Post-course assessment

The questions are grouped into easy and moderate/hard questions. Most questions consist of both a "theory" part and an "applied" part. Choose the questions you want to work on based on your strengths and interests. You should try at least two moderate/hard questions.

All data used in the questions on this assignment sheet can be loaded into R using the command

```
load(url("http://www.stats.gla.ac.uk/~levers/aptspca.RData"))
```

This file also contains the function `bbase` *used to construct the B-spline basis in the first practical session.*

Easy questions

Question 1 (Kernel-density estimation). Consider the distribution represented by a density estimate $\hat{f}$, constructed from a sample of data $\{y_1, \ldots, y_n\}$. What is the mean and variance of this distribution? What do these expressions indicate about the nature of smoothing?

Note this question does not refer to $\mathbb{E}(\hat{f}(y))$ and $\text{Var}(\hat{f}(y))$ at specific values of y, as discussed in the lectures. It refers to the mean and variance of a random variable whose density function is $\hat{f}$.

Question 2 (Local regression). In section 2.1 we have defined the local mean estimator as

$$\hat{m}(x) = \frac{\sum_{k=1}^n w(x_k - x; h) y_k}{\sum_{k=1}^n w(x_k - x; h)},$$

which implies that

$$\hat{y}_i = \hat{m}(x_i) = \frac{\sum_{k=1}^n w(x_k - x_i; h) y_k}{\sum_{k=1}^n w(x_k - x_i; h)}.$$

Thus the fitted values are a linear function of the observed response and we can write $\hat{\mathbf{y}} = \mathbf{S}\mathbf{y}$.

(a) Write down the entries of the matrix $\mathbf{S}$.

(b) Consider the radio-carbon dating data (available in the data frame `radiocarbon`), in which we aim to predict the age from radio-carbon dating (`rc.age`) from the calendar age (`cal.age`).

  (i) Construct the matrix $\mathbf{S}$ for a Gaussian kernel with bandwidth $h = 0.05$ and use it to compute the fitted values.

  (ii) Plot the data and add the fitted values.

  (iii) In section 2.4 of the notes we have seen that we can define the effective degrees of freedom as $\text{tr}(\mathbf{S})$. Compute the trace.

  (iv) Change the value of h. How does this change the fitted function and the effective degrees of freedom?

Question 3 (Penalised least-squares for P-splines). In section 3.3 of the notes we have introduced P-splines, which minimise the penalised least-squares criterion

$$\sum_{i=1}^n (y_i - m(x_i))^2 + \lambda \|\mathbf{D}\boldsymbol{\beta}\|^2 = \|\mathbf{y} - \mathbf{B}\boldsymbol{\beta}\|^2 + \lambda \|\mathbf{D}\boldsymbol{\beta}\|^2 = (\mathbf{y} - \mathbf{B}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{B}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^\top \mathbf{D}^\top \mathbf{D}\boldsymbol{\beta},$$

where $\mathbf{B}$ is the matrix of B-spline basis functions, $\mathbf{D}$ is the difference matrix used in the penalty, and $\lambda \geq 0$ is the smoothing parameter.

(a) Show, by taking the derivative with respect to $\boldsymbol{\beta}$, that the minimiser of the penalised least-squares criterion is given by

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

(b) Explain why we can rewrite the objective function as

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

(c) *(harder)* How can we exploit this to estimate $\boldsymbol{\beta}$ using a QR decomposition, which is numerically more stable than inverting the matrix $\mathbf{B}^\top\mathbf{B} + \lambda\mathbf{D}^\top\mathbf{D}$?

Question 4 (Clyde dissolved oxygen data). The Clyde DO data (available in the data frame `clyde`) were used in one of the practical sessions.

Subset these data to focus only on Station 10. Fit an additive model to describe the relationship between DO and the three potential explanatory variables, Doy (day of the year), Year and Salinity. Describe the results of fitting three additive terms. Now fit a model which includes interaction (a bivariate term) between Doy and Year. Is there any evidence that the seasonal effect has changed over the years at this station?

