ignoring the term h^2 which cancels in the ratio, the numerator can be expressed as

$$\left\{g(x)^2 + \frac{h^2}{2} g(x)g'(x) - h^2 g'(x)^2\right\} f(x) + \frac{h^2}{2} g(x)^2 f''(x)^2,$$

after an integral approximation, a change of variable and a Taylor series expansion. By a similar argument, the denominator of $\mathbb{E}\left\{\hat{f}(x)\right\}$ can be approximated by

$$g(x)^2 + \frac{h^2}{2} g(x)g''(x) - h^2 g'(x)^2.$$

The principal term of the ratio then gives

$$\mathbb{E}\left\{\hat{f}(x)\right\} \approx f(x) + \frac{h^2}{2} f''(x).$$

So, after considerable work, a very simple expression has been achieved. Similar manipulations for the variance produces an expression which is exactly the same as that for the variance of the local mean estimator.

A comparison of the expressions for the local mean and local linear estimators is interesting. For example, the principal terms in the expression for the mean of the local linear estimator is not only simpler but also does not involve $g(x)$, both of which are attractive properties. This is one of the reasons that the local linear estimator is generally preferred over the local mean.

However, another issue concerns edge effects. These require more careful analysis to identify so, instead, we will use a simple illustration based on simulation, Figure 2.2.

The figure shows the results of repeatedly simulating 50 data points, equally spaced over $[0, 1]$, from the model $Y = x + \varepsilon$, where the standard deviation of the error terms is 0.1. For each set of simulated data, a nonparametric regression curve is plotted, using local mean (left) and local linear (right) estimators. Notice that at the ends of the sample space the local mean has strong bias, because there is data only on one side of the estimation point of interest. In contrast, the local linear method is unaffected. The same pattern is displayed in the lower plots, using the model $Y = x^2 + \varepsilon$ over the range $[-1, 1]$.

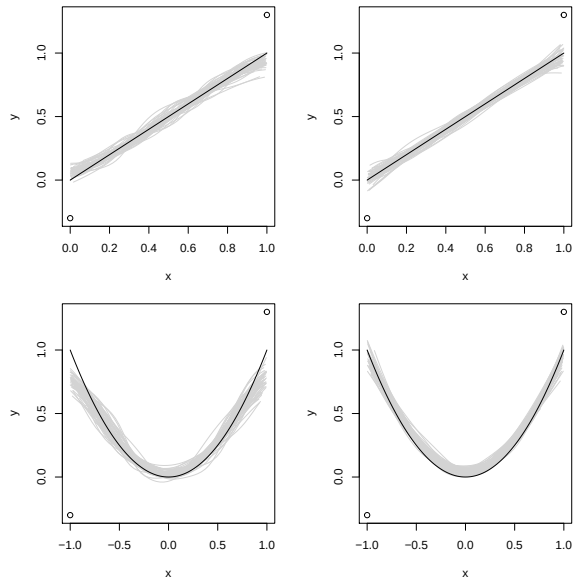


Figure 2.2. Repetition of 50 simulated data points [0,1] from a line of equality with error (top) and quadratic line with error (bottom). Local mean estimators (left) and local linear estimators (right) are used for the fitted lines.

With a little more theoretical work, a central limit theorem can be constructed to show that

$$\frac{\hat{f}(x) - f(x) - b(x)}{\sqrt{v(x)}} \to N(0, 1),$$

where $b(x)$ and $v(x)$ denote the bias and variance of $\hat{f}(x)$.

Following on from the discussions in density estimation, the performance of a non-parametric estimator can be summarised in the *mean integrated squared error*, defined as

$$\text{MISE} = \int \mathbb{E}\Big\{\hat{f}(x) - f(x)\Big\}^2 g(x)dx$$

and an optimal smoothing parameter can be defined as the value of h which minimises the asymptotic approximation of MISE, namely

$$h_{\text{opt}} = \left\{\frac{\gamma(w)\sigma^2}{\int [f''(x)]^2 g(x)dx}\right\}^{1/5} n^{-1/5}.$$

If we use this optimal smoothing parameter, then both the bias and the square root of the variance, which determines the rate of convergence, are of order $n^{-2/5}$. Notice that this rate of convergence is slower than the $n^{-1/2}$ which applies for parametric models.

Fan and Gijbels (1996) give further details on the theoretical aspects of local polynomial smoothing, also see Bowman and Azzalini (1997) for more details and examples for local polynomial regression.

2.2.2 Bias and variance trade-off

The properties derived and discussed above highlight that both bias and variance have to be considered when it comes to fitting a smooth curve to model the relationship between y and x. There is a trade-off between following the data closely (low bias, possibly large variance) and obtaining a smooth function (low variance, possibly large bias). However, the presence of bias in the estimator $\hat{f}$ will have the effect of inflating the size of the residual sum of squares. We will return to this when we consider 'how much to smooth?' in Chapter 4.

2.2.3 Local linear regression in R

Example 2.2. A local linear regression fit for the Reef data can be obtained using the R library `sm`.

```
library(sm)
sm.regression(trawl$Longitude, trawl$Score1, se = TRUE)
```

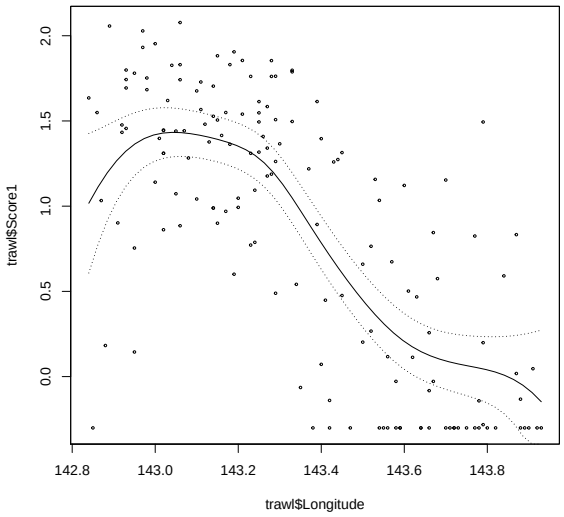


Figure 2.3. A flexible regression curve for the Reef data, with variability bands indicated.

With this, a normal density is 'typically' specified as the kernel to define the weights,

$$w(x_i - x; h) = \exp\left(-0.5\left(\frac{x_i - x}{h}\right)^2\right).$$

The **smoothing parameter** h is the standard deviation of a normal density.

Unfortunately, we can't easily produce confidence intervals for the curve because of the bias mentioned above. However, by adding and subtracting two standard errors at each point on the curve we can produce *variability bands* which express the variation in the curve estimate. In fact, we don't need to rely on the asymptotic formula for variance. More details are provided later.

◁

2.3 Regression splines

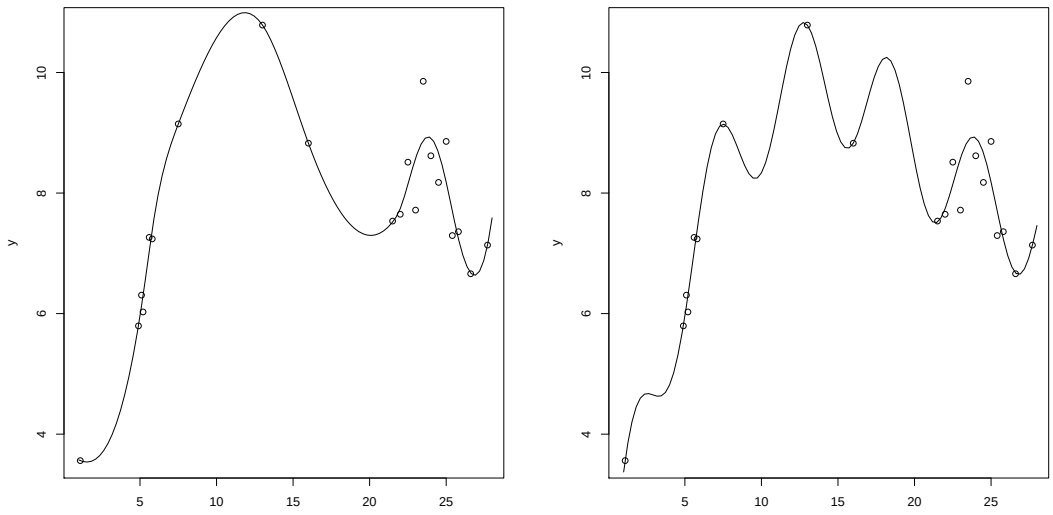
The second part of this chapter covers splines, which are one of the most popular tools for flexible modelling. This section discusses a number of more philosophical concepts, some of which we have already touched upon, initially starting with splines in one dimension.

Example 2.3. Figure 2.4 shows two smooth functions describing the relationship between the response y_i and the covariate x_i. In this example both functions yield the same fitted values $\hat{y}_i = \hat{f}(x_i)$. This also implies that the least-squares loss $\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$ is the same for both functions, i.e. the data alone does not tell us which function does a better job. There is no global answer to this question.

Which of the two functions appears better suited to us depends on the context and also to some extent our subjective choice. In most circumstances we would prefer the function in the left-hand panel as it is the "simpler" function. However, if we expect the signal to have a periodic component (say we are expecting a day-of-the-week effect) then we might prefer the function shown in the right-hand panel. ◁

What we have seen in the example is simply that the family of smooth functions is so large that observing a finite sample alone will not tell us enough to learn the function of interest $f(\cdot)$.

We need to provide additional information, which can be of different types:



(a) (b)

Figure 2.4. Two possible smooth functions modelling the relationship between the response Y_i and the covariate x_i. Note that both functions yield the same fitted values $\hat{y}_i = \hat{f}(x_i)$ and thus the same least-squares loss $\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$.

- We can assume that the function of interest $f(\cdot)$ comes from a more restricted family of functions. We might even assume a rich class of parametric models. We will use this idea when we are looking at splines based on truncated power series and B-splines.
- We express a preference for some functions over others (without looking at the data) and use this in the model fitting procedure. Typically we prefer a smooth function to a more wiggly function. In a frequentist setting, this leads to a penalty-based approach, or can be viewed as a Bayesian prior over the space of functions.

2.3.1 Polynomial regression

We will start by revising polynomial regression. To fix notation, we quickly state the simple linear regression model

$$\mathbb{E}(Y_i) = \beta_0 + \beta_1 x_i \qquad \text{for } i = 1, \ldots, n,$$

or equivalently, in matrix-vector notation,

$$\mathbb{E}(\mathbf{y}) = \mathbf{B}\boldsymbol{\beta} \qquad \text{with } \mathbf{y} = (Y_1, \ldots, Y_n)^\top \text{ and } \mathbf{B} = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}.$$

In this we call $\mathbf{B}$ our matrix of basis functions and $\boldsymbol{\beta}$ our vector of basis coefficients, and for the simple linear regression we have the basis functions:

$$B_0(x) = 1, B_1(x) = x.$$

The simple linear regression model can be extended into a polynomial regression model by including powers of the covariates x_i in the design matrix. The polynomial regression model

$$\mathbb{E}(Y_i) = \beta_0 + \beta_1 x_i + \ldots + \beta_r x_i^r \qquad \text{for } i = 1, \ldots, n,$$

just corresponds to linear regression using the expanded design matrix (matrix of basis functions)

$$\mathbf{B} = \begin{pmatrix} 1 & x_1 & \ldots & x_1^r \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \ldots & x_n^r \end{pmatrix},$$

where the basis functions are:

$$B_0(x) = 1, B_1(x) = x \ldots, B_r(x) = x^r.$$

We can then estimate $\boldsymbol{\beta}$ using the same techniques as used in multiple linear regression, i.e. the least-squares estimator is

$$\hat{\boldsymbol{\beta}} = (\mathbf{B}^\top \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{y}.$$

Polynomial regression is a very simple example of a basis expansion technique. We have simply replaced the design matrix of simple linear regression by an augmented design matrix. In the case of polynomial regression we have simply added powers of the x_i's.
The figures below illustrate a fitted simple linear regression with the corresponding basis:

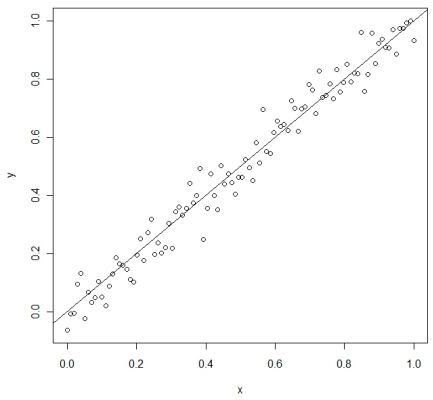


Figure 2.5. A simple linear regression line with underlying simulated data

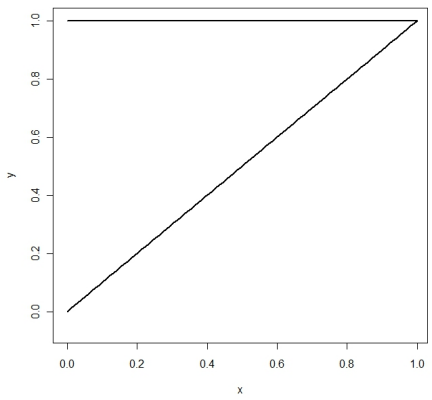


Figure 2.6. The basis functions for simple linear regression $1, x$

Many of the techniques covered in this section will be based on this idea of basis expansions. Polynomial regression can be a useful tool if a polynomial of very low order yields a sufficient fit to the data.

Example 2.4 (Glucose levels in potatoes). Figure 2.7 shows a quadratic regression model fitted to a simple data set from an experiment in which the glucose level in potatoes was measured over the course of several weeks. Given the small number of observations there is little need to go beyond a simple quadratic regression model. ◁

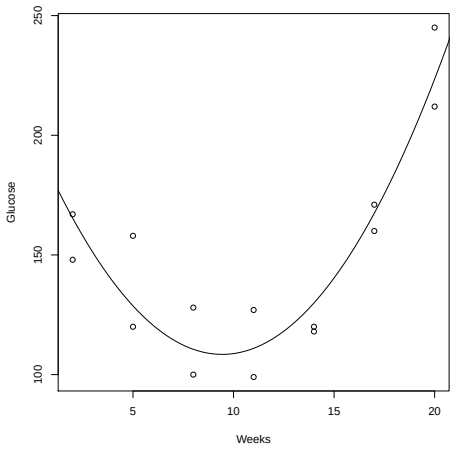


Figure 2.7. Glucose level in potatoes. The solid line is the fitted regression function obtained from quadratic regression.

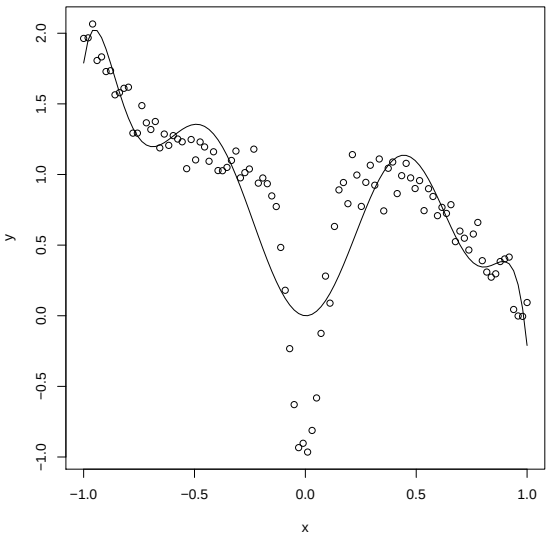
However, polynomial regression is not very well suited for modelling more complex relationships, as the following example shows.

Example 2.5. Consider the dataset simulated using the model

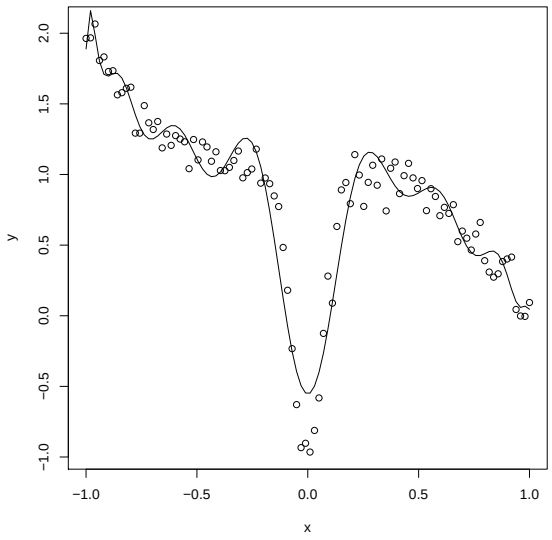
$$Y_i = 1 - x_i^3 - 2\exp(-100x_i^2) + \varepsilon_i$$

with $\mathbf{x} = (-1, -0.98, \ldots, 0.98, 1)$ and $\varepsilon_i \sim \mathsf{N}(0, 0.1^2)$. Figure 2.8(a) shows the data together with the fitted function obtained for a polynomial regression model of degree 10. The polynomial model of degree 10 is not flexible enough to capture the sharp dip around 0. If we increase the degree to 17 we can capture the dip better (panel (b)). However, the polynomial fit of degree 17 shows strong oscillations which are not supported by the data. Panel (c) shows the fitted regression function using a spline

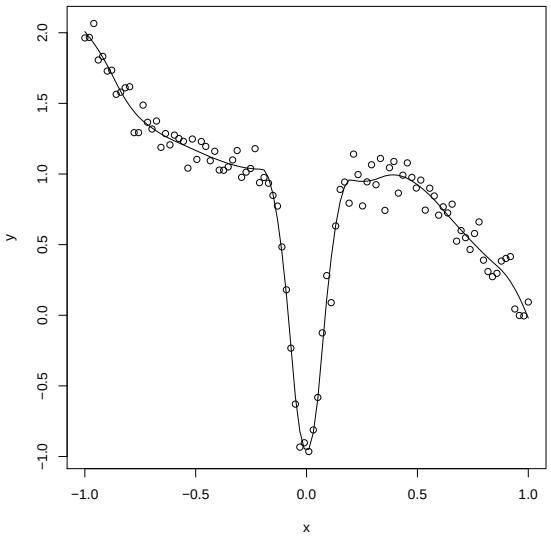
based model, which we will discuss later on in this chapter. The spline-based approach can capture the sharp dip much better and without yielding any oscillations.



(a) Polynomial regression of degree 10



(b) Polynomial regression of degree 17



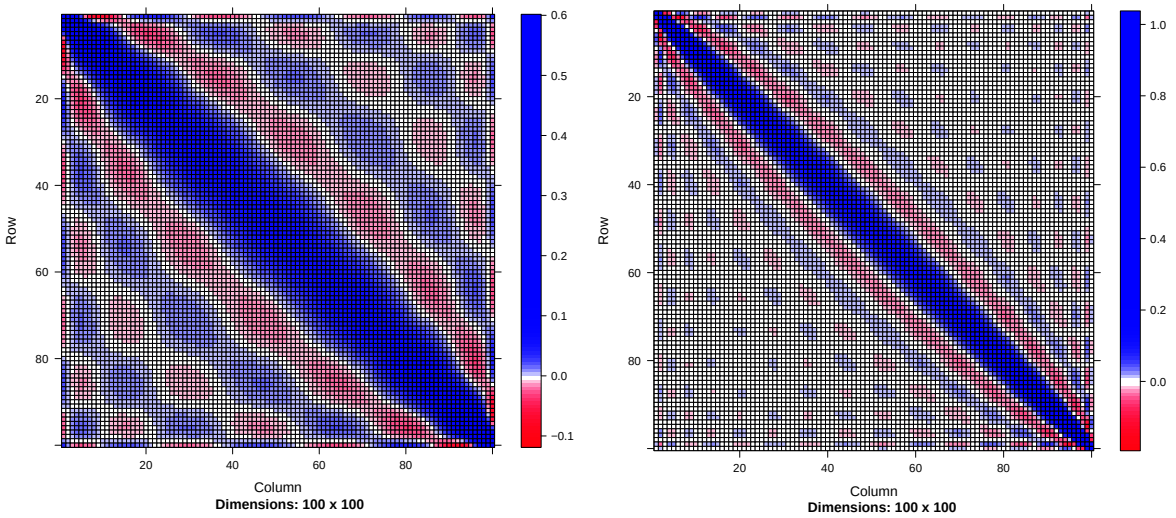
(c) Quadratic-spline-based regression

Figure 2.8. Data and fitted function for the simulated data from example 2.5 for polynomial regression of degrees 10 and 17 as well as for a spline-based model.

Figure 2.9 allows some insight into why the polynomial model struggles. It shows `image` plots of the hat matrix $\mathbf{S} = \mathbf{B}(\mathbf{B}^\top\mathbf{B})^{-1}\mathbf{B}^\top$ for the three models under consideration. The hat matrix maps the observed response to the fitted response, i.e.

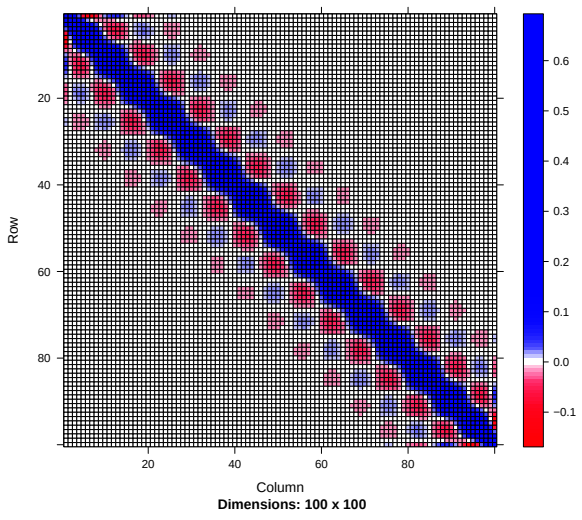
$$\hat{\mathbf{y}} = \mathbf{B}\hat{\boldsymbol{\beta}} = \mathbf{B}(\mathbf{B}^\top\mathbf{B})^{-1}\mathbf{B}^\top\mathbf{y} = \mathbf{S}\mathbf{y}$$

When performing flexible regression we would expect the prediction at x_i to almost only depend on observations close to x_i, i.e. we would expect the hat matrix $\mathbf{S}$ to be largely



(a) Polynomial regression of degree 10

(b) Polynomial regression of degree 17



(c) Quadratic-spline-based regression

Figure 2.9. Hat matrix $\mathbf{S} = \mathbf{B}(\mathbf{B}^\top\mathbf{B})^{-1}\mathbf{B}^\top$ for polynomial regression of degrees 10 and 17 as well as for splines applied to the simulated data from example 2.5.

band-diagonal with a rather narrow band width. However, polynomials are not "local". As one can see from a Taylor series expansion, the coefficients of the polynomial can be learnt from higher order derivatives observed at a single point. The problem is that a sharp dip provides more information than the data on either side of it, yielding to a poor fit on both sides. This is known as Runge's phenomenon in Numerical Analysis.

Figure 2.8(a) and Figure 2.8(b) shows another drawback of polynomial regression. As $x \to \pm\infty$ the polynomial must go to $\pm\infty$ as well. This often leads to very high curvature at both ends of the range, which is typically not supported by the data.

Yet another reason for avoiding polynomial regression is that it is highly likely to be numerically unstable. Due to the large correlations between the powers of the x_i, which make up the columns of the design matrix, the design matrix $\mathbf{B}$ and the matrix of cross-products $\mathbf{B}^\top\mathbf{B}$ is very likely to be ill-conditioned. The condition number[1] of $\mathbf{B}^\top\mathbf{B}$ for the polynomial regression model of degree 17 is 1.56×10^{12}, i.e. $\mathbf{B}^\top\mathbf{B}$ is barely invertible. For comparison, the corresponding condition number for the spline-based model is 32.49.

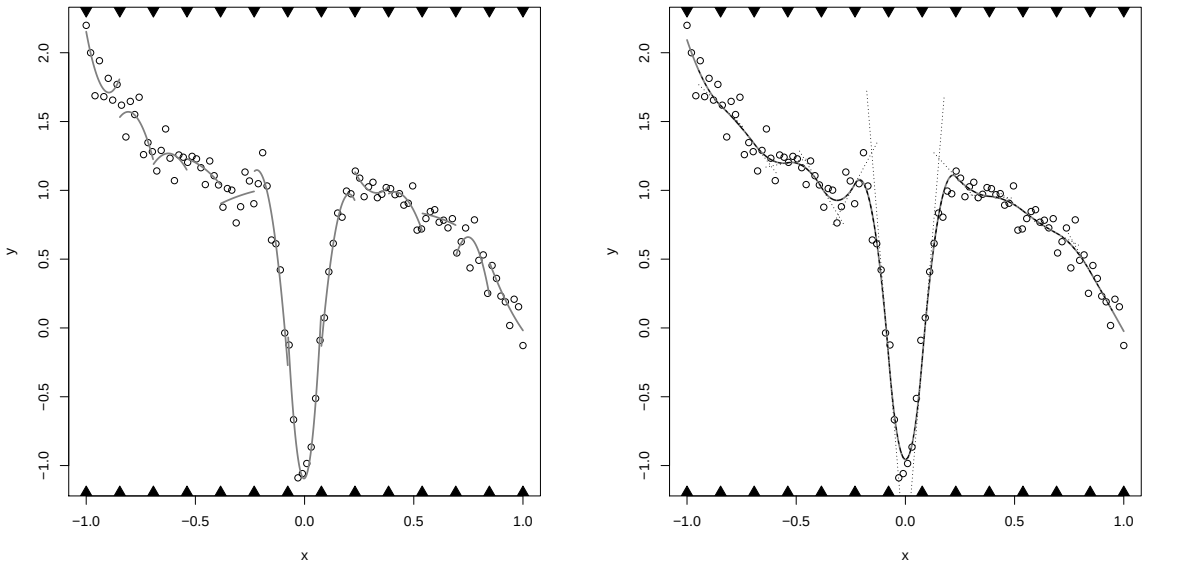
Instead of using monomials it would be numerically more stable to use so-called Tchebychev polynomials (as produced for example by the R function `poly`). Both sets of basis functions are equivalent, i.e. they span the same linear subspace and thus yield identical predictions. Though numerically more stable, Tchebychev polynomials suffer from all the other problems just as much as monomials. ◁

As we have seen in the example above, polynomial regression is, unless modelling very simple relationships, not a suitable tool for flexible regression. In the next section we will consider piecewise polynomial models, which are better suited for flexible regression. These are based on the idea of splitting the input domain and fitting low-order polynomials in each interval. As we can see from Figure 2.10(a) fitting polynomials independently of each other does not yield satisfactory results. We will thus introduce additional constraints which make the function continuous and (potentially) differentiable (cf. panel (b)).

2.3.2 Polynomial splines

In this section we will introduce polynomial splines (see, for example, de Boor (1978) for more introductory details) which are piecewise polynomials, which are "glued together" at the knots so that the resulting function is r-times continuously differentiable.

[1] The condition number of a matrix is defined as the ratio of the largest singular value divided by the smallest singular value. For a symmetric positive-definite matrix this is the same as the ratio of the largest over the smallest eigenvalue. The condition number is of measure of how numerically unstable matrix operations like taking the inverse will be.



(a) Discontinuous piecewise polynomials

(b) Piecewise polynomials which form a continuously differentiable function (derivatives at knots shown as dashed lines)

Figure 2.10. Piece-wise polynomials fitted to the data from example 2.5 with an without smoothness constraints. The back triangles show the positions of the knots.

Definition 2.1 (Polynomial spline). *Given a set of knots* $a = \kappa_1 < \kappa_2 < \ldots < \kappa_l = b$, *a function* $f : [a, b] \to \mathbb{R}$ *is called a (polynomial) spline of degree* r *if*

- $f(\cdot)$ *is a polynomial of degree* r *on each interval* (κ_j, κ_{j+1}) $(j = 1, \ldots, l-1)$.
- $f(\cdot)$ *is* $r-1$ *times continuously differentiable.*[2]

Historically, a spline was an elastic ruler used to draw technical designs, notably in shipbuilding and the early days of aircraft engineering. Figure 2.11 shows such a spline.[3]

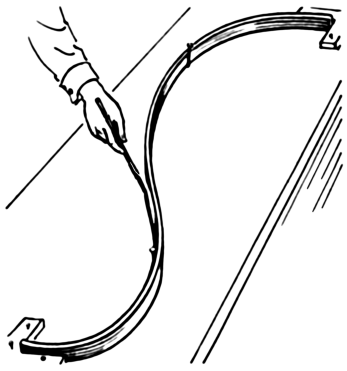
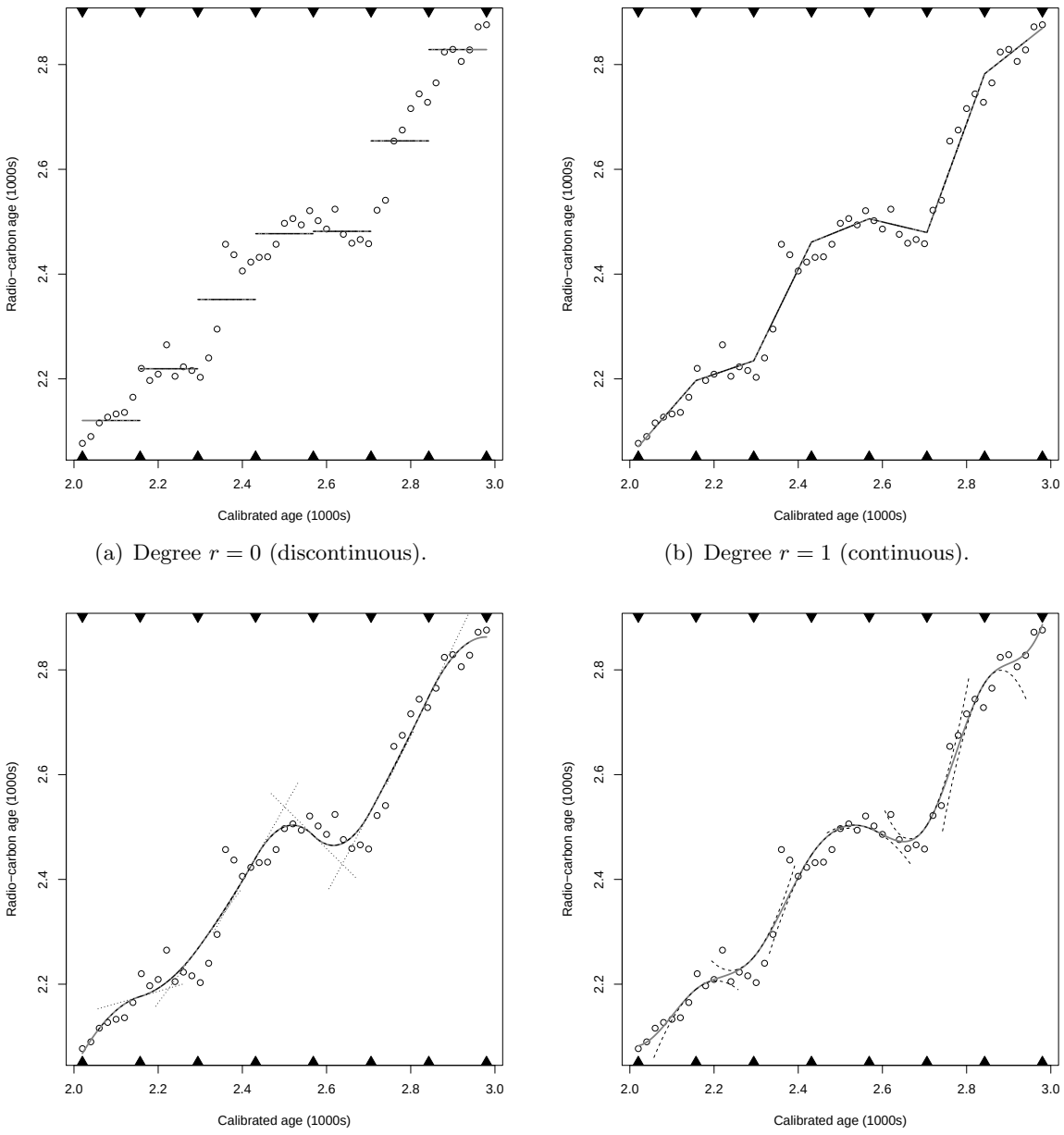


Figure 2.11. A spline.

[2] For a spline of degree 0 the function $f(\cdot)$ does not need to be continuous. For a spline of degree 1 the function $f(\cdot)$ needs to be continuous, but does not need to be differentiable.

[3] See `http://pages.cs.wisc.edu/~deboor/draftspline.html` for a picture (probably from the 1960's) of a Boeing engineer using a spline.

Choice of degree r. The degree r of the spline controls the smoothness in the sense of controlling its differentiability. For $r = 0$ the spline is a discontinuous step function. For $r = 1$ the spline is a polygonal line. For larger values of r the spline is increasingly smooth, but also behaves more and more like one global polynomial. It is worth noting that assuming too smooth a function can have significant detrimental effects on the fitted regression function (*e.g.* oscillations, ballooning). In practice it is rarely necessary to go beyond $r = 3$.



(a) Degree $r = 0$ (discontinuous).

(b) Degree $r = 1$ (continuous).

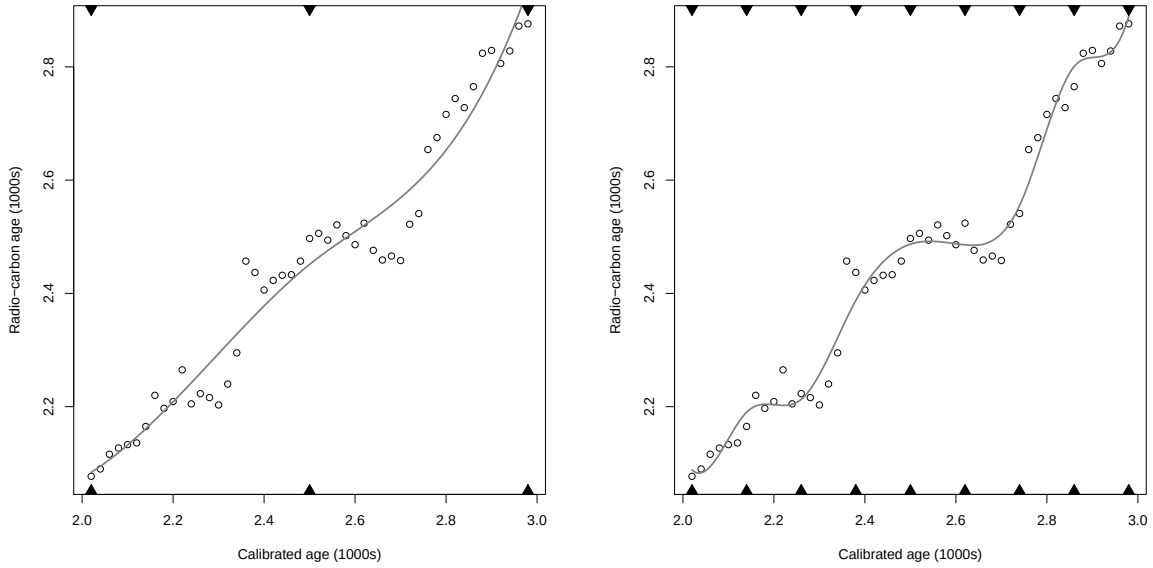
(c) Degree $r = 2$ (continuous first derivative).

(d) Degree $r = 3$ (continuous second derivative).

Figure 2.12. Splines of degree $r \in \{0, 1, 2, 3\}$ fitted to the radiocarbon data.

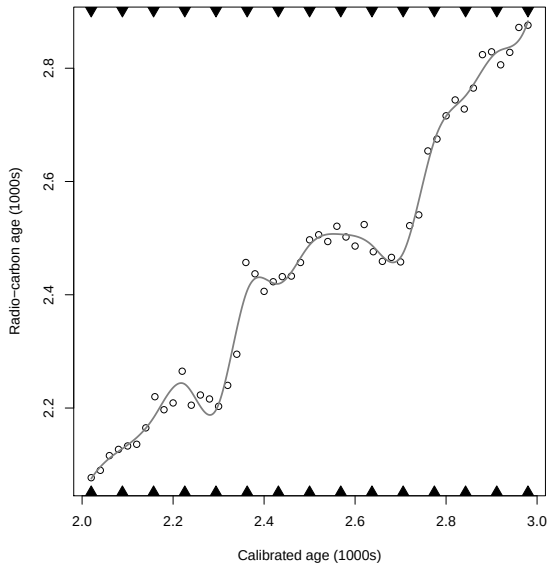
Example 2.6 (Radiocarbon dating). In a scientific experiment high-precision measurements of radiocarbon were performed on Irish oak. To construct a calibration curve we need to learn the relationship between the radiocarbon age and the calendar age. Figure 2.12 shows spline fits to the data using splines of different degrees. ◁

Choice of the number of knots l. In an (unpenalised) spline the number of knots acts as a smoothing parameter. The more knots that are used, the more flexible the regression function can become. A more flexible regression function has a lower bias, but a higher variance.

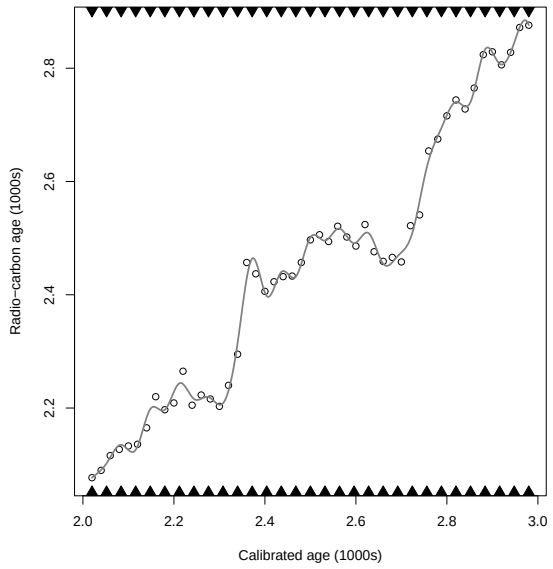


(a) $l = 3$ knots.

(b) $l = 9$ knots.



(c) $l = 15$ knots.



(d) $l = 31$ knots.

Figure 2.13. Cubic spline with different number of knots $l \in \{3, 9, 15, 31\}$ fitted to the radiocarbon data.

Example 2.7 (Radiocarbon dating (continued)). Figure 2.13 shows a cubic spline fitted to the radiocarbon data using an increasing number of knots. Too few knots lead to an underfit to the data: the fitted function does not fully represent the relationship between radiocarbon age and calendar age. Too many knots on the other hand lead to an overfit: the spline does not only pick up the signal, but also adapts to artefacts in the noise. ◁

Especially when the number of knots is small, the positioning of the knots can be important. The simplest strategy consists of using a set of equally spaced knots; this is computationally the simplest. Alternatively, we can place the knots according to the quantiles of the covariate. This makes the spline more flexible in regions with more data (and thus potentially in regions with more information) and less flexible in areas with less data (and potentially less information). A third strategy consists of trying to find an optimal placement of the knots. This usually is computationally very demanding.

Yet another approach consists of using "too many" knots — one knot per observation in the most extreme case — and then using a penalty term to control for the smoothness. This avoids the need to select the number of knots altogether. We will study two such approaches in sections 2.3.3 and 2.3.5.

Splines as a vector space. For a given set of l knots and given degree r, the space of polynomial splines is a vector space, i.e. the sum of two splines, as well as a scalar multiple of each spline, are again splines. To find the dimension of the vector space we have to find the number of "free parameters".

- Each polynomial has $r+1$ parameters and there are $l-1$ polynomials. Thus the spline model has $(r+1)\cdot(l-1)$ parameters. However we cannot choose all these parameters freely, as the resulting function needs to be $r-1$ times continuously differentiable.
- At the $l-2$ interior knots we have to guarantee that $f(\cdot)$ is $r-1$ times continuously differentiable. This corresponds to r constraints ($r-1$ constraints for each derivative and one for $f(\cdot)$ to be continuous). Thus there are $r\cdot(l-2)$ constraints (which are all linearly independent).

Thus there are $(r+1)\cdot(l-1) - r\cdot(l-2) = r+l-1$ free parameters. Thus the vector space of polynomial splines of degree r with l knots is $r+l-1$.

In section 2.3.4 we will explore different ways of constructing a basis for this space. The dimension will come in handy when proving that a given set of basis functions is indeed a basis of this space, as we only need to show that the basis functions are independent and that we use the correct number of basis functions.

Natural cubic splines. Finally, we will introduce the concept of a natural cubic spline. It is based on the idea that it is "safer" (or more "natural") to assume that the curvature of the spline at the first and last knot is zero. If we were to extrapolate, we would then extrapolate linearly.

Definition 2.2 (Natural cubic spline). *A polynomial spline $f : [a, b] \to \mathbb{R}$ of degree 3 is called a natural cubic spline if $f''(a) = f''(b) = 0$.*

Given a set of l knots the vector space of all cubic splines has dimension $l+2$. Natural cubic splines introduce two additional constraints, thus they form a vector space of dimension l. This makes natural cubic splines perfectly suited for interpolation.

Proposition 2.3. *A set of l points (x_i, y_i) can be exactly interpolated using a natural cubic spline with the $x_1 < \ldots < x_l$ as knots. The interpolating natural cubic spline is unique.*

Proof. The space of natural cubic splines with knots at $x_1, \ldots, x_l$ is a vector space of dimension l. Introducing l additional constraints ($y_i = f(x_i)$ for $i = 1, \ldots, l$) yields a system of l equations and l free parameters, which yields a unique solution.[4] □

Natural cubic splines can be generated using the function `ns` in the package `splines` in R.