Moderate/hard questions

Question 5 (B-spline basis functions for equally-spaced knots). The B-spline basis functions are defined recursively. Given a set of l knots the B-spline basis of degree 0 is given by the functions $(B_1^0(x), \dots, B_{l-1}^0(x))$ with

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

The B-spline basis of degree $r > 0$ is given by the functions $(B_1^r(x), \dots, B_{l+r-1}^r(x))$ with

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

Important: Note there was a typo in the notes (subscript in bold above).

We will now turn to the important special case that the knots are equally spaced, i.e. $\kappa_{j+1} - \kappa_j = \delta$ for all j. In this case the basis functions can be computed as r-th order differences of truncated polynomials.

We start by defining the coefficients of the difference of r-th order

$$\Delta_j^r = (-1)^j \binom{r}{j} \qquad \text{for } j = 0, \dots, r$$

For a first-order difference we obtain $\Delta_0^1 = 1$ and $\Delta_1^1 = -1$.
For a second order difference we obtain $\Delta_0^2 = 1$, $\Delta_1^2 = -2$ and $\Delta_2^2 = 1$.
For a third order difference we obtain $\Delta_0^3 = 1$, $\Delta_1^3 = -3$, $\Delta_2^3 = 3$ and $\Delta_3^3 = -1$.
These are also the numbers appearing inside the differencing matrix used in P-splines.

(a) Show that $B_j^r(x) = \sum_{i=0}^{r+1} \frac{\Delta_i^{r+1}}{r!\delta^r}(x - \kappa_{j-r+i})_+^r$

The proof is by induction using the recursive definition of B-splines.

The following two properties of Δ_j^r will help in the proof.

- $\Delta_0^{r+1}=1$, $\Delta_{r+1}^{r+1}=(-1)^{r+1}$ *and* $\Delta_j^{r+1}=\Delta_j^r-\Delta_{j-1}^r$ *for* $j=1,\dots,r$.
  For first-order and second-order differences this corresponds to:

$$\begin{array}{ccc} 1 & -1 & \\ & -1 & 1 \\ \hline\hline 1 & -2 & 1 \end{array}$$

  For second-order and third-order differences this corresponds to:

$$\begin{array}{cccc} 1 & -2 & 1 & \\ & -1 & 2 & -1 \\ \hline\hline 1 & -3 & 3 & -1 \end{array}$$

  If we define $\Delta_{-1}^r=\Delta_{r+1}^r=0$, *the formula* $\Delta_j^{r+1}=\Delta_j^r-\Delta_{j-1}^r$ *holds for all* $j=0,\dots,r+1$.

- $(r-i+1)\Delta_{i-1}^r=(r-i+1)(-1)^{i-1}\frac{r!}{(i-1)!(r-i+1)!}=(-1)^{i+1}\frac{r!}{(i-1)!}(r-i)!=-(-1)^i i\frac{r!}{i!(r-i)!}=-i\Delta_i^r$

(b) Write an R script (or function) which uses the above method to generate a B-spline basis of degrees 0, 1, and 2. Can you generalise your script (or function) so that it generate a B-spline basis of any order r?

Question 6 (Derivative estimation). In this question we will focus on estimating the derivative $m'(x)$ of the regression function. We will initially be focusing on B-splines.

(a) Show that

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

In the special case of equally-spaced knots ($\kappa_j-\kappa_{j-1}=\delta$) this becomes

$$\frac{\partial}{\partial x}B_j^r(x)=\frac{1}{\delta}B_{j-1}^{r-1}(x)-\frac{1}{\delta}B_j^{r-1}(x).$$

Hint: The proof in the general case is done by induction using the recursive definition of B-splines. In the special case of equally spaced knot, one can simply take the derivative of the formula from question 5.