In the next section we will show that natural cubic splines have an important optimality property.

2.3.3 Optimality of splines

This section provides a theoretical justification for the choice of splines for flexible regression.

In this section we will ask a rather general question. Given a data set (x_i, y_i) with $a \leq x_i \leq b$ we try to find, amongst all twice continuously differentiable functions, the function which "best" models the relationship between response y_i and covariate x_i.

First of all, we need to specify what we mean by "best". We could look for the function $f(\cdot)$ which yields the smallest least-squares criterion

[4] Strictly speaking, we would need to show that the system of equations cannot be rank-deficient, which could cause the solution to be either non-unique or non-existing.

$$\sum_{i=1}^{n}(y_i - f(x_i))^2.$$

This is however not a good idea. Any function which interpolates all the observations (x_i, y_i) would be optimal in this sense, yet such a function would typically not describe the relationship between x_i and y_i but rather model the artefacts of the random noise. Thus we will consider a so-called *penalised* (or *regularised*) criterion which tries to balance out two aspects which are important to us:

Fit to the data. We want $f(\cdot)$ to follow the data closely.

Simplicity/smoothness. We want the function $f(\cdot)$ not to be too complicated so that it generalises well to future unseen data.

We will thus minimise the following penalised fitting criterion

$$\underbrace{\sum_{i=1}^{n}(y_i - f(x_i))^2}_{\text{Fit to the data}} + \lambda \underbrace{\int_a^b f''(x)^2\, dx}_{\text{Roughness penalty}}, \tag{2.1}$$

where $\lambda > 0$ is a tuning parameter which balances between following the data and preventing $f(\cdot)$ from being too rough.

We will now establish that the minimiser of (2.1) over all twice continuously differentiable functions has to be a natural cubic spline, i.e. natural cubic splines with knots at each of the unique x_i are in this sense the optimal class functions.

We will start by stating that natural cubic splines are optimal interpolators, in the sense that they minimise the roughness penalty $\int_a^b f''(x)^2\, dx$.

Lemma 2.4. *Amongst all functions on $[a, b]$ which are twice continuously differentiable and which interpolate the set of points (x_i, y_i), a natural cubic spline with knots at the x_i yields the smallest roughness penalty*

$$\int_a^b f''(x)^2\, dx.$$

Proof. Let $f(\cdot)$ be the natural cubic spline with knots at the x_i, interpolating the data. Suppose there is another function $g(\cdot)$, which is twice continuously differentiable and which also interpolates the data. Denote by $h(x) = g(x) - f(x)$ the difference between the two functions.

1. We will first of all show that we can decompose

$$\int_a^b g''(x)^2\,dx = \int_a^b f''(x)^2\,dx + \int_a^b h''(x)^2\,dx$$

i. As both $f(\cdot)$ and $g(\cdot)$ interpolate the (x_i, y_i) we have that $f(x_i) = g(x_i) = y_i$, thus $h(x_i) = g(x_i) - f(x_i) = 0$.

ii. Using integration by parts we obtain that

$$\begin{aligned}
\int_a^b f''(x)h''(x)\,dx &= \underbrace{[f''(x)h'(x)]_{x=a}^b}_{=0 \text{ (as } f''(a)=f''(b)=0)} - \int_a^b f'''(x)h'(x)\,dx \\
&= -\sum_{i=1}^{n-1} \int_{x_i}^{x_{i+1}} f'''(x)h'(x)\,dx \\
&= -\sum_{i=1}^{n-1} f'''\left(\frac{x_i + x_{i+1}}{2}\right) \cdot \underbrace{\int_{x_i}^{x_{i+1}} h'(x)\,dx}_{=\underbrace{h(x_{i+1})}_{=0} - \underbrace{h(x_i)}_{=0} = 0} \\
&= 0
\end{aligned}$$

In the second line we have used that the natural cubic spline is piece-wise cubic polynomial, i.e. between two knots x_i and x_{i+1} the third derivative $f'''(x)$ is constant.

iii. Thus

$$\begin{aligned}
\int_a^b g''(x)^2\,dx &= \int_a^b (g''(x) - f''(x) + f''(x))^2\,dx = \int_a^b (h''(x) + f''(x))^2\,dx \\
&= \int_a^b h''(x)^2\,dx + 2\underbrace{\int_a^b h''(x)f''(x)\,dx}_{=0} + \int_a^b f''(x)^2\,dx \\
&= \int_a^b h''(x)^2\,dx + \int_a^b f''(x)^2\,dx
\end{aligned}$$

2. Because of $\int_a^b h''(x)^2\,dx \geq 0$ we have that

$$\int_a^b g''(x)^2\,dx = \int_a^b f''(x)^2\,dx + \underbrace{\int_a^b h''(x)^2\,dx}_{\geq 0} \geq \int_a^b f''(x)^2,$$

i.e. the natural cubic spline cannot have a larger roughness penalty.

3. In the above inequality equality holds if and only if $\int_a^b h''(x)^2\,dx = 0$, which, given that $h(x_i) = 0$, can only be the case if $g(x) = f(x)$ for all $x \in [a, b]$. □

Spline-based interpolation is implemented in the R functions `spline` and `splinefun`.

Theorem 2.5. *The minimiser of*

$$\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \cdot \int_a^b f''(x)^2 \, dx$$

amongst all twice continuously differentiable functions on $[a, b]$ *is given by a a natural cubic spline with knots in the unique* x_i*.*

This is an extremely powerful theorem. Even though we consider the entire infinite-dimensional vector space of all twice continuously differentiable functions, we only need to consider the finite-dimensional vector space of natural cubic splines. We have thus reduced the complexity of the optimisation problem to the comparatively simple problem of finding the optimal coefficients of the natural cubic spline. This can be done using least-squares.

Proof. Let $g(\cdot)$ be a twice continuously differentiable function. We will now create a competitor to $g(\cdot)$, which is a natural cubic spline with knots in the x_i. We will now show that, unless $g(\cdot)$ is already a natural cubic spline, $f(\cdot)$ leads to a smaller value of the objective function. We choose the natural cubic spline $f(\cdot)$ such that it interpolates the fitted values $g(\cdot)$ generates, i.e. $f(x_i) = g(x_i)$. Thus $\sum_{i=1}^{n}(y_i - f(x_i))^2 = \sum_{i=1}^{n}(y_i - g(x_i))^2$, i.e. both functions model the data equally well, however as we have shown in Lemma 2.4 the natural cubic spline $f(\cdot)$ has the smaller roughness penalty. □

Note that the proof did not make use of the fact that we have used the least-squares loss function. In fact, the theorem holds for any pointwise loss function.

The technique of ***smoothing splines*** is based on this theoretical result and finds the natural cubic spline minimising (2.1), and, due to the theorem, the optimal function amongst all twice continuously differentiable functions. This approach is implemented in the R function `smooth.spline`, illustrated here in Figure 2.14.

We will revisit the idea of regularisaton in more detail in section 2.3.5.

```
radiocarbon <- radioc[(radioc$Cal.age>=2000)&(radioc$Cal.age<=3000),]
smsp <- with(radiocarbon, {
    plot(Cal.age, Rc.age)
    smooth.spline(Cal.age, Rc.age)
})
smsp

   ## Call:
   ## smooth.spline(x = Cal.age, y = Rc.age)
   ##
   ## Smoothing Parameter  spar= 0.3707624  lambda= 2.777959e-06 (15 iterations)
   ## Equivalent Degrees of Freedom (Df): 24.10109
   ## Penalized Criterion (RSS): 10466.32
   ## GCV: 737.7253

lines(smsp)
```

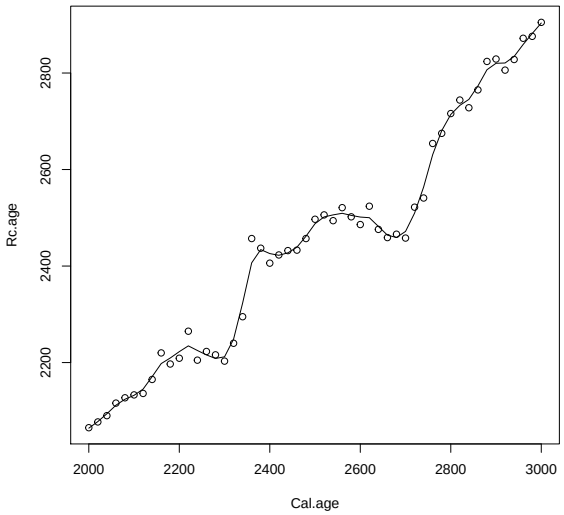


Figure 2.14. Smoothing spline fit to the radiocarbon data.

2.3.4 Constructing splines

In this section we will study two ways of constructing a basis for the vector space of polynomial splines: the truncated power basis and the B-spline basis. We will only cover the case of generic polynomial splines. However one can modify these bases to only span the space of natural cubic splines.

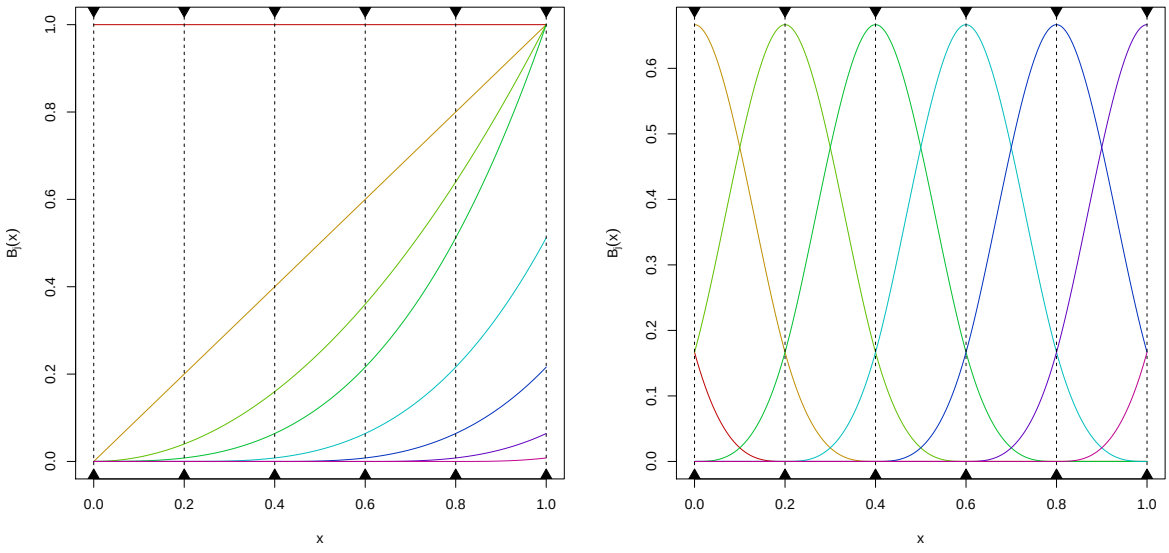
Truncated power basis. The simplest basis for polynomial splines is the truncated power basis.

Definition 2.6 (Truncated power basis). *Given a set of knots* $a = \kappa_1 < \ldots < \kappa_l = b$ *the truncated power basis of degree* r *is given by*

$$\left(1, x, \ldots, x^{r-1}, (x-\kappa_1)_+^r, (x-\kappa_2)_+^r, \ldots, (x-\kappa_{l-1})_+^r\right),$$

where $(z)_+^r = \begin{cases} z^r & \text{for } z > 0 \\ 0 & \text{otherwise.} \end{cases}$

The truncated power basis has $r + l - 1$ basis functions. It is easy to see that they are linearly independent. Thus the truncated power basis is indeed a basis of the vector space of polynomial splines. Figure 2.15(a) shows the truncated power series basis of degree 3 for six equally spaced knots.



(a) Truncated power series (b) B-splines

Figure 2.15. Basis functions $B_j(x)$ of the cubic truncated power series basis (left panel) and B-splines (right panel). The vertical lines indicate the location of the knots.

To fit a polynomial spline to data we can exploit the fact the truncated power basis is a basis of the vector space of polynomial splines of the given degree and with the given set of knots. Thus we can write any spline $f(\cdot)$ as a linear combination of the basis functions, i.e.

$$f(x) = \beta_0 + \beta_1 x + \ldots + \beta_{r-1} x^{r-1} + \beta_r (x - \kappa_1)_+^r + \ldots + \beta_{r+l-2} (x - \kappa_{l-1})_+^r$$

We can thus find the optimal spline $f(\cdot)$ by just finding the optimal set of coefficients β_j, which is nothing other than a linear regression problem with design matrix

$$\mathbf{B} = \begin{pmatrix} 1 & x_1 & \ldots & x_1^{r-1} & (x_1 - \kappa_1)_+^r & \ldots & (x_1 - \kappa_{l-1})_+^r \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \ldots & x_n^{r-1} & (x_n - \kappa_1)_+^r & \ldots & (x_n - \kappa_{l-1})_+^r \end{pmatrix}$$

We can use the design matrix $\mathbf{B}$ in exactly the same way as we would use the design matrix of a classical linear model.

We can interpret the truncated power series as a regression model in which the leading coefficient changes at each knot. At each knot, the remaining coefficients change as well. However they are fully constrained by the condition that the spline has to be $r-1$ times continuously differentiable at each knot.

Example 2.8 (Radiocarbon data (continued)). Figure 2.19 illustrates the use of a truncated power series basis for fitting a spline-based flexible regression model for the radiocarbon data.

As one can see from the middle panel of Figure 2.19 and from Figure 2.17, some of the estimated coefficients are very large: some of the basis functions are scaled up by a factor of more than 1000, with "neighbouring" basis functions having opposite signs. The reason for this is the high correlation between the columns of the design matrix of the truncated power series. The largest correlation between columns is 0.99921, which is very close to 1.

Figure 2.18 shows a scree plot of the singular values of the design matrix $\mathbf{B}$. The condition number of the matrix $\mathbf{B}$ is 225333.0, with the condition number of $\mathbf{B}^\top \mathbf{B}$ being $5,857,413,839$, i.e. $\mathbf{B}^\top \mathbf{B}$ is close to being numerically singular. This suggests that finding the least-squares estimate of the coefficients is close to being numerically unstable. ◁

We can generate a truncated power basis in R as follows.

```
tpower <- function(x, t, p)
    (x - t) ^ p * (x > t)

tbase <- function(x, xl = min(x), xr = max(x), n.knots = 10, deg = 3) {
  nseg <- n.knots - 1
  dx <- (xr - xl) / nseg
  knots <- seq(xl, xr,  len=n.knots)
  B <- cbind(outer(x-xl,0:(deg-1),"^"),
             outer(x,knots[-length(knots)],function(x,y) pmax(x-y,0))^deg)
  B
}

B <- tbase(radiocarbon$Cal.age, n.knots=10)
y <- radiocarbon$Rc.age
beta <- qr.coef(qr(B), y)
y.hat <- B%*%beta
with(radiocarbon, {
    plot(Cal.age, Rc.age)
    lines(Cal.age, y.hat)
})
```

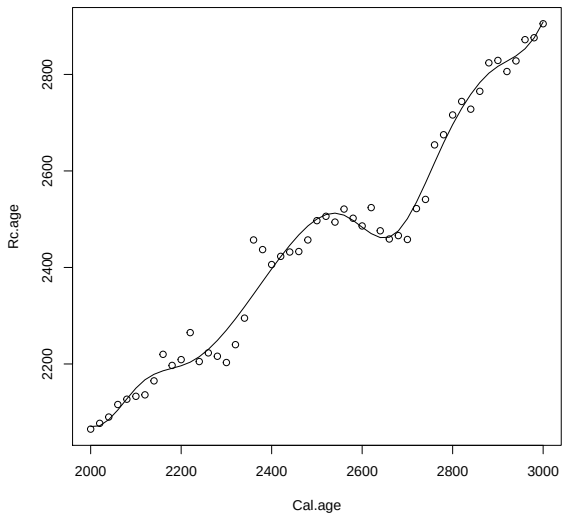


Figure 2.16. Truncated power basis fit to the radiocarbon data.

As we have seen in the above example the truncated power basis can easily lead to numerical instability. Thus we will turn to an alternative basis, the so-called B-spline basis.

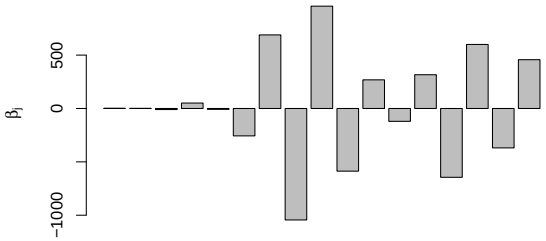


Figure 2.17. Bar plot of the coefficients $\hat{\boldsymbol{\beta}}$ estimated using the truncated power series regression model shown in Figure 2.19.

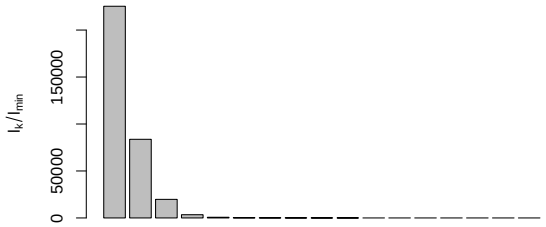


Figure 2.18. Scree plot of the singular values of the design matrix $\mathbf{B}$ (square root of the eigenvalues of the cross-product matrix $\mathbf{B}'\mathbf{B}$) for the truncated power series regression model shown in Figure 2.19.

B-splines. B-splines form a numerically more stable basis. They also make the definition of meaningful penalty matrices easier, which we will exploit in section 2.3.5.

The key idea of B-splines is to use basis functions which are local, i.e. only non-zero for a "small" proportion of the range of the covariate and which are bounded above. We can think of B-splines as a sequence of "bumps".

Definition 2.7 (B-spline basis). *(a) Given a set of l knots the B-spline basis of degree 0 is given by the functions $(B_1^0(x), \ldots, B_{l-1}^0(x))$ with*

$$B_j^0(x) = \begin{cases} 1 & \text{for } \kappa_j \leq x < \kappa_{j+1} \\ 0 & \text{otherwise.} \end{cases}$$

(b) Given a set of l knots the B-spline basis of degree $r > 0$ is given by the functions $(B_1^r(x), \ldots, B_{l+r-1}^r(x))$ with

$$B_j^r(x) = \frac{x - \kappa_{j-r}}{\kappa_j - \kappa_{j-r}} B_{j-1}^{r-1}(x) + \frac{\kappa_{j+1} - x}{\kappa_{j+1} - \kappa_{j+1-r}} B_j^{r-1}(x).$$

In order to be able to construct the splines recursively we have to introduce additional outside knots to the left of κ_1 and to the right of κ_l. In order to be able to construct a

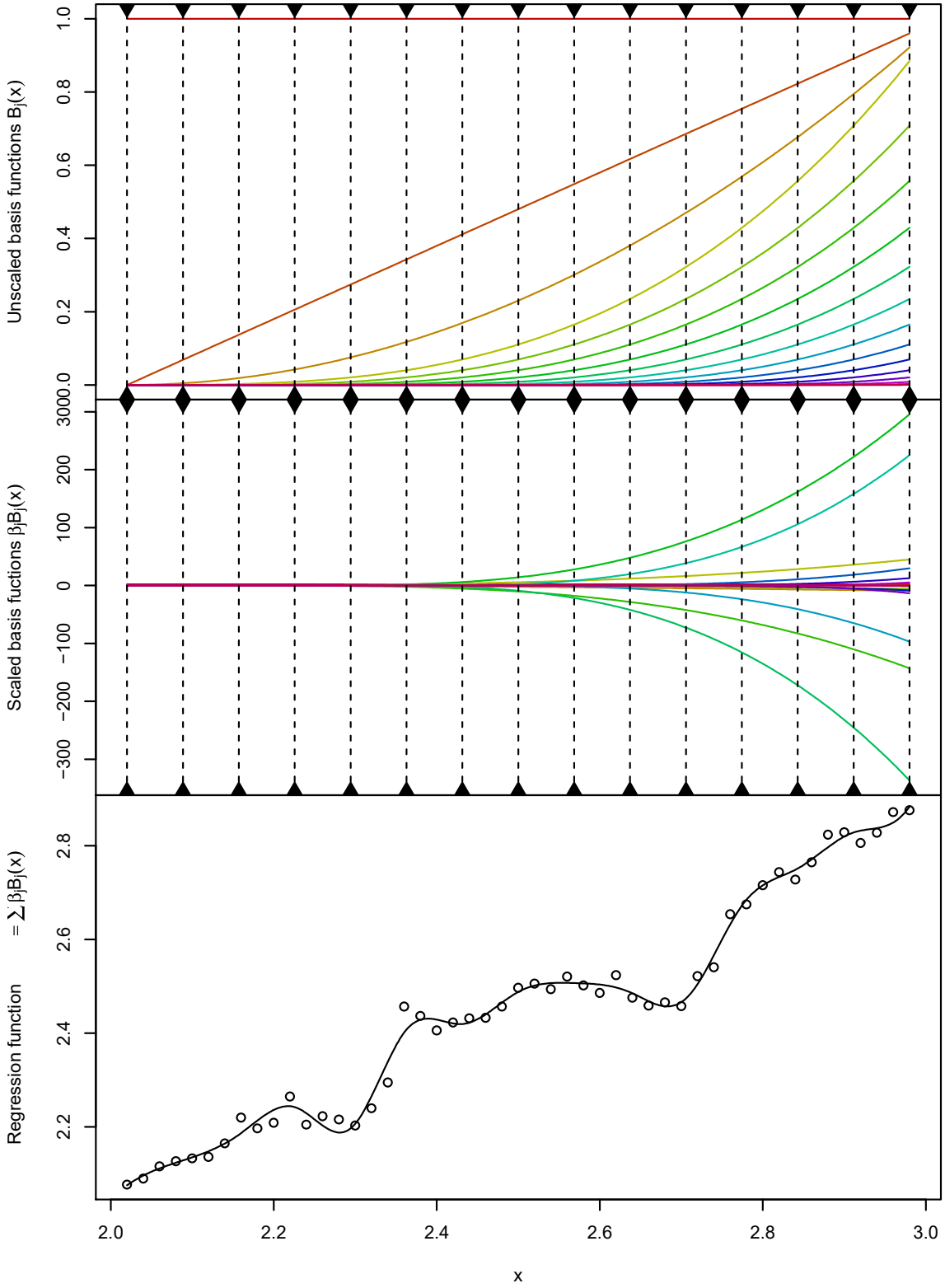
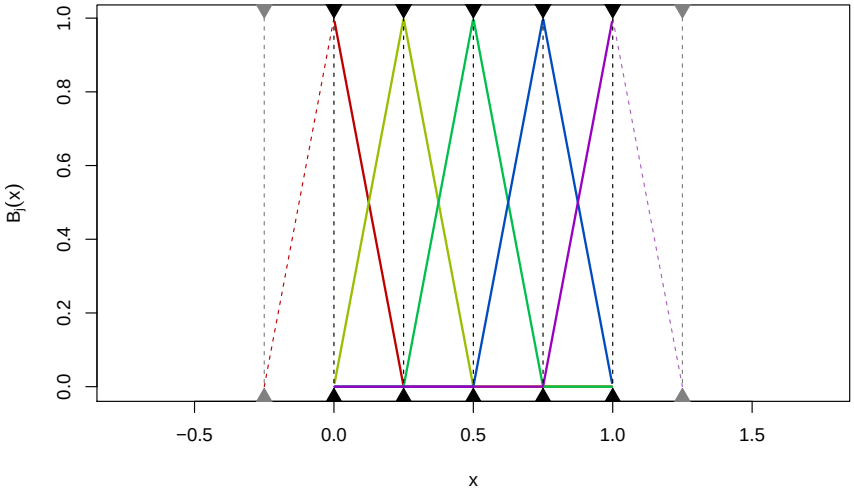
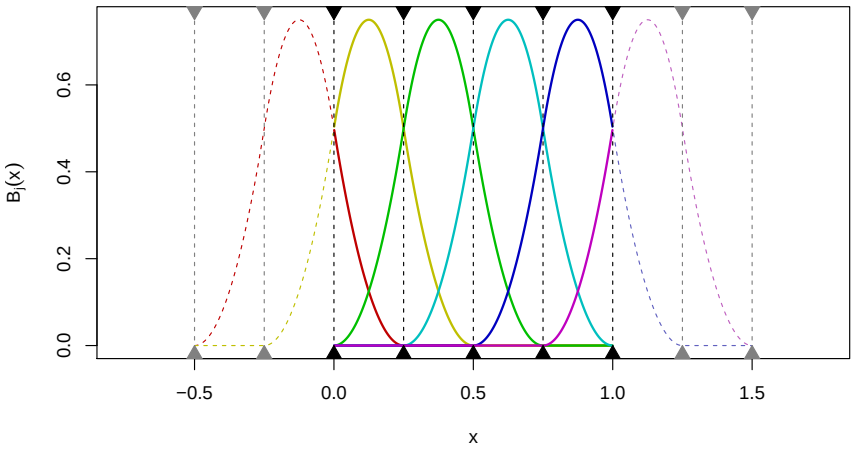


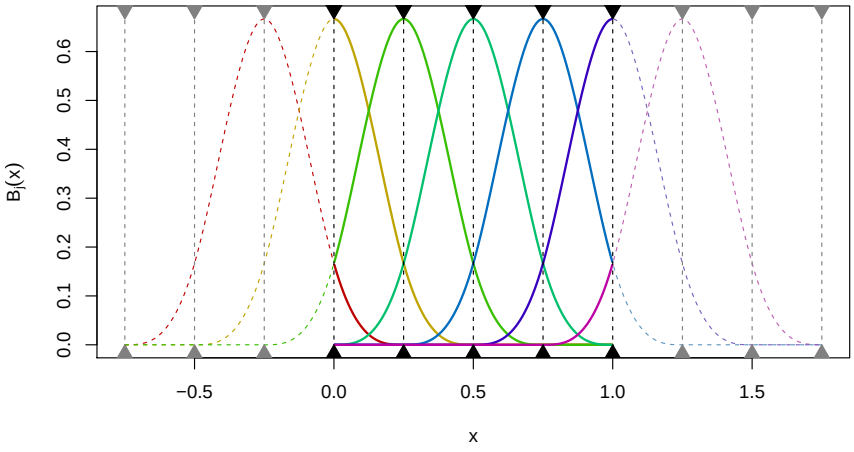
Figure 2.19. Illustration of flexible regression using the truncated power series basis of degree 3 applied to the radiocarbon data. The top panel shows the unscaled basis functions $B_j(x)$. The middle panel shows the scaled basis functions $\hat{\beta}_j B_j(x)$. The bottom panel shows a scatter plot of the data together with the fitted function $\hat{f}(x) = \sum_j \hat{\beta}_j B_j(x)$.



(a) Degree $r = 1$



(b) Degree $r = 2$



(c) Degree $r = 3$

Figure 2.20. B spline bases for degrees $r \in \{1, 2, 3\}$.

basis of degree r we need r additional outside knots on each side. Figure 2.20 illustrates this idea. These outside knots are just used to construct the basis.

From their recursive definition one can derive that B-splines have the following properties. These can also be seen in Figure 2.20.

- A B-spline basis function of degree r is made up of $r+1$ polynomials of degree r. Outside these $r+1$ intervals, the basis function is zero. This makes the basis functions local.
- At every $x \in (a, b)$ only $r+1$ basis functions are non-zero.
- The basis functions sum to 1 for all $x \in [a, b]$. This implies that we do not need to include an intercept in the design matrix.
- One can show that the derivative of a B-spline of degree r is a B-spline of degree $r-1$.

We can fit a B-spline model to data by using the design matrix

$$\mathbf{B} = \begin{pmatrix} B_1^r(x_1) & \dots & B_{l+r-1}^r(x_1) \\ \vdots & \ddots & \vdots \\ B_1^r(x_n) & \dots & B_{l+r-1}^r(x_n) \end{pmatrix}.$$

Example 2.9 (Radiocarbon data (continued)). Figure 2.24 illustrates the use of a B-spline basis for fitting a spline-based flexible regression model for the radiocarbon data.

The B-spline basis is numerically much better behaved. The coefficient values (cf. Figure 2.22) are not too large and the columns of the design matrix $\mathbf{B}$ are much less correlated than the columns of the truncated power basis; the maximum correlation is 0.8309. The condition number of $\mathbf{B}$ is 25.664 (cf. Figure 2.23) and the condition number of $\mathbf{B}^\top\mathbf{B}$ is 358.263. ◁

The R function `bs` from the package `splines` can generate a **B-spline basis** and can be used inside `lm`. The number of basis functions needs to be chosen manually when using `bs`. This is done by selecting a value for `df` - here the degrees of freedom for the basis - we'll return to this idea in Chapter 4.

```
model <- lm(Rc.age~bs(Cal.age, df=10), data=radiocarbon)
with(radiocarbon, {
    plot(Cal.age, Rc.age)
    lines(Cal.age, predict(model))
})
```

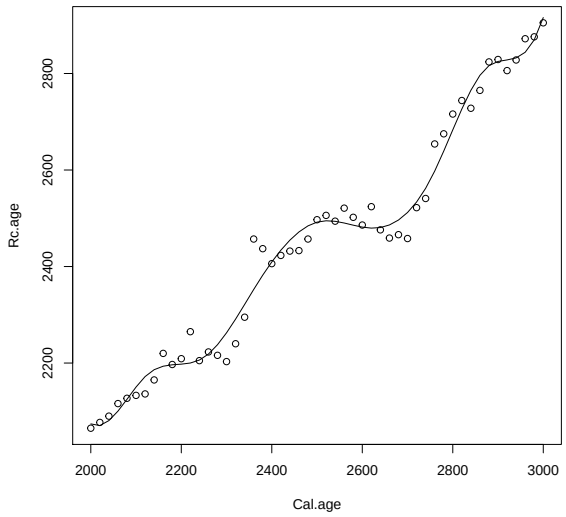


Figure 2.21. B-spline fit to the radiocarbon data.

However, in terms of scaling and properties on the boundary, the basis returned by `bs` differs slightly from the definitions above. The function given below (based on a function written by Paul Eilers) generates a B spline basis which looks exactly like the ones shown above.

```
bbase <- function(x, xl = min(x), xr = max(x), n.knots = 10, deg = 3) {
# Construct B-spline basis (based on a function written by Paul Eilers)
    nseg <- n.knots-1
    dx <- (xr - xl) / nseg
    knots <- seq(xl - deg * dx, xr + deg * dx, len = n.knots + 2*deg )
    P <- outer(x, knots, tpower, deg)
    n <- dim(P)[2]
    D <- diff(diag(n), diff = deg + 1) / (gamma(deg + 1) * dx ^ deg)
    B <- (-1) ^ (deg + 1) * P %*% t(D)
    B
}
```

The function `bbase` can be used in the same way as the function `tbase`.

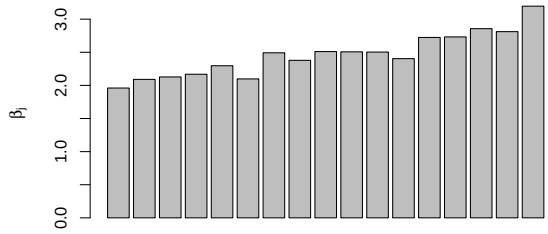


Figure 2.22. Bar plot of the coefficients $\hat{\boldsymbol{\beta}}$ estimated using the B-spline regression model shown in Figure 2.20.

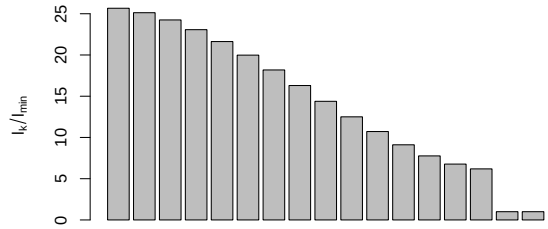


Figure 2.23. Scree plot of the singular values of the design matrix $\mathbf{B}$ (square root of the eigenvalues of the cross-product matrix $\mathbf{B}'\mathbf{B}$) for the B-spline regression model shown in Figure 2.20. The condition number of $\mathbf{B}'\mathbf{B}$ is 395.661.

2.3.5 Penalised splines (P-splines)

A reminder of ridge regression

Ridge regression solves the penalised (or regularised) least-squares criterion

$$\|\mathbf{y} - \mathbf{B}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|^2,$$

where $\mathbf{B}$ is the matrix of covariates. The solution of this problem is given by

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = (\mathbf{B}^\top\mathbf{B} + \lambda\mathbf{I}_p)^{-1}\mathbf{B}^\top\mathbf{y}$$

To compute $\hat{\boldsymbol{\beta}}_{\text{ridge}}$ it is numerically more stable to use a QR decomposition (as mentioned in the preliminary material) to minimise the augmented system

$$\left\| \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} - \begin{pmatrix} \mathbf{B} \\ \sqrt{\lambda}\mathbf{I} \end{pmatrix} \boldsymbol{\beta} \right\|^2$$

When using splines the positioning of the knots can have a large influence on the fitted function, especially if a comparatively small number of basis functions is used.

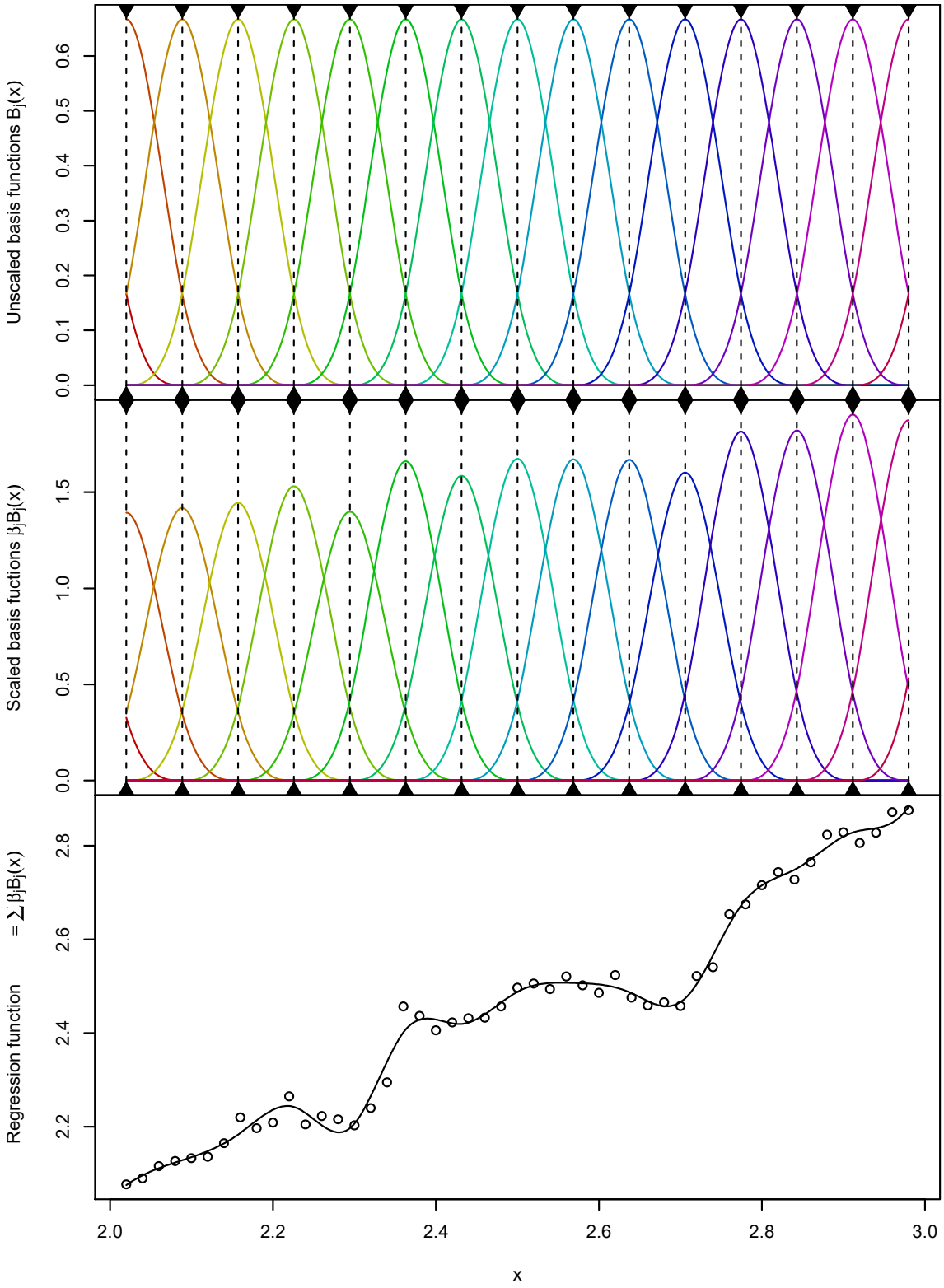


Figure 2.24. Illustration of flexible regression using the B-spline basis applied to the radiocarbon data. The top panel shows the unscaled basis functions $B_j(x)$. The middle panel shows the scaled basis functions $\hat{\beta}_j B_j(x)$. The bottom panel shows a scatter plot of the data together with the fitted function $\hat{f}(x) = \sum_j \hat{\beta}_j B_j(x)$.

One way of avoiding this problem is to use *penalised splines*. They are based on the idea of *not* using the number of basis functions to control the smoothness of the estimate, but to use a roughness penalty to this end. This is similar in spirit to the approach discussed in section 2.3.3, though in most cases it is not necessary to use one basis function per observation. Around 20 to 30 basis functions should be sufficient. Without including a penalty in the fitting criterion this would most likely lead to an overfit to the data. Thus we need to consider a penalised criterion which, just like in section 2.3.3, contains a roughness penalty. In this section we will use $\|\mathbf{D}\boldsymbol{\beta}\|^2$ as roughness penalty, i.e. we choose the regression coefficients $\boldsymbol{\beta}$ by minimising

$$\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda\|\mathbf{D}\boldsymbol{\beta}\|^2. \tag{2.2}$$

This objective function is, with the exception of the inclusion of the matrix $\mathbf{D}$, the objective function of ridge regression. As before, λ controls the trade-off between following the data (small λ) and obtaining a strongly regularised curve (large λ). In analogy with ridge regression one can show that the optimal $\boldsymbol{\beta}$ is given by

$$\boldsymbol{\beta} = (\mathbf{B}^\top\mathbf{B} + \lambda\mathbf{D}^\top\mathbf{D})^{-1}\mathbf{B}^\top\mathbf{y},$$

where $\mathbf{B}$ is the design matrix corresponding to the B-spline basis used for $f(\cdot)$. Numerically, it is more advantageous to represent the penalty term $\lambda\|\mathbf{D}\boldsymbol{\beta}\|^2$ by including it into an expanded design matrix, i.e. to solve

$$\left\| \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} - \begin{pmatrix} \mathbf{B} \\ \sqrt{\lambda}\mathbf{D} \end{pmatrix} \boldsymbol{\beta} \right\|^2$$

using a QR decomposition.

There are (at least) two possible approaches for choosing $\mathbf{D}$. We can choose $\mathbf{D}$ to be a difference matrix, or we can choose $\mathbf{D}$ such that $\|\mathbf{D}\boldsymbol{\beta}\|^2 = \int_a^b f''(x)^2\,dx$. The former is both conceptually and computationally simpler; the latter is closer to what the theory suggests as optimal.

2.3.6 Difference penalties

The simplest choice of $\mathbf{D}$ is to use a difference penalty. Using the identity matrix for $\mathbf{D}$, as we would in ridge regression, is usually not appropriate: it shrinks all coefficients to zero, i.e. it shrinks the regression function $f(\cdot)$ to zero as well, which is rarely desirable (cf. Figure 2.25(a)). As we can see from the middle panel of Figure 2.24, we obtain a smooth function when neighbouring β_j's are similar.

This can be achieved by using one of the following choices. We assume that we are using equally-spaced knots.

First-order differences. We can set

$$\mathbf{D}_1 = \begin{pmatrix} 1 & -1 & \dots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 1 & -1 \end{pmatrix}.$$

This calculates the roughness penalty as the sum of the squared first-order differences between the neighbouring β_j, i.e.

$$\|\mathbf{D}_1\boldsymbol{\beta}\|^2 = \sum_{j=1}^{l+r-2} (\beta_{j+1} - \beta_j)^2$$

This penalty shrinks the coefficients towards a common constant (cf. Figure 2.25(b)) and thus shrinks the regression function $f(\cdot)$ towards a constant function. Adding a constant to $f(\cdot)$ does thus not change the penalty.

This penalty is the natural choice if B-splines of order 2 are used.

Second-order differences. We can set

$$\mathbf{D}_2 = \begin{pmatrix} 1 & -2 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 1 & -2 & 1 \end{pmatrix}.$$

This calculates the roughness penalty as the sum of the squared second-order differences between the neighbouring β_j, i.e.

$$\|\mathbf{D}_2\boldsymbol{\beta}\|^2 = \sum_{j=1}^{l+r-3} (\beta_{j+2} - 2\beta_{j+1} + \beta_j)^2$$

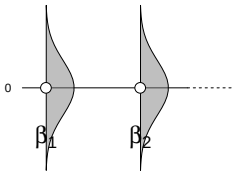
This penalty shrinks the coefficients towards a linear sequence (cf. Figure 2.25(c)) and thus shrinks the regression function $f(\cdot)$ towards a linear function. Adding a linear function to $f(\cdot)$ does thus not change the penalty.

This penalty is the natural choice if B-splines of order 3 are used.

Higher-order differences. Higher-order difference matrices can be constructed using the recursive formula $\mathbf{D}_r = \mathbf{D}_1\mathbf{D}_{r-1}$ where $\mathbf{D}_r$ denotes the penalty matrix of order r.

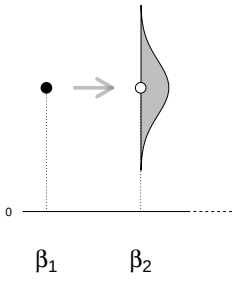
Example 2.10 (Radiocarbon dating (continued)). Figure 2.26 shows the model fit obtained when fitting a P-spline model with different values of the smoothing parameter λ. The

Penalty interpretation: Only an all-zero coefficient vector incurs no penalty.
Bayesian interpretation: Independent zero-mean Gaussian prior.



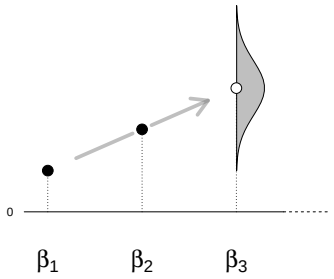
(a) Illustration of a 0-th order penalty (ridge regression).

Penalty interpretation: Only an all constant coefficient vector incurs no penalty.
Bayesian interpretation: Conditional distribution of β_2 given β_1 is Gaussian with mean β_1.
(First-order random walk)



(b) Illustration of a first-order penalty.

Penalty interpretation: Only a coefficient vector which forms a linear sequence incurs no penalty.
Bayesian interpretation: Conditional distribution of β_3 given β_1 and β_2 is Gaussian with mean $2 \cdot \beta_2 - \beta_1$.
(Second-order random walk)



(c) Illustration of a second-order penalty.

Figure 2.25. Illustration of difference penalties of order 0 to 2.

smaller λ the closer the fitted function $\hat{f}(\cdot)$ is to the data, which leads for very small values of λ to an overfit to the data. ◁

For more details on this **p-splines** approach see Marx and Eilers (1998).

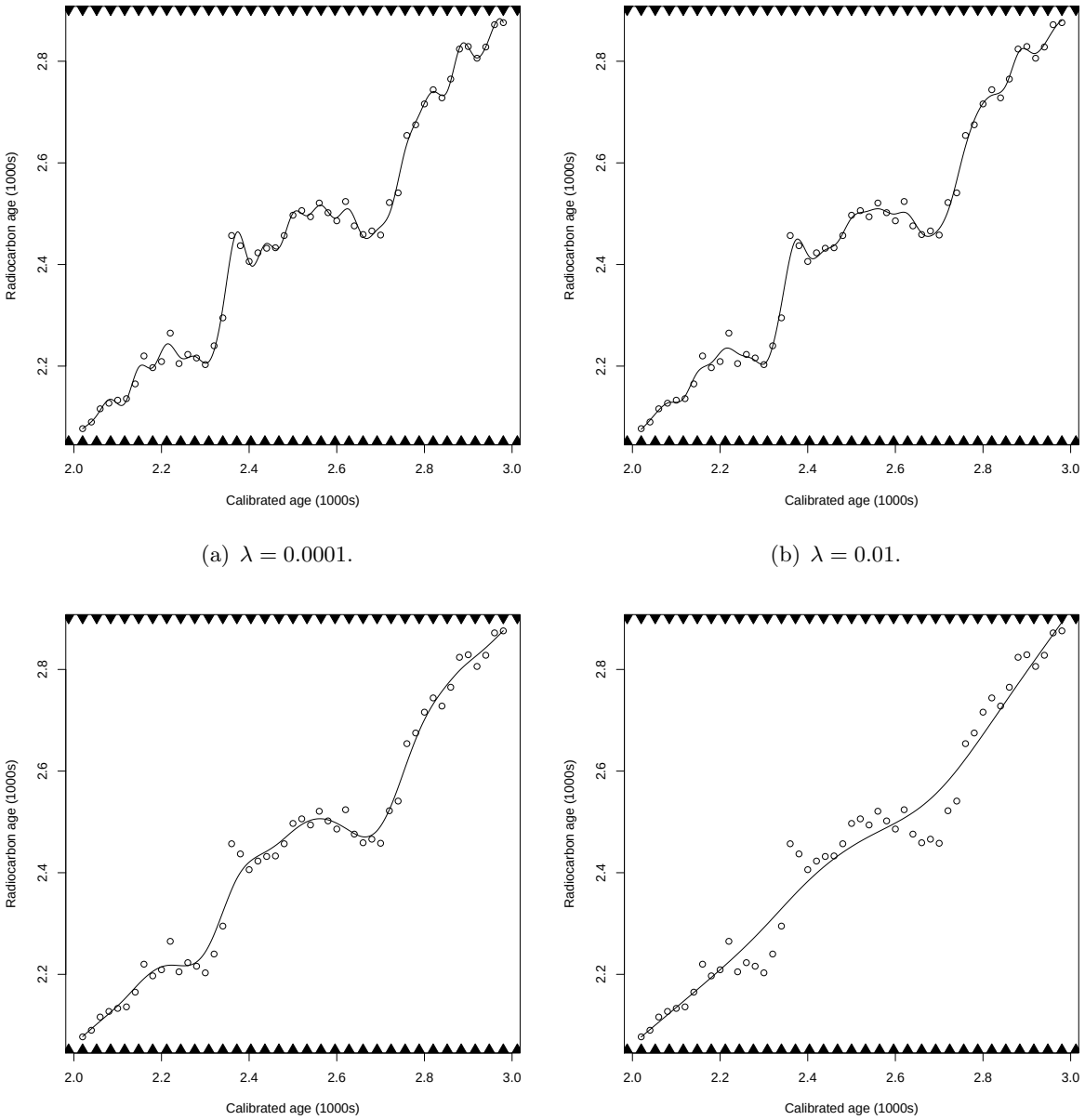
Other penalties Difference penalties are not the only choice of penalty matrix. An alternative choice consists of choosing $\mathbf{D}$ such that $\|\mathbf{D}\boldsymbol{\beta}\|^2 = \int_a^b f''(x)^2\,dx$, which is the roughness penalty we have used in section 2.3.3.

Using that $f''(x) = \sum_{j=1}^{l+r-1} \beta_j B_j''(x)$ we have that

$$\begin{aligned}\int_a^b f''(x)^2\,dx &= \sum_{j=1}^{l+r-1}\sum_{k=1}^{l+r-1} \beta_j\beta_k \int_a^b B_j''(x)B_k''(x)\,dx \\ &= \boldsymbol{\beta}^\top \begin{pmatrix} \int_a^b B_1''(x)B_1''(x)\,dx & \dots & \int_a^b B_1''(x)B_{l+r-1}''(x)\,dx \\ \vdots & \ddots & \vdots \\ \int_a^b B_1''(x)B_{l+r-1}''(x)\,dx & \dots & \int_a^b B_{l+r-1}''(x)B_{l+r-1}''(x)\,dx \end{pmatrix} \boldsymbol{\beta}\end{aligned}$$

Thus we just need to choose $\mathbf{D}$ such that

$$\mathbf{D}^\top\mathbf{D} = \begin{pmatrix} \int_a^b B_1''(x)B_1''(x)\,dx & \dots & \int_a^b B_1''(x)B_{l+r-1}''(x)\,dx \\ \vdots & \ddots & \vdots \\ \int_a^b B_1''(x)B_{l+r-1}''(x)\,dx & \dots & \int_a^b B_{l+r-1}''(x)B_{l+r-1}''(x)\,dx \end{pmatrix}.$$



(a) $\lambda = 0.0001$.

(b) $\lambda = 0.01$.

(c) $\lambda = 1$.

(d) $\lambda = 10$.

Figure 2.26. P-spline with different values of the smoothing parameter λ fitted to the radiocarbon data.

2.3.7 Penalised splines in R

We can fit a penalised spline from "first principles" in R as follows.

```
B <- bbase(radiocarbon$Cal.age, n.knots=25)
D <- diff(diag(ncol(B)), diff=2)
y <- radiocarbon$Rc.age
lambda <- 1
beta <- qr.coef(qr(rbind(B, lambda*D)), c(y, rep(0, nrow(D))))
y.hat <- B%*%beta
with(radiocarbon, {
    plot(Cal.age, Rc.age)
    lines(Cal.age, y.hat)
})
```

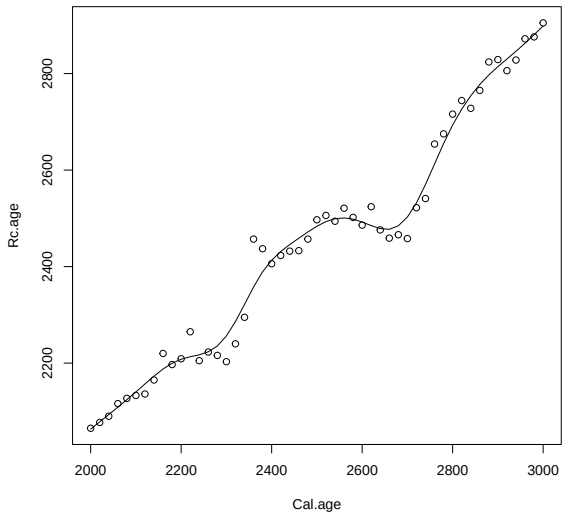


Figure 2.27. P-spline fit to the radiocarbon data.

The parameter λ would need to be tuned manually.

It is however simpler to use the function `gam` from the package `mgcv`, see Figure 2.28, which automatically tunes the smoothing parameters (though these can also be set manually, if needed).

The function `s` uses by default a penalty based on the integrated squared second derivative, but can be set to use a difference penalty by using the additional argument `bs='ps'`,

```
model <- gam(Rc.age~s(Cal.age), data=radiocarbon)
model

   ##
   ## Family: gaussian
   ## Link function: identity
   ##
   ## Formula:
   ## Rc.age ~ s(Cal.age)
   ##
   ## Estimated degrees of freedom:
   ## 7.56  total = 8.56
   ##
   ## GCV score: 1470.6

plot(model, residuals=TRUE)
```

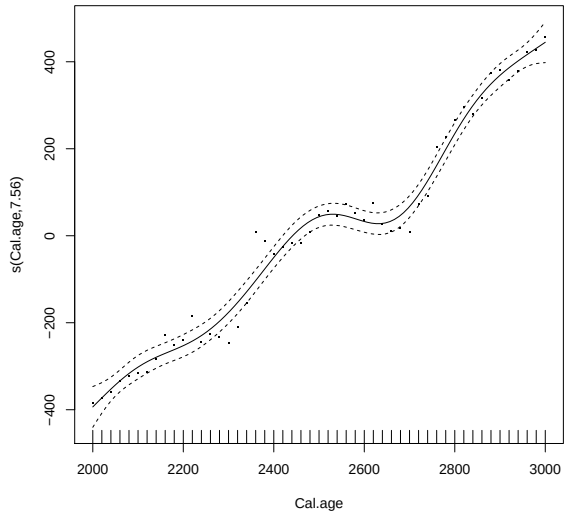


Figure 2.28. Penalised spline fit using `mgcv` to the radiocarbon data.

3

Quantile regression

When we talk about regression, we usually refer to mean regression which describes how the expected value of a response variable of interest varies with explanatory variables. Sometimes it is also useful to consider the effect of explanatory variables on the entire conditional distribution of the variable of interest. Quantile regression is one way of achieving this. In this chapter we will introduce quantile regression by first considering what is a quantile and what is meant by quantile regression, before presenting the main properties of quantiles and the theory of estimation and inference for quantile regression models. Throughout the chapter we will show examples of implementation of quantile regression in R.

To explain what we mean by the term *quantile regression*, we must first define the term *quantile*.

Definition 3.1. *Suppose that the random variable Y has cumulative distribution function (cdf) $F_Y(y) = P(Y \leq y)$. The τth quantile of Y is defined as*

$$Q_\tau(Y) = \inf\{y : F_Y(y) \geq \tau\},$$

where $0 < \tau < 1$ is the quantile level.

From the definition of a quantile we can see that $Q_{0.5}(Y)$ is the median, also referred to to as the second quartile, while $Q_{0.25}(Y)$ is the first quartile or 25th percentile and $Q_{0.75}(Y)$ is the third quartile or 75th percentile. Quantiles and percentiles are essentially the same thing, except that the former refer to proportions while the latter to percentages.

The quantile function $Q_\tau(Y)$ is a non-decreasing function of τ, i.e. $Q_{\tau_1}(Y) \leq Q_{\tau_2}(Y)$ for $\tau_1 < \tau_2$. In a regression setting we are actually interested in the τth *conditional* quantile, defined as follows.

Definition 3.2. *Suppose that we have a response variable Y, and that $\mathbf{x}$ is a p-dimensional predictor. Let $F_Y(y|\mathbf{x}) = P(Y \leq y|\mathbf{x})$ denote the conditional cdf of Y given $\mathbf{x}$. Then the τth conditional quantile of Y is defined as*

$$Q_\tau(Y|\mathbf{x}) = \inf\{y : F_Y(y|\mathbf{x}) \geq \tau\}.$$

This allows us to consider a model of the form

$$Q_\tau(Y|\mathbf{x}) = \mathbf{x}^\mathsf{T}\boldsymbol{\beta}(\tau), \quad 0 < \tau < 1, \tag{3.1}$$

which is similar to the more familiar model for the mean $\mathbb{E}(Y|\mathbf{x}) = \mathbf{x}^\mathsf{T}\boldsymbol{\beta}$, where $\boldsymbol{\beta}$ measures the marginal change in the mean of Y due to a marginal change in $\mathbf{x}$.

In the linear quantile regression model defined by equation (3.1), the regression coefficients are allowed to depend on the quantile level τ: $\boldsymbol{\beta} = (\beta_1(\tau), \ldots, \beta_p(\tau))^\mathsf{T}$ is the quantile coefficient that may depend on τ. The first element of $\mathbf{x}$ is equal to 1, corresponding to the intercept so that $Q_\tau(Y|\mathbf{x}) = \beta_1(\tau) + x_2\beta_2(\tau) + \cdots + x_p\beta_p(\tau)$. We can interpret $\boldsymbol{\beta}(\tau)$ as the marginal change in the τth quantile due to the marginal change in $\mathbf{x}$. Note that this is for a particular value of τ and that different quantiles can have coefficients that differ from each other in magnitude, sign, or both. Also note the monotonicity property of conditional quantiles: $Q_\tau(Y|\mathbf{x})$ is a nondecreasing function of τ for any given $\mathbf{x}$.

Situations in which the quantile coefficients may vary with τ can arise in a number of ways, as can be seen in the following examples.

Example 3.1 (Location-scale shift model). Consider random variabes Y_i, $i = 1, \ldots, n$ where

$$Y_i = \alpha + \mathbf{z}_i^\mathsf{T}\boldsymbol{\beta} + (1 + \mathbf{z}_i^\mathsf{T}\boldsymbol{\gamma})\varepsilon_i,$$

with $\varepsilon \overset{\text{i.i.d}}{\sim} F(\cdot)$. Then the conditional quantile function can be written as

$$Q_\tau(Y|\mathbf{x}_i) = \alpha(\tau) + \mathbf{z}_i^\mathsf{T}\boldsymbol{\beta}(\tau),$$

where $\alpha(\tau) = \alpha + F^{-1}(\tau)$ is non-decreasing in τ and $\boldsymbol{\beta}(\tau) = \boldsymbol{\beta} + \boldsymbol{\gamma}F^{-1}(\tau)$ may depend on τ. That is, the explanatory variable is allowed to have a different impact on different

quantiles of the Y distribution. If, however, $\boldsymbol{\gamma} = 0$, $\boldsymbol{\beta}(\tau) = \boldsymbol{\beta}$ becomes constant across quantile levels, and the model simplifies to a **location shift** model.

◁

Example 3.2 (Quantile treatment effect). Suppose that we are interested in exploring a treatment effect where the explanatory variable takes values $X_i = 0$ for the control group and $X_i = 1$ for the treatment group. Then the conditional distribution of Y given X will be $Y_i|X_i = 0 \sim F$ for the control group and $Y_i|X_i = 1 \sim G$ for the treatment group. The mean treatment effect is given by

$$\Delta = \mathbb{E}(Y_i|X_i = 1) - \mathbb{E}(Y_i|X_i = 0) = \int ydG(y) - \int ydF(y).$$

The *quantile treatment effect* can be thought of as the horizontal distance between F and G at y: $F(y) = G(y + \Delta(y))$ (Doksum, 1974). Then $\Delta(y)$ is uniquely defined as $\Delta(y) = G^{-1}(F(y)) - y$. Changing variables so that $\tau = F(y)$ (or $\tau = F(Y|x)$ to be more precise) we have the quantile treatment effect (QTE)

$$\delta(\tau) = \Delta(F^{-1}(\tau)) = G^{-1}(\tau) - F^{-1}(\tau) = Q_\tau(Y|X_i = 1) - Q_\tau(Y|X_i = 0).$$

Thus,

$$\Delta = \int_0^1 G^{-1}(u)du - \int_0^1 F^{-1}(u)du = \int_0^1 \delta(u)du.$$

We can write this as a quantile regression model with a binary explanatory variable x:

$$Q_\tau(Y|x) = \alpha(\tau) + \delta(\tau)x.$$

Assuming a location shift where the relationship between F and G is shown in the leftmost panel of Figure 3.1, we have

$$F(y) = G(y + \delta) \Rightarrow \delta(\tau) = \Delta = \delta,$$

resulting in a constant quantile treatment effect as can be seen in the rightmost panel of Figure 3.1.

In the case of a scale shift, with a cdf and density as shown in the left and middle panel of Figure 3.2 below, we have $\delta(0.5) = 0$ but $\delta(\tau) \neq 0$ at other quantiles, as shown in the right hand panel of Figure 3.2.

Finally the cdf, density and quantile treatment effect in the case of a location and scale shift are shown in Figure 3.3.

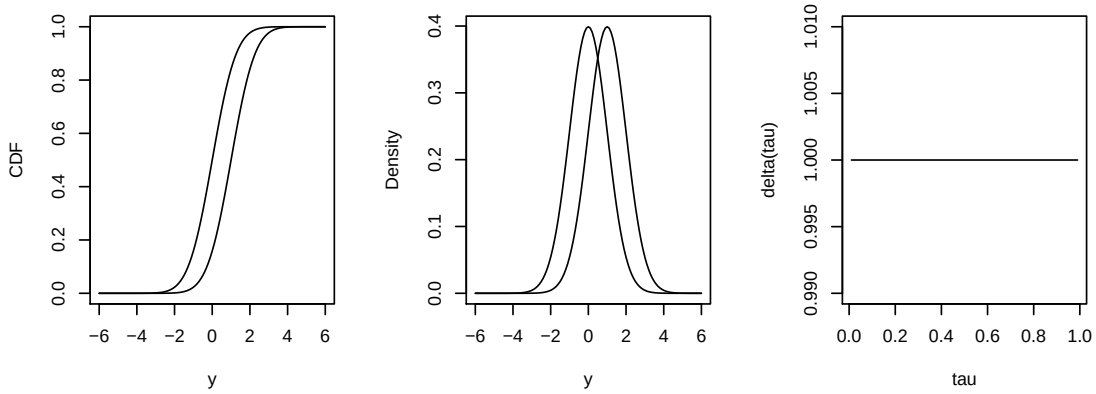


Figure 3.1. Cumulative distribution function, density and quantile treatment effect in the case of a location shift.

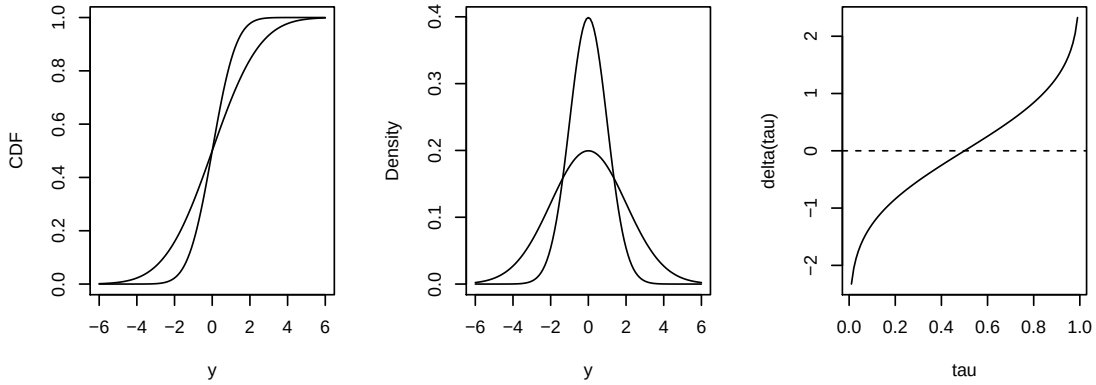


Figure 3.2. Cumulative distribution function, density and quantile treatment effect in the case of a scale shift.

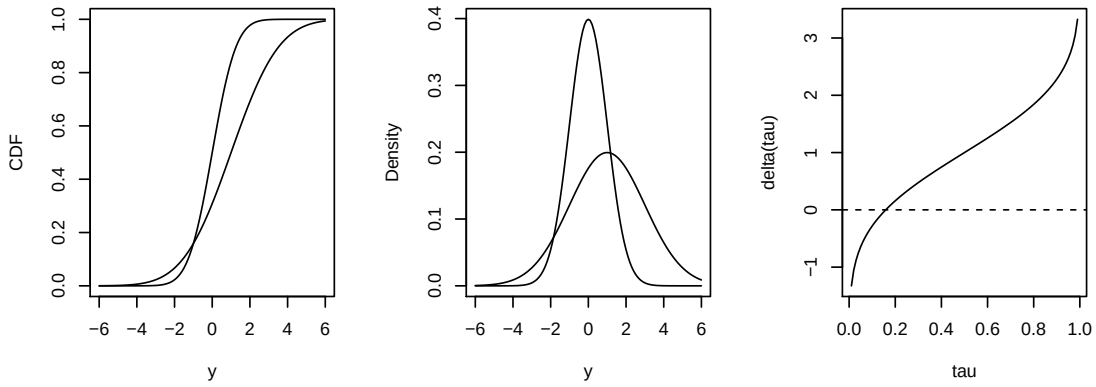


Figure 3.3. Cumulative distribution function, density and quantile treatment effect for a location and scale shift model.

◁

There are several reasons why using quantile regression might be a good idea. The most important one is that quantile regression allows us to study the impact of predictors on different quantiles of the response distribution, which provides a more complete picture of the relationship between Y and $\mathbf{x}$.

Example 3.3 (Are tropical cyclones becoming more severe?). The maximum wind speed of tropical cyclones in the North Atlantic was recorded over the years 1978 to 2009. A plot of the maximum wind speed against the year is shown in Figure 3.4, with the least squares regression line shown in red.

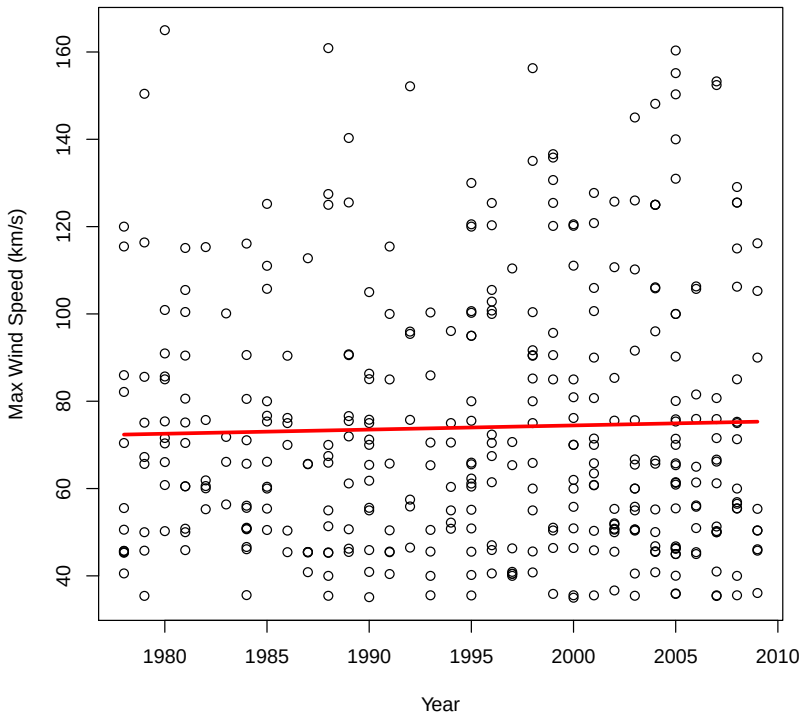


Figure 3.4. Maximum wind speed of North Atlantic cyclones over the period 1978-2009 with least squares regression line (solid red line).

Here a linear model with maximum wind speed Y as the response and the year x as the explanatory variable gives a slope estimate of 0.095 with a p-value of 0.569 indicating no significant trend in the mean. However, we might want to explore whether the *quantiles* of maximum wind speed change over time. Figure 3.5 shows the fitted quantile regression lines for $\tau = 0.25, 0.5, 0.75$ and 0.95. The coefficient of `year` is not significant for the

first three quantile levels (p-values of 0.100, 0.718 and 0.659 respectively), however for $\tau = 0.95$ the `year` coefficient is significant with a p-value of 0.009.

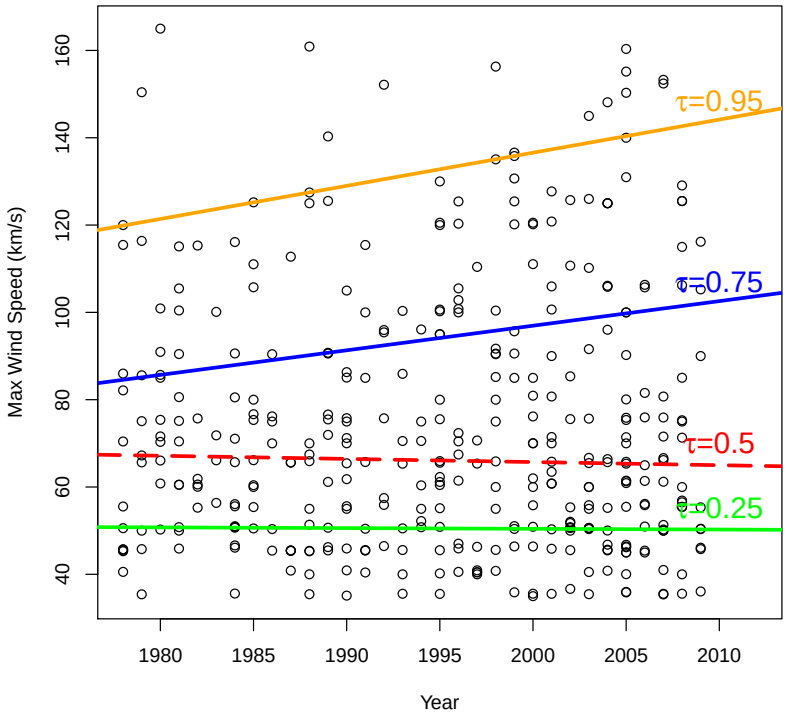


Figure 3.5. Quantile regression fit for maximum wind speed of North Atlantic cyclones as a function of time over the period 1978-2009 for $\tau = 0.25, 0.5, 0.75$ and 0.95.

So there is a significant trend for the cyclones with the highest wind speeds (95th percentile of wind speed) – they are getting faster over time.

◁

Another reason quantile regression might be useful is that it is robust to outliers in y observations, as can be seen in Figure 3.6.

A third reason quantile regression might be preferred in certain situations is that it does not make any distributional assumptions and therefore estimation and inference are distribution-free. This differentiates it from models which estimate conditional quantiles by assuming a distributional form, *e.g.* GAMLSS (see Chapter 5 for an example).

3.1 Properties of quantiles and quantile regression

In this section we will explore some of the basic properties of quantiles and quantile regression, beginning with some basic equivariance properties.

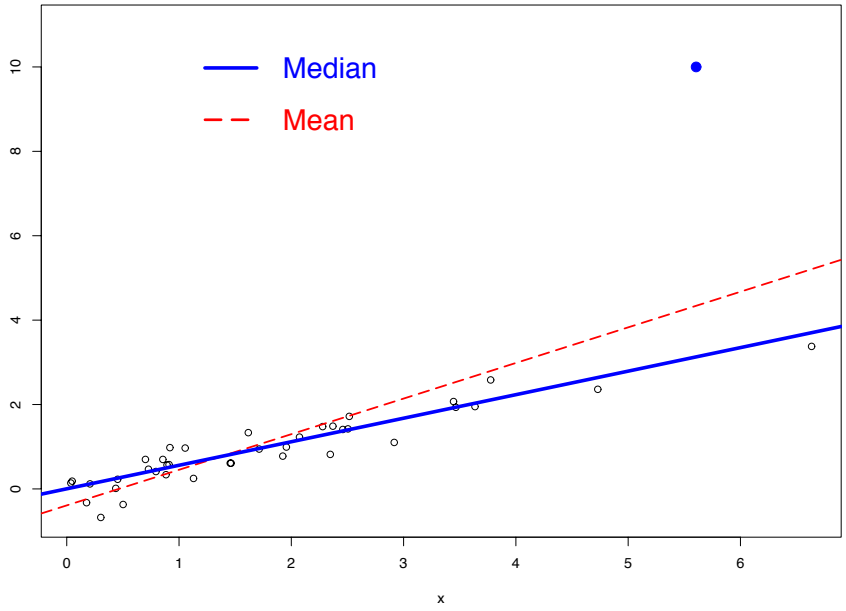


Figure 3.6. Fitted mean and median regression lines in the presence of an outlier.

Consider $\hat{\beta}(\tau; y, X)$, the estimator for the τth quantile regression based on observations (y, X) and let A be any $p \times p$ non-singular matrix, $\gamma \in \mathbb{R}^p$, and $a > 0$ be constant. Then for any $\tau \in [0, 1]$,

1. $\hat{\beta}(\tau; ay, X) = a\hat{\beta}(\tau; y, X)$ and $\hat{\beta}(\tau; -ay, X) = -a\hat{\beta}(1-\tau; y, X)$ (scale equivariance);
2. $\hat{\beta}(\tau; y + X\gamma, \mathbf{x}) = \hat{\beta}(\tau; y, X) + \gamma$ (regression shift);
3. $\hat{\beta}(\tau; y, XA) = A^{-1}\hat{\beta}(\tau; y, X)$ (reparameterisation of design).

In addition, conditional quantile functions are equivariant to monotone transformations. Suppose that $h(\cdot)$ is an increasing function on $\mathbb{R}$. Then for any variable Y,

$$Q_\tau(h(Y|X)) = h(Q_\tau(Y|X)).$$

That is, the quantiles of the transformed random variable $h(Y)$ are simply the transformed quantiles on the original scale. This is useful, for instance, when we log-transform the response. In the case of quantile regression $Q_\tau(\log(Y|X)) = \log(Q_\tau(Y|X)$, while the same is not true in mean regression as $\mathbb{E}(\log(Y)|X) \neq \log(\mathbb{E}(Y|X))$ in general.

Another interesting property of linear quantile regression is that it exactly fits p observations. If the first column of the design matrix is a column of ones corresponding to the intercept, then there are roughly p zero, $n\tau$ negative and $n(1-\tau)$ positive residuals $y_i - \mathbf{x}_i^\mathsf{T}\hat{\boldsymbol{\beta}}(\tau)$. We will look at this property in more detail in Section 3.3.2.

3.2 Estimation and computation

We have seen why quantile regression is useful and have explored some of its properties. Now let us turn our attention to coefficient estimation. We begin by comparing the estimation procedure for mean and median regression before we generalise to other quantiles. Suppose that we observe realisations $\{y_i\}$ with corresponding values of the explanatory variables $\{\mathbf{x}_i,\, i = 1, \ldots, n\}$. In mean regression using ordinary least squares estimation (OLS), we estimate the regression coefficients by minimising the sum of squares. The simplest case is that of an intercept-only model, where $\mathbb{E}(Y) = \mu_Y = \text{argmin}_a \mathbb{E}(Y - a)^2$. The sample mean solves $\min_a \sum_{i=1}^n (y_i - a)^2$. When other explanatory variables are present, the least squares estimates are obtained by minimising $\sum_{i=1}^n (y_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta})^2$. This gives estimates that are consistent for the conditional mean $\mathbb{E}(Y|\mathbf{x}) = \mathbf{x}^\mathsf{T}\boldsymbol{\beta}$. Similarly we can estimate the sample median and the regression coefficients in median regression using the method of least absolute deviations (LAD). The median is defined as $Q_{0.5}(Y) = \text{argmin}_a \mathbb{E}|Y - a|$ and the sample median solves $\min_a \sum_{i=1}^n |y_i - a|$. If we assume that the conditional median of Y given $\mathbf{x}$ is equal to $\mathbf{x}^\mathsf{T}\boldsymbol{\beta}(0.5)$, then $\hat{\boldsymbol{\beta}}(0.5)$ can be obtained by solving

$$\min_\beta \sum_{i=1}^n |y_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}|.$$

Now let us consider quantile regression at quantile level $0 < \tau < 1$. The τth quantile of Y is given by

$$Q_\tau(Y) = \underset{a}{\text{argmin}} \;\; \mathbb{E}\left[\rho_\tau(Y - a)\right],$$

where $\rho_\tau(u) = u\tau - I(u < 0)$ is the quantile loss function, shown in Figure 3.7.

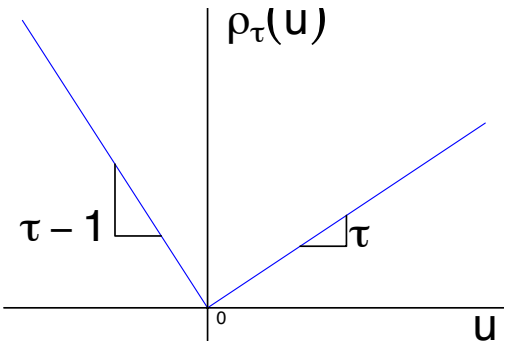


Figure 3.7. The objective function for the τth conditional quantile.

The τth sample quantile of Y solves

$$\min_a \sum_{i=1}^{n} \rho_\tau(y_i - a).$$

If we assume that $Q_\tau(Y|\mathbf{x}) = \mathbf{x}^\mathsf{T}\boldsymbol{\beta}(\tau)$, then

$$\hat{\boldsymbol{\beta}}(\tau) = \operatorname*{argmin}_\beta \sum_{i=1}^{n} \rho_\tau(y_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}).$$

The solution that minimises the objective function is typically found using linear programming. The main idea of using linear programming for solving a standard minimisation problem can be summarised as follows.

Suppose that we wish to solve

$$\min_{\boldsymbol{y}\in\mathbb{R}^m} \boldsymbol{y}^\mathsf{T}\mathbf{b},$$

subject to constraints

$$\boldsymbol{y}^\mathsf{T}\boldsymbol{A} \geq \mathbf{c}^\mathsf{T},$$

and $y_1 \geq 0, \ldots, y_m \geq 0$. Here $\boldsymbol{A}$ is an $m \times n$ matrix, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{c} \in \mathbb{R}^n$.

The minimisation problem above has a **dual maximisation problem** which can be expressed as

$$\max_{\mathbf{x}\in\mathbb{R}^n} \mathbf{c}^\mathsf{T}\mathbf{x},$$

subject to constraints $\boldsymbol{A}\boldsymbol{x} \geq \mathbf{b}$ and $\mathbf{x} \geq 0$.

Noting that the linear quantile regression model can be rewritten as

$$y_i = \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau) + e_i = \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau) + (u_i - v_i),$$

where $u_i = e_i I(e_i > 0)$ and $v_i = |e_i| I(e_i < 0)$, we see that the minimisation problem

$$\min_{\mathbf{b}} \sum_{i=1}^{n} \rho_\tau(y_i - \mathbf{x}_i^\mathsf{T}\mathbf{b})$$

becomes

$$\min_{\mathbf{b},\mathbf{u},\mathbf{v}} \ \tau 1_n^\mathsf{T}\mathbf{u} + (1-\tau)1_n^\mathsf{T}\mathbf{v}$$

$$\text{such that } \ \boldsymbol{y} - \mathbf{x}^\mathsf{T}\mathbf{b} = \mathbf{u} - \mathbf{v}$$

where

$$\mathbf{b} \in \mathbb{R}^p, \quad \mathbf{u} \geq 0, \quad \mathbf{v} \geq 0.$$

This is a minimisation problem which can be solved using linear programming methods. In the remainder of this section we will briefly decribe the solving algorithms that are relevant for computation in quantile regression.

1. **Simplex method**: An essential linear programming theory result is that solutions are focused on the vertices of the constraint set. This is the subset that provides an exact fit to the p observations. The first phase of the simplex algorithm finds an initial feasible vertex. This can be achieved by choosing any subset h such that $\mathbf{x}(h)$ is of full rank and $\mathbf{b}(h) = \mathbf{x}(h)^{-1}\boldsymbol{y}$. In the second phase the algorithm travels from one vertex to another until optimality is achieved. The moves are along the direction of "steepest descent", i.e. the one with the most negative directional derivative; see Barrodale and Roberts (1974) for more details. The simplex method is the default method for estimating quantile coefficients in R as it is efficient for problems with modest sample size (n up to several thousands) and its speed is comparable to the least squares estimator for n up to several hundreds. However, the algorithm is very slow relative to OLS for larger sample sizes. For more details on the simplex algorithm for obtaining quantile regression coefficients see Keener and d'Orey (1987).
2. **Frisch-Newton interior point method**: In contrast to the simplex, this algorithm traverses the interior of the feasible region, which makes it more efficient than the simplex algorithm for larger sample sizes. For more details see Portnoy and Koenker (1997).
3. **Sparse regression quantile fitting**: This is a sparse implementation of the Frisch-Newton interior-point algorithm which is efficient when the design matrix has many zeros, as is often the case when the predictors contain several factors. This method is implemented in the R function `rq.fit.sfn()` in `library(quantreg)`. For more details see Koenker and Ng (2003).

The three algorithms described above are implemented in `library(quantreg)` in RṪhe default method of estimation is `method=br` for the simplex (Barrodale-Roberts) algorithm. Other options are `method="fn"` for the Frisch-Newton interior point method which may be better for larger problems and `method="pfn"` (Frisch-Newton approach with preprocessing) for very large problems. Finally `method="sfn"` uses the interior point algorithm in the case of a sparse design matrix, *e.g.* when the model includes factors.

Example 3.4 (Quantile regression in R *estimation and computation of quantile coefficients).* We illustrate quantile regression coefficient estimation in R by simulating heteroscedastic data and fitting quantile regression models for different values of τ.

First we generate the data, setting the seed for reproducibility:

```
x <- seq(0,100,length.out = 100)
sig <- 0.1 + 0.05*x
b_0 <- 6
b_1 <- 0.1
set.seed(9873)
e <- rnorm(100,mean = 0, sd = sig)
y <- b_0 + b_1*x + e
dat <- data.frame(x,y)
```

We can plot the data along with the least squares regression line as shown in Figure 3.8. Notice how this fit is not particularly helpful for x values that lie away from zero.

```
library(ggplot2)
ggplot(dat, aes(x,y)) + geom_point() + geom_smooth(method="lm")
```

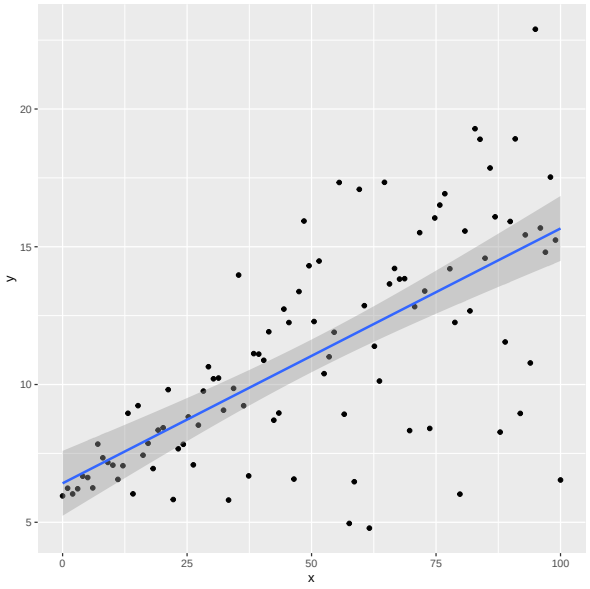


Figure 3.8. Simulated data with least squares regression line and confidence intervals.

We fit a quantile regression model using the `rq()` function which takes a `tau` argument in addition to the usual `formula` and `data` arguments. The summary function gives a summary of the fit including coefficient estimates, standard errors and p-values. Here we print the results for $\tau = 0.9$. The fitted quantile regression line is shown in Figure 3.9.

For a more complete picture we might wish to fit quantile regression models for several τ values. The regression lines can be plotted against the data as shown in Figure 3.10.