(b) Show that, in the case of equally-spaced knots,

$$\frac{\partial}{\partial x}m(x)=\sum_{j=1}^{l+r-2}B_j^{r-1}(x)\frac{\beta_{j+1}-\beta_j}{\delta},$$

i.e. we can estimate the derivative by multiplying the matrix of basis functions of degree $r-1$ with the vector $\hat{\gamma}=(\hat{\gamma}_1,\dots,\hat{\gamma}_{r+l-2})$, where $\hat{\gamma}_j=\dfrac{\hat{\beta}_{j+1}-\hat{\beta}_j}{\delta}$.

(c) The data frame `follicle` contains data on the number of ovarian follicles counted from sectioned ovaries of women of various ages. It has two columns.

`age`	age of the women
`log.count`	logarithm of the follicle count

(i) Fit a B-spline model with a suitable number of knots to the `follicle` data.
Hint: You can use the function `bbase and the code from the first practical session.`

(ii) Suppose we are interested in the rate by which the number of follicles is reducing. We can estimate this rate by computing the derivative of the fitted regression function. Use the formula from part (b) to compute the derivative and plot it.

(iii) Can you construct a confidence interval for the estimated derivative? From what age onward is there a significant decrease in the number of follicles

(d) Suppose you wanted to use a truncated power basis instead of a B-spline basis. How would you estimate the derivative?

(e) Suppose you wanted to use a local estimate like the one studied in question 2. How would you estimate the derivative?

Question 7 (Monotonic smoothing). Consider again the radio-carbon example in which tried to relate the observed radio-carbon age to the calibrated age. It seems natural to impose the constraint that the function describing the relationship between the two is non-decreasing. In this question you will learn how this can be achieved by using equally-spaced B-splines.

(a) Explain why the estimated regression function $\hat{m}(x) = \sum_{j=1}^{l+r-1} B_j(x)\beta_j$ is non-decreasing if $\beta_j \leq \beta_{j+1}$ for all j.
Hint: Use the derivative formula from question 6(b).

(b) This fact can be exploited to construct a monotonic regression function. "All" we need to do is to introduce the additional constraint that $\beta_j \leq \beta_{j+1}$ for all j. These additional constraints however make finding $\hat{\boldsymbol{\beta}}$ much more difficult: we have to resort to quadratic programming methods[1]. We will use a simpler approach, based on modifying the penalty in a P-splines approach.[2] When using first-order differences to construct the penalty we use

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

i.e. the penalty becomes

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

In order to penalise lack of monotonicity we only want to penalise differences between the β_j if $\beta_j > \beta_{j+1}$, i.e. we would like to use the penalty

$$\sum_{j:\ \beta_j > \beta_{j+1}} (\beta_{j+1} - \beta_j)^2$$

This corresponds to modifying the matrix, setting all rows to zero that correspond to pairs with $\beta_j \leq \beta_{j+1}$.

Now there is of course the problem that we need the differencing matrix to estimate $\hat{\boldsymbol{\beta}}$, but it, in turn, depends on $\boldsymbol{\beta}$. The way around this problem is to simply iterate between these two steps, which gives the following algorithm.

1. Set $\mathbf{D}^{(1)} = \begin{pmatrix} 0 & \dots & 0 \\ \vdots & \ddots & 0 \\ 0 & \dots & 0 \end{pmatrix}$.

2. For $h = 1, 2, \dots$ until convergence ...

   i. Compute $\boldsymbol{\beta}^{(h)} = (\mathbf{B}^\top\mathbf{B} + \lambda\mathbf{D}^{(h)\top}\mathbf{D}^{(h)})^{-1}\mathbf{B}^\top\mathbf{y}$ (or use a QR decomposition to compute $\boldsymbol{\beta}^{(h)}$).

   ii. Set $\delta_j^{(h)} = \begin{cases} 1 & \text{if } \beta_j^{(h)} > \beta_{j+1}^{(h)} \\ 0 & \text{otherwise.} \end{cases}$

[1] These are for example implemented in the R package `quadprog`.