In the case of multiple τ values a summary plot of the β coefficient estimates gives a quick impression of whether they are constant across quantile levels. Figure 3.11 shows that the quantile coefficients are not constant for all τ as the grey bands for the quantiles do not overlap with those for least squares regression (dashed red lines).

```
library(quantreg)
fit <- rq(y ~ x, data=dat, tau = 0.9)

summary(fit)

   ##
   ## Call: rq(formula = y ~ x, tau = 0.9, data = dat)
   ##
   ## tau: [1] 0.9
   ##
   ## Coefficients:
   ##             coefficients lower bd upper bd
   ## (Intercept) 6.76819      6.14746  9.04763
   ## x           0.15111      0.12035  0.17442

library(ggplot2)
ggplot(dat, aes(x,y)) + geom_point() + geom_abline(intercept=coef(fit)[1], slope=coef(fit)[2])
```

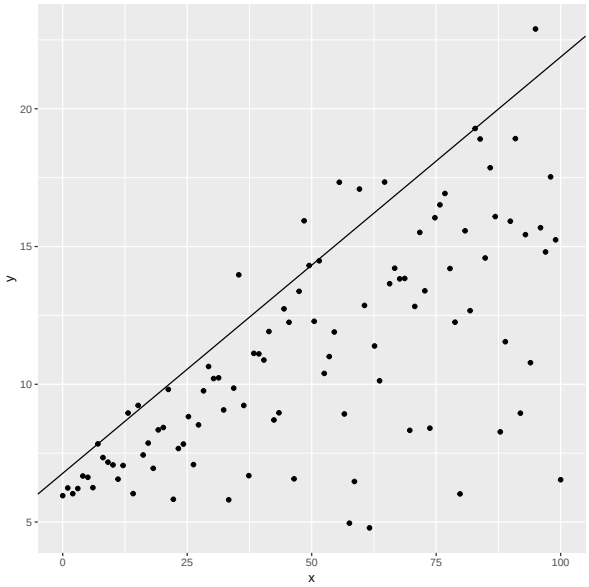


Figure 3.9. Quantile regression fit for $\tau = 0.9$.

```
taus <- 1:9/10
fit2 <- rq(y ~ x, data=dat, tau = taus)

ggplot(dat, aes(x,y)) + geom_point() + geom_quantile(quantiles = taus)
```

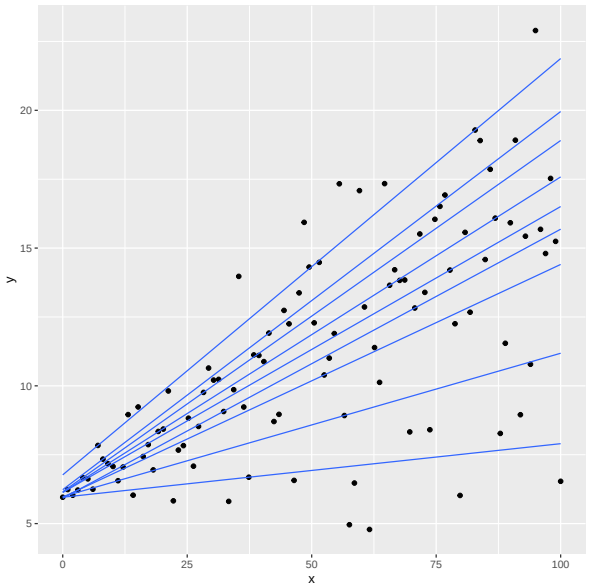


Figure 3.10. Fitted quantile regression lines for $\tau = 0.1, 0.2, \ldots, 0.9$.

```
plot(summary(fit2), parm="x")
```

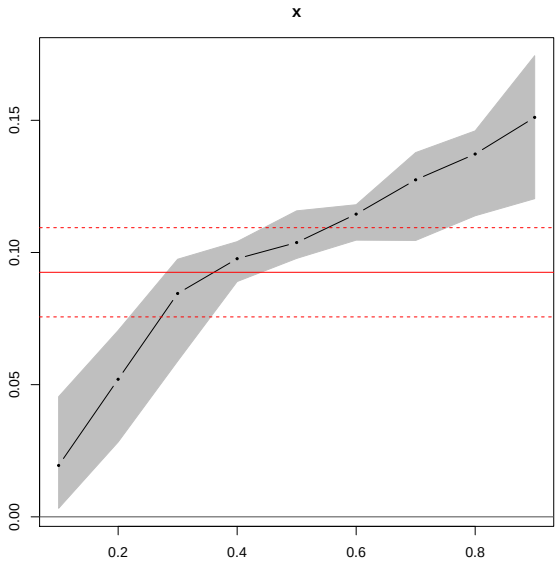


Figure 3.11. Coefficient estimates for $\tau = 0.1, 0.2, \ldots, 0.9$ along with confidence intervals in grey. The confidence intervals do not overlap with the least squares regression line (horizontal solid line in red).

We will discuss confidence intervals in more detail in Section 3.4. ◁

3.3 Statistical properties of quantile regression coefficient estimates

3.3.1 Consistency and asymptotic normality

The quantile regression coefficient estimates can be shown to be consistent and to follow an asymptotic normal distribution. The coefficient estimator in a linear quantile regression model is given by

$$\hat{\boldsymbol{\beta}}(\tau) = \underset{\mathbf{b}\in\mathbb{R}^p}{\arg\min} \sum_{i=1}^{n} \rho_\tau(y_i - \mathbf{x}_i^\mathsf{T}\mathbf{b}).$$

The coefficient estimates are consistent under the following regularity conditions.

A1. The distribution functions of Y given $\mathbf{x}_i, F_i(\cdot)$, are absolutely continuous with continuous densities $f_i(\cdot)$ that are uniformly bounded away from 0 and ∞ at $\xi_i(\tau) = Q_\tau(Y|\mathbf{x}_i)$.

A2. There exist positive definite matrices D_0 and D_1 such that

(i) $\lim_{n\to\infty} n^{-1} \sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\mathsf{T} = D_0$;

(ii) $\lim_{n\to\infty} n^{-1} \sum_{i=1}^{n} f_i\left(\xi_i(\tau)\right) \mathbf{x}_i\mathbf{x}_i^\mathsf{T} = D_1(\tau)$;

(iii) $\max_{i=1,\ldots,n}||\mathbf{x}_i|| = o(n^{\frac{1}{2}})$.

Theorem 3.3 (Consistency). *Under conditions A1 and A2(i),* $\hat{\boldsymbol{\beta}}(\tau) \xrightarrow{p} \boldsymbol{\beta}(\tau)$.

Proof (Sketch). Following Pollard (1991), use the uniform law of large numbers to show that

$$\sup_{\mathbf{b}\in\mathcal{B}} n^{-1} \sum_{i=1}^{n} \left[\rho_\tau(y_i - \mathbf{x}_i^\mathsf{T}\mathbf{b}) - \mathbb{E}\{\rho_\tau(y_i - \mathbf{x}_i^\mathsf{T}\mathbf{b})\}\right] \to 0,$$

where $\mathcal{B}$ is a compact subset of $\mathbb{R}^p$. Note that $\hat{\boldsymbol{\beta}}(\tau) \to \boldsymbol{\beta}(\tau)$ holds if for any $\varepsilon > 0$, $\bar{Q}(\mathbf{b}) \equiv n^{-1}\sum_{i=1}^{n} \mathrm{E}\left[\rho_\tau(y_i - \mathbf{x}_i^\mathsf{T}\mathbf{b})\right]$ is bounded away from zero with probability approaching one for any $||\mathbf{b} - \boldsymbol{\beta}(\tau)|| \geq \varepsilon$.

Under Conditions A1 and A2(i), $\bar{Q}(\mathbf{b})$ has a unique minimiser $\boldsymbol{\beta}(\tau)$.

□

Next we will discuss asymptotic normality of the quantile regression coefficient estimates.

Theorem 3.4. *Under Conditions A1 and A2,*

$$\sqrt{n}\left(\hat{\boldsymbol{\beta}}(\tau) - \boldsymbol{\beta}(\tau)\right) \xrightarrow{d} N\left(0, \tau(1-\tau)D_1^{-1}D_0D_1^{-1}\right).$$

For i.i.d. error models, i.e. $f_i(\xi_i(\tau)) = f_\varepsilon(0)$, the above result can be simplified to

$$\sqrt{n}\left(\hat{\boldsymbol{\beta}}(\tau) - \boldsymbol{\beta}(\tau)\right) \xrightarrow{d} N\left(0, \frac{\tau(1-\tau)}{f_\varepsilon^2(0)}D_0^{-1}\right).$$

Proof (Sketch). Define $\hat{\boldsymbol{\delta}}_n = \sqrt{n}(\hat{\boldsymbol{\beta}}(\tau) - \boldsymbol{\beta}(\tau))$, which is the minimiser of

$$Z_n(\boldsymbol{\delta}) = \sum_{i=1}^{n} \left[\rho_\tau(\varepsilon_i - n^{-\frac{1}{2}}\mathbf{x}_i^\mathsf{T}\boldsymbol{\delta}) - \rho_\tau(\varepsilon_i)\right],$$

where $\varepsilon_i = y_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau)$.

By Knight's identity (Knight (1998))

$$\rho_\tau(u - v) - \rho_\tau(u) = -v\psi_\tau(u) + \int_0^v \{I(u \leq s) - I(u \leq 0)\}ds,$$

where $\psi_\tau(u) = \tau I(u < 0)$ (see Figure 3.12), it can be shown that

$$Z_n(\boldsymbol{\delta}) = -\boldsymbol{\delta}^\mathsf{T} W_n + Z_{2n}(\boldsymbol{\delta}),$$

where

$$W_n = n^{-\frac{1}{2}} \sum_{i=1}^{n} \mathbf{x}_i \psi_\tau(\varepsilon_i) \xrightarrow{d} W, \quad W \sim N(0, \tau(1-\tau)D_0),$$

and

$$Z_{2n}(\boldsymbol{\delta}) = \sum_{i=1}^{n} Z_{2ni}(\boldsymbol{\delta}) = \sum_{i=1}^{n} \int_0^{\mathbf{x}_i^\mathsf{T}\boldsymbol{\delta}/\sqrt{n}} [I(\varepsilon_i \le s) - I(\varepsilon_i \le 0)]\, ds.$$

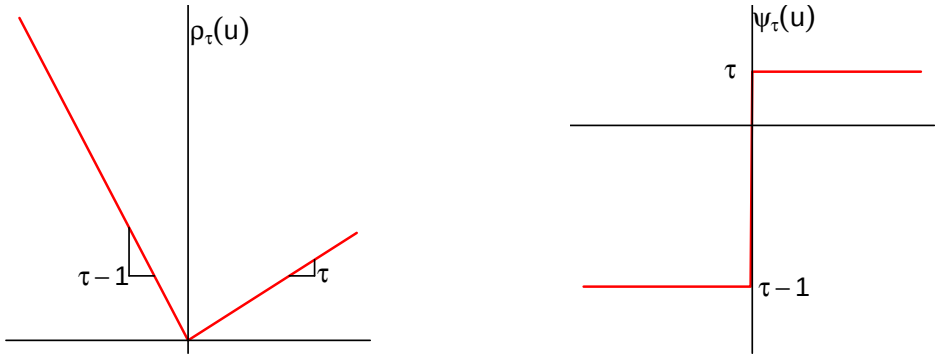


Figure 3.12. The objective function $\rho(\tau)$ (left panel) and function $\psi(\tau)$ (right panel).

Using a Taylor series expansion,

$$\mathbb{E}(Z_{2n}(\boldsymbol{\delta})) = \frac{1}{2}\boldsymbol{\delta}^\mathsf{T} D_1 \boldsymbol{\delta} + o(1),$$

and

$$\mathrm{Var}(Z_{2n}(\boldsymbol{\delta})) \le \frac{1}{\sqrt{n}} \max_i |\mathbf{x}_i^\mathsf{T}\boldsymbol{\delta}| \mathbb{E}\left[Z_{2n}(\boldsymbol{\delta})\right] = o(\|\boldsymbol{\delta}\|)$$

Thus

$$Z_n(\boldsymbol{\delta}) \to Z_0(\boldsymbol{\delta}) = -\boldsymbol{\delta}^\mathsf{T} W + \frac{1}{2}\boldsymbol{\delta}^\mathsf{T} D_1 \boldsymbol{\delta}.$$

$Z_0(\boldsymbol{\delta})$ is convex and thus has a unique minimiser

$$\operatorname{argmin}_{\boldsymbol{\delta}} Z_0(\boldsymbol{\delta}) = D_1^{-1} W.$$

By the complexity lemma of Pollard (1991), the result can be strengthened to be uniform over $\boldsymbol{\delta}$. Thus, $\hat{\boldsymbol{\delta}}_n = \operatorname{argmin} Z_n(\boldsymbol{\delta}) \xrightarrow{d} D_1^{-1} W$.

An alternative approach based on the score function can be found in He and Shao (1996). □

3.3.2 Subgradient condition

Let us now explore the condition that ensures basic optimality in a quantile regression problem. Define

$$R(\boldsymbol{\beta}) = \sum_{i=1}^{n} \rho_\tau(y_i - \mathbf{x}_i^\mathsf{T} \boldsymbol{\beta})$$

noting that $R(\boldsymbol{\beta})$ is piecewise linear, continuous and differentiable except at points such that $y_i - \mathbf{x}_i^\mathsf{T} \boldsymbol{\beta} = 0$. The directional derivative of $R(\boldsymbol{\beta})$ in direction $\mathbf{w}$ is given by

$$\nabla R(\boldsymbol{\beta} \boldsymbol{w}) = \frac{d}{dt} R(\boldsymbol{\beta} + \mathbf{x}_i^\mathsf{T} \mathbf{w})|_{t=0}.$$

We have

$$\begin{aligned}
&\frac{d}{dt} \rho_\tau(y - \mathbf{x}^\mathsf{T} \boldsymbol{\beta} - \mathbf{x}^\mathsf{T} \mathbf{w} t)|_{t=0} \\
&\qquad = \frac{d}{dt} (y - \mathbf{x}^\mathsf{T} \boldsymbol{\beta} - \mathbf{x}^\mathsf{T} \mathbf{w}) \{\tau - I(y - \mathbf{x}^\mathsf{T} \boldsymbol{\beta} - \mathbf{x}^\mathsf{T} \mathbf{w} < 0)\}|_{t=0} \\
&\qquad = \begin{cases} \mathbf{x}^\mathsf{T} \mathbf{w} \tau, & y - \mathbf{x}^\mathsf{T} \boldsymbol{\beta} > 0 \\ -\mathbf{x}^\mathsf{T} \mathbf{w} (1 - \tau), & y - \mathbf{x}^\mathsf{T} \boldsymbol{\beta} < 0 \\ -\mathbf{x}^\mathsf{T} \mathbf{w} \{\tau - I(-\mathbf{x}^\mathsf{T} \mathbf{w} < 0)\}, & y - \mathbf{x}^\mathsf{T} \boldsymbol{\beta} = 0 \end{cases} \\
&\qquad = \mathbf{x}^\mathsf{T} \mathbf{w} \psi_\tau^*(y - \mathbf{x}^\mathsf{T} \boldsymbol{\beta}, -\mathbf{x}^\mathsf{T} \mathbf{w}),
\end{aligned}$$

where

$$\psi_\tau^*(u, v) = \begin{cases} \tau - I(u < 0), & u \neq 0 \\ \tau - I(v < 0), & u = 0. \end{cases}$$

Thus

$$\nabla R(\boldsymbol{\beta}, \boldsymbol{w}) = \sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{w} \psi_\tau^*(y_i - \mathbf{x}_i^\mathsf{T} \boldsymbol{\beta}, -\mathbf{x}_i^\mathsf{T} \mathbf{w}).$$

Noting that $\nabla R(\hat{\boldsymbol{\beta}}, \mathbf{w}) \geq 0$ for all $\mathbf{w} \in \mathbb{R}^p$ with $\|\mathbf{w}\| = 1$ we have

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} R(\boldsymbol{\beta}).$$

Theorem 3.5. *If $(y, \mathbf{x})$ are in general position (i.e. if any p observations of them yield a unique exact fit), then there exists a minimiser of $R(\boldsymbol{\beta})$ of the form $\mathbf{b}(h) = \mathbf{x}(h)^{-1}y(h)$ if and only if, for some $h \in \mathcal{H}$,*

$$-\tau 1_p \leq \psi(h) \leq (1-\tau)1_p,$$

where $\xi(h)^T = \sum_{i \in \bar{h}} \psi_\tau(y_i - \mathbf{x}_i^T\mathbf{b}(h))\mathbf{x}_i^T\mathbf{x}(h)^{-1}$, $\psi_\tau(u) = \tau - I(u < 0)$ and $\bar{h}$ is the complement of h.

What this result says is that the quantile regression line fits p points exactly. This may seem surprising, but of course we shouldn't forget that all the points were used to determine which p points should be interpolated.

Proof. In linear programming, vertex solutions, also known as basic solutions, correspond to points at which p observations are interpolated, i.e. $(y(h), \mathbf{x}(h)) = \{(y_i, \mathbf{x}_i), i \in h\}$. That is, the basic solutions pass through these n points as

$$\mathbf{b}(h) = \mathbf{x}(h)^{-1}y(h), \quad h \in \mathcal{H}^* - \{h \in \mathcal{H}^* : |\mathbf{x}(h)| \neq 0\}.$$

For any $\mathbf{w} \in \mathbb{R}^p$, reparameterise to get $\mathbf{v} = \mathbf{x}(h)\mathbf{w}$, i.e. $\mathbf{w} = \mathbf{x}(h)^{-1}\mathbf{v}$.

For a basic solution $\mathbf{b}(h)$ to be the minimiser, we need for all $\mathbf{v} \in \mathbb{R}^p$,

$$-\sum_{i=1}^{n} \psi_\tau^*\{y_i - \mathbf{x}_i^\mathsf{T}\mathbf{b}(h), -\mathbf{x}_i^\mathsf{T}\mathbf{x}(h)^{-1}\mathbf{v}\}\mathbf{x}_i^\mathsf{T}\mathbf{x}(h)^{-1}\mathbf{v} \geq 0.$$

Without loss of generality, assume that $\mathbf{x}(h) = (\mathbf{x}_1^\mathsf{T}, \ldots, \mathbf{x}_p^\mathsf{T})^\mathsf{T}$. If $i \in h, \mathbf{x}_i^\mathsf{T}\mathbf{x}(h) = \mathbf{e}_i^\mathsf{T}$, where $\mathbf{e}_i$ is a p-dimensional vector containing all zeros except the ith element being of 1. Thus $\mathbf{e}_i\mathbf{v} = v_i$. If $(y, \mathbf{x})$ are in general position, none of the residuals $y_i - \mathbf{x}_i^\mathsf{T}\mathbf{b}(h)$ with $i \in \bar{h}$ is zero. If the y_i's have a density wrt Lesbesgue measure, then with probability one $(y, \mathbf{x})$ are in general position. The space of directions $\mathbf{v} \in \mathbb{R}^p$ is spanned by $\mathbf{v} = \pm\mathbf{e}_k, k = 1, \ldots, p$, so the equation above holds for any $\mathbf{v} \in \mathbb{R}^p$ if and only if the inequality holds for $\pm\mathbf{e}_k, k = 1, \ldots, p$.

Therefore, the equation above becomes

$$0 \leq -\sum_{i \in h} \psi_\tau^*\{0, -v_i\}v_i - \boldsymbol{\xi}(h)^\mathsf{T}\mathbf{v},$$

where $\boldsymbol{\xi}(h)^\mathsf{T} = \sum_{i\in\bar{h}} \psi_\tau(y_i - \mathbf{x}_i^\mathsf{T}\mathbf{b}(h))\mathbf{x}_i^\mathsf{T}\mathbf{x}(h)^{-1}$.

If $\mathbf{v} = \mathbf{e}_i$, we have

$$0 \leq -(\tau - 1) - \xi_i(h), \quad i = 1, \ldots, p.$$

If $\mathbf{v} = -\mathbf{e}_i$, we have

$$0 \leq \tau + \xi_i(h), \quad i = 1, \ldots, p.$$

That is,

$$-\tau 1_p \leq \boldsymbol{\xi}(h) \leq (1 - \tau)1_p.$$

□

3.3.3 Bahadur representation

A linear representation for the quantile regression estimator can be obtained under the following regularity conditions.

C1. The distribution functions of Y given $\mathbf{x}_i$, $F_i(\cdot)$, have continuous densities $f_i(\cdot)$ that are bounded away from zero and uniformly bounded away from infinity in a neighbourhood of $\xi_i(\tau) = Q_\tau(Y|\mathbf{x}_i), \quad i = 1, \ldots, n$. In addition, the first derivative of $f_i(\cdot)$ is uniformly bounded in a neighbourhood of $\xi_i(\tau), \ i = 1, \ldots, n$.

C2. $\max_{i=1,\ldots,n}||\mathbf{x}_i|| = O(n^{\frac{1}{4}}\{\log(n)\}^{\frac{1}{2}})$.

C3. $n^{-1}\sum_{i=1}^n ||\mathbf{x}_i||^4 \leq B$ for some finite constant B.

C4. There exist positive definite matrices D_0 and D_1 such that

$$\lim_{n\to\infty} n^{-1}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^\mathsf{T} = D_0 \text{ and}$$

$$\lim_{n\to\infty} n^{-1}\sum_{i=1}^n f_i(\mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau))\mathbf{x}_i\mathbf{x}_i^\mathsf{T} = D_1.$$

Theorem 3.6. *Assuming that Conditions C1-C4 above hold, then*

$$\sqrt{n}(\hat{\boldsymbol{\beta}}(\tau) - \boldsymbol{\beta}(\tau)) = D_1^{-1}\sqrt{n}\sum_{i=1}^n \mathbf{x}_i\psi_\tau(y_i - \mathbf{x}_i^T\boldsymbol{\beta}(\tau)) + O_p(n^{\frac{1}{4}}\sqrt{\log n}).$$

For a linear regression model with *i.i.d.* errors $\varepsilon_i = y_i - \mathbf{x}^\mathsf{T}\boldsymbol{\beta}(\tau) \sim F_\varepsilon$, we have

$$\sqrt{n}\{\hat{\boldsymbol{\beta}}(\tau) - \boldsymbol{\beta}(\tau)\} = \frac{1}{f_\varepsilon(0)}D_0^{-1}\sqrt{n}\sum_{i=1}^n \mathbf{x}_i\psi_\tau(\varepsilon_i) + O_p(n^{-\frac{1}{4}}\sqrt{\log n}).$$

Proof (Sketch). **Step 1 (uniform approximation)**: Let C be some fixed constant. Define

$$R_n(\boldsymbol{\delta}) = \sum_{i=1}^{n} \mathbf{x}_i \big[\psi_\tau(\varepsilon_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\delta}) - \psi_\tau(\varepsilon_i)\big],$$

where $\varepsilon_i = y_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau)$. Then

$$\sup_{\boldsymbol{\delta}:||\boldsymbol{\delta}||\leq C} ||R_n(\boldsymbol{\delta}) - \mathbb{E}\left[R_n(\boldsymbol{\delta})\right]|| = O_p(n^{\frac{1}{2}}(\log n)||\boldsymbol{\delta}||^{\frac{1}{2}}).$$

The uniform approximation can be obtained by applying exponential inequality and chaining arguments, *e.g.* applying Lemma 4.1 of *He and Shao (1996).*

Step 2: The consistency of $\hat{\boldsymbol{\beta}}(\tau)$ was proven earlier (Theorem 3.3).

Step 3: Define $\boldsymbol{\delta} = (\mathbf{b} - \beta(\tau))$. Then

$$\begin{aligned} R_n(\boldsymbol{\delta}) &= \sum_{i=1}^{n} \mathbf{x}_i \left[\psi_\tau(\varepsilon_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\delta}) - \psi_\tau(\varepsilon_i)\right] \\ &= \sum_{i=1}^{n} \mathbf{x}_i \big[\psi_\tau(y_i = \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau) - \mathbf{x}_i^\mathsf{T}\boldsymbol{\delta}) - \psi_\tau(y_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau))\big] \end{aligned}$$

and

$$\begin{aligned} \mathbb{E}(R_n(\boldsymbol{\delta})) &= \sum_{i=1}^{n} \mathbf{x}_i \big[\tau - F_i(\mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau) + \mathbf{x}_i^\mathsf{T}\boldsymbol{\delta})\big] \\ &= \sum_{i=1}^{n} \mathbf{x}_i \big[F_i(\mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau)) - F_i(\mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau) + \mathbf{x}_i^\mathsf{T}\boldsymbol{\delta})\big] \\ &= -\sum_{i=1}^{n} \mathbf{x}_i \big[f_i(\mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau))\mathbf{x}_i^\mathsf{T}\boldsymbol{\delta} + f_i'(\eta_i)(\mathbf{x}_i^\mathsf{T}\boldsymbol{\delta})^2\big] \\ &= -\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\mathsf{T} f_i(\mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau))\boldsymbol{\delta} + O\big(\sum_{i=1}^{n} ||\mathbf{x}_i||^3||\boldsymbol{\delta}||^2\big). \end{aligned}$$

Define $\hat{\boldsymbol{\delta}} = \hat{\boldsymbol{\beta}}(\tau)$. By the uniform approximation and the root-n consistency of $\hat{\boldsymbol{\beta}}(\tau)$, under Conditions C1-C3 we get

$$\begin{aligned} &\sum_{i=1}^{n} \mathbf{x}_i\psi_\tau(y_i - \mathbf{x}_i^\mathsf{T}\hat{\boldsymbol{\beta}}(\tau)) - \sum_{i=1}^{n} \mathbf{x}_i\psi_\tau(y_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau)) \\ &= -\Big[n^{-1}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\mathsf{T} f_i(\mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau))\Big] n \\ &\quad + O_p(n\|\hat{\boldsymbol{\delta}}\|^2) + O_p(n^{1/2}(\log n)\|\hat{\boldsymbol{\delta}}\|^{1/2}). \end{aligned}$$

By the subgradient condition and Condition C2,

$$\sum_{i=1}^{n} \mathbf{x}_i \psi_\tau(y_i - \mathbf{x}_i^\mathsf{T} \hat{\boldsymbol{\beta}}(\tau)) = O_p(p \max_{i=1,\ldots,n} ||x_i||) = O_p(n^{\frac{1}{4}}/\sqrt{\log n}).$$

Combining with the previous equation, we get $\hat{\boldsymbol{\delta}} = O_p(n^{-1/2} \log n)$.

Therefore,

$$\sqrt{n}(\hat{\boldsymbol{\beta}}(\tau) - \boldsymbol{\beta}(\tau)) = D_1^{-1} n^{-\frac{1}{2}} \sum_{i=1}^{n} \mathbf{x}_i \psi_\tau(y_i - \mathbf{x}_i^\mathsf{T} \boldsymbol{\beta}(\tau)) + O_p(n^{\frac{-1}{4}} \sqrt{\log n})$$

□

3.4 Statistical inference

Having obtained regression coefficient estimates, the next step is to say something about their significance. This can be done via hypothesis tests and confidence intervals and we will discuss several ways of obtaining these in a quantile regression setting.

3.4.1 Wald-type tests

The first test we consider is a Wald-type test based on the asymptotic normality of the quantile regression coefficients.

In an *i.i.d.* setting we have the asymptotic normality result

$$\sqrt{n}(\hat{\boldsymbol{\beta}}(\tau) - \boldsymbol{\beta}(\tau)) \xrightarrow{d} N\Big(0, \frac{\tau(1-\tau)}{f_\varepsilon^2(0)} D_0^{-1}\Big),$$

where $D_0 = \lim_{n\to\infty} n^{-1} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\mathsf{T}$.

In non- *i.i.d.* settings the asymptotic normality result becomes

$$\sqrt{n}(\hat{\boldsymbol{\beta}}(\tau) = \boldsymbol{\beta}(\tau)) \xrightarrow{d} N(0, \tau(1-\tau) D_1(\tau)^{-1} D_0 D_1(\tau)),$$

where

$$D_1(\tau) = \lim_{n\to\infty} n^{-1} \sum_{i=1}^{n} f_i(\mathbf{x}_i^\mathsf{T} \boldsymbol{\beta}(\tau)) \mathbf{x}_i \mathbf{x}_i^\mathsf{T}.$$

Also the asymptotic covariance between quantiles is

$$\mathrm{Acov}\big(\sqrt{n}(\hat{\boldsymbol{\beta}}(\tau_i) - \boldsymbol{\beta}(\tau_i)), \sqrt{n}(\hat{\boldsymbol{\beta}}(\tau_j) - \boldsymbol{\beta}(\tau_j))\big) = (\tau_i \wedge \tau_j - \tau_i \tau_j) D_1(\tau_i)^{-1} D_0 D_1(\tau_j).$$

A Wald test for general linear hypotheses is as follows. Define the coefficient vector $\boldsymbol{\theta} = (\boldsymbol{\beta}(\tau_1)^\mathsf{T}, \ldots, \boldsymbol{\beta}(\tau_m)^\mathsf{T})^\mathsf{T}$.

The null hypothesis is $H_0 : R\boldsymbol{\theta} = \mathbf{r}$. The test statistic is

$$T_n = n(R\hat{\boldsymbol{\theta}} - \mathbf{r})^\mathsf{T}(RV^{-1}R^\mathsf{T})^{-1}(R\hat{\boldsymbol{\theta}} - \mathbf{r}),$$

where V is the $mp \times mp$ matrix with the ijth block

$$V(\tau_i, \tau_j) = (\tau_i \wedge \tau_j - \tau_i\tau_j)D_1(\tau_i)^{-1}D_0D_1(\tau_j).$$

Under H_0, $T_n \xrightarrow{d} \chi^2_q$, where q is the rank of R. One problem with implementing this test in practice is that the covariance matrix involves the unknown density functions (nuisance parameters), i.e. $f_i(\mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau))$ in non-*i.i.d.* settings, and $f_\varepsilon(0)$ in *i.i.d.* settings.

So how can we estimate the asymptotic covariance matrix?

In *i.i.d.* settings

$$\mathrm{Var}(\sqrt{n}\hat{\boldsymbol{\beta}}(\tau)) = \frac{\tau(1-\tau)}{\hat{f}_\varepsilon^2(0)}\hat{D}_0^{-1},$$

where $\hat{D}_0 = n^{-1}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^\mathsf{T}$.

To estimate $f_\varepsilon(0) = f_\varepsilon(f_\varepsilon^{-1}(\tau))$ we use the *sparsity parameter* $s(\tau) = \dfrac{1}{f(F^{-1}(\tau))}$. Note that $F(F^{-1}(t)) = t$. Differentiating both sides with respect to t, we get

$$f(F^{-1}(t))\frac{d}{dt}F^{-1}(t) = 1 \Leftrightarrow \frac{d}{dt}F^{-1}(t) = s(t).$$

That is, the sparsity parameter $s(t)$ is simply the derivative of the quantile function $F^{-1}(t)$ with respect to t. We can estimate $s(t)$ using the *difference quotient estimator* proposed by Siddiqui (1960):

$$\hat{s}_n(t) = \frac{\hat{F}_n^{-1}(t + h_n|\bar{\mathbf{x}}) - \hat{F}_n^{-1}(t - h_n|\bar{\mathbf{x}})}{2h_n},$$

where $h_n \to 0$ as $n \to \infty$, and $\hat{F}_n^{-1}(t|\bar{\mathbf{x}})$ is the estimated tth conditional quantile of Y given $\bar{\mathbf{x}} = n^{-1}\sum_{i=1}^n \mathbf{x}_i$.

In non-*i.i.d.* settings

$$\mathrm{Var}(\sqrt{n}\hat{\boldsymbol{\beta}}(\tau)) = \tau(1-\tau)\hat{D}_1(\tau)^{-1}\hat{D}_0(\tau)\hat{D}_1(\tau).$$

We can use a "sandwich formula" (Hendricks-Koenker sandwich) for estimating $D_1(\tau)$. Suppose the conditional quantiles of Y given $\mathbf{x}$ are linear at quantile levels around τ. Then we can fit quantile regressions at the $(\tau \pm h_n)$th quantiles, resulting in $\hat{\boldsymbol{\beta}}(\tau - h_n)$ and $\hat{\boldsymbol{\beta}}(\tau + h_n)$. We estimate $f_i(\xi_i(\tau))$ by

$$\tilde{f}_i(\xi_i(\tau)) = \frac{2h_n}{\mathbf{x}_i^\mathsf{T}\hat{\boldsymbol{\beta}}(\tau + h_n) - \mathbf{x}_i^\mathsf{T}\hat{\boldsymbol{\beta}}(\tau - h_n)},$$

where $\xi_i(\tau) = Q_\tau(Y|\mathbf{x}_i)$. In finite samples quantiles may cross so that the upper quantiles may be estimated to be smaller than lower quantiles. A modified estimator to account for this issue is

$$\hat{f}_i(\xi_i(\tau)) = \max\left(0, \frac{2h_n}{\mathbf{x}_i^\mathsf{T}\hat{\boldsymbol{\beta}}(\tau+h_n) - \mathbf{x}_i^\mathsf{T}\hat{\boldsymbol{\beta}}(\tau-h_n) - \varepsilon}\right),$$

where ε is a small positive constant to avoid zero denominator. Barney *et al.* (1991) proposed the following kernel estimator of $D_1(\tau)$:

$$\hat{D}_1(\tau) = n^{-1}\sum_{i=1}^{n} K\Big(\frac{\hat{\varepsilon}_i(\tau)}{h_n}\Big)\mathbf{x}_i\mathbf{x}_i^\mathsf{T},$$

where $\hat{\varepsilon}_i(\tau) = y_i - \mathbf{x}_i^\mathsf{T}\hat{\boldsymbol{\beta}}(\tau)$, and h_n is a bandwidth parameter satisfying $h_n \to 0$ and $n^{\frac{1}{2}}h_n \to \infty$ as $n \to \infty$. Under certain continuity conditions on f_i, $\hat{D}_1(\tau) \xrightarrow{P} D_1(\tau)$.

```
fit <- rq(y x, tau=0.9) summary.rq(fit, se="iid")  assuming iid errors

   Call: rq(formula = y   x, tau = 0.9)   tau: [1]  0.9
   ##
   ## Coefficients:
   ##             Value    Std. Error t value  Pr(>|t|)
   ## (Intercept)  6.76819  0.72282    9.36355  0.00000
   ## x            0.15111  0.01249   12.10025  0.00000

summary.rq(fit, se="nid") # assuming non-iid errors, Hendricks-Koenker sandwich

   ##
   ## Call: rq(formula = y ~ x, tau = 0.9)
   ##
   ## tau: [1] 0.9
   ##
   ## Coefficients:
   ##             Value   Std. Error t value Pr(>|t|)
   ## (Intercept) 6.76819 0.91973    7.35890 0.00000
   ## x           0.15111 0.01872    8.07143 0.00000

summary.rq(fit, se="ker") # based on Powell kernel estimator

   ##
   ## Call: rq(formula = y ~ x, tau = 0.9)
   ##
   ## tau: [1] 0.9
   ##
   ## Coefficients:
   ##             Value    Std. Error t value  Pr(>|t|)
   ## (Intercept)  6.76819  0.55162   12.26969  0.00000
   ## x            0.15111  0.01462   10.33560  0.00000
```

◁

3.4.2 Rank score test

Another type of test is the **rank score test**. Consider the model

$$Q_\tau(Y|\mathbf{x}_i, \mathbf{z}_i) = \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau) + \mathbf{z}_i^\mathsf{T}\boldsymbol{\gamma}(\tau)$$

and the hypotheses

$$H_0 : \boldsymbol{\gamma}(\tau) = 0 \quad \text{vs} \quad H_1 : \boldsymbol{\gamma}(\tau) \neq 0.$$

Here $\boldsymbol{\beta}(\tau) \in \mathbb{R}^p$ and $\boldsymbol{\gamma}(\tau) \in \mathbb{R}^q$. The score function is

$$S_n = \sqrt{n}\sum_{i=1}^{n} z_i^* \psi_\tau(y_i - \mathbf{x}_i^\mathsf{T}\hat{\boldsymbol{\beta}}(\tau)),$$

where $\psi_\tau(u) = \tau - I(u < 0)$, $\mathbf{z}^* = (\mathbf{z}_i^*) = \mathbf{z} - \mathbf{x}(\mathbf{x}^\mathsf{T}\boldsymbol{\Psi}\mathbf{x})^{-1}\mathbf{x}^\mathsf{T}\boldsymbol{\Psi}\mathbf{z}$, $\boldsymbol{\Psi} = \text{diag}(f_i(Q_\tau(Y|\mathbf{x}_i, \mathbf{z}_i))$ and $\hat{\boldsymbol{\beta}}(\tau)$ is the quantile coefficient estimator obtained under H_0.

Under H_0, as $n \to \infty$,

$$S_n = AN(0, M_n^{\frac{1}{2}}),$$

where $M_n = n^{-1}\sum_{i=1}^n \mathbf{z}_i^*\mathbf{z}_i^{*\mathsf{T}}\tau(1-\tau)$.

Then the rank-score test statistic $T_n = S_n^\mathsf{T} M_n^{-1} S_n \xrightarrow{d} \chi_q^2$, under H_0.

In *i.i.d.* settings $\mathbf{z}^* = (\mathbf{z}_i^*) = \{\boldsymbol{I} - \mathbf{x}(\mathbf{x}^\mathsf{T}\mathbf{x})^{-1}\mathbf{x}^\mathsf{T}\}\mathbf{z}$ are the residuals by projecting $\mathbf{z}$ on $\mathbf{x}$ and $M_n = \tau(1-\tau)n^{-1}\sum_{i=1}^n \mathbf{z}_{i*}\mathbf{z}_i^{*\mathsf{T}}$, so there is no need to estimate the nuisance parameters $f_i\{Q_\tau(Y|\mathbf{x}_i, z_i)\}$.

We can invert the rank score test to construct a confidence interval (CI) of $\gamma(\tau)$. Consider the hypotheses

$$H_0 : \gamma(\tau) = \gamma_0 \quad \text{vs} \quad H_1 : \gamma(\tau) \neq \gamma_0,$$

where γ_0 is a prespecified scalar. The test rejects H_0 if $T_n \geq \chi_\alpha^2(1)$, the $(1-\alpha)$th quantile of $\chi^2(1)$. The collection of all the γ_0 for which H_0 is not rejected is taken to be the $(1-\alpha)$th confidence interval of $\gamma(\tau)$. For more detail see Gutenbrunner *et al.* (1993).

```
fit <- rq(y x, tau=0.9) summary.rq(fit, se="rank", alpha=0.05, iid=TRUE)  assuming iid errors

   Call: rq(formula = y   x, tau = 0.9)   tau: [1]  0.9
   ##
   ## Coefficients:
   ##             coefficients lower bd upper bd
   ## (Intercept)  6.76819      6.04781 11.22792
   ## x            0.15111      0.10823  0.17721

summary.rq(fit, se="rank", alpha=0.05, iid=FALSE) # assuming non-iid errors
```

```
##
## Call: rq(formula = y ~ x, tau = 0.9)
##
## tau: [1] 0.9
##
## Coefficients:
##             coefficients lower bd upper bd
## (Intercept)  6.76819      6.04702 11.26816
## x            0.15111      0.10730  0.17725
```

◁

3.4.3 Resampling and bootstrap methods

A way to get around the issues with the unknown densities appearing in the formulae for standard errors is to use resampling methods. First consider the case of *i.i.d.* errors in a location-shift model. A bootstrap method can be implemented as follows.

1. Obtain the estimator $\hat{\boldsymbol{\beta}}(\tau)$ using the observed sample, and residuals $\hat{\varepsilon}_i = y_i - \mathbf{x}_i^\mathsf{T}\hat{\boldsymbol{\beta}}(\tau)$.
2. Draw bootstrap samples $\varepsilon_i^*, i = 1, \ldots, n$ from $\{\hat{\varepsilon}_i, i = 1, \ldots, n\}$ with replacement, and define $y_i^* = \mathbf{x}_i^\mathsf{T}\hat{\boldsymbol{\beta}}(\tau) + \varepsilon_i^*$.
3. Compute the bootstrap estimator $\hat{\boldsymbol{\beta}}^*(\tau)$ by quantile regression using the bootstrap sample.
4. Carry out inference by calculating the covariance of $\hat{\boldsymbol{\beta}}(\tau)$ by the sample covariance of bootstrap estimators or contruct a confidence interval using percentile methods.

For more detail on bootstrap methods see De Angelis *et al.* (1993) and Knight (2003).

An alternative method is **paired bootstrap**, which can be implemented as follows:

1. Generate a bootstrap sample $(y_i^*, \mathbf{x}_i^*)$ by drawing with replacement from the n pairs $\{(y_i, \mathbf{x}_i), i = 1, \ldots, n\}$.
2. Obtain the bootstrap estimator $\hat{\boldsymbol{\beta}}^*(\tau)$ by quantile regression using the bootstrap sample.

For more detail see Andrews and Buchinsky (2000, 2001).

Another option is to use Markov chain marginal bootstrap (MCMB) He and Hu (2002) and Kocherginsky *et al.* (2005). Instead of solving a p-dimensional estimating equation for each bootstrap replication, MCMB solves p one-dimensional estimating equations. In MCMB we consider the model

$$Q_\tau(Y_i|\mathbf{x}_i) = \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau), \quad \boldsymbol{\beta}(\tau) \in \mathbb{R}^p,$$

where $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,p})^\mathsf{T}$. Calculate $r_i = y_i - \mathbf{x}_i^\mathsf{T}\hat{\boldsymbol{\beta}}(\tau)$. Define $z_i = \mathbf{x}_i\psi_\tau(r_i) - \bar{z}$ with $\bar{z} = n^{-1}\sum_{i=1}^n \mathbf{x}_i\psi_\tau(r_i)$, where $\psi_\tau(r) = \tau - I(r < 0)$.

1. Step 1: let $\boldsymbol{\beta}^{(0)} = \hat{\boldsymbol{\beta}}(\tau)$.
2. Step k: for each integer $1 \leq j \leq p$ in ascending order, draw with replacement from $z_1, \ldots, z_n$ to obtain $z_1^{j,k}, \ldots, z_n^{j,k}$. Obtain $\beta_j^{(k)}$ as the solution to

$$\sum_{i=1}^{n} \mathbf{x}_i \psi_\tau \left\{ y_i - \sum_{l<j} x_{i,l} \beta_l^{(k)} - \sum_{l>j} x_{i,l} \beta l^{(k-1)} - x_{i,j} \beta_j^{(k)} \right\} = \sum_{i=1}^{n} z_i^{j,k}.$$

3. Repeat until K replications $\boldsymbol{\beta}^k, k = 1, \ldots, K$ are obtained. The variance of $\hat{\boldsymbol{\beta}}(\tau)$ is then estimated by the sample variance of $\{\boldsymbol{\beta}^{(k)}, k = 1, \ldots, K\}$.

In addition to the bootstrap methods described here, there are other options such as the bootstrap estimating equations of Parzen and Ying (1994) and the wild bootstrap method of Feng *et al.* (2011). All of the above bootstrap methods are implemented in R – see function `boot.rq` from `library(quantreg)` for details.

3.5 Nonparametric quantile regression

So far we have restricted our attention to linear quantile regression models, but we can also fit models with a smooth term for the explanatory variable using the methods described in Chapter 4. We will illustrate how these methods can be applied in a quantile regression setting using a well-known simulated dataset, the "motorcycle" data (`mcycle` from `library(MASS)`).

Example 3.7 (Quantile regression for the motorcycle data). We begin by considering a locally linear approach using the `lprq` function from `library(quantreg)`. This function computes a quantile regression fit at each of m equally spaced x-values over the support of the observed x points. The median regression fits for different values of the smoothing parameter can be seen in Figure 3.13 below.

Another approach is to use regression splines such as B-splines. We can do this directly, using the `bs()` function from `library(splines)` and choosing the order of the spline, the number and even placements of knots ourselves. The fit is shown in Figure 3.7.

An even better approach might be to use the `rqss` function from `library(quantreg)` which applies a smoothness penalty λ and also allows various constraints such as monotonicity and convexity for the fitted smooth curves. Here the user is responsible for selecting the value of λ as there is no automatic selection criterion in the current implementation. In our example we set $\lambda = 1$ (default) and we do not impose any monotonicity or convexity constraint (`constraint="N"`). The resulting fit is shown in Figure 3.7.

```
library(MASS)
par(mfrow=c(1,2))
plot(mcycle$times, mcycle$accel, xlab="milliseconds", ylab="acceleration (in g)", pch=20)
fit1 <- lprq(mcycle$times,mcycle$accel,h=0.5,tau=0.5)
lines(fit1$xx,fit1$fv,col="blue",lwd=2)
plot(mcycle$times, mcycle$accel, xlab="milliseconds", ylab="acceleration (in g)", pch=20)
fit2 <- lprq(mcycle$times,mcycle$accel,h=2,tau=0.5)
lines(fit2$xx,fit2$fv,col="blue",lwd=2)
```

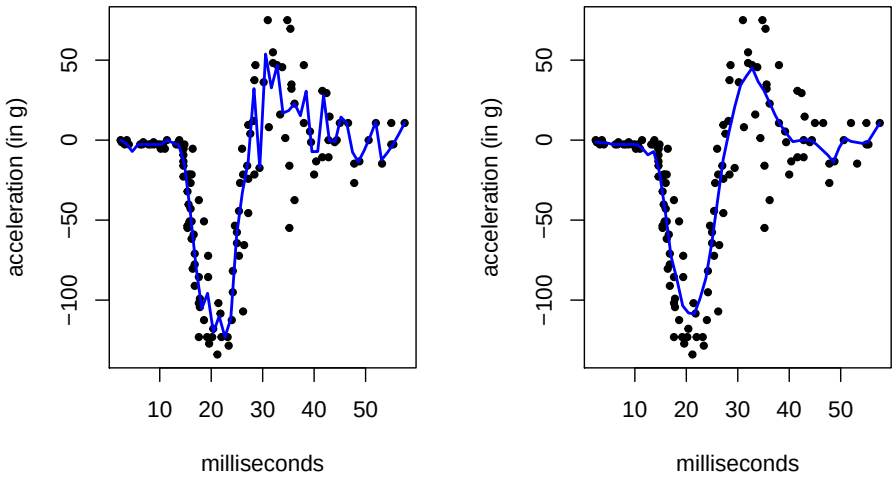


Figure 3.13. Local linear median regression fit for the motorcycle data with smoothing parameter $h = 0.5$ (left) and $h = 2$ (right).

```
fit3 <- rq(accel~bs(times,df=10),tau=0.5, data=mcycle)
fit4 <- rqss(accel~qss(times,constraint="N", lambda=1),tau=0.5, data=mcycle)
par(mfrow=c(1,2))
plot(mcycle$times, mcycle$accel, xlab="milliseconds", ylab="acceleration (in g)", pch=20)
lines(mcycle$times,fit3$fitted.values,col="blue",lwd=2)
plot(mcycle$times, mcycle$accel, xlab="milliseconds", ylab="acceleration (in g)", pch=20)
lines(mcycle$times,fitted(fit4),col="blue",lwd=2)
```

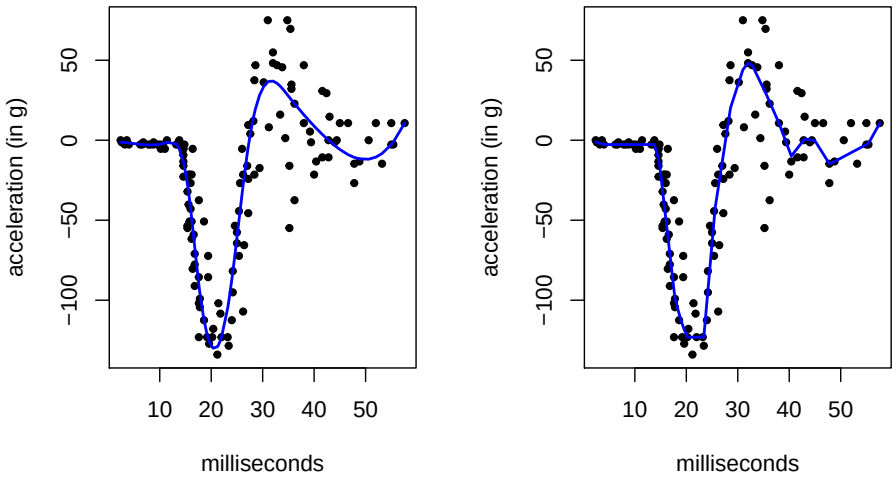


Figure 3.14. Median regression fit for the motorcycle data using cubic B-splines with 5 degrees of freedom (left panel) and using quantile smoothing splines with penalty $\lambda = 1$. (right panel)

◁

Hopefully by now it is clear that flexible regression models for the mean can be extended in a very straightforward way to the conditional quantiles of a response variable of interest. The issues of choosing the smoothing parameter and the potential of over/under-smoothing remain, and in addition we have to worry about quantile crossing. This is a risk with any quantile regression model but especially so when fitting non-parametric smooth curves to the data. In Chapter 5 we will see how we can avoid this problem by using models that make distributional assumptions.

4

Generalised Additive Models (GAMs)

In this chapter, methods of extending flexible regression to more than one covariate will be explored by introducing a more general approach known as *additive modelling*. Firstly, though we will extend the initial concepts introduced in Chapter 2 by considering:

- How much to smooth?
- How to select smoothing parameters?
- Nonparametric regression in higher dimensions

4.1 How much to smooth?

One of the key questions with nonparametric regression models is how much smoothing to apply to the data. For exploratory work, it can often be helpful simply to experiment with different degrees of smoothing. One appealing way to do that is to specify how many *degrees of freedom* you would like to have (for the model, or smooth covariate of interest).

4.1.1 Effective degrees of freedom and standard errors

For this, we introduce the notion of effective degrees of freedom, also sometimes called the effective number of parameters.

As illustrated earlier, it is helpful to express the fitted values of the nonparametric regression as

$$\hat{\mathbf{y}} = \hat{f} = \mathbf{S}\mathbf{y},$$

where $\hat{f}$ denotes the vector of fitted values, $\mathbf{S}$ denotes a *smoothing matrix* whose rows consist of the weights appropriate to estimation at each evaluation point, and $\mathbf{y}$ denotes

the observed responses in vector form. This linear structure applies with both local fitting and spline approaches and it is very helpful.

For example, it gives us a route to defining *degrees of freedom for the model* by analogy with what happens with the usual linear model, where the number of parameters is the trace of the projection matrix. An approximate version of these can be constructed for nonparametric models as

$$\mathrm{df}_{\mathrm{mod}} = \mathrm{tr}\left\{\mathbf{S}\right\}.$$

In an un-penalised regression problem, the number of parameters provides us with information about the complexity of the model. More complex models have more parameters than simpler models. For penalised regression problems counting the parameters is however not meaningful. Due to the roughness penalty not all parameters are "free". Recall that in linear regression the hat matrix $\mathbf{S} = \mathbf{B}(\mathbf{B}^\top\mathbf{B})^{-1}\mathbf{B}^\top$ (where $\mathbf{B}$ would usually be the standard design matrix $\mathbf{X}$) is a projection matrix and thus the trace $\mathrm{tr}(\mathbf{S})$ equals the number of parameters. We can generalise this to penalised models and define the *effective degrees of freedom for the model* as

$$\mathrm{edf}_{\mathrm{mod}(\lambda)} = \mathrm{tr}(\mathbf{S}_\lambda),$$

where $\mathbf{S}_\lambda = \mathbf{B}(\mathbf{B}^\top\mathbf{B} + \lambda\mathbf{D}^\top\mathbf{D})^{-1}\mathbf{B}^\top$.

Error variance and standard errors Similarly, we can construct an estimate of the error variance σ^2 through the residual sum-of-squares, which in a nonparametric setting is simply $\mathrm{RSS} = \sum\{y_i - \hat{f}(x_i)\}^2$. This leads to the estimator of the error variance $\hat{\sigma}^2 = \mathrm{RSS}/\mathrm{df}_{\mathrm{err}}$, where $\mathrm{df}_{\mathrm{err}} = n - \mathrm{tr}(\mathbf{S})$ if $\mathbf{S}^\top = \mathbf{S}$ and $\mathbf{S}^2 = \mathbf{S}$.

The linear structure of the fitted values also makes it very easy to produce standard errors which quantify the variability of the estimate at any value of x.

If $\hat{f}$ denotes the estimated values of f at a set of evaluation points then

$$\mathrm{Var}\left\{\hat{f}\right\} = \mathrm{Var}\{\mathbf{S}\mathbf{y}\} = \mathbf{S}\mathbf{S}^\mathsf{T}\sigma^2$$

and so, by plugging in $\hat{\sigma}^2$ and taking the square root of the diagonal elements, the standard errors at each evaluation point are easily constructed.

4.2 Automatic methods for smoothing

In more complicated situations only using degrees of freedom to determine the appropriate level of smoothing can be difficult and it is helpful to have an automatic way of producing a suitable level of smoothing. There are several ways to do this, some of which are carefully tailored to particular models. Here we will outline a method called *cross-validation* which, although it has some difficulties, has the advantage that the generality of its definition allows it to be applied to quite a wide variety of settings. In the present setting, the idea is to choose the smoothing parameter to minimise

$$\text{CV}: \quad \sum_{i=1}^{n} \{y_i - \hat{f}_{-i}(x_i)\}^2.$$

The subscript $-i$ denotes that the estimate of the smooth curve at x_i is constructed from the remainder of the data, excluding x_i. The aim then is to evaluate the level of smoothing through the extent to which each observation is predicted from the smooth curve produced by the rest of the data. The level of smoothing which minimises the expression above should provide a suitable level of smoothing. The linearity of smoothing operations allows the computations to be performed in a very efficient manner.

It is often convenient to use an approximation known as *generalised cross-validation* (GCV) which has the efficient computational form

$$\text{GCV}: \quad n\text{RSS}/\{\text{tr}\,\{I - \mathbf{S}\}^2\}.$$

Altering the smoothing parameters changes the entries of $\mathbf{S}$, which in turns affects the value of GCV.

The degree of smoothing can also be selected automatically by minimising a quantity based on *Akaike's information criterion*, namely

$$\text{AIC}: \quad \frac{\text{RSS}}{n} + 1 + \frac{2(\nu+1)}{(n-\nu-2)},$$

where ν denotes the degrees of freedom.

The following two approaches provide interpretations of a penalised regression spline model, which enable automatic selection of the level of smoothing.

4.2.1 Random effects interpretation

Random effect models – Likelihood

In the random effects model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\alpha} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}$$

with error term $\boldsymbol{\varepsilon} \sim \mathsf{N}(\mathbf{0}, \sigma^2\mathbf{I})$ and random effect $\boldsymbol{\gamma} \sim \mathsf{N}(\mathbf{0}, \tau^2\mathbf{I})$, twice the loglikelihood is (ignoring the variance parameters) given by

$$-\frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^\top\boldsymbol{\alpha} - \mathbf{z}_i^\top\boldsymbol{\gamma})^2 - \frac{1}{\tau^2}\sum_{j=1}^{q}\gamma_j^2$$

**note, the change in notation here from the preliminary material from $\mathbf{X}\boldsymbol{\beta}$ to $\mathbf{X}\boldsymbol{\alpha}$ so that we can reserve $\boldsymbol{\beta}$ to be a vector of basis coefficients later.

Comparing the penalised least squares criterion (2.2) to the loglikelihood suggests that we can interpret the penalised regression model as a random effects model with no fixed effect and random effect $\boldsymbol{\beta}$. However the problem is that, at least for difference matrices, $\mathbf{D}^\top\mathbf{D}$ is not of full rank, thus we cannot take its inverse matrix square root. In order to obtain a proper random-effects representation we need to "split" $\boldsymbol{\beta}$ into an (unpenalised) fixed effect and a (penalised) random effect.

In the following we will only consider the case of a difference penalty of order 1 or 2. In the case of a first-order difference penalty we define $\mathbf{G} = (1, \ldots, 1)$. For a second-order difference penalty we define $\mathbf{G} = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ 1 & 2 & \ldots & l+r-1 \end{pmatrix}$. The rows in $\mathbf{G}$ are parameter sequences which do not incur a penalty, i.e. $\mathbf{GD} = \mathbf{0}$. We also define $\mathbf{H} = \mathbf{D}^\top(\mathbf{D}\mathbf{D}^\top)^{-1}$. We can now write

$$\boldsymbol{\beta} = \mathbf{G}\boldsymbol{\alpha} + \mathbf{H}\boldsymbol{\gamma}$$

Because $\mathbf{D}\mathbf{D}^\top$ is of full rank we have that $\mathbf{D}\mathbf{H} = \mathbf{D}\mathbf{D}^\top(\mathbf{D}\mathbf{D}^\top)^{-1} = \mathbf{I}$. Plugging this into the objective function (2.2) gives

$$\|\mathbf{y} - \mathbf{B}\mathbf{G}\boldsymbol{\alpha} - \mathbf{B}\mathbf{H}\boldsymbol{\gamma}\|^2 + \lambda\Big(\underbrace{\boldsymbol{\alpha}\mathbf{G}^\top\mathbf{D}^\top\mathbf{D}\mathbf{G}\boldsymbol{\alpha}}_{=0} + 2\underbrace{\boldsymbol{\alpha}\mathbf{G}^\top\mathbf{D}^\top\mathbf{D}\mathbf{H}\boldsymbol{\gamma}}_{=0} + \boldsymbol{\gamma}^\top\underbrace{\mathbf{H}^\top\mathbf{D}^\top\mathbf{D}\mathbf{H}}_{=\mathbf{I}}\boldsymbol{\gamma}\Big)$$

$$= \|\mathbf{y} - \mathbf{B}\mathbf{G}\boldsymbol{\alpha} - \mathbf{B}\mathbf{H}\boldsymbol{\gamma}\|^2 + \lambda\|\boldsymbol{\gamma}\|^2$$

Defining $\mathbf{X} = \mathbf{BG}$ and $\mathbf{Z} = \mathbf{BH}$ and denoting rows of $\mathbf{X}$ and $\mathbf{Z}$ by $\mathbf{x}_i$ and $\mathbf{z}_i$ respectively, this is equivalent to

$$\sum_{i=1}^{n}(y_i - \mathbf{x}_i^\top \boldsymbol{\alpha} - \mathbf{z}_i^\top \boldsymbol{\gamma})^2 + \lambda \sum_{j=1}^{q} \gamma_j^2,$$

which is $-\sigma^2$ times the loglikelihood of a random-effects model, which we have stated above. Hereby we have used $\lambda = \sigma^2/\tau^2$.

Thus the penalised regression model is nothing other than a random effects effect and we can use standard mixed model software to fit these models. Most importantly we can estimate the variances σ^2 and τ^2 in a mixed model (using (restricted) maximum likelihood - see the preliminary material), which gives us a way of estimating the otherwise rather elusive smoothing parameter $\hat{\lambda} = \hat{\sigma}^2/\hat{\tau}^2$.

This approach is used in the `mgcv` package in R that we have used to illustrate examples in Chapter 2 of fitting penalised regression spline models. For the radiocarbon data considered before in Chapter 2, we have:

```
model <- gam(Rc.age~s(Cal.age), method="REML", data=radiocarbon)
model

   ##
   ## Family: gaussian
   ## Link function: identity
   ##
   ## Formula:
   ## Rc.age ~ s(Cal.age)
   ##
   ## Estimated degrees of freedom:
   ## 7.44  total = 8.44
   ##
   ## REML score: 258.6108

plot(model, residuals=TRUE)
```

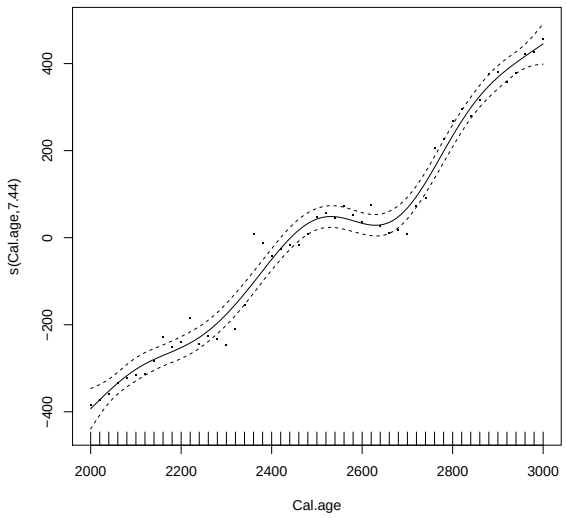


Figure 4.1. Automatic smoothness detection using `mgcv`.

4.2.2 Bayesian interpretation

Rather than interpreting the penalised fitting criterion as a random effects model we can treat the penalised regression model as a fully Bayesian model with the following prior and data model.

$$\mathbf{D}\boldsymbol{\beta}|\tau^2 \sim \mathsf{N}(\mathbf{0}, \tau^2\mathbf{I})$$
$$\mathbf{y}|\boldsymbol{\beta}, \sigma^2 \sim \mathsf{N}(\mathbf{B}\boldsymbol{\beta}, \sigma^2\mathbf{I})$$

The prior distribution of $\boldsymbol{\beta}$ is improper if $\mathbf{D}$ is not of full rank, which is the case for all difference penalties. However in the case of difference penalties the prior distribution of $\boldsymbol{\beta}$ can be expressed in terms of random walks (cf. Figure 2.25).

First-order random walk The first-order penalty corresponds to an improper flat prior on β_1 and $\beta_j|\beta_{j-1} \sim \mathsf{N}(\beta_{j-1}|\tau^2)$ (for $j \geq 2$).

Second-order random walk The second-order penalty corresponds to an improper flat prior on β_1 and β_2 and $\beta_j|\beta_{j-1}, \beta_{j-2} \sim \mathsf{N}(2\beta_{j-1} - \beta_{j-2}|\tau^2)$ (for $j \geq 3$).

It seems natural to complement the model with priors for σ^2 and τ^2

$$\sigma^2 \sim \mathsf{IG}(a_{\sigma^2}, b_{\sigma^2})$$

$$\tau^2 \sim \mathsf{IG}(a_{\tau^2}, b_{\tau^2})$$

Inference can then be carried out efficiently using a Gibbs sampler. This model and many other Bayesian smoothing models are implemented in the software `BayesX`.

Rather than placing an independent inverse-gamma prior on τ^2 we can set $\tau^2 = \sigma^2/\lambda$ and place a prior of our choice on λ. In this model the posterior distribution of λ does not follow a known distribution, but can be evaluated efficiently, as all the other parameters can be integrated out in closed form. The drawback is that the integration over λ would need to be carried out numerically, which suggests that this approach is better suited for an empirical Bayes strategy for estimating λ.

We can also use `BayesX`(see `www.bayesx.org`) to estimate a penalised spline model in the Bayesian framework (using the package `R2BayesX`).

```
library(R2BayesX)
model <- bayesx(Rc.age ~ sx(Cal.age), data = radiocarbon)
model

   ## Call:
   ## bayesx(formula = Rc.age ~ sx(Cal.age), data = radiocarbon)
   ## Summary:
   ## N = 51  burnin = 2000  method = MCMC  family = gaussian
   ## step = 10

plot(model)
```

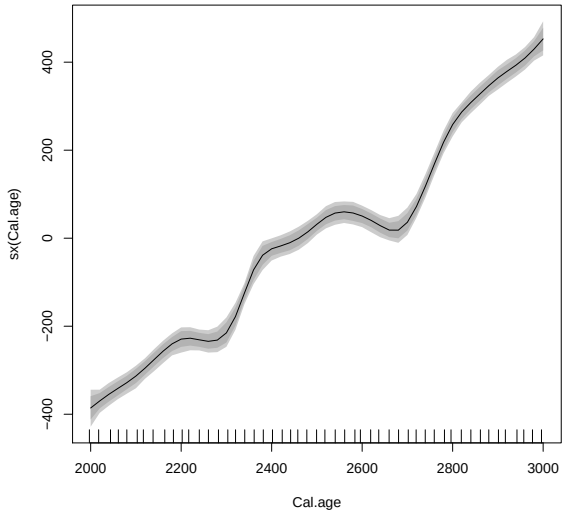


Figure 4.2. Penalised spline fit using a Bayesian framework.

4.3 Nonparametric regression in higher dimensions

4.3.1 Local fitting with bivariate smoothing

It is rare to have problems which involve only a single covariate. For the Reef data, Chapter 2, a natural extension is to look at the relationship between the catch score and both latitude (x_1) and longitude (x_2), in a model

$$Y_i = f(x_{1i}, x_{2i}) + \varepsilon_i.$$

The local linear approach is particularly easy to extend to this setting. If the observed data are denoted by $\{x_{1i}, x_{2i}, y_i; i = 1, \ldots, n\}$, then for estimation at the point (x_1, x_2) the weighted least squares formulation is

$$\min_{\alpha,\beta,\gamma} \sum_{i=1}^{n} \{y_i - \alpha - \beta(x_{1i} - x_1) - \gamma(x_{2i} - x_2)\}^2 \, w(x_{1i} - x_1; h_1) \, w(x_{2i} - x_2; h_2).$$

The value of the fitted surface at (x_1, x_2) is simply $\hat{\alpha}$. With careful thought, the computation can be performed efficiently.

This will be illustrated using the Reef data, a reminder is below:

Example 4.1 (Great Barrier Reef data). A survey of the fauna on the sea bed lying between the coast of northern Queensland and the Great Barrier Reef was carried out. The sampling region covered a zone which was closed to commercial fishing, as well as neighbouring zones where fishing was permitted. The variables are:

Zone	an indicator for the closed (1) and open (0) zones
Year	an indicator of 1992 (0) or 1993 (1)
Latitude	latitude of the sampling position
Longitude	longitude of the sampling position
Depth	bottom depth
Score1	catch score 1
Score2	catch score 2

◁

Investigating one year of the Reef data, the effect of longitude dominates, as we see from the earlier nonparametric regression in Chapter 2. However, a small effect of latitude is also suggested.

```
trawl1 <- subset(trawl, Year == 0)
sm.regression(trawl1[ , c("Longitude", "Latitude")], trawl1$Score1, theta = 120)
```

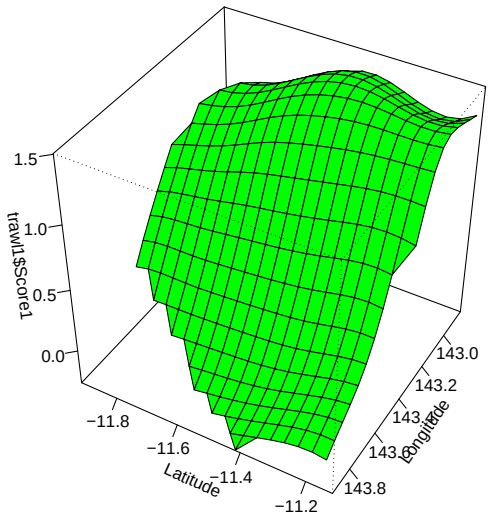


Figure 4.3. Reef data with two covariates for one year.

Notice that two smoothing parameters, h_1 and h_2, are now required - one for each covariate.

4.3.2 Bivariate splines

Tensor-product splines So far we have only covered the construction of spline bases in one dimension. In this section we will see how we can turn a one-dimensional spline basis into a spline basis of any dimension. To keep things simple we shall start with the bivariate case.

Suppose we have two covariates and want to fit a regression model of the form

$$\mathbb{E}(Y_i) = f(x_{i1}, x_{i2}),$$

where $f(\cdot, \cdot)$ is a bivariate surface.

We start by placing a basis on each dimension separately. Denote by $B_1^{(1)}(x_1), \ldots, B_{l_1+r-1}^{(1)}(x)$ the basis functions placed on the first covariate, and by $B_1^{(2)}(x_1), \ldots, B_{l_2+r-1}^{(2)}(x)$ the basis functions placed on the second covariate. We now define a set of basis functions

$$B_{jk}(x_1, x_2) = B_j^{(1)}(x_1) \cdot B_k^{(2)}(x_2)$$

for $j \in 1, \ldots, l_1 + r - 1$ and $k \in 1, \ldots, l_2 + r - 1$. Figure 4.4 shows how one such bivariate basis function looks like for different degrees of the underlying univariate B-spline. Figure 4.5 shows all 36 bivariate basis functions resulting from two B-spline bases with six basis functions each.

We will now use the basis expansion

$$f(x_{i1}, x_{i2}) = \sum_{j=1}^{l_1+r-1} \beta_{jk} B_{jk}(x_1, x_2)$$

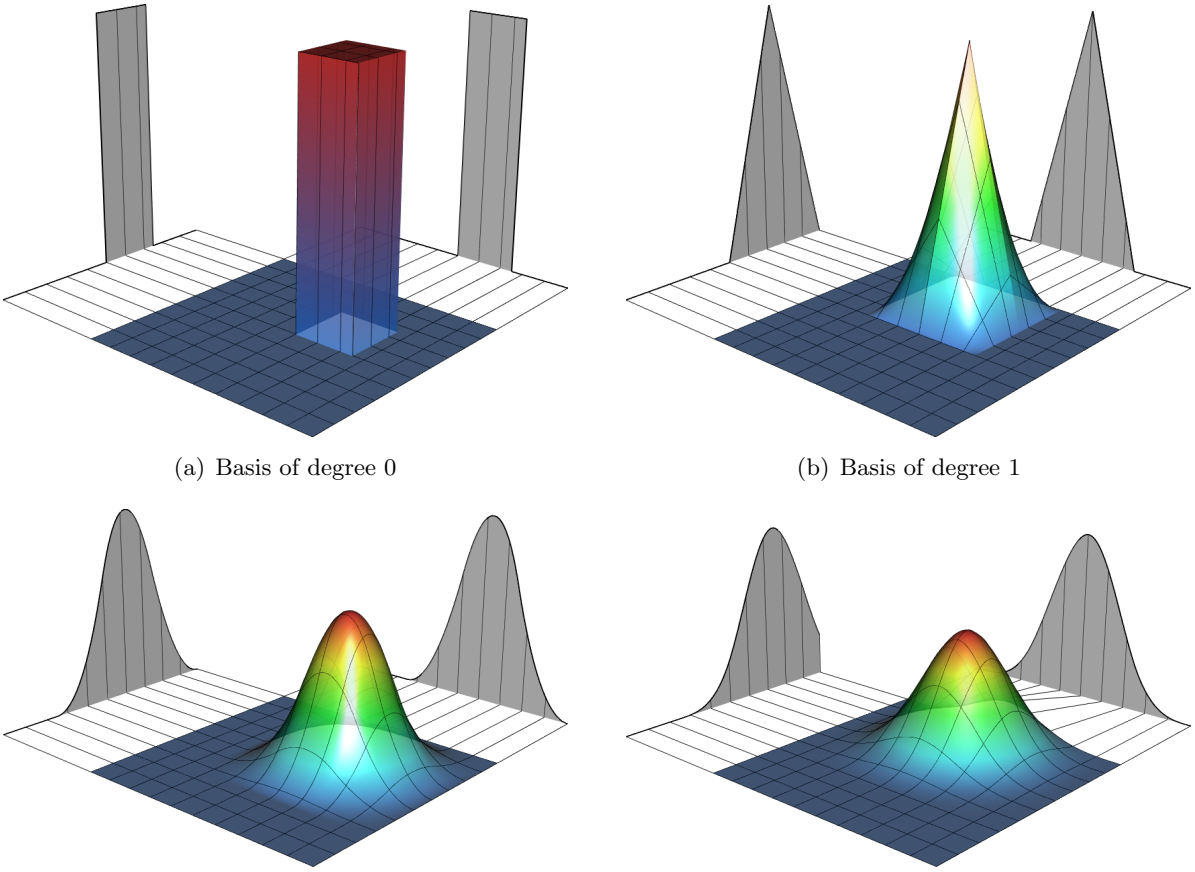
which corresponds to the design matrix

$$\mathbf{B} = \begin{pmatrix} B_{11}(x_{11}, x_{12}) & \ldots & B_{l_1+r-1,1}(x_{11}, x_{12}) & B_{12}(x_{11}, x_{12}) & \ldots & B_{l_1+r-1,2}(x_{11}, x_{12}) & \ldots & B_{l_1+r-1,l+2+r-1}(x_{11}, x_{12}) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ B_{11}(x_{n1}, x_{n2}) & \ldots & B_{l_1+r-1,1}(x_{n1}, x_{n1}) & B_{12}(x_{n1}, x_{n2}) & \ldots & B_{l_1+r-1,2}(x_{n1}, x_{n2}) & \ldots & B_{l_1+r-1,l+2+r-1}(x_{n1}, x_{n2}) \end{pmatrix}$$

and coefficient vector $\boldsymbol{\beta} = (\beta_{11}, \ldots, \beta_{l_1+r-1,1}, \beta_{12}, \ldots, \beta_{l_2+r-1,2}, \ldots, \beta_{l_1+r-1,l_2+r-1})^\top$.

We can generalise this principle of constructing a basis to dimension p by multiplying all combinations of basis functions of the p covariates.

Finally, we need to explain how a penalty matrix can be constructed for this bivariate spline basis. We will explain the basic idea using Figure 4.5. A simple way of constructing a roughness penalty consists of applying the univariate roughness penalties to the rows



(a) Basis of degree 0

(b) Basis of degree 1

(c) Basis of degree 2

(d) Basis of degree 3

Figure 4.4. Illustration of the construction of a single bivariate B-spline basis function $B_{jk}(x_1, x_2) = B_j(x_1) \cdot B_k(x_2)$ for B-spline bases of different degree.

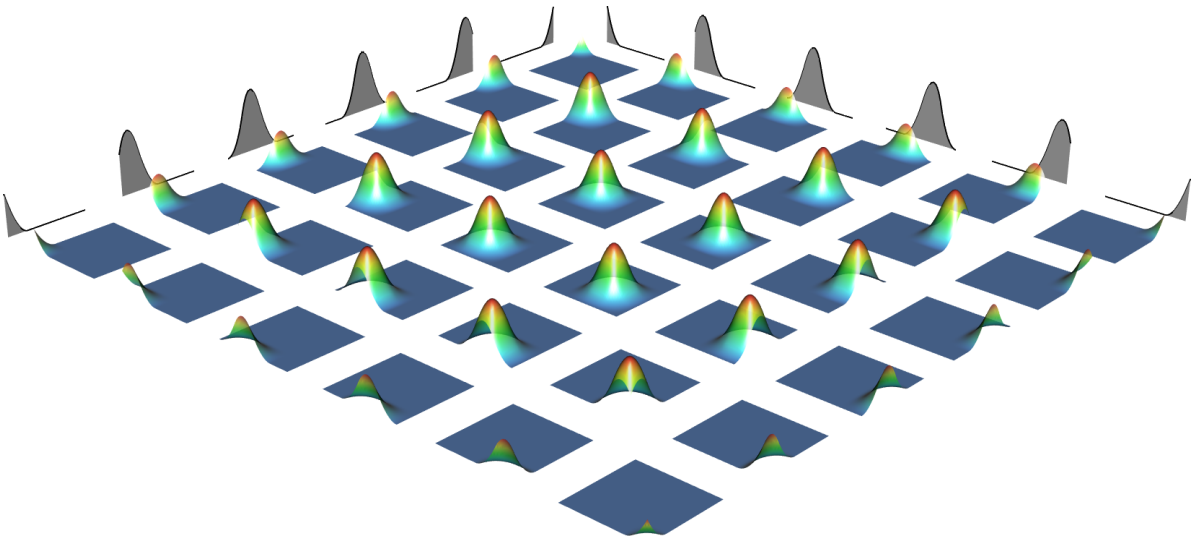


Figure 4.5. Illustration of the construction of a bivariate B-spline basis created from a univariate B-spline basis.

and columns of the basis functions. More mathematically, this corresponds to taking Kronecker products, i.e. using the difference matrix

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}^{(2)} \otimes \mathbf{I}_{l_1+r-1} \\ \mathbf{I}_{l_2+r-1} \otimes \mathbf{D}^{(1)} \end{pmatrix},$$

where $\mathbf{D}^{(1)}$ is the univariate difference matrix used for the first dimension and $\mathbf{D}^{(2)}$ is the univariate difference matrix used for the second dimension.

Example 4.2 (Great Barrier Reef (continued)). Figure 4.6 shows the result of fitting a tensor-product-spline model to the data from example 4.1. The objective is to model a score which represents the composition of the catch as a function of longitude and latitude. ◁

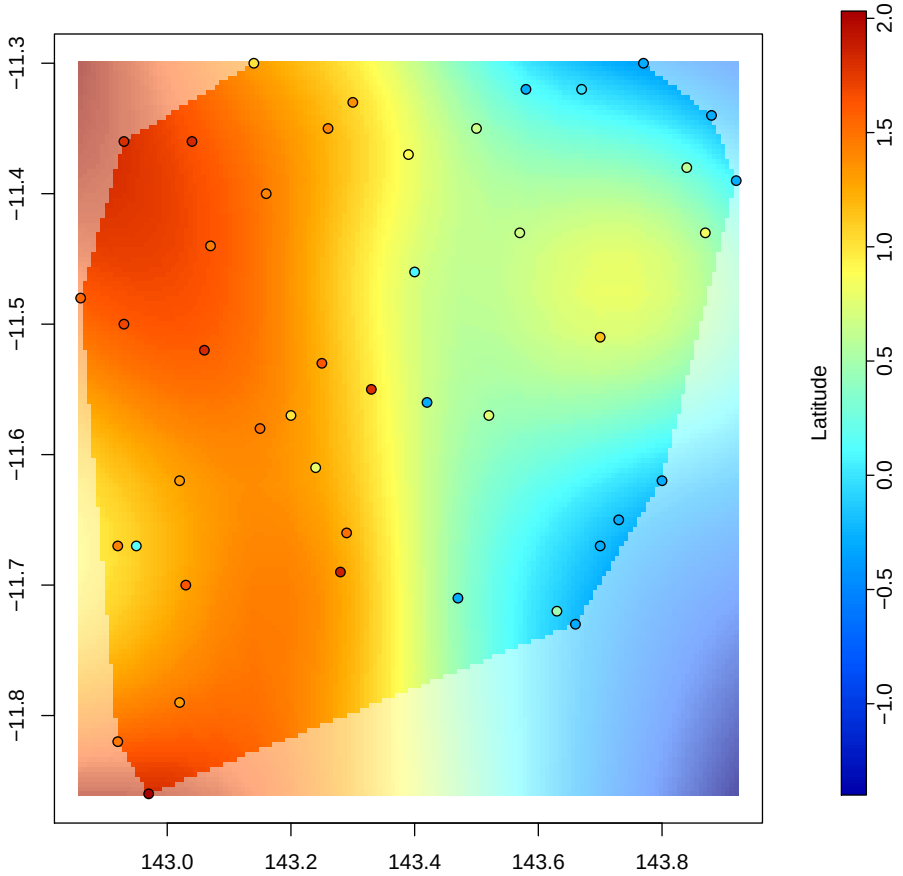


Figure 4.6. Predicted score obtained from a tensor-product-spline model fitted to the Great Barrier Reef data.

In principle, tensor-product spline bases can be constructed for any dimension, however the number of basis functions scales exponentially in the dimension, so tensor-

product splines do not scale well as the dimension is increased. The number of basis functions increases exponentially in the dimension. Thus they cannot be used for dimensions beyond three (and in some cases even two).

Thin-plate splines In this section we generalise natural cubic splines to the bivariate case, which provides an alternative way of bivariate spline smoothing. In section 2.3.3 we have seen that the minimiser of

$$\underbrace{\sum_{i=1}^{n}(y_i - f(x_i))^2}_{\text{Fit to the data}} + \lambda \underbrace{\int_a^b f''(x)^2 \, dx}_{\text{Roughness penalty}},$$

has to be a natural cubic spline.

Generalising this variational problem to the bivariate case leads to the objective function

$$\underbrace{\sum_{i=1}^{n}(y_i - f(x_{i1}, x_{i2}))^2}_{\text{Fit to the data}} + \lambda \underbrace{\int\int \left(\frac{\partial^2}{\partial x_1^2} f(x_1, x_2) + 2\frac{\partial^2}{\partial x_1 \partial x_2} f(x_1, x_2) + \frac{\partial^2}{\partial x_2^2} f(x_1, x_2) \right)^2 dx_2 \, dx_1}_{\text{Roughness penalty}},$$

The roughness penalty can be interpreted as the bending energy of a thin plate of metal. One can show that the solution to this problem has to be a so-called *thin-plate* spline of the form

$$f(\xi_1, \xi_2) = \beta_0 + \beta_1 \xi_1 + \beta_2 \xi_2 + \sum_{i=1}^{n} \beta_{2+i} K\left((\xi_1, \xi_2), (x_{i1}, x_{i2})\right),$$

where $K\left((\xi_1, \xi_2), (\zeta_1, \zeta_2)\right) = \frac{1}{2}\left((\zeta_1 - \xi_1)^2 + (\zeta_2 - \xi_2)^2\right) \cdot \log\left((\zeta_1 - \xi_1)^2 + (\zeta_2 - \xi_2)^2\right)$.

Similar to what we have discussed in section 2.3.5 we can estimate the coefficients β_j using a penalised least squares criterion. In fact, we need to minimise the objective function

$$\sum_{i=1}^{n} \left(y_i - f(x_{i1}, x_{i2})\right)^2 + \lambda \boldsymbol{\beta}' \boldsymbol{\beta}$$

subject to the constraints that $\sum_{i=1}^{n} \beta_{2+i} = \sum_{i=1}^{n} x_{i1}\beta_{2+i} = \sum_{i=1}^{n} x_{i2}\beta_{2+i} = 0$, where

$$= \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & K\left((x_{11}, x_{12}), (x_{11}, x_{12})\right) & \dots & K\left((x_{11}, x_{12}), (x_{n1}, x_{n2})\right) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & K\left((x_{n1}, x_{n2}), (x_{11}, x_{12})\right) & \dots & K\left((x_{n1}, x_{n2}), (x_{n1}, x_{n2})\right) \end{pmatrix}$$

Thin-plate splines scale much better in dimensionality, however they do not scale as well as tensor-product splines in the number of data points. Thin-plate splines are the default in `mgcv`'s function `gam`.

```
model <- gam(Score1~s(Latitude, Longitude), data=trawl)
vis.gam(model, plot.type="contour")
```

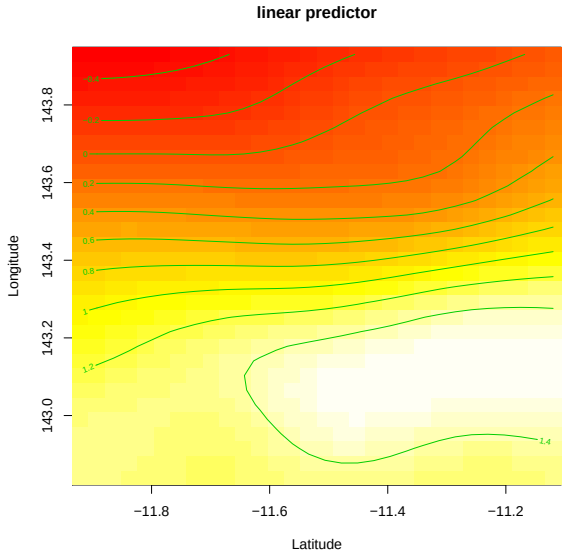


Figure 4.7. Bivariate splines using thin plate splines in `mgcv`

4.4 A simple additive model

We will now use all of the concepts above and from Chapter 2 to extend nonparametric regression.