[2] This idea was first suggested by Bollaerts *et al.* (British Journal of Mathematical and Statistical Psychology (2006), 59, 451–469). Bollaerts *et al.* use a monotonicity penalty in conjunction with a smoothness penalty, but for simplicity we will omit the smoothness penalty and control the smoothness by choosing a small enough number of basis functions.

iii. Set $\mathbf{D}^{(h+1)} = \begin{pmatrix} \delta_1^{(h)} & -\delta_1^{(h)} & \dots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & \delta_{l+r-2}^{(h)} & -\delta_{l+r-2}^{(h)} \end{pmatrix}$.

We will now turn to the radiocarbon data.

(i) Fit a "classical" B-spline model with 25 equally spaced knots to the data.
Hint: You can use the function `bbase` *to create the matrix* $\mathbf{B}$.

(ii) Implement the above algorithm to estimate a non-decreasing regression function modelling the relationship between radio-carbon age and calibrated age. Use the same basis function as in part (i).

Compare the results to the B-spline model fitted in part (i).

(c) In this part we will return to the follicle data from question 6. All (primordial) follicles are developed before birth, so, after birth, the number of follicles must decrease over time.

When using a moderate number of knots (say 10) for modelling the log-follicle counts the estimated regression function is oscillating.

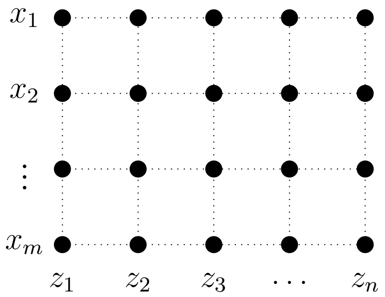
How can you use the approach set out above to estimate a *non-increasing* regression function?

Question 8 (SO$_2$ concentrations). The data frame `SO2station` contains measurements of SO$_2$ from the air at a sampling station in mainland Europe. The variables are:

`logSO2`	SO2 measurement on a log scale
`Year`	year, with a fractional component to reflect week within year
`Week`	week within the year
`Rain`	rainfall
`Temp`	temperature
`Humidity`	air humidity
`Flow`	a measurements of air flow

1. Explore the relationships between the variables, and specifically between SO$_2$ and potential explanatory variables, by any graphical means you consider suitable.
2. Fit an additive model and refine this into a model which you believe gives a good description of the data. (At this stage, do not worry about temporal correlation in the data.)
3. Consider a model which uses only `Year` and `Week` as explanatory variables. This model is of interest because meteorological information is not always easy to obtain, so an understanding of whether it is needed at this station may help in decisions on whether to collect it at others. Compare this reduced model with one which makes use of the meteorological information. In the comparison, interest lies particularly in any effect on the estimate of trend in SO$_2$ over the years.
4. Consider the additive model which contains only `Year` and `Week` as explanatory variables. Examine the residuals from this model for evidence of serial correlation. How would you adjust your model to account for this? Even if you don't do that, can you say what the effects of a suitable adjustment would be?

Question 9 (Marginal smoothing using local smoothers). Consider a bivariate smoothing problem in which the data has been collected on a regular grid, i.e. the response was observed at each combination (x_i, z_j) $(i = 1, \dots, m, j = 1, \dots, n)$. The figure below illustrates this setup.



In this case it is easiest to write the observed response as a matrix $\mathbf{Y} = (y_{ij})$ with y_{ij} being the response associated with (x_i, z_j). Alternatively we can write the observed response as a long vector, stacking the columns of $\mathbf{Y}$ on top of each other, i.e.

$$\mathbf{y} = (y_{11}, \dots, y_{m1}, y_{12}, \dots, y_{m2}, \dots, y_{mn})^\top .$$

Suppose we wish to use a bivariate local smoother, i.e.