Now that we have tools available to estimate smooth curves and surfaces, linear regression models can be extended to *additive models* as

$$Y_i = \beta_0 + f_1(x_{1i}) + \ldots + f_p(x_{pi}) + \varepsilon_i, \qquad i = 1, \ldots, n,$$

where the f_i are functions whose shapes are unrestricted, apart from an assumption of smoothness. This gives a very flexible set of modelling tools. To see how these models can be fitted, consider the case of only two covariates,

$$Y_i = \beta_0 + f_1(x_{1i}) + f_2(x_{2i}) + \varepsilon_i, \qquad i = 1, \ldots, n.$$

A rearrangement of this as $y_i - \beta_0 - f_2(x_{2i}) = f_1(x_{1i}) + \varepsilon_i$ suggests that an estimate of component f_1 can then be obtained by smoothing the residuals of the data after fitting $\hat{f}_2$,

$$\hat{f}_1 = S_1(\mathbf{y} - \bar{\mathbf{y}} - \hat{f}_2)$$

and that, similarly, subsequent estimates of f_2 can be obtained as

$$\hat{f}_2 = S_2(\mathbf{y} - \bar{\mathbf{y}} - \hat{f}_1).$$

Repetition of these steps gives a simple form of the *backfitting* algorithm. The same idea applies when we have more than two components on the model. At each step we smooth over a particular variable using as response the y variable with the current estimates of the other components subtracted.

If a spline basis is used, then the backfitting algorithm is not required as we have a form of linear model with a penalty term. This can be written as

$$Y_i = \mathbf{B}\boldsymbol{\beta} + \varepsilon_i$$

where, as usual, the columns of the matrix $\mathbf{B}$ evaluate the basis functions at each observation. This time $\mathbf{B}$ is constructed by stacking together the columns of a basis matrix for each covariate. The model is fitted by choosing the vector of weights $\boldsymbol{\beta}$ to minimise

$$(\mathbf{y} - \mathbf{B}\boldsymbol{\beta})^\mathsf{T}(\mathbf{y} - \mathbf{B}\boldsymbol{\beta}) + \boldsymbol{\beta}^\mathsf{T} P \boldsymbol{\beta}, \tag{4.1}$$

where the penalty matrix P is of block-diagonal form, constructed from the penalties from the individual model components, with the jth component $\lambda_j \mathbf{D}_j^\mathsf{T}\mathbf{D}_j$, where $\mathbf{D}_j$ is a differencing matrix. This leads to the direct solution

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{B}^\mathsf{T}\mathbf{B} + P\right)^{-1}\mathbf{B}^\mathsf{T}\mathbf{y}.$$

The terms of an additive model are unidentifiable without imposing some constraint, as a constant can be added and subtracted from the individual components without changing the resulting value. A simple solution is to require that $\sum_i f_j(x_{ij}) = 0$ for each component j.

A simple example of an additive model for the Reef data is shown in Figure 4.8.

4.5 More general additive models

A more general formulation of an additive model is:

$$Y_i = \beta_0 + f_1(x_{1i}) + \ldots + f_p(x_{pi}) + \varepsilon_i.$$

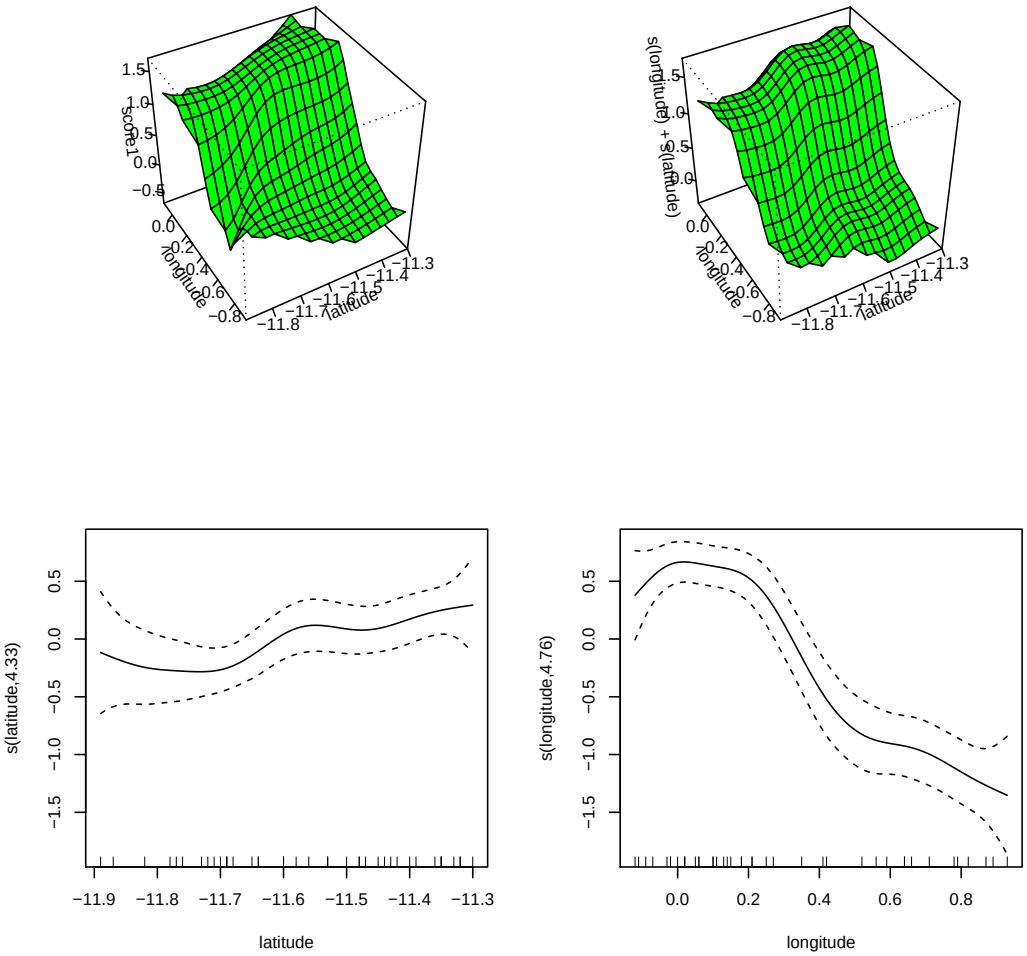


Figure 4.8. The top left hand plot shows a two-dimensional smooth estimate of the combined effects of latitude and longitude on the catch score for the Reef data after fitting $Y_i = f(\text{lat}_i, \text{long}_i) + \varepsilon_i$. The lower plots show the estimated components from a GAM after fitting $Y_i = \beta_0 + f_1(\text{lat}_i) + f_2(\text{long}_i) + \varepsilon_i$. The top right hand plot shows the surface produced by the combination of the two GAM components.

Further generality can be achieved by the use of a link function to create a *generalised additive model* or GAM for short. At the moment we will consider additive models, with link functions considered later, but it is convenient to use the terminology GAM in this case too. A simple extension of the steps outlined for two covariates gives a form of the *backfitting* algorithm. In order to ensure identifiability, we assume that $\sum_i f_j(x_{ji}) = 0$, for each j. At each step we smooth over a particular variable using as response the y variable with the current estimates of the other components subtracted.

The backfitting algorithm can be expressed as:

$$\hat{f}_j^{(l)} = \mathbf{S}_j \left(\mathbf{y} - \hat{\beta}_0 \mathbf{1} - \sum_{k<j} \hat{f}_k^{(l)} - \sum_{k>j} \hat{f}_k^{(l-1)} \right).$$

We can also express these in terms of projection matrices.

$$\mathbf{P}_j^{(l)} = (I_n - \mathbf{P}_0)\mathbf{S}_j(I_n - \sum_{k<j} \mathbf{P}_k^{(l)} - \sum_{k>j} \mathbf{P}_k^{(l-1)}),$$

$$\hat{\mathbf{y}} = \mathbf{P}\mathbf{y} = (\mathbf{P}_0 + \sum_{j=1}^{p} \mathbf{P}_j)\mathbf{y}$$

If a regression splines or p-splines model is adopted, then each of the functions $f_i(x)$ is represented by a linear expression and so the model itself remains linear. It can then be fitted by standard linear regression, incorporating a set of penalties in the p-splines case. This has the advantage of direct, rather than iterative, fitting but it has the potential disadvantage of needing to invert very large matrices if the model has many terms.

Example 4.3. The plots in Figure 4.9 show data from a survey of dissolved oxygen (DO) in the River Clyde at a single sampling station, related to potential explanatory variables of interest (top). In the bottom row is the output from fitting the following `GAM`:

$$\mathrm{DO}_i = \beta_0 + f_1(\mathrm{Year}_i) + f_2(\mathrm{Temp}_i) + f_3(\mathrm{logSalinity}) + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2)$$

The additive terms usefully capture the underlying trends.

The points on the plots in the bottom row of the figure are partial residuals (see the preliminary material). These help us to assess whether or not the fitted function appears to be appropriate. ◁

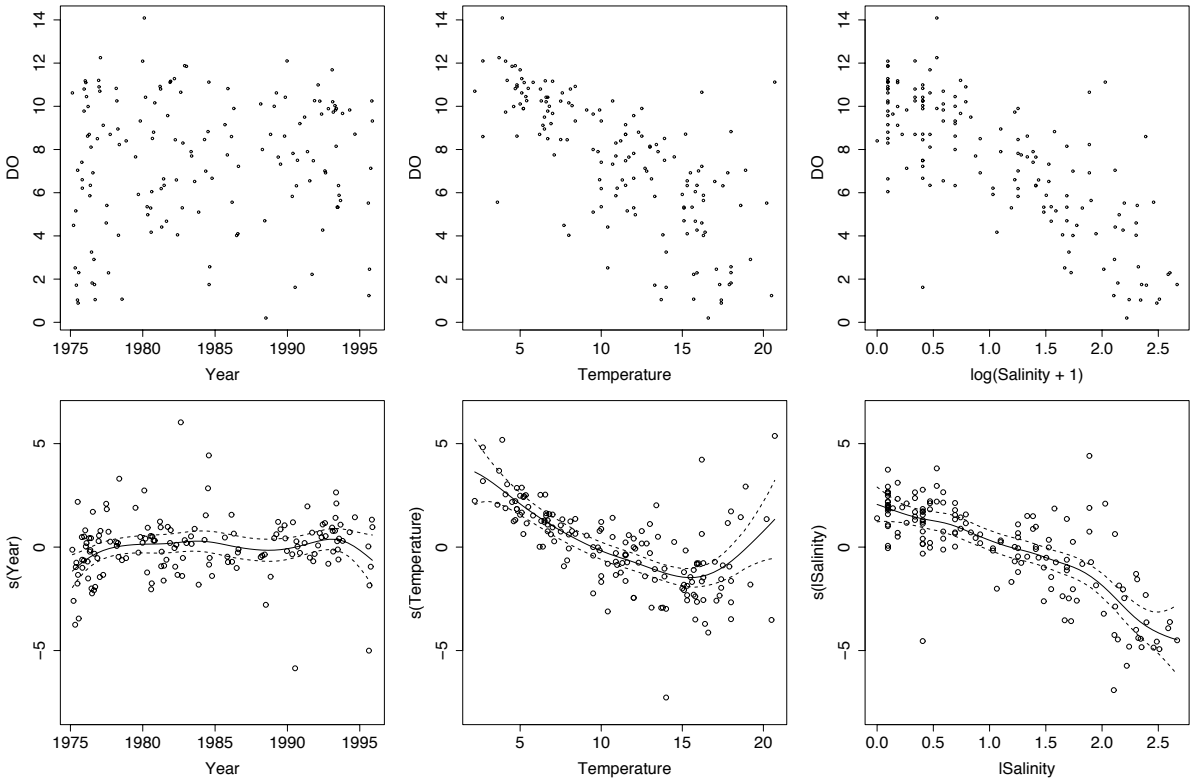


Figure 4.9. The top row of plots show Dissolved Oxygen against three covariates. The bottom row of plots show the fitted components, and partial residuals, of a GAM model.

As ever, a method of determining the level of smoothing in an additive model is required. Cross-validation provides a convenient option. The `mgcv` package for `R` has efficient algorithms for identifying the optimal smoothing parameters. In particular, the options of selecting smoothing parameters using maximum likelihood (ML) and restricted maximum likelihood (REML) are provided, which make use of the random effects formulation presented previously to estimate smoothing parameters. In recent literature, Wood (2011), these latter methods have been seen to outperform other methods such as GCV and AIC.

4.6 Fitting (G)AMs

As illustrated previously, one way to fit (Generalised) Additive Models is to use the `mgcv` library in R. A few notes on this are:

- the function `bam` fits a generalised additive model to a very large dataset;
- diagnostic plots can be obtained by using the function `plot(model)` for the fitted model. Note that these are in terms of deviance residuals (see the preliminary material for more details here);
- the Table 4.1 contains some useful notes on the type of smoothers that can be used;
- a generalised additive model is of the form:

$$g(\mu_i) = \beta_0 + \sum_{j=1}^{p} f_p(x_{pi})$$

where the mean $\boldsymbol{\mu} = \mathbb{E}(\mathbf{y}|\mathbf{x_1}, ...\mathbf{x_p})$ and $g()$ is a link function such that $\eta_i = g(\mu_i)$. This can be fitted using a local scoring procedure or penalised iteratively re-weighted least squares (see for example Hastie and Tibshirani (1990) and Wood (2017)). The fitting procedure is similar to that of a generalised linear model with a weighted linear model replaced by a weighted additive model in the fitting algorithm.

– for response distributions other than the normal, some examples of family arguments for GAMs are provided in Table 4.2;
– The default number of basis function used by `mgcv` is 9 (accounting for the identifiability constraint). This can be increased inside the call to the smooth function by specifying a value for `k` i.e. `s(x, k=15)`;
– The basis dimension can be assessed by using the function `gam.check`.

spline	**description**
cubic regression spline	Efficient and straight forward to interpret
cubic regression spline with shrinkage	The smoothness selection can set a covariate that is not important completely to zero
cyclic cubic regression spline	Constrains the start point to be the same as the end point
P-splines	Eilers & Marx p-splines with a difference penalty
thin plate regression splines	The default for `mgcv`
this plate regression splines with shrinkage	The smoothness selection can set a covariate that is not important completely to zero

Table 4.1. Alternative spline functions available in `library(mgcv)`

See Wood (2006) and Wood (2017) for a much fuller description of the `mgcv` package.

4.7 Comparing additive models

While models of this type provide very flexible and visually informative descriptions of the data, it is also necessary to consider how models can be compared and inferences drawn. Hastie and Tibshirani (1990) recommend the use of residual sums-of-squares

Distribution	Description
Gamma	strictly positive real valued data
Poisson	count data
Binomial	binary data, or number of successes from a trial
Inverse gaussian	strictly positive real valued response data
Quasi distributions	e.g. quasibinomial and quasipoisson, allows inference when the full distribution does not hold but the mean variance relationship is well approximated
Negative binomial	overdispersed count data
Tweedie	when the power parameter relating the variance to the mean is to be estimated
ocat	ordered categorical data
betar	for proportions data (0,1) when binomial is not appropriate
ziP	for zero inflated poisson data

Table 4.2. Family arguments in `gam`

and their associated approximate degrees of freedom to provide guidance for model comparisons.

For an additive model, the residual sum-of-squares can easily be defined as

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2,$$

where $\hat{y}_i$ denotes the fitted value, produced by evaluating the additive model at the observation x_i. We can write the residual sum-of-squares as

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \mathbf{y}^\top (I - \mathbf{P})^\top (I - \mathbf{P}) \mathbf{y},$$

where $\mathbf{P}$ denotes the projection matrix discussed earlier. The approximate *degrees of freedom for error* can be defined as

$$\text{df} = \text{tr}\left\{ (I - \mathbf{P})^\top (I - \mathbf{P}) \right\}.$$

In an obvious notation, comparisons of two models can be expressed quantitatively in

$$F = \frac{(\mathrm{RSS}_2 - \mathrm{RSS}_1)/(\mathrm{df}_2 - \mathrm{df}_1)}{\mathrm{RSS}_1/\mathrm{df}_1},$$

by analogy with the F-statistic used to compare linear models. Unfortunately, this analogy does not extend to distributional calculations and no general expression for the distribution of this test statistic is available. However, Hastie and Tibshirani (1990, sections 3.9 and 6.8) suggest that at least some approximate guidance can be given by referring the observed nonparametric F-statistic to an F distribution with $(\mathrm{df}_2 - \mathrm{df}_1)$ and df_1 degrees of freedom.

There are corresponding analogies for the Wald approach to testing, using quadratic forms associated with individual terms in an additive model to assess their significance. Wood (2006) and Wood (2017) describes the details in the context of testing whether relevant spline coefficients might be 0.

The reef data provide a simple illustration of how model comparisons may be made, using the `mgcv` package. The table below indicates that the evidence for a latitude effect is not compelling.

```
## model2  <- gam(score1 ~ s(latitude) + s(longitude))
## anova(model2)

   Approximate significance of smooth terms:
                 edf Ref.df      F p-value
   s(latitude)  4.329  5.284  2.131  0.0822
   s(longitude) 4.763  5.791 29.386  <2e-16
```

4.8 Further examples of additive models

Example 4.4. Mackerel eggs in the Eastern Atlantic

A further example uses data from a multi-country survey of mackerel eggs in the Eastern Atlantic [1]. Figure 4.10 shows the locations at which samples were taken. An additive model for egg density might reasonably contain terms for depth and temperature, plus a joint term for latitude and longitude, to reflect spatial position. This leads to the model

$$Y_i = \beta_0 + f_{12}(x_{1i}, x_{2i}) + f_3(x_{3i}) + f_4(x_{4i}) + \varepsilon_i,$$

where f_{12} represents a smooth two-dimensional function of latitude (x_1) and longitude (x_2), and f_3 and f_4 represent additive terms of the usual type for depth (x_3) and temperature (x_4). Two-dimensional terms require restrictions to define the functions uniquely, as in the one-dimensional case. A simple convention is $\sum_{i=1}^{n} f_{12}(x_{1i}, x_{2i}) = 0$.

[1] available in the `sm` package in R

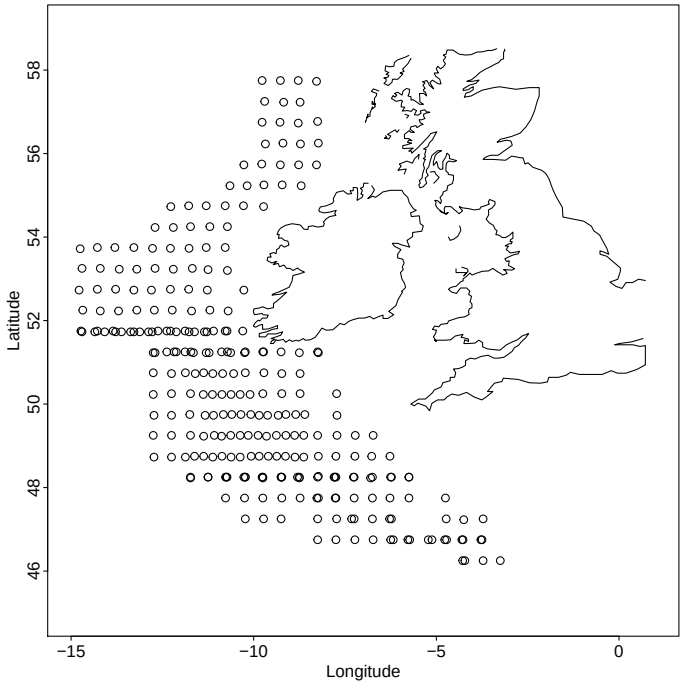


Figure 4.10. Locations of mackerel egg samples.

Figure 4.11 gives the details of a fitted GAM for the mackerel data.

◁

4.8.1 Interactions in GAM models

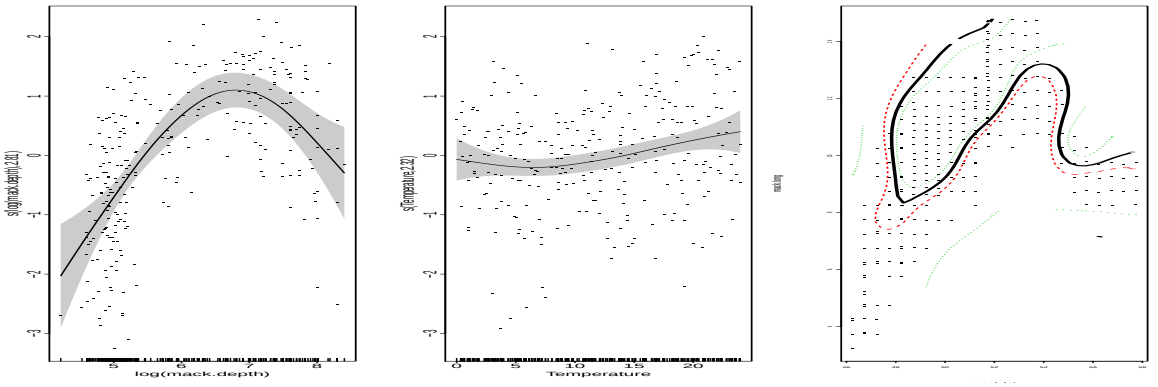
What does an interaction mean in a GAM model? A broad interpretation of an interaction between two covariates is that the effect of one depends on the setting of the other. For a GAM, this means that we need a smooth surface to describe the combined effects of the two covariates (just as we used for the spatial term in the mackerel data above). Two one-dimensional functions to capture the effects of the separate (marginal) covariates is no longer enough.

A model for the dissolved oxygen in the River Clyde illustrates this, expressed here in R syntax:

```
DO ~ s(lSalinity, Station) + s(Temperature, Station) + s(Year, Station)
```

This builds a model for the whole river, using data at many sampling stations. (Some care has to be taken here because of the repeated measures nature of the data. We will ignore this complication for the moment). We might reasonably expect that the effects of salinity, temperature and year will be different at different locations on the river. The interaction terms are shown in the surface plots, see Figure 4.12.

```
model1  <- gam(log(Density) ~ s(log(mack.depth)) + s(Temperature)
           + s(mack.lat, mack.long), data = mackerel)
par(mfrow=c(1,3), mar = c(3, 3, 1, 1), mgp = c(1.2, 0.2, 0), tcl = -0.2)
plot.gam(model1, se = TRUE, shade = TRUE, residuals = TRUE)
```




```
anova(model1)


   Family: gaussian
   Link function: identity

   Formula:
   log(Density) ~ s(log(mack.depth)) + s(Temperature) + s(mack.lat,
       mack.long)

   Approximate significance of smooth terms:
                           edf Ref.df      F  p-value
   s(log(mack.depth))    2.815  3.538 18.055 9.55e-12
   s(Temperature)        2.316  2.904  3.872   0.0147
   s(mack.lat,mack.long) 20.197 24.788  5.060 1.03e-12
```

Figure 4.11. A GAM model for the density of mackerel eggs. The right hand plot uses contours to indicate the spatial effect, with shaded contours to indicate variability.

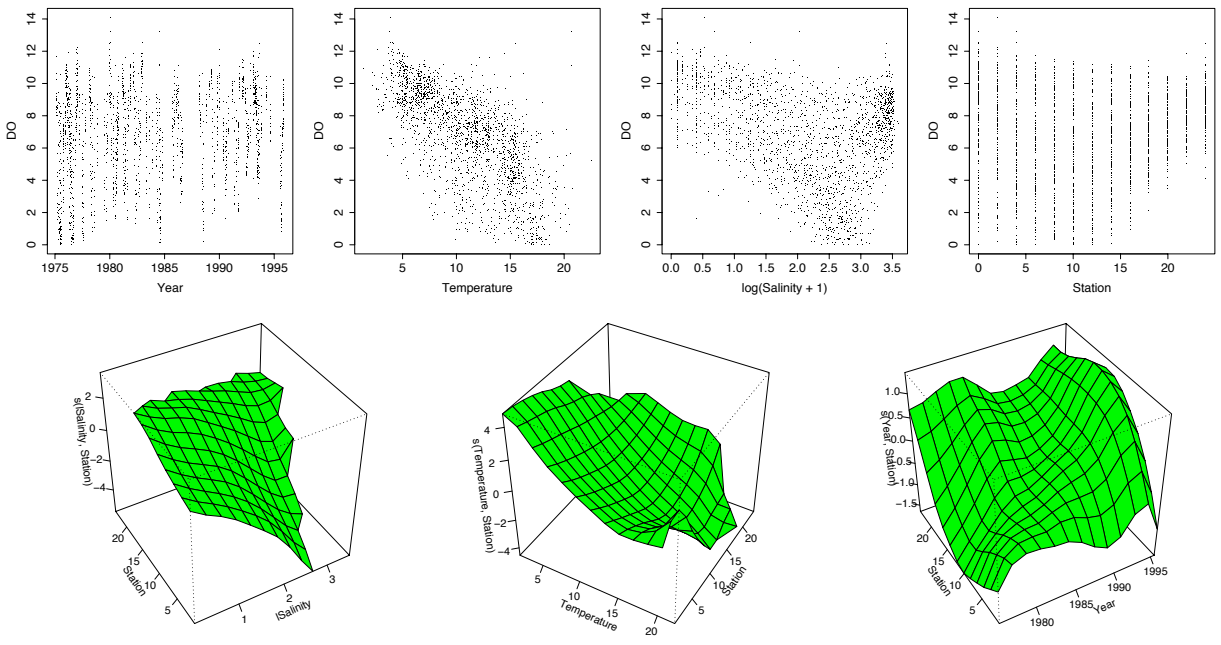


Figure 4.12. The top row of plots show DO against four covariates. The lower row of plots show interaction terms from a fitted GAM model.

4.8.2 Correlation in GAMs

The random effects framework introduced earlier can also be used in order to incorporate, and account for, correlation in GAMs.

Example 4.5. Daily river flow data were collected for a Scottish river between 1997 and 2001. It was of interest to investigate the long-term trend and any cyclical patterns in the data. ◁

The natural log transform of the data over the years and over the days of the year is displayed in the plots below.

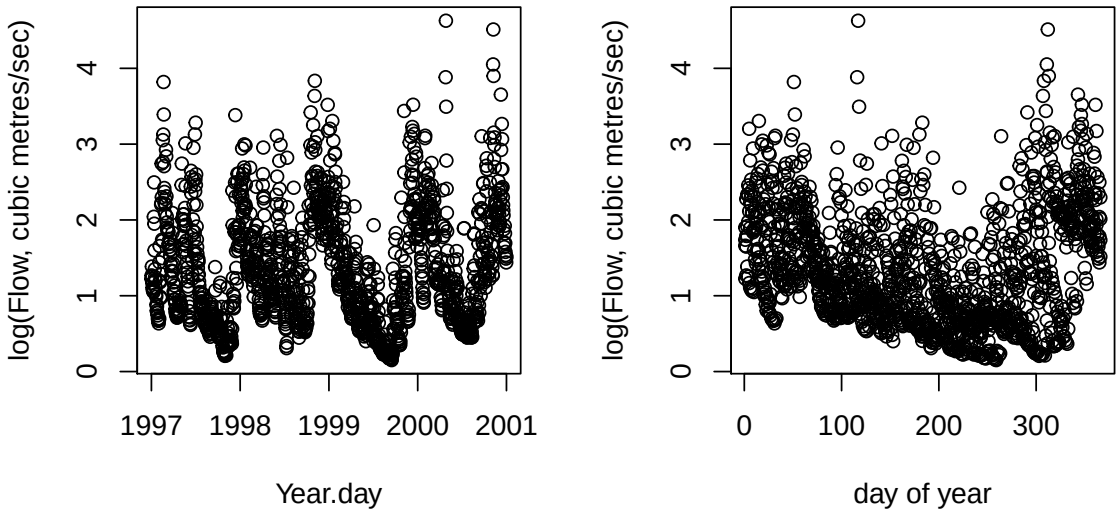


Figure 4.13. Flow data over year and day of year

We could then fit an additive model to these data to try to describe the long-term trend (Year) and the seasonal pattern (Day of Year, doy).

For example, we could fit the model:

$$\log(\text{flow}_i) = \beta_0 + s(\text{Year}_i) + s(\text{Day of Year}_i) + \varepsilon_i \tag{4.2}$$

This model can be fitted using the `mgcv` libray and the following commands:

```
gam.ind<-gam(log(Flow)~s(Year,bs="cr")+s(doy,bs="cc"))
summary(gam.ind)

   ##
```

```
## Family: gaussian
## Link function: identity
##
## Formula:
## log(Flow) ~ s(Year, bs = "cr") + s(doy, bs = "cc")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.36328    0.01473   92.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##           edf Ref.df     F p-value
## s(Year) 8.911  8.994 52.09  <2e-16 ***
## s(doy)  6.408  8.000 50.46  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.45   Deviance explained = 45.6%
## GCV = 0.32055  Scale est. = 0.31697   n = 1461
```

Note here that a circular smoother has been used for the day of year term, (`bs=''cc''`).

We can then plot the fitted model with partial residuals and a shaded band to illustrate the standard errors of the estimates.

```
plot(gam.ind, residuals=T,lty=2,lwd=2,pages=1, se=TRUE, shade.col="light blue", shade=TRUE)
```

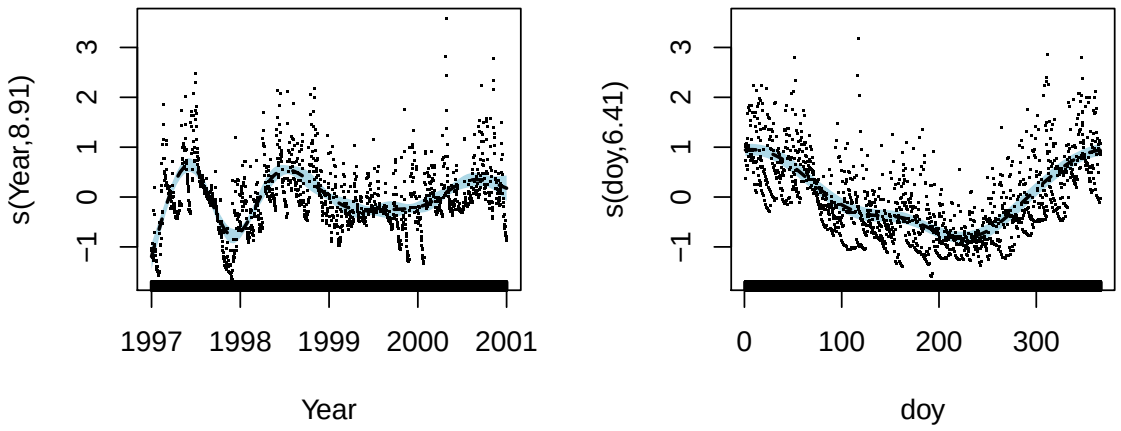


Figure 4.14. Fitted smooth components for year and day of year for a response of log(flow) for Model 4.2

Since these data are recorded over time, it is likely that there is still some correlation remaining in the residuals. This can be assessed by investigating the autocorrelation and partial autocorrelation functions of the residuals.

```
par(mfrow=c(1,2))
acf(gam.ind$residuals,plot=T,main="")
pacf(gam.ind$residuals,plot=T,main="")
```

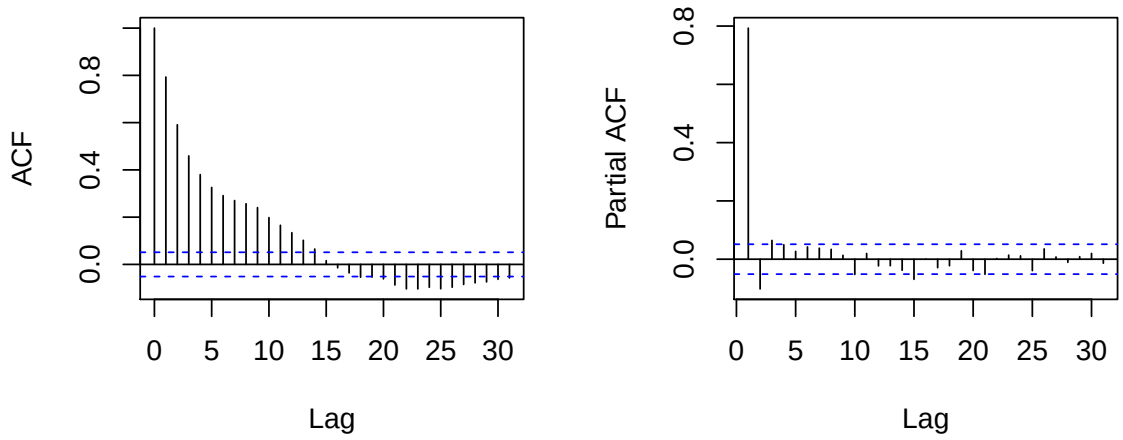


Figure 4.15. Autocorrelation (left) and partial autocorrelation function (right) of residuals after fitting Model 4.2.

It is clear from the acf and pacf (Figure 4.15) that there is autocorrelation remaining in the residuals after we have removed a long-term trend component and smooth seasonal pattern.

Instead of assuming $\varepsilon \sim N(0, \sigma^2)$, we have that, $\varepsilon \sim N(0, \Sigma)$, and one approach to incorporating the correlation is to specify this as, $\varepsilon \sim N(0, V\sigma^2)$ for a correlation matrix V, with V specified using a time series model.

We will not go into the different types of time series model here that can be used to account for correlation. However, Auto-Regressive Moving Average (ARMA) models are a wide class of models that can be used. Here we will consider how to incorporate an AR(1) correlation structure for the errors into the modelling. In such a situation, we do not wish to model the autocorrelation explicitly - it is essentially a nuisance parameter. Therefore, very often a simple ARMA model will 'mop up' a large amount of the correlation in the errors. It is generally considered that when autocorrelation is present it is better to account for it 'wrongly' than not at all.

Why do we want to do this? Well, our standard assumption is that the observations we are working with are independent. When autocorrelation is present, it has the effect of reducing our sample size. Our effective sample size is, therefore, smaller than the actual

sample size. This has the effect, that if we do not account for the autocorrelation then our standard errors are underestimated and hence we are more likely to find statistical significance, when it is not truly present.

Therefore, here we will fit:

$$\varepsilon_i = \phi\varepsilon_{i-1} + \epsilon_i,$$

with $\epsilon_i \sim N(0, \sigma^2)$.

This can be fitted in R by using the function `gamm` which gives access to the correlation structures available in the package `lme`.

```
gam.corr<-gamm(log(Flow)~s(Year,bs="cr")+s(doy, bs="cc"),correlation=corAR1(form=~1))
summary(gam.corr$gam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(Flow) ~ s(Year, bs = "cr") + s(doy, bs = "cc")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.35934    0.04968   27.36   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##           edf Ref.df     F  p-value
## s(Year) 6.493  6.493 3.921  0.00742 **
## s(doy)  3.997  8.000 5.678 5.57e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.414
##   Scale est. = 0.35879   n = 1461
```

The fitted model is shown in Figure 4.16 and residuals after accounting for the correlation can be assessed, as shown in Figure 4.17.

If we compare the two fitted models using the plots in Figure 4.18 below we can see that the standard errors for the model incorporating the correlation are now much wider than from the model which assumes independent errors.

A variety of correlation structures can be incorporated in this way. (Additionally, random effects can be incorporated in smooths by using `bs="re"`)

```
plot(gam.corr$gam,residuals=TRUE,pages=1, se=TRUE, shade=TRUE, shade.col="light blue")
```

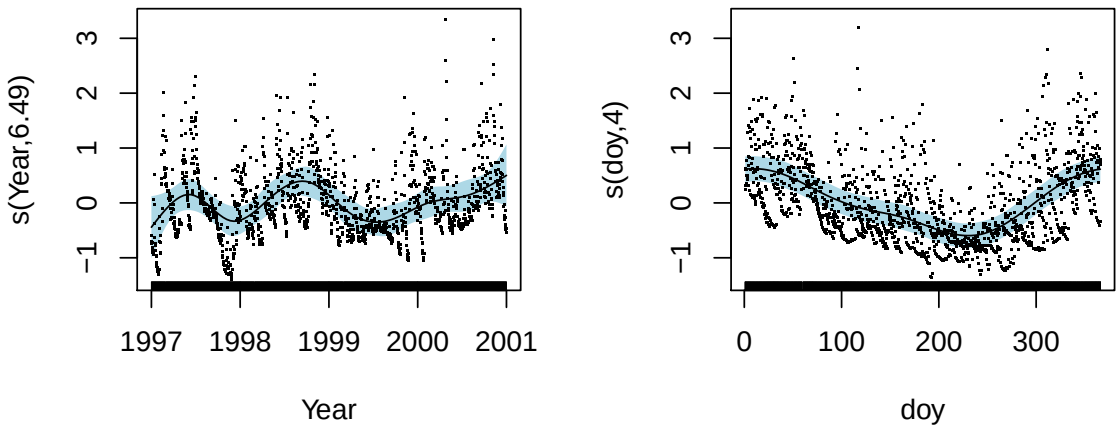


Figure 4.16. Fitted smooth components for Model 4.2 after accounting for correlated errors

```
par(mfrow=c(1,2))
acf(resid(gam.corr$lme,type="normalized"),main="")
pacf(resid(gam.corr$lme,type="normalized"),main="")
```

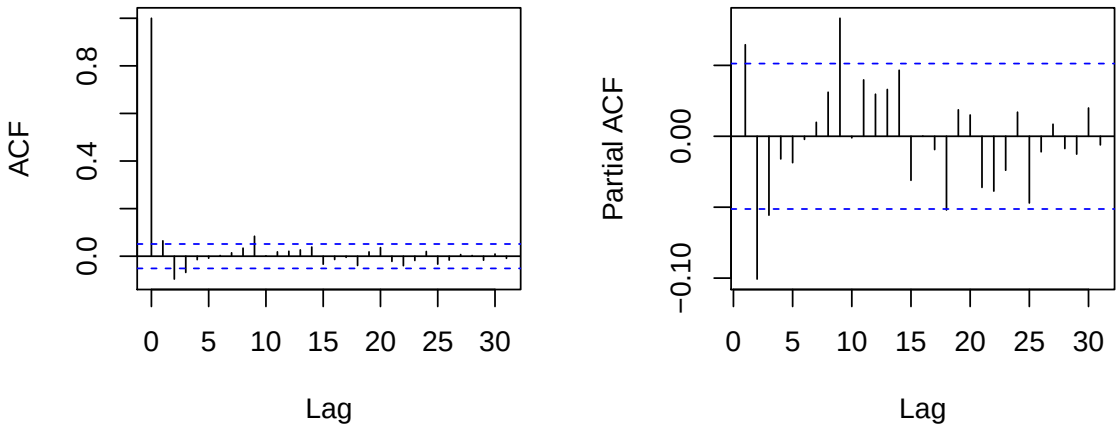


Figure 4.17. Autocorrelation (left) and partial autocorrelation function (right) of residuals after fitting model 4.2 accounting for correlation

```
par(mfrow=c(2,2))
plot(gam.ind, se=TRUE, shade=TRUE, shade.col="light blue", main="Independent case")
plot(gam.corr$gam, se=TRUE, shade=TRUE, shade.col="light blue", main="AR(1)", ylim=c(-1.5,1))
```

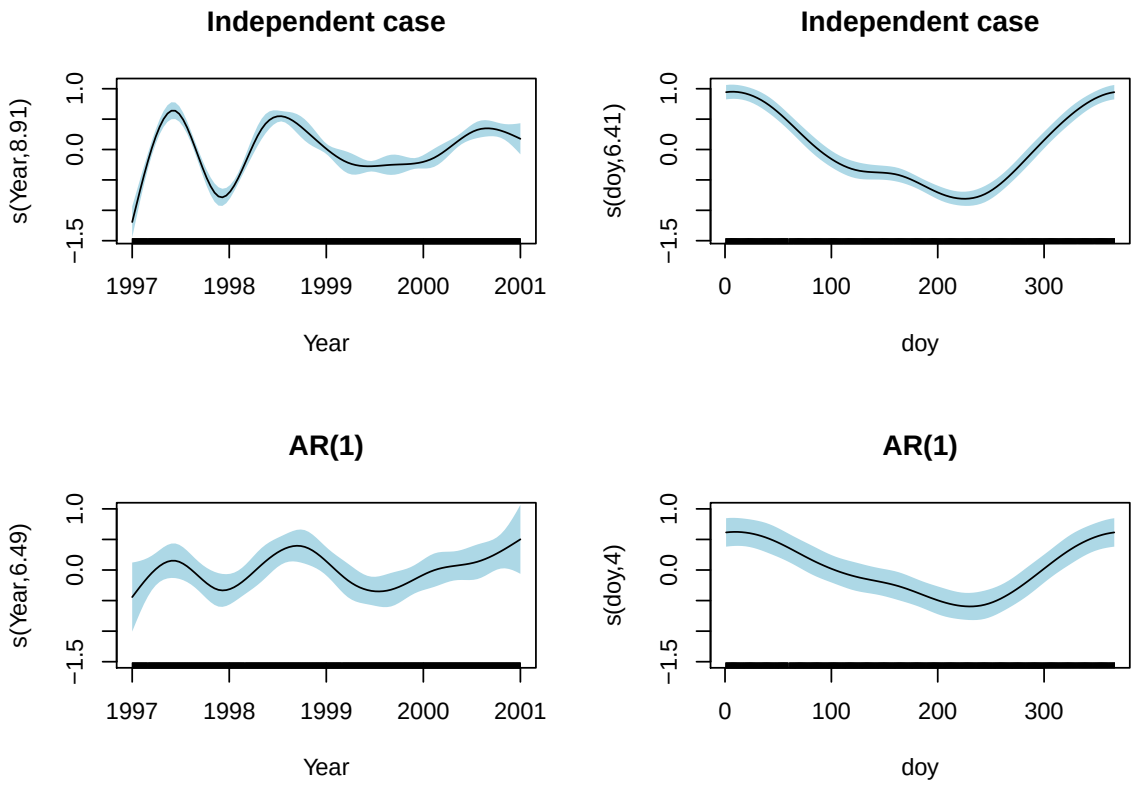


Figure 4.18. Fitted smooth components for Model 4.2 (top) and after accounting for correlation (bottom)

4.8.3 Bayesian additive models

The penalised spline approach to fitting flexible regression curves and surfaces, and additive models, is strongly suggestive of a Bayesian approach. Expression (4.1) can be viewed as the combination of a log-likelihood (quantifying how well the model fits the data) and a prior for the parameters $\boldsymbol{\beta}$ (expressing correlation between neighbouring values). This can be developed into a fully Bayesian approach, including priors for the unknown hyperparameter λ. For example, we could use the `R2BayesX` package to experiment with this approach. Figure 4.19 and Figure 4.20 show two models for the Reef data which are (reassuringly) very similar to the models produced earlier.

```
library(R2BayesX)
model1 <- bayesx(Score1 ~ sx(Longitude), data = trawl)
```

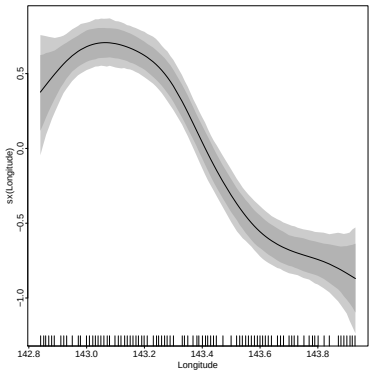
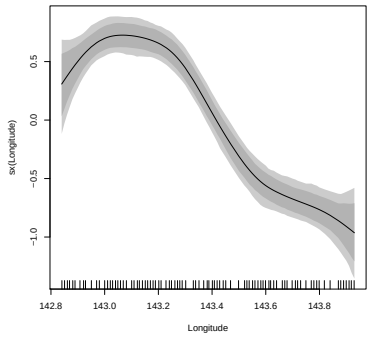


Figure 4.19. Flexible regression models for the Reef data, using a fully Bayesian approach implemented in the `BayesX` package - a model for longitude alone

```
model2 <- bayesx(Score1 ~ sx(Longitude) + sx(Latitude), data = trawl)
plot(model2, term = 1)
```




```
plot(model2, term = 2)
```

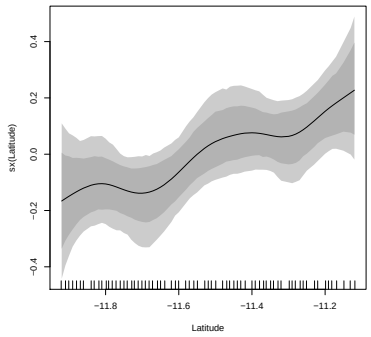


Figure 4.20. Flexible regression models for the Reef data, using a fully Bayesian approach implemented in the `BayesX` package - components of an additive model for longitude and latitude.

5

Quantile regression extensions

In this chapter we will look at extensions and alternatives to the quantile regression methods introduced in Chapter 3. Sections 5.1 and 5.2 introduce alternative methods for computing conditional quantile functions. Section 5.3 describes how quantile regression can be applied to censored data. For more information on recent extensions to quantile regression models see Koenker *et al.* (2017).

5.1 Generalised Additive Models for Location, Scale and Shape

Generalised Additive Models for Location, Scale and Shape or GAMLSS (Rigby and Stasinopoulos, 2005) are an extension to generalised additive models (GAMs, see Chapter 4). They allow us to focus not just on the conditional mean, but also on how the spread and the shape of the distribution of the response depend on explanatory variables.

GAMLSS have up to four parameters which can be influenced by explanatory variables. The μ parameter controls the location, the σ parameter controls the spread, the τ parameter controls the skewness and the ν parameter controls the kurtosis.

The `gamlss` package implements a very large number of distributions. One such distribution is the so-called Box-Cox Cole and Green distribution (BCCG) given by

$$f(y|\mu,\sigma,\nu) = \frac{1}{\sqrt{2\pi}\sigma}\frac{y^{\nu-1}}{\mu^{\nu}}\exp\left(-\frac{z^2}{2}\right)$$

where

$$z = \begin{cases} \frac{(y/\mu)^{\nu}-1}{\nu\sigma} & \text{if } \nu \neq 0 \\ \frac{\log(y/\mu)}{\sigma} & \text{if } \nu = 0. \end{cases}$$

For a tutorial on GAMLSS see Stasinopoulos *et al.* (2018).

In this section we will see how GAMLSS models can be used to obtain conditional quantile functions. Note that in contrast to the quantile regression methods we studied in Chapter 3, these models make distributional assumptions. We will illustrate the use of GAMLSS for quantile regression through an example.

Example 5.1 (Effect of age on obesity in the US). The National Health and Nutrition Examination Survey (NHANES) is a program of studies designed to assess the health and nutritional status of adults and children in the United States. As part of the NHANES data (available in the R package `NHANES`) both the age and the body mass index (BMI) are collected.

Suppose we want to study the effect of age on obesity. If the objective of our investigation is obesity, we may be more interested in how large quantiles, rather than mean BMI, change with age. This requires modelling the effect of age on all aspects of the distribution of the BMI and not just its mean.

The function `gamlss` from the package `gamlss` lets us fit such a model. In the R code that follows, we fit P-splines to each of the GAMLSS parameters. The left panel of Figure 5.1 shows the fitted centile curves. The R code for fitting this model is given below.

```
model <- gamlss(BMI~ps(Age), sigma.formula=~ps(Age),
                tau.formula=~ps(Age), data=NHANES, family="BCCG")
```

When the quantity of interest is just one quantile it is easiest to fit a quantile regression model, as discussed in Chapter 3. To find the 0.9th and 0.98th quantile of the conditional distribution of the BMI given age using the function `rq` from `quantreg` we need to fit two separate quantile regression models. The fitted quantile curves are shown in the right panel of Figure 5.1. The R code for the fits is given below.

```
library(quantreg)
model.90 <- rq(BMI~bs(Age, df=10), data=NHANES, tau=0.9)
model.98 <- rq(BMI~bs(Age, df=10), data=NHANES, tau=0.98)
```

◁

As we saw in Chapter 3, quantile regression models are run separately for each quantile which can sometimes lead to problems with quantile crossing, especially with extreme quantiles such as the above. This can be avoided by estimating the entire conditional distribution in one go, which is what GAMLSS does. But this again comes at a cost as we are making distributional assumptions which may or may not hold.

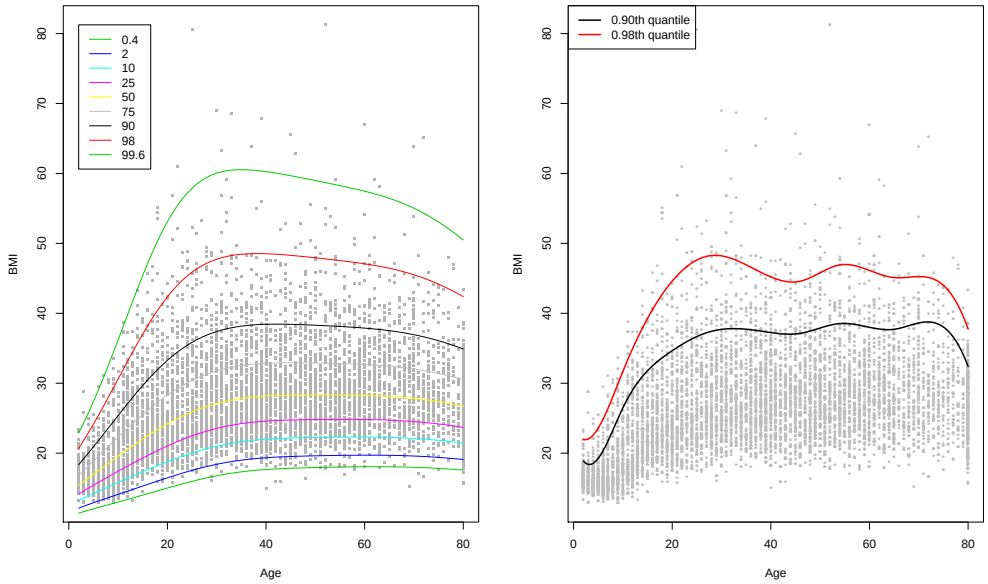


Figure 5.1. Left panel: Conditional centile functions obtained using GAMLSS. Right panel: Conditional quantile functions using `rq()` for $\tau = 0.9$ and $\tau = 0.98$.

5.2 Bayesian quantile regression

Another way to estimate conditional quantile functions is by using Bayesian computation. This has the advantage that it computes point estimates and confidence intervals simultaneously from the posterior sequences, and that it takes advantage of Markov chain Monte Carlo methods for computation. But how can Bayesian methods be implemented in the absence of a likelihood? A working likelihood is necessary. The most commonly used one is that proposed by Yu and Moyeed (2001) which utilises the asymmetric Laplace distribution. This approach uses a parametric working likelihood,while others have utilised a nonparametric/semiparametric working likelihood, *e.g.* Kottas and Krnjajić (2009), Reich *et al.* (2010) or an empirical likelihood Lancaster and Jun (2010); Otis (2008); Yang and He (2012). In what follows we will describe the approach of Yu and Moyeed (2001).

5.2.1 Asymmetric Laplace (AL) likelihood

You may have encountered the Laplace distribution with density $f(z) = \frac{1}{2\sigma}\exp\left(-\frac{|z-\mu|}{\sigma}\right)$, and may remember that the maximum likelihood estimator (MLE) of μ is the sample median. The asymmetric Laplace (AL) distribution is a generalisation of the Laplace distribution. A random variable Z is said to follow an asymmetric Laplace distribution $AL(\mu, \sigma, \tau)$ if its density is given by

$$f(z) = \frac{\tau(1-\tau)}{\sigma}\exp\left\{-\rho_\tau\left(\frac{z-\mu}{\sigma}\right)\right\}.$$

Therefore, assuming τ is known, the MLE of μ is

$$\underset{\mu}{\operatorname{argmin}} \sum_{i=1}^{n} \rho_\tau \left(\frac{z_i - \mu}{\sigma} \right) = \underset{\mu}{\operatorname{argmin}} \sum_{i=1}^{n} \rho_\tau \left(z_i - \mu \right).$$

That is, the MLE of μ is just the sample quantile of $(z_1, \cdots, z_n)$.

Yu and Moyeed (2001) developed a Bayesian quantile regression method assuming an AL likelihood for $\boldsymbol{y} = (y_1, \cdots, y_n)$:

$$L(\boldsymbol{y}|\boldsymbol{\beta}) = \{\tau(1-\tau)\}^n \exp\left\{ -\sum_{i=1}^{n} \rho_\tau\{y_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}) \right\},$$

that is, assuming $Y|\mathbf{x}_i \sim AL(\mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}, 1, \tau)$. The posterior distribution of $\boldsymbol{\beta} = \boldsymbol{\beta}(\tau)$ is

$$\pi(\boldsymbol{\beta}|\boldsymbol{y}) \propto L(\boldsymbol{y}|\boldsymbol{\beta})\pi(\boldsymbol{\beta}),$$

where $\pi(\boldsymbol{\beta})$ is the prior distribution of $\boldsymbol{\beta}$. This is a clever way to create a likelihood which makes Bayesian computation possible, in order to obtain conditional quantiles. To differentiate from other Bayesian methods for density regression, this method is sometimes also called *Laplace quantile regression*. The R package `bayesQR` implements this method.

Example 5.2. Here we illustrate the use of function `bayesQR()` from `library(bayesQR)` by fitting a quantile regression model to the data we simulated in Example 3.4. In addition to the usual arguments (`formula`, `data` and `tau`) we also have to specify the number of MCMC draws. The option to use adaptive lasso variable selection is also available. The R code to fit the model is shown below with the resulting quantile fits in Figure 5.2. The OLS fit is plotted alongside the conditional quantile functions for comparison.

```
fit.b <- bayesQR(y~x, quantile=c(.1,.25,.5,.75,.9), ndraw=5000)
```

◁

Although this method is attractive due to its conceptual simplicity and implementation in R , a word of caution is needed. Wang *et al.* (2016) show that the posterior variance is not the right approximation to the sampling variance of $\hat{\boldsymbol{\beta}}(\tau)$ and propose an adjusted posterior variance which can be used to construct asymptotically valid posterior intervals using a normal approximation.

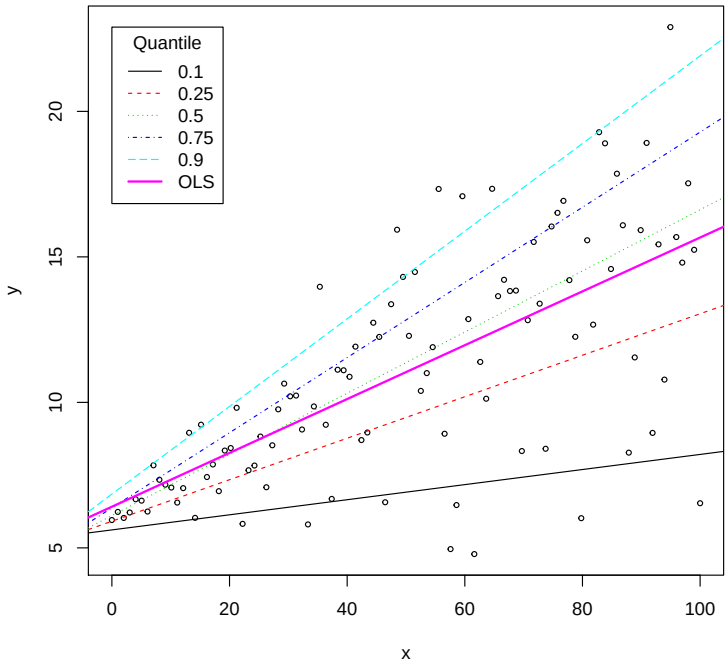


Figure 5.2. Fitted conditional quantile curves using `bayesQR` along with the OLS regression line for the data introduced in Example 3.4.

The asymmetric Laplace distribution has been used by others in various implementations of quantile regression. For example, Geraci and Bottai (2014) estimate linear quantile mixed models, which allow for random effects, by optimising an asymmetric Laplace distribution. Their method is implemented in `library(lqmm)` in R .

5.2.2 Other Bayesian approaches

In recent years several other Bayesian approaches to quantile regression have been proposed. We have discussed that of Yu and Moyeed (2001), which is implemented in the package `bayesQR` and which relies on the asymmetric Laplace distribution as an auxiliary distribution to facilitate computation. Other approaches are less restrictive in terms of distribution assumptions. For instance Yang and He (2012) propose a Bayesian empirical likelihood approach which allows joint modelling of multiple quantiles, while Kottas and Krnjajić (2009) use mixtures with Dirichlet process priors. Reich *et al.* (2010) also use mixtures in their approach for independent and clustered data (R code available from `https://blogs.gwu.edu/judywang/software/`). Finally the Bayesian density regression approach of Dunson *et al.* (2007), based on Dirichlet process mixtures, is also worth mentioning.

5.3 Quantile regression for censored and survival data

In survival analysis we are interested in questions such as how long patients with a certain condition will live with or without treatment, or in estimating unemployment duration according to various predictors. Here the quantiles (*e.g.* median duration) provide a very natural way to answer such questions, but most of the methods widely used for analysis of such data do not model the quantiles directly. Instead, a **transformation model** is typically used.

A wide variety of survival analysis models such as the Cox proportional hazards model may be written as

$$h(T_i) = \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta} + u_i,$$

where T_i is an observed survival time, h is a monotone transformation, $\mathbf{x}_i$ is a vector of covariates, $\boldsymbol{\beta}$ is an unknown parameter vector, and $\{u_i\}$ are $i.i.d. \sim F$.

Example 5.3 (Accelerated failure time model). This model can be written as $\log(T_i) = \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta} + u_i$, which is clearly of the above form. ◁

Example 5.4 (Cox proportional hazards model). For the proportional hazards model with

$$\log \lambda(t|\mathbf{x}) = \log \lambda_0(t) - \mathbf{x}^\mathsf{T}\boldsymbol{\beta}$$

the conditional survival function in terms of the integrated baseline hazard

$$\Lambda_0(t) = \int_0^\mathsf{T} \lambda_0(s)ds = \log\{S(t)\}$$

as,

$$\log[-\log\left(S(t|x)\right)] = \log \Lambda_0(t) - \mathbf{x}^\mathsf{T}\boldsymbol{\beta}$$

so, evaluating at $t = T_i$, we have the model

$$\log \Lambda_0(T) = \mathbf{x}^\mathsf{T}\boldsymbol{\beta} + u$$

for u_i iid with distribution function $F_0(u) = 1 - e^{-e^u}$. ◁

Example 5.5 (Bennett proportional odds model). For the proportional odds model, where the conditional odds of death

$$\Gamma(t|\mathbf{x}) = F(t|\mathbf{x})/(1 - F(t|\mathbf{x}))$$

are written as

$$\log \Gamma(t|\mathbf{x}) = \log \Gamma_0(t) - \mathbf{x}^\mathsf{T}\boldsymbol{\beta},$$

we have, similarly,

$$\log \Gamma_0(T) = \mathbf{x}^\mathsf{T}\boldsymbol{\beta} + u$$

for u i.i.d. logistic with $F_0(u) = (1 + e^{-u})^{-1}$. ◁

The common feature of all these transformation models is that after transformation of the observed survival times we have a pure location-shift, iid-error regression model, where the effect of the explanatory variables is to shift the centre of the distribution of $h(T)$. However, the explanatory variables cannot affect the scale or shape of this distribution.

Consider, in contrast, the quantile regression model

$$Q_\tau\{h(T_i)|\mathbf{x}_i\} = \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau),$$

where $h(\cdot)$ is a monotone transformation. By the equivariance property of quantile regression to monotone transformation (suppose h is increasing),

$$Q_\tau(T_i|\mathbf{x}_i) = h^{-1}\left(\mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau)\right).$$

Quantile regression allows the explanatory variable to influence not only the location but also the scale and shape of the conditional distribution. What's more, interpretation is simpler as we can directly interpret the effect of a predictor on the median survival time etc.

A common feature of survival or duration data is the presence of censoring. Without loss of generality assume that T_i is the transformed survival/duration and suppose we have data $(\mathbf{x}_i, Y_i, \delta_i)$, $i = 1, \cdots, n$, where

$$Y_i = \min(T_i, C_i), \quad \delta_i = I(T_i \leq C_i).$$

The equation for censored quantile regression then is

$$Q_\tau(T_i|\mathbf{x}_i) = \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}_0(\tau).$$

Example 5.6 (Student earnings against study hours). Figure 5.3 shows the weekly earnings (in dollars) of US college students plotted against the number of hours of study. The points in red are censored at 200 dollars, meaning that the students earn at least this much, but we don't know the exact amount. The green lines show the conditional quartiles estimated by ignoring censoring, while the blue lines are estimated by taking censoring into account. We see that ignoring censoring underestimates the students' earnings.

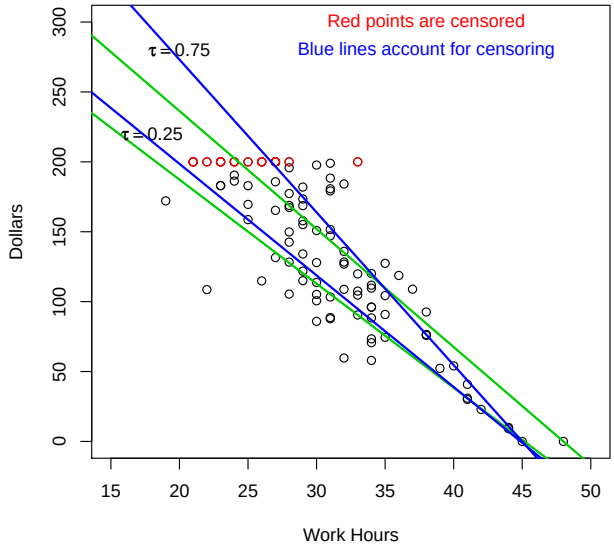


Figure 5.3. Average weekly earnings against study hours for US college students.

◁

While the conditional mean $E(T|X)$ is not identifiable in the presence of censoring, the conditional quantiles $Q_T(\tau|\mathbf{x})$ are identifiable for some τ.

Taking censoring into account is important to avoid biasing the results of survival analysis. There are various types of censoring that could be encountered:

1. **Fixed censoring**: the censoring times C_i are known for all observations, even for those subjects that are not censored. Without log of generality assume $C_i = C$. In Example 5.6 we have fixed censoring at $C = 200$ dollars. This type of censoring is quite common in survey data.
2. **Random censoring**: censoring points are unknown for uncensored observations. This is more common in biomedical studies.

In the case of random censoring, censoring points are unobserved for uncensored observations. Two different assumptions can be made, either that C is independent of T

and $\mathbf{x}$ (Assumption A) or that C and T are independent conditional on $\mathbf{x}$ (Assumption B).

A common approach under Assumption A involves re-weighting the quantile estimating equation to take censoring into account. For instance Ying *et al.* (1995) propose using the estimating equation

$$\sum_{i=1}^{n} \mathbf{x}_i \left\{ \frac{I(Y_i > \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta})}{\hat{G}(\mathbf{x}_i^\mathsf{T}\boldsymbol{\beta})} - (1-\tau) \right\} \approx 0,$$

where G is the Kaplan-Meier estimate of the survival function of C_i. The idea behind this is that

$$\begin{aligned} & P\{Y_i > \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}_0(\tau)|\mathbf{x}_i\} \\ &= P\{\min(T_i, C_i) > \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}_0(\tau)|\mathbf{x}_i\} \\ &= P\{T_i > \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}_0(\tau)|\mathbf{x}_i\} P\{C_i > \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}_0(\tau)|\mathbf{x}_i\} \\ &= (1-\tau) G\{\mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}_0(\tau)\}, \end{aligned}$$

thus the estimating function in the equation above is unbiased if G is known.

Bang *et al.* (2002) use

$$\sum_{i=1}^{n} \frac{\delta_i}{\hat{G}(Y_i)} \mathbf{x}_i \{ I(Y_i < \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}) - \tau \} \approx 0.$$

The idea behind this is that

$$\begin{aligned} & E\left[\frac{\delta_i}{G(Y_i)} \{ I(Y_i < \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}) - \tau \} | \mathbf{x}_i \right] \\ &= E\left(E\left[\frac{I(T_i < C_i)}{G(T_i)} \{ I(T_i < \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}) - \tau \} | \mathbf{x}_i, T_i \right] \right) \\ &= 0 \end{aligned}$$

when $\boldsymbol{\beta} = \boldsymbol{\beta}_0(\tau)$.

A common approach under Assumption B is to distribute the mass of censored observations to the right (*e.g.* Portnoy (2003), Wang and Wang (2009)):

$$\min \sum_{i=1}^{n} \left[w_i \rho_\tau(Y_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}) + (1-w_i)\rho_\tau(+\infty - \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}) \right],$$

where each right censored observation is split into two points

- at Y_i with mass point w_i;
- at infinity with mass $1 - w_i$.

So how does redistribution of mass work?

When there is no censoring, $Y_i = T_i$ and $\boldsymbol{\beta}_0(\tau)$ can be estimated by minimising

$$S_n(\boldsymbol{\beta}) = n^{-1} \sum_{i=1}^{n} \rho_\tau(T_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}), \tag{5.1}$$

where $\rho_\tau(u) = u\,(\tau - I(u < 0))$. The minimizer of $S_n(\boldsymbol{\beta})$ is a root of the estimating equation

$$D_n(\boldsymbol{\beta}) = n^{-1} \sum_{i=1}^{n} \mathbf{x}_i \left\{\tau - I(T_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta} \leq 0)\right\} \approx 0. \tag{5.2}$$

Here $D_n(\boldsymbol{\beta})$ is the gradient function.

Note that the gradient depends only on the signs of $T_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}_0(\tau)$. So the weights are as follows:

- Uncensored: $w_i = 1$.
- Censored and not yet crossed (above the τth quantile): i.e., $Y_i = C_i > \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}_0(\tau)$. Treat it as uncensored: $w_i = 1$.
- Censored and crossed: $\delta_i = 0$ and $Y_i = C_i < \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}_0(\tau)$, i.e. $\tilde{\tau}_i \doteq F(C_i|\mathbf{x}_i) < \tau$,

$$E\left\{I(T_i - \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}_0(\tau) < 0)|T_i > C_i, C_i, \mathbf{x}_i\right\} = \frac{\tau - \tilde{\tau}_i}{1 - \tilde{\tau}_i} \doteq w_i.$$

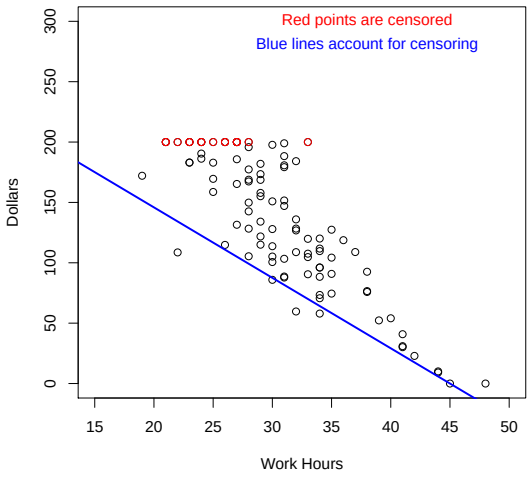
How to estimate $\tilde{\tau}_i = P(T_i \leq C_i|\mathbf{x}_i)$, the quantile level at which the conditional quantile crosses C_i?

Portnoy (2003) proposes estimating quantiles at a fine grid starting from $\tau = 0$ and then moving up step by step.

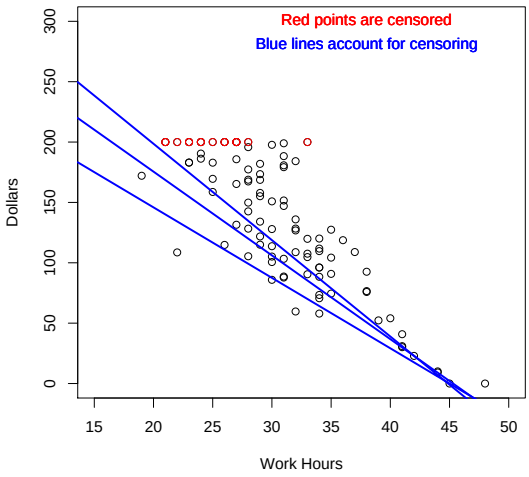
Example 5.7 (Portnoy's reweighting-to-the-right algorithm). Figures 5.4 shows a graphical illustration of the reweighting-to-the-right algorithm of Portnoy (2003) using the earnings data from Example 5.6. The process starts at $\tau = 0$ and it continues until all observations have been crossed or until only censored observations remain.

◁

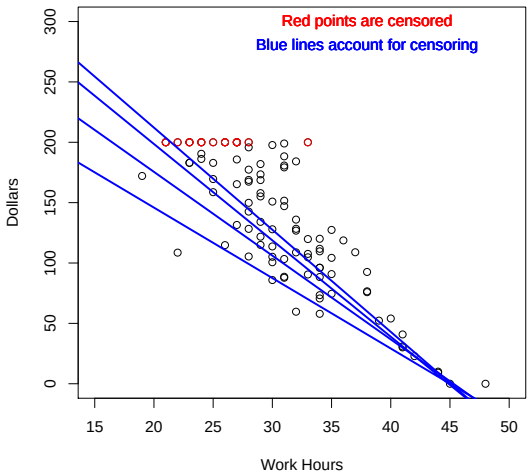
Portnoy's approach relies on redistribution-of-mass idea of Efron (1967): all observations are assigned weights depending on whether they are uncensored, censored but not yet crossed, or censored and crossed. It reduces to Kaplan Meier's estimator for the univariate case (if X has finitely many distinct values) and it also allows more general censoring. In this algorithm each update is a weighted quantile regression problem which



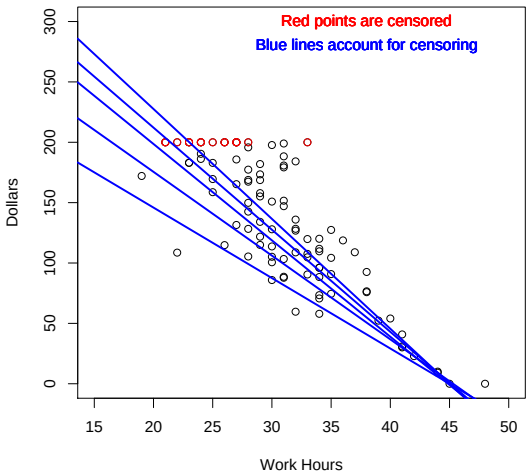
(a) Start at $\tau = 0.05$. No censored observations crossed at this point.



(b) $\tau = 0.05, 0.15, 0.25$: still no censored observations crossed.



(c) $\tau = 0.05, 0.15, 0.25, 0.35$. For the left two censored and crossed points the weights are calculated as $\tilde{\tau}_i = 0.35$, $w_i = \dfrac{\tau - 0.35}{1 - 0.35}$ for $\tau > 0.35$.



(d) $\tau = 0.05, 0.15, 0.25, 0.35, 0.45$. For the third censored and crossed point the weight is calculated as $\tilde{\tau}_i = 0.45$, $w_i = \dfrac{\tau - 0.45}{1 - 0.45}$ for $\tau > 0.45$.

Figure 5.4. Illustration of the reweighting-to-the-right algorithm of Portnoy (2003) using the earnings data from Example 5.6.

has to estimate all the quantiles below τ and assumes all the quantile functions are linear in covariates.

Wang and Wang (2009) provide an alternative method to estimate $\tilde{\tau}_i = F(C_i|\mathbf{x}_i)$ by using a local Kaplan-Meier estimator of $F(\cdot|\mathbf{x})$. In contrast to the Portnoy method, no recursive fitting is required and linearity of quantile function is needed only at the quantile level of interest. However, computation is challenging for high dimensional data. An implementation of this method in R can be found at `https://blogs.gwu.edu/judywang/software/`.

Peng and Huang (2008) extend the martingale representation of the Nelson-Aalen estimator of the cumulative hazard function to produce an "estimating equation" for conditional quantiles. Consider $\Lambda_T(t|\mathbf{x}) = -\log\{1 - F_T(t|\mathbf{x})\}$, the cumulative hazard function of T conditional on $\mathbf{x}$. Let $N_i(t) = I(Y_i \leq t, \delta_i = 1)$ be a counting process and $M_i(t) = N_i(t) - \Lambda_T\{t \wedge Y_i|\mathbf{x}_i)$ be a martingale process so that $E\{M_i(t)|\mathbf{x}_i\} = 0$ for all $t \geq 0$.

So

$$E\left[N_i\{\mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}_0(\tau)\} - \Lambda_T\{\mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}_0(\tau) \wedge Y_i\}|\mathbf{x}_i\right] = 0.$$

The following connection exists between Λ_T and the quantile functions:

$$\begin{aligned}&\Lambda_T\{\mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}_0(\tau) \wedge Y_i|\mathbf{x}_i\} = H(\tau) \wedge H\{F_T(Y_i|\mathbf{x}_i)\}\\ &= \int_0^\tau I\{Y_i \geq \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}_0(u)\}dH(u),\end{aligned}$$

where $H(u) = -\log(1-u)$ for $0 \leq u \leq 1$.

The estimating equation becomes

$$n^{-1/2}\sum_{i=1}^n \mathbf{x}_i\left[N_i(\mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}) - \int_0^\tau I\{Y_i \geq \mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(u)\}dH(u)\right] = 0.$$

Approximating the integral on a grid, $0 = \tau_0 < \tau_1 < \cdots < \tau_J < 1$, yields a simple linear programming formulation to be solved at the gridpoints,

$$\alpha_i(\tau_j) = \sum_{k=0}^{j-1} I\{Y_i \geq \mathbf{x}_i^\mathsf{T}\hat{\boldsymbol{\beta}}(\tau_k)\}\{H(\tau_{k+1}) - H(\tau_k)\},$$

yielding Peng and Huang's final estimating equation,

$$n^{-1/2}\sum \mathbf{x}_i\left[N_i(\mathbf{x}_i^\mathsf{T}\boldsymbol{\beta}(\tau)) - \alpha_i(\tau)\right] = 0.$$

Setting $r_i(\mathbf{b}) = Y_i - \mathbf{x}_i^\mathsf{T}\mathbf{b}$, this convex function for the Peng and Huang problem takes the form

$$R(\mathbf{b}, \tau_j) = \sum_{i=1}^{n} r_i(\mathbf{b}) \left[\alpha_i(\tau_j) - I\{r_i(\mathbf{b}) < 0\}\delta_i\right] = \min!$$

The estimation procedure estimates the quantile coefficients sequentially from the lower quantiles to upper quantiles. The resulting estimator is closely related to Nelson-Aalen estimator in the univariate case.

Simulation evidence from Koenker (2008) confirms that the Portnoy and Peng-Huang estimators are asymptotically similar. Both methods require estimation of all the quantiles below τ, and they assume global linearity of quantile functions.

Example 5.8 (Censored quantile regression in R *).* The function `crq()` from the package `quantreg` fits the Portnoy and Peng-Huang estimators for censored data. A third method (`method="powell"`) for fixed censoring exists, but even for fixed censoring where the C_i are observed, it is better to use the Portnoy or Peng-Huang estimators than the Powell estimator. We will illustrate the use of this function with the `uis` data example of Hosmer and Lemeshow following Koenker (2008). The response is the logarithm of time to relapse of subjects in a drug treatment program. The explanatory variables are the number of prior treatments, ND1 and ND2; the treatment indicator, TREAT taking the value 1 for subjects taking the "long" course, and 0 for subjects taking the "short" course; an indicator for prior intravenous drug use, IV3; a compliance variable, FRAC; subject's race; and the main and interaction effects of age and site of treatment. To allow a comparison with the Cox proportional hazards model, we use function `coxph` from `library(survival)` with the same formula argument. The results are plotted in Figure 5.5. The quantile coefficients are shown in blue, with a pointwise 95% confidence band estimated using bootstrap. The red line corresponds to the quantile effect according to the Cox proportional hazards model. A feature of the Cox model is that all of the red lines are proportional to one another as they are forced to have the same shape determined by the estimate of the baseline hazard function. There is agreement between the Cox estimates and some of the quantile regression coefficient estimates, but for `TREAT` and `FRAC` the estimates differ quite a bit. Another feature of the Cox estimates is that they must lie entirely above the horizontal axis or entirely below it, while quantile regression allows for the possibility that treatments may increase hazard and then decrease it or vice versa.

The R code for fitting the models is given below.

```
library(quantreg)
library(survival)
data(uis)
fit <- crq(Surv(log(TIME), CENSOR) ~  ND1 + ND2 + IV3 +
             TREAT + FRAC + RACE + AGE * SITE,
```

```
            method = "Portnoy", data = uis)                    # crq using Portnoy method
Sfit <- summary(fit,1:19/20)
PHfit <- coxph(Surv(TIME, CENSOR) ~  ND1 + ND2 + IV3 +
                TREAT + FRAC + RACE + AGE * SITE, data = uis) # Cox PH model
```

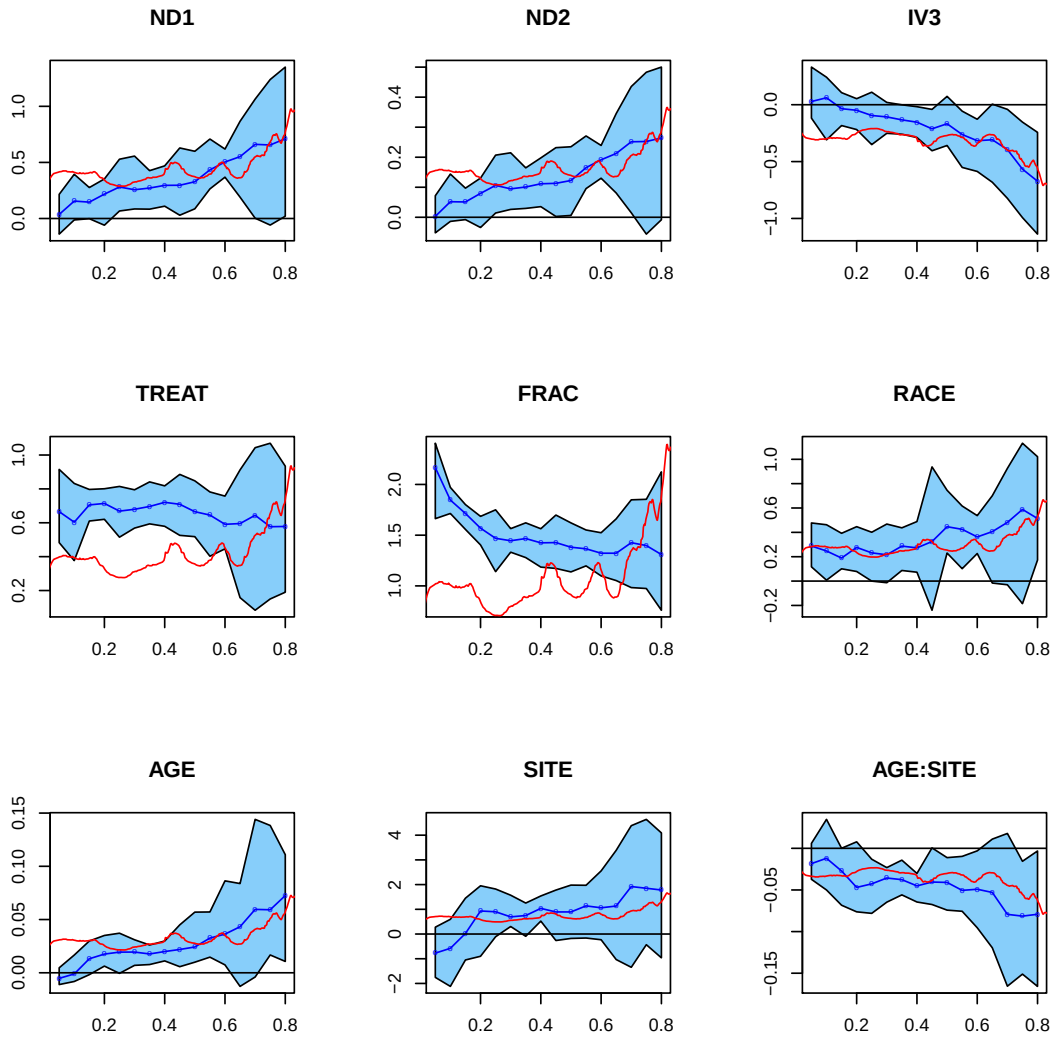


Figure 5.5. Censored quantile regression coefficients (solid blue lines) for the `uis` data with pointwise 95% confidence band (light blue). The red curve is an estimate of the conditional quantile effect from the Cox proportional hazards model.

◁

6

Flexible regression extensions

In previous chapters we have introduced a broad class of models that enable us to have flexibility in both the nature of the response and the mean regression function, and we've considered both frequentist and Bayesian approaches.

In this section we will introduce 2 further topics which provide alternative approaches to modelling random functions (and, in particular, collections of random functions):

– the Bayesian nonparametric method of Gaussian processes;
– and functional data analysis.

Gaussian processes are a Bayesian model for function estimation. A Gaussian process defines a distribution over functions and the object for which we want to perform inference is an infinite-dimensional object rather than a finite-dimensional vector of parameters.

Functional data analysis is about the analysis of information on functions or curves. We are usually interested in saying something about the combined information over a set of functions.

After briefly introducing these approaches we'll finish with a brief mention of other related approaches that the methods we have developed are related to.

6.1 Gaussian processes

6.1.1 Definition of a Gaussian process

We start by defining what a Gaussian process actually is. We define a Gaussian process to be a collection of random variables indexed by a continuous variable $Y_i = Y(\mathbf{x}_i)$ $(i = 1, 2, 3, \ldots)$ depending on covariates $\mathbf{x}_i$ such that any *finite* subset of random vari-

ables $\mathbf{y} = (y_1, \ldots, y_n) = (Y(\mathbf{x}_1), \ldots, Y(\mathbf{x}_n))$ has a multivariate normal distribution. In geostatistics, this model is known as a kriging model[1].

We assume that we can only make observations subject to noise and assume (without any loss of generality) a mean of 0, i.e.

$$\mathbf{y} \sim \mathsf{N}\left(\mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I}\right). \tag{6.1}$$

with

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \ldots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \ldots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix}. \tag{6.2}$$

If we write $Y_i = f_i + \varepsilon_i$ with $f_i = f(\mathbf{x}_i)$ and $\varepsilon_i \sim \mathsf{N}(0, \sigma^2)$ then (6.1) is equivalent to

$$\mathbf{y}|\mathbf{f} \sim \mathsf{N}(\mathbf{f}, \sigma^2\mathbf{I}) \qquad\qquad \mathbf{f} \sim \mathsf{N}\left(\mathbf{0}, \mathbf{K}\right).$$

i.e. $\mathrm{Cov}(f_i, f_j) = k(\mathbf{x}_i, \mathbf{x}_j)$

The function $k(\cdot, \cdot)$ is called *covariance function* or *kernel function* i.e. here the smoothness is specified through the covariance matrix. We are free to choose any $k(\cdot, \cdot)$ as long as it is symmetric in its arguments and the matrix $\mathbf{K}$ from (6.2) is positive semi-definite.