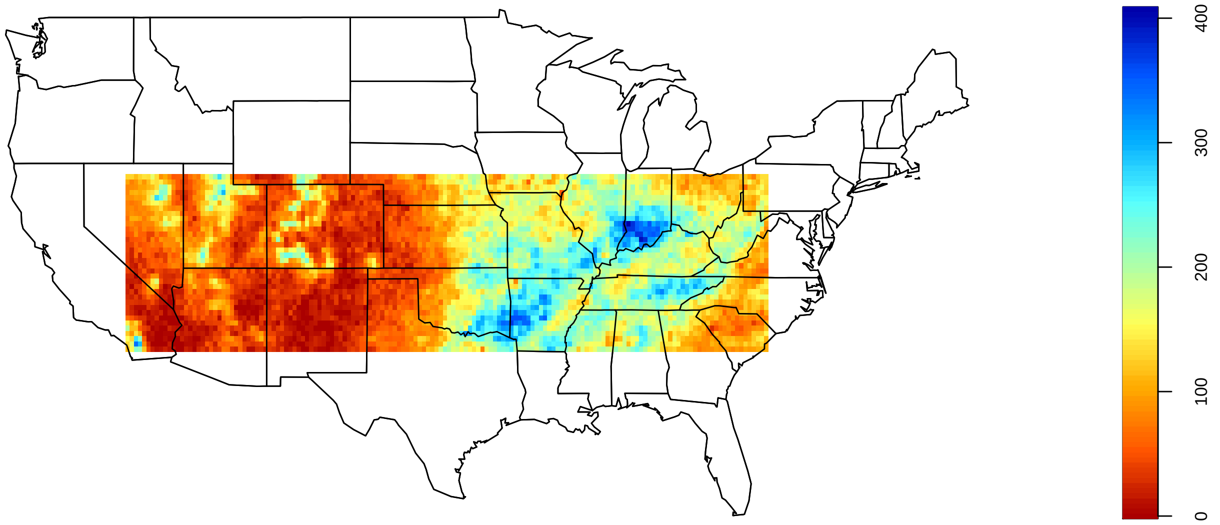
$$\hat{y}_{ij} = \hat{m}(x_i, z_j) = \frac{\sum_{k=1}^m \sum_{l=1}^n w(x_k - x_i; h) w(z_l - z_j; h) y_{kl}}{\sum_{k=1}^m \sum_{l=1}^n w(x_k - x_i; h) w(z_l - z_j; h)}.$$

(a) Just like in question 2 we can write $\hat{\mathbf{y}} = \mathbf{S}\mathbf{y}$. Write down one entry of $\mathbf{S}$.

(b) Show that $\mathbf{S} = \mathbf{S}^{(2)} \otimes \mathbf{S}^{(1)}$ where $\mathbf{S}^{(2)}$ is the smoothing matrix associated with a univariate smoothing problem with observed covariate values $z_1, \dots, z_n$ and $\mathbf{S}^{(1)}$ is the smoothing matrix associated with a univariate smoothing problem with observed covariate values $x_1, \dots, x_m$.
Hint: Important properties of the Kronecker product ("$\otimes$") are summarised on page 8.

(c) Using the properties of the Kronecker product show that we can also write $\hat{\mathbf{Y}} = \mathbf{S}^{(1)}\mathbf{Y}\mathbf{S}^{(2)\top}$. What is the advantage of this representation?
Hint: Think about the dimensions of the matrices involved in the calculation.

(d) The data frame `us.rain` contains noisy observations of the total rainfall in March/April 2006 in a rectangular area covering most of the central US (see figure below). The vectors `us.northing` and `us.easting` contain the corresponding latitudes and longitudes. Construct the smoothing matrices $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$ and use the formula derived in part (c) to construct the fitted values. Use an R function like `image` to plot the resulting smoothed surface.