Note that the Bayesian linear model (revised in the preliminary material) is a special case of a Gaussian process with covariance function $k(\mathbf{x}_i, \mathbf{x}_j) = \tau^2 \cdot \mathbf{x}_i^\top \mathbf{x}_j$.

We often make the assumption that the Gaussian process is *stationary*, which is the case if and only if

$$k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i - \mathbf{x}_j).$$

Figure 6.1 shows a draw from a non-stationary Gaussian process.

An additional simplifying assumption is that the process is *isotropic*, which is the case if and only if

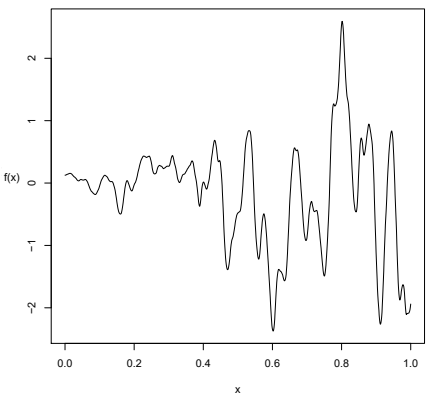
$$k(\mathbf{x}_i, \mathbf{x}_j) = k(\|\mathbf{x}_i - \mathbf{x}_j\|),$$

i.e. only distance, but not direction matters. For the remainder we will assume that the Gaussian process is stationary and isotropic. Figure 6.2 shows a draw from a non-isotropic Gaussian process.

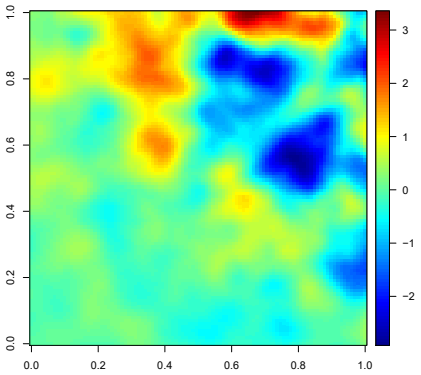
Furthermore, a process is called separable if

$$k(\mathbf{x}_i, \mathbf{x}_j) = k_1(x_{i1} - x_{j1}) \cdot k_2(x_{i2} - x_{j2}) \cdots k_p(x_{p1} - x_{p2}).$$

[1] named after Daniel Gerhardus Krige, a South African mining engineer and professor at the University of the Witwatersrand, who first suggested kriging to model mineral deposits.



(a) The variance of $f(\cdot)$ is smaller to the left than to the right.



(b) The variance of $f(\cdot)$ is smaller at the bottom-left.

Figure 6.1. Draw from a one-dimensional and two-dimensional non-stationary Gaussian process.

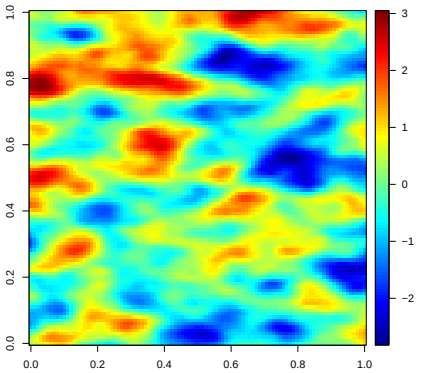


Figure 6.2. Draw from a non-isotropic two-dimensional Gaussian process. The variability in the horizontal direction is less than the one in the vertical direction.

If the covariance function is separable and the data are observed on a regular grid then we can write the covariance matrix $\mathbf{K}$ of the process as a Kronecker product

$$\mathbf{K} = \mathbf{K}_1 \otimes \mathbf{K}_2 \otimes \ldots \otimes \mathbf{K}_m,$$

where $\mathbf{K}_m$ is the covariance matrix constructed using the unique values of the m-th block of covariance only. In this case one can evaluate the posterior distribution without ever having to compute $\mathbf{K}$, which is a rather large matrix. The matrices $\mathbf{K}_j$ are of much smaller dimensions allowing for very efficient computations.
The idea of separability can also be used to define a covariance function by multiplying different covariance functions acting on separate sub-vectors of $\mathbf{x}_i$. Separability is often assumed in spatio-temporal models, where data is observed over time in space. In this case $\mathbf{x}_i = (s_{i1}, s_{i2}, t_i) = (\mathbf{s}_i, t_i)$, the covariates consist of the spatial coordinates $\mathbf{s}_i = (s_{i1}, s_{i2})$ and time t_i. Such models often make the separability assumption that

$$k((\mathbf{s}_i, t_i), (\mathbf{s}_j, t_j)) = k_1(\mathbf{s}_i, \mathbf{s}_j)k_2(t_i, t_j)$$

with $k_1(\cdot, \cdot)$ being a covariance function for space and $k_2(\cdot, \cdot)$ being a covariance function for time.

We will discuss different choices of $k(\cdot, \cdot)$ later on in section 6.1.3. In geostatistics it is quite common to use a different parametrisation and work with the so-called *(semi-)variogram*

$$\gamma(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}\text{Var}\,(m_i - m_j) = \frac{1}{2}(k(\mathbf{x}_i, \mathbf{x}_i) + k(\mathbf{x}_j, \mathbf{x}_j)) - k(\mathbf{x}_i, \mathbf{x}_j)$$

instead of the covariance function. There is a one-to-one mapping between the two, so you can either work with the (semi-)variogram or the covariance function.

6.1.2 Predictions for Gaussian processes

Conditionals of Gaussian distributions

Assume that

$$\begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} \sim \mathsf{N}\left(\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}\right)$$

Then the conditional distribution of $\mathbf{y}_2$ given $\mathbf{y}_1$ is

$$\mathbf{y}_2|\mathbf{y}_1 \sim \mathsf{N}\left(\boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}(\mathbf{y}_1 - \boldsymbol{\mu}_1), \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}\right)$$

We can compute predictions for a new observation with covariates $\mathbf{x}_0$ by looking at the joint distribution

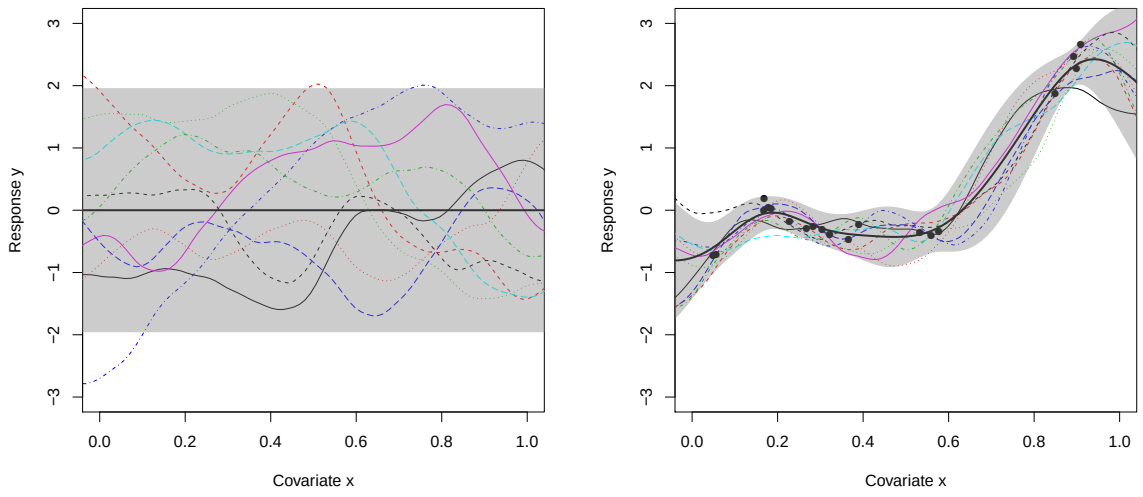
$$\begin{pmatrix} \mathbf{y} \\ y_0 \end{pmatrix} \sim \mathsf{N}\left(\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{K} + \sigma^2\mathbf{I} & \mathbf{k}_0 \\ \mathbf{k}_0^\top & k_{00} + \sigma^2 \end{pmatrix}\right),$$

where $\mathbf{K}$ is as defined in the preceding section, $\mathbf{k}_0 = (k(\mathbf{x}_0, \mathbf{x}_1), \ldots, k(\mathbf{x}_0, \mathbf{x}_n))$ is the covariance between the training data and the test case, and $k_{00} = k(\mathbf{x}_0, \mathbf{x}_0)$. Then using the formula for the conditional distribution of a Gaussian we obtain

$$y_0|\mathbf{y} \sim \mathsf{N}\left(\mathbf{k}_0^\top \left(\mathbf{K} + \sigma^2\mathbf{I}\right)^{-1} \mathbf{y}, \left(k_{00} - \mathbf{k}_0^\top \left(\mathbf{K} + \sigma^2\mathbf{I}\right)^{-1} \mathbf{k}_0\right) + \sigma^2\right)$$

The mean of the posterior distribution of y_0 can be shown to be the best linear unbiased predictor (BLUP). The formula above gives the variance to be used for a prediction interval for a new observation. If we want to get the variance for a confidence interval for its mean we have to omit the "$+\sigma^2$" term accounting for the error on the unseen data, i.e. the variance of the predicted mean is $\left(k_{00} - \mathbf{k}_0^\top \left(\mathbf{K} + \sigma^2\mathbf{I}\right)^{-1} \mathbf{k}_0\right)$.

Figure 6.3 shows five draws each from the prior distribution (panel (a)) and the posterior distribution (panel (b)) from a simple Gaussian process fitted to data.



(a) Samples from the prior distribution.

(b) Samples from the posterior distribution.

Figure 6.3. Draws from the prior distribution and the posterior distribution of a simple Gaussian process (Matérn covariance with $\kappa = 2.5$). The bold line corresponds to the mean, the shaded area corresponds to pointwise 95% credible intervals.

6.1.3 Covariance functions (kernel functions)

Squared exponential (SE) The squared exponential (or, Gaussian) covariance function is defined as

$$k(\mathbf{x}_i, \mathbf{x}_j) = \tau^2 \cdot \exp(-\rho \|\mathbf{x}_i - \mathbf{x}_j\|^2).$$

The squared exponential covariance function generates very smooth processes: their paths are infinitely differentiable, which is often unrealistically smooth.

Exponential covariance function – Ornstein-Uhlenbeck (OU) process The exponential covariance function is defined as

$$k(\mathbf{x}_i, \mathbf{x}_j) = \tau^2 \cdot \exp(-\rho \|\mathbf{x}_i - \mathbf{x}_j\|).$$

It leads to a continuous, but not a differentiable process, which is often unrealistically rough. The OU process is the continuous equivalent of an $AR(1)$ process.

γ-exponential One can generalise the above two covariance functions by considering

$$k(\mathbf{x}_i, \mathbf{x}_j) = \tau^2 \cdot \exp(-\rho \|\mathbf{x}_i - \mathbf{x}_j\|^\gamma)$$

with $0 < \gamma \leq 2$, which allows choosing any model between the rough OU process and the squared exponential. However it is less flexible than the Matérn class.

Matérn class The Matérn covariance function[2] is more flexible than the γ-exponential covariance function, however also much more complex.

$$k(\mathbf{x}_i, \mathbf{x}_j) = \tau^2 \cdot \frac{1}{\Gamma(\kappa) 2^{\kappa-1}} (2\sqrt{\kappa}\rho \|\mathbf{x}_i - \mathbf{x}_j\|)^\kappa K_\kappa \left(2\sqrt{\kappa}\rho \|\mathbf{x}_i - \mathbf{x}_j\|\right),$$

where $K_\kappa(\cdot)$ is the modified Bessel function of the second kind. Special cases of the Matérn covariance function are the OU process ($\kappa = \frac{1}{2}$) and the squared exponential ($\kappa \to +\infty$).

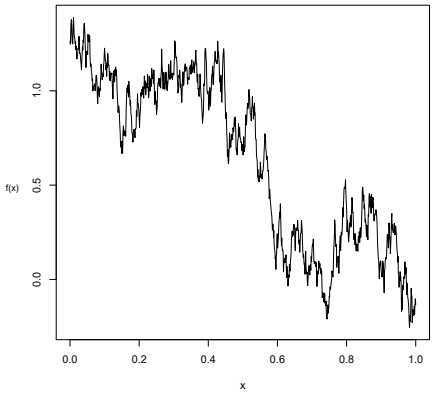
Figure 6.4 shows functions drawn from the Matérn class for different values of κ.

For all of the above covariance functions, the parameter ρ controls how fast the correlation decays. The larger ρ the quicker the decay of the correlation. The parameter τ^2 controls the prior variance of the signal. All of the above covariance functions are stationary and isotropic, as all are based only on $\|\mathbf{x}_i - \mathbf{x}_j\|$.

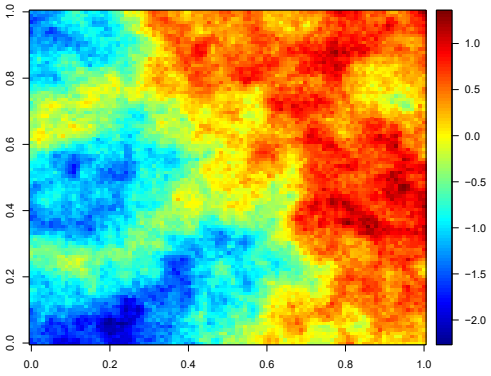
6.1.4 Estimation of hyperparameters

We have so far assumed that the hyperparameters (σ^2 and parameters of the kernel function) are known. In practice however, these need to be estimated from the data.

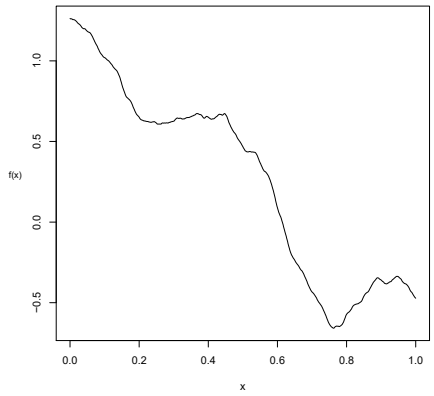
[2] named after Bertil Matén, a Swedish statistician.



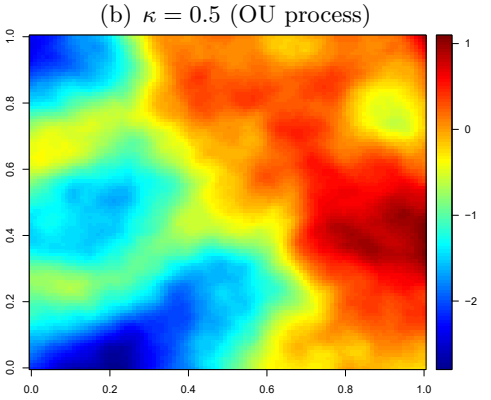
(a) $\kappa = 0.5$ (OU process)



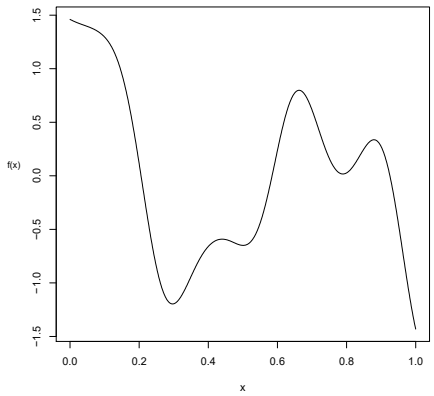
(b) $\kappa = 0.5$ (OU process)



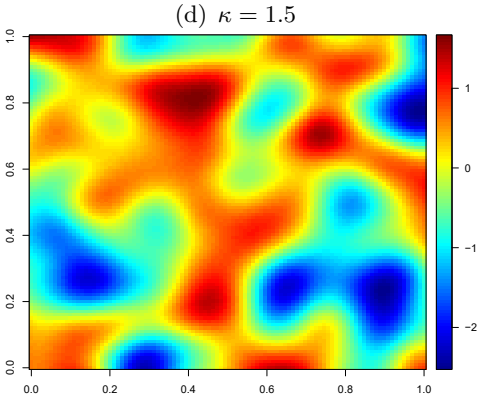
(c) $\kappa = 1.5$



(d) $\kappa = 1.5$



(e) $\kappa = +\infty$ (SE)



(f) $\kappa = +\infty$ (SE)

Figure 6.4. Samples drawn from the prior distribution of a Gaussian process with a Matérn covariance function for of $\kappa \in \{0.5, 1.5, +\infty\}$. The parameter ρ was chosen so that the covariance at lag $\frac{1}{2}$ is the same for all plots.

This is best done using the marginal log-density of $\mathbf{y}$,

$$\log f(\mathbf{y}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log\det(\mathbf{K}+\sigma\mathbf{I}) - \frac{1}{2}\mathbf{y}^\top(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{y}$$

We could use an empirical Bayes strategy (sometimes also referred to as maximum-likelihood) and maximise the density with respect to the hyperparameters.

However, a Gaussian process can use many hyperparameters and there is often little information in the data about the hyperparameters. This is especially true for the parameter κ of the Matérn covariance function. Full Bayesian models thus typically fare better as they take into account the uncertainty about the values of the hyperparameters. However, with the possible exception of σ^2, none of the hyperparameters can be integrated out in closed form, thus one has to resort to either using a discrete grid or sampling techniques such as Markov Chain Monte Carlo (MCMC).

6.1.5 Gaussian processes in R

Gaussian processes (with maximum-likelihood estimation of the hyperparameters) can be fitted using the packages `GPfit` or `mlegp`. The example below uses the latter.

We fit a GP to the Great Barrier Reef data (of Chapters 2 & 4), initially using one covariate only.

```
library(mlegp)
fit <- mlegp(trawl$Longitude , trawl$Score1)
newdata <- data.frame(Longitude = seq(min(trawl$Longitude), max(trawl$Longitude), len=50))
predictions <- predict(fit, newdata)
plot(Score1~Longitude, data=trawl)
lines(newdata$Longitude, predictions)
```

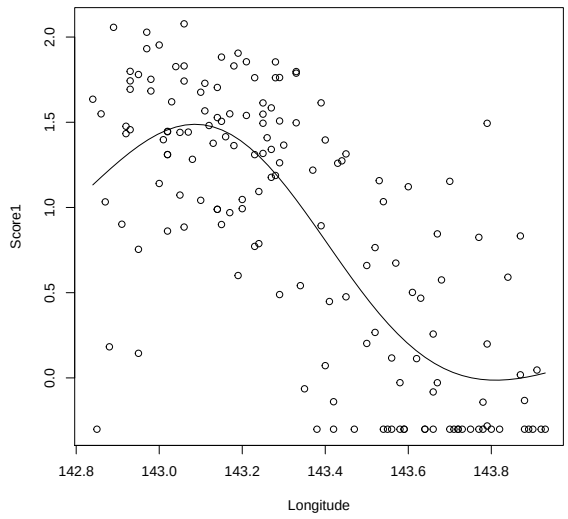


Figure 6.5. Gaussian process fit to Reef data.

A Gaussian process smooth can also be used in the `mgcv` function in R to fit a spatially smooth term. See Wood (2017) for examples.

6.2 Functional data analysis

The type of data which are now routinely collected can have quite complex structures, rather than simply having single measurements of a response variable. For example, a response might be in the form of a function collected by a monitoring device which effectively collects data continuously over time at several different locations. Although in practice the data may be discretised on a grid of time points for each location, it can be helpful to think of this as representing a function. This leads to the concept of *functional data analysis* which has attracted considerable interest over the last couple of decades. There are strong links here with the techniques we have been discussing, as methods of flexible regression provide curve descriptions which reduce noise or have compact representations through basis functions. For further details on FDA see the following references: Ramsay and Silverman (1997), Ramsay and Silverman (2002), Ramsay and Silverman (2005), Ramsay *et al.* (2009).

The data to be analysed are assumed to come from a smooth function and are modelled using a smooth function. There are two central ideas:

- The functions are smooth, usually meaning that one or more derivatives can be estimated and are useful.
- No assumptions, such as stationarity, low dimensionality, equally spaced sampling points, etc, are made about the functions or the data.

6.2.1 Functional data methods

There are functional counterparts to standard statistical approaches:

- summary statistics;
- analysis of variance;
- multiple regression analysis;
- principal components analysis;
- canonical correlation analysis;
- cluster and classification analysis;

and one way to think of functional data analysis is that it combines ideas of smoothing and multivariate statistics. Because the functions we estimate are assumed smooth, we can model the dynamic behaviour of the data. This means using differential equations

to model how the output of an input/output system changes in response to changes in the input. Let's consider an example:

Example 6.1 (Mediterranean fruit flies). A dataset containing the number of eggs laid from fifty Mediterranean fruit flies ("medflies", Ceratitis capitata) during the first 25 days of their lives is of interest here [3]. In addition to the number of eggs laid, the dataset also contains the lifespan of each fly. Our objective is to investigate whether fecundity can predict the future lifespan of a fly.

In this example the covariate is functional. Rather than having a single egg count we have a time series of 25 counts, i.e. our covariate is a function of time $x_i(t)$ for each of fifty fruit flies.

This suggests using a regression model of the form

$$\mathbb{E}(Y_i) = \int_0^{25} x_i(t)\beta(t)\, dt$$

to predict the future lifetime of the fly. Because the covariate is functional we also have to use a functional regression coefficient. ◁

The starting point is to estimate smooth functions from discrete noisy data. To do this, we use basis function expansions to model functions, and we impose smoothness using roughness penalties. This is all using similar ideas to those seen in earlier parts of the course. This produces a set of curves, one for each 'individual' in your study.

Because most functional data show variation in both phase and amplitude, the next step is often to learn how to separate phase from amplitude variation and then we can use functional versions of standard multivariate data analyses to reduce dimensionalty further and summarise key features in the data.

Every function, whether directly fit to data, or estimated from non-functional data, is assumed to have one or more derivatives available for an analysis. These derivatives themselves can then be analysed to investigate e.g. velocity and acceleration.

6.2.2 Curve fitting

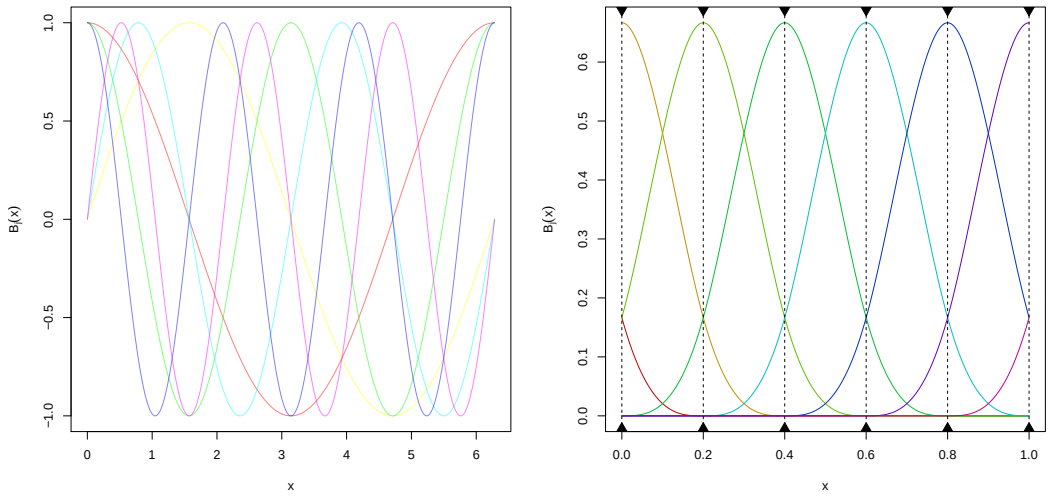
The usual starting poing is to estimate a smooth cuve for each 'individual' using basis functions. Earlier we had that:

$$f(x) = \sum_j \beta_j B_j(x).$$

[3] Available from `http://faculty.bscb.cornell.edu/~hooker/FDA2008/medfly.Rdata"`

with basis functions $\mathbf{B}$ and basis coeffcients $\boldsymbol{\beta}$.

The main choices of basis functions are B-splines (introduced in Chapter 2 and the basis of choice for most non-periodic data) and Fourier series (best for periodic data).



(a) Fourier basis (b) B-spline basis

Figure 6.6. Examples of a Fourier basis and a B-spline basis.

Fourier series was one of the first basis approaches to approximating a smooth function, which is based on the expansion

$$f(x_i) \approx \frac{a_0}{2} + \sum_{j=1}^{r} a_j \cos\left(\frac{2\pi j x_i}{P}\right) + b_j \sin\left(\frac{2\pi j x_i}{P}\right),$$

where $x_i \in (0, P)$. This approximation corresponds to using the design matrix

$$\mathbf{B} = \begin{pmatrix} \frac{1}{2} & \cos\left(\frac{2\pi x_1}{P}\right) & \sin\left(\frac{2\pi x_1}{P}\right) & \dots & \cos\left(\frac{2\pi r x_1}{P}\right) & \sin\left(\frac{2\pi r x_1}{P}\right) & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{1}{2} & \cos\left(\frac{2\pi x_n}{P}\right) & \sin\left(\frac{2\pi x_n}{P}\right) & \dots & \cos\left(\frac{2\pi r x_1}{P}\right) & \sin\left(\frac{2\pi r x_1}{P}\right) & \end{pmatrix}$$

with $\boldsymbol{\beta} = (a_0, a_1, b_1, \dots, a_r, b_r)$.

Panel (a) in Figure 6.6 shows the basis functions of a Fourier basis with $r = 3$, with a B-spline basis in panel (b) for comparison.

Each basis function in a Fourier expansion has effects across the entire range of the data.

6.2.3 FDA

Summary statistics The functional mean and functional standard deviation can be computed as follows:

mean $\bar{x} = \frac{1}{n}\sum x_i(t)$

covariance $\sigma(s,t) = \frac{1}{n}\sum(x_i(s) - \bar{x}(s))(x_i(t) - \bar{x}(t))$

For functional PCA, instead of a covariance matrix $\boldsymbol{\Sigma}$ we have a surface $\sigma(x,t)$ for functions and the eigendecomposition is re-interpreted through the Karhunen-Loève decomposition:

$$\sigma(s,t) = \sum_{i=1}^{\infty} d_i \xi_i(s)\xi_i(t)$$

with the ξ_i orthonormal, and providing the principal components, and the d_i providing the variance. The principal component scores are:

$$f_{ij} = \int \xi_i(t)[x_j(t) - \bar{x}(t)]dt.$$

The best way to obtain an idea of the variation for each component is to plot:

$$\bar{x}(t) \pm 2\sqrt{d_i}\xi_i(t).$$

This can then be followed up by fitting functional regression models. There are three types of model to consider:

– Response is a function; covariates are multivariate.
– Response is scalar or multivariate; covariates are functional.
– Both response and covariates are functional.

6.2.4 An example of FDA in R

Let's explore an appropriate functional data analysis for the fruit flies example. The package `fda` can be used to generate appropriate basis functions, explore the data using functional principal components analysis and to fit a functional regression model.

We start by creating a set of 10 basis functions which are then used to represent the data. From this we then create an `fd` object to hold the functional data.

```
basisfd <- create.bspline.basis(rangeval=c(0, 25), 10)
xfd <- Data2fd(medfly$eggcount, argval=0:25, basisobj=basisfd)
lifetime <- as.numeric(medfly$lifetime)
plot(xfd)
```

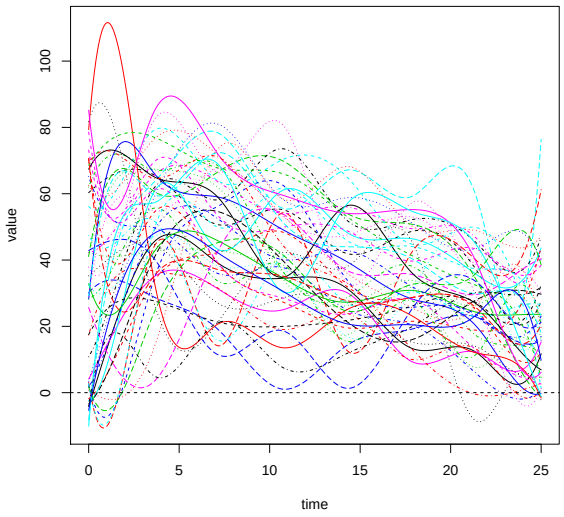


Figure 6.7. Created functions for the medfly data

To further explore the data, we can compute the functional principal components.

```
par(mfrow=1:2)
plot(pca.fd(xfd),pointplot=FALSE)
```

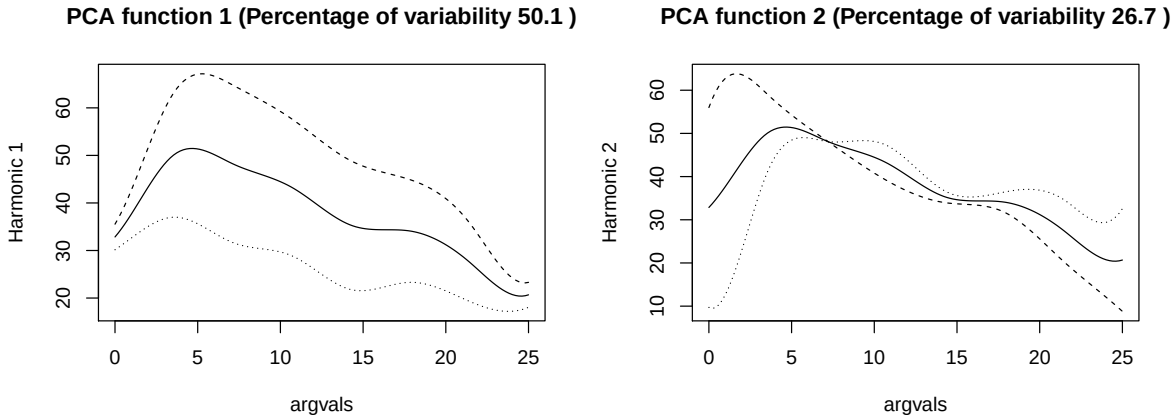


Figure 6.8. Functional PCA for medfly data

Finally we fit the functional regression model. Note that the regression coefficient is now itself a function (represented as a B-spline). In this example, the response *lifetime* is a scalar and the covariate is functional and so we have a model of the form:

$$Y_i = \beta_0 + \int_0^t \beta(t) x_i(t) dt + \varepsilon_i$$

```
basisfd <- create.bspline.basis(rangeval=c(0, 25), 10)
xfd <- Data2fd(medfly$eggcount, argval=0:25, basisobj=basisfd)
lifetime <- as.numeric(medfly$lifetime)
model <- fRegress(lifetime ~ xfd)
plot(model$betaestlist[[2]])
```

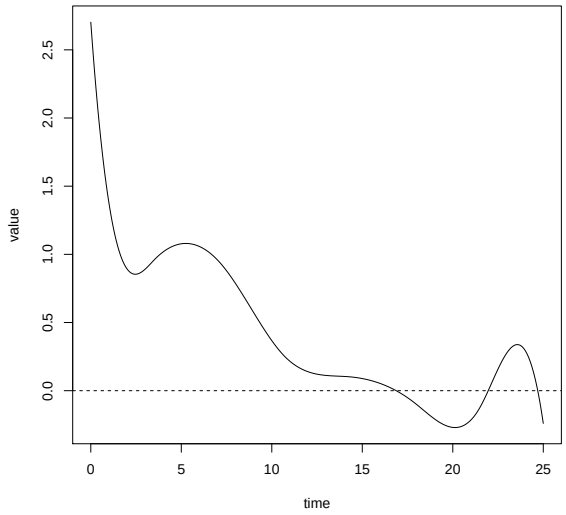


Figure 6.9. Plotted functional coefficients for medfly data.

Since the coefficients before around 15 days are positive, the number of eggs laid towards the beginning of the 25 day period is positively related to the further survival of the fly.

6.3 Other flexible regression models

We have come to the end of our treatment of flexible regression models for this course. However, there are several other extensions here that we have not introduced.

Spatial data We have seen how we can use bivariate smooths and Gaussian processes to develop smooth functions for geostatistical data. We may also have data collected as point processes or discrete/areal units and this opens up two other broad classes of models, the latter of which are very often investigated using CAR models. Additionally, data may be obtained from a connected network e.g. from a river and it can be important to account for the flow strength, direction and connectedness of the river.

Mixture models In the preliminary material we introduced estimation using density functions, this can be extended along with work on Dirichlet processes to provide a nonparametric representation for mixture models.

Neural networks From a computational viewpoint the artifical intelligence approach of neural networks can be seen as an alternative (similar and sometimes more flexible)

approach to fitting additive models. Neural networks are basically nonlinear models [4]. A neural network [5] is a two-stage regression or classification model, typically represented by a network diagram that takes nonlinear functions of linear combinations of the inputs. They can approximate any nonlinear function.

Example 6.2. For example, if we consider the simple case of linear regression. Suppose we are interested in predicting a response (Y) at data point i (the output layer with one node). We will do this using the covariates $x_1, \ldots x_p$ (p nodes in the input layer), and we will combine them using a weighted sum with coefficients β_j (as the weights).

This is an example of a very simple neural network with no hidden layer and no non-linear computations.

◁

In neural networks, in general, non-linear computations can be applied in different layers, and we have additional hidden layers between the inputs and outputs that apply non-linear computations to provide a flexible response.

The approach is most useful when prediction rather than interpretation is the goal (hence a key difference from the additive models, and our approach here, where our goal has been to describe and interpret the nature of relationships).

Therefore, there is very much more to explore and develop in this field for those of you that are interested.

[4] see Hastie *et al.* (2001) for more details here, this text also pulls together well the smoothing/additive model ideas covered in this course along with unsupervised and supervised learning approaches

[5] first developed as models for the human brain to illustrate electrical signals passing from one layer to another but then later recognised as useful tools to provide nonlinear models

References

Andrews, D. and Buchinsky, M. (2000). A three-step method for choosing the number of bootstrap repetitions. *Econometrica 68*, 23–51.

Andrews, D. and Buchinsky, M. (2001). Evaluation of a three-step method for choosing the number of bootstrap repetitions. *Journal of Econometrics 103*, 345–386.

Bang, O., Krolikowski, W., Wyller, J., and Rasmussen, J. (2002). Collapse arrest and soliton stabilization in nonlocal nonlinear media. *Physical Review E 66*, 046619.

Barney, W., Powell, J., and Tauchen, G. (1991). *Nonparametric ad Semiparametric Methods in Econometrics*. Cambridge: Cambridge University Press.

Barrodale, I. and Roberts, F. (1974). Solution of an overdetermined system of equations in the l1 norm. *Communications of the ACM 17*, 319–320.

Bowman, A. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis*. Oxford: Oxford University Press.

Cook, B. L. and Manning, W. G. (2013). Thinking beyond the mean: a practical guide for using quantile regression methods for health services research. *Shanghai Arch Psychiatry 25*, 55–59.

De Angelis, D., Hall, P., and Young, G. (1993). Analytical and bootstrap approximations to estimator distributions in l1 regression. *Journal of the American Statistical Association 88*, 1310–1316.

de Boor, C. (1978). *A Practical Guide to Splines*. New York: Springer-Verlag.

Dunson, D., Pillai, N., and J-H., P. (2007). Bayesian density regression. *Journal of the Royal Statistical Society: Series B 69*, 163–183.

Efron, B. (1967). The two sample problem with censored data. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability 4*, 831–853.

Fan, J. and Gijbels, I. (1996). *Local polynomial modelling and its applications*. London: Chapman and Hall.

Feng, X., He, X., and Hu, J. (2011). Wild bootstrap for quantile regression. *Biometrika 98*, 995–999.

Geraci, M. and Bottai, M. (2014). Linear quantile mixed models. *Statistics & Computing 24*, 461–479.

Gutenbrunner, C., Jurečková, J., Koenker, R., and Portnoy, S. (1993). Tests of linear hypotheses based on regression rank scores. *Journal of Nonparametric Statistics 2*, 307–333.

Hastie, T. and Tibshirani, R. (1990). *Generalized Additive Models*. London: Chapman and Hall.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

Hattab, M. W., de Souza, R. S., Ciardi, B., Paardekooper, J. P., Khochfar, S., and Vecchia, C. D. (2018). A case study of hurdle and generalized additive models in astronomy: the escape of ionizing radiation.

He, X. and Hu, F. (2002). Markov chain marginal bootstrap. *Journal of the American Statistical Association 97*, 783–795.

He, X. and Shao, Q. (1996). A general bahadur representation of m-estimators and its application to linear regression with nonstochastic designs. *The Annals of Statistics 24*, 2608–2630.

Keener, R. and d'Orey, V. (1987). Computing regression quantiles. *Applied Statistics 36*, 383–393.

Knight, K. (1998). Limiting distributions for l1 regression estimators under general conditions. *The Annals of Statistics 26*, 755–770.

Knight, K. (2003). On the second order behaviour of the bootstrap of l1 regression estimators. *Journal of the Iranian Statistical Society 2*, 21–42.

Kocherginsky, M., He, X., and Mu, Y. (2005). Practical confidence intervals for regression quantiles. *Journal of Computational and Graphical Statistics 14*, 41–55.

Koenker, R. (2005). *Quantile Regression(Econometric Society Monographs)*. Cambridge University Press.

Koenker, R. (2008). Censored quantile regression redux. *Journal of Statistical Software 27*.

Koenker, R., Chernozhukov, V., He, X., and Peng, L. (2017). *Handbook of Quantile Regression*. Chapman and Hall/CRC.

Koenker, R. and Ng, P. (2003). Sparsem: A sparse matrix package for r. *Journal of Statistical Software 8*, 1–9.

Kottas, A. and Krnjajić, M. (2009). Bayesian semiparametric modelling in quantile regression. *Scandinavian Journal of Statistics 36*, 297–319.

Lancaster, T. and Jun, S. (2010). Bayesian quantile regression methods. *Journal of Applied Econometrics 25*, 287–307.

Marx, B. and Eilers, P. (1998, August). Direct generalized additive modeling with penalized likelihood. *Computational Statistics & Data Analysis 28*(2), 193–209.

Otis, T. (2008). Conditional empirical likelihood estimation and inference for quantile regression models. *Journal of Econometrics 142*, 508–538.

Parzen, M. and Ying, Z. (1994). A resampling method based on pivotal estimating functions. *Biometrika 81*, 341–350.

Peng, L. and Huang, Y. (2008). Survival analysis with quantile regression models. *Journal of the American Statistical Association 103*, 637–649.

Pollard, D. (1991). Asymptotics for least absolute deviation regression estimators. *Econometrics Theory 7*, 186–199.

Portnoy, S. (2003). Censored regression quantiles. *Journal of the American Statistical Association 98*, 1001–1012.

Portnoy, S. and Koenker, R. (1997). The gaussian hare and the laplacian tortoise: Computability of squared-error versus absolute-error estimators, with discussion. *Statistical Science 12*, 279–300.

Ramsay, J. O., Hooker, G., and S., G. (2009). *Applied Functional Data Analysis with R and MATLAB (2009)*. Springer.

Ramsay, J. O. and Silverman, B. W. (1997). *Functional Data Analysis*. Springer.

Ramsay, J. O. and Silverman, B. W. (2002). *Applied Functional Data Analysis*. Springer.

Ramsay, J. O. and Silverman, B. W. (2005). *Functional Data Analysis SECOND EDITION*. Springer.

Reich, B., Bondell, H., and Wang, H. (2010). Flexible bayesian quantile regression for independent and clustered data. *Biostatistics 11*, 337–352.

Rigby, R. A. and Stasinopoulos, D. M. (2005). Generalized additive models for location, scale and shape,(with discussion). *Applied Statistics 54*, 507–554.

Ruppert, D., Wand, M. P., and Carroll, R. (2003). *Semiparametric regression*. London: Cambridge University Press.

Siddiqui, M. (1960). Distribution of quantiles from a bivariate population. *Journal of Research of the National Bureau of Standards 64*, 145–150.

Stasinopoulos, M., Rigby, R., and de Bastiani, F. (2018). A distributional regression approach using gamlss. *Statistical Modelling 18*, 248–273.

Wang, H. and Wang, L. (2009). Locally weighted censored quantile regression. *Journal of the American Statistical Association 104*, 1117–1128.

Wang, Y., Wang, H., and He, X. (2016). Posterior inference in bayesian quantile regression with asymmetric laplace likelihood. *International Statistical Review 84*, 327–344.

Wood, S. (2006). *Generalized Additive Models: an introduction with R*. London: Chapman and Hall/CRC.

Wood, S. (2017). *Generalized Additive Models: an introduction with R, SECOND EDITION*. CRC Press, Taylor & Francis Group.

Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society Series B-Statistical Methodology 73*, 3–36.

Yang, Y. and He, X. (2012). Bayesian empirical likelihood for quantile regression. *The Annals of Statistics 40*, 1102–1131.

Ying, Z., Jung, H., and Wei, L. (1995). Survival analysis with median regression models. *Journal of the American Statistical Association 90*, 178–184.

Yu, K. and Moyeed, R. (2001). Bayesian quantile regression. *Statistics & Probability Letters 54*, 437–447.