Question 10 (Marginal smoothing using B-splines and P-splines). Consider again the bivariate smoothing problem with gridded data set out in question 8. In this question we will consider the tensor-product-based spline approach, set out in section 3.4.1 of the notes.

(a) Explain that we can write the design matrix

$$\mathbf{B} = \begin{pmatrix} B_{11}(x_1,z_1) & \dots & B_{l_1+r-1,1}(x_1,z_1) & B_{12}(x_1,z_1) & \dots & B_{l_1+r-1,2}(x_1,z_1) & \dots & B_{l_1+r-1,l+2+r-1}(x_1,z_1) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ B_{11}(x_m,z_1) & \dots & B_{l_1+r-1,1}(x_m,z_1) & B_{12}(x_m,z_1) & \dots & B_{l_1+r-1,2}(x_m,z_1) & \dots & B_{l_1+r-1,l+2+r-1}(x_m,z_1) \\ B_{11}(x_1,z_2) & \dots & B_{l_1+r-1,1}(x_1,z_2) & B_{12}(x_1,z_2) & \dots & B_{l_1+r-1,2}(x_1,z_2) & \dots & B_{l_1+r-1,l+2+r-1}(x_1,z_2) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ B_{11}(x_m,z_n) & \dots & B_{l_1+r-1,1}(x_m,z_n) & B_{12}(x_m,z_n) & \dots & B_{l_1+r-1,2}(x_m,z_n) & \dots & B_{l_1+r-1,l+2+r-1}(x_m,z_n) \end{pmatrix}$$

of the bivariate tensor-product-splines as $\mathbf{B} = \mathbf{B}^{(2)} \otimes \mathbf{B}^{(1)}$, where $\mathbf{B}^{(1)}$ is a univariate B-spline basis on the x_i, $\mathbf{B}^{(2)}$ is a univariate B-spline basis on the z_j, and $B_{i,j}(x,z) = B_i^{(1)}(x)B_j^{(2)}(z)$.

(b) Suppose **S** is the bivariate smoothing matrix, i.e.

$$\mathbf{S} = \mathbf{B}(\mathbf{B}^\top\mathbf{B})^{-1}\mathbf{B}^\top$$

and $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$ are the corresponding univariate smoothing matrices, i.e.

$$\mathbf{S}^{(1)} = \mathbf{B}^{(1)}(\mathbf{B}^{(1)\top}\mathbf{B}^{(1)})^{-1}\mathbf{B}^{(1)\top} \qquad \mathbf{S}^{(2)} = \mathbf{B}^{(2)}(\mathbf{B}^{(2)\top}\mathbf{B}^{(2)})^{-1}\mathbf{B}^{(2)\top}.$$

Show that $\mathbf{S} = \mathbf{S}^{(2)} \otimes \mathbf{S}^{(1)}$ and that we can write $\hat{\mathbf{y}} = \mathbf{S}\mathbf{y}$ as $\hat{\mathbf{Y}} = \mathbf{S}^{(1)}\mathbf{Y}\mathbf{S}^{(2)\top}$, just like in question 8.

(c) Is the above formula also valid if P-splines are used?

(d) Use the formula derived in part (c) to analyse the US rainfall data from question 8(d).

The Kronecker Product

Given a $m \times n$ matrix $\mathbf{C}$ and a $p \times q$ matrix $\mathbf{D}$ the *Kronecker product* of the two matrices is defined as the following $mp \times nq$ matrix:

$$\mathbf{C}\otimes\mathbf{D} = \begin{bmatrix} c_{11}\mathbf{D} & \cdots & c_{1n}\mathbf{D} \\ \vdots & \ddots & \vdots \\ c_{m1}\mathbf{D} & \cdots & c_{mn}\mathbf{D} \end{bmatrix} = \begin{bmatrix} c_{11}d_{11} & c_{11}d_{12} & \cdots & c_{11}d_{1q} & \cdots & c_{1n}d_{11} & c_{1n}d_{12} & \cdots & c_{1n}d_{1q} & \\ c_{11}d_{21} & c_{11}d_{22} & \cdots & c_{11}d_{2q} & \cdots & c_{1n}d_{21} & c_{1n}d_{22} & \cdots & c_{1n}d_{2q} & \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ c_{11}d_{p1} & c_{11}d_{p2} & \cdots & c_{11}d_{pq} & \cdots & c_{1n}d_{p1} & c_{1n}d_{p2} & \cdots & c_{1n}d_{pq} & \\ \vdots & \vdots & & \vdots & \ddots & \vdots & \vdots & & \vdots & \\ c_{m1}d_{11} & c_{m1}d_{12} & \cdots & c_{m1}d_{1q} & \cdots & c_{mn}d_{11} & c_{mn}d_{12} & \cdots & c_{mn}d_{1q} & \\ c_{m1}d_{21} & c_{m1}d_{22} & \cdots & c_{m1}d_{2q} & \cdots & c_{mn}d_{21} & c_{mn}d_{22} & \cdots & c_{mn}d_{2q} & \\ \vdots & \vdots & \ddots & \vdots & & \vdots & \vdots & \ddots & \vdots & \\ c_{m1}d_{p1} & c_{m1}d_{p2} & \cdots & c_{m1}d_{pq} & \cdots & c_{mn}d_{p1} & c_{mn}d_{p2} & \cdots & c_{mn}d_{pq} & \end{bmatrix}$$

For this assignment you will need the following properties of the Kronecker product.

- The Kronecker product is, just like the standard matrix product, *not* commutative, i.e. $\mathbf{C}\otimes\mathbf{D} \neq \mathbf{D}\otimes\mathbf{C}$. It is however associative, i.e. $(\mathbf{C}\otimes\mathbf{D})\otimes\mathbf{E} = \mathbf{C}\otimes(\mathbf{D}\otimes\mathbf{E})$.
- $\mathbf{C}\otimes(\mathbf{D}+\mathbf{E}) = \mathbf{C}\otimes\mathbf{D} + \mathbf{C}\otimes\mathbf{E}$ and $(\mathbf{C}+\mathbf{D})\otimes\mathbf{E} = \mathbf{C}\otimes\mathbf{E} + \mathbf{D}\otimes\mathbf{E}$.
- $(\mathbf{C}\otimes\mathbf{D})(\mathbf{E}\otimes\mathbf{F}) = (\mathbf{CE})\otimes(\mathbf{DF})$.
- $(\mathbf{C}\otimes\mathbf{D})^{-1} = \mathbf{C}^{-1}\otimes\mathbf{D}^{-1}$.
- $(\mathbf{C}\otimes\mathbf{D})^\top = \mathbf{C}^\top\otimes\mathbf{D}^\top$.
- $(\mathbf{A}\otimes\mathbf{B})\mathbf{c} = \mathbf{d}$ if and only if $\mathbf{BCA}^\top = \mathbf{D}$, where the vectors $\mathbf{c}$ (and $\mathbf{d}$) simply consist of the columns of $\mathbf{C}$ (and $\mathbf{D}$) stacked on top of each other, i.e.

  $$\mathbf{C} = \begin{pmatrix} c_{11} & \dots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{m1} & \dots & c_{mn} \end{pmatrix} \qquad \mathbf{c} = (c_{11},\dots,c_{m1},c_{12},\dots,c_{m2},\dots,c_{mn})^\top$$

  $$\mathbf{D} = \begin{pmatrix} d_{11} & \dots & d_{1q} \\ \vdots & \ddots & \vdots \\ d_{p1} & \dots & d_{pq} \end{pmatrix} \qquad \mathbf{d} = (d_{11},\dots,d_{p1},d_{12},\dots,d_{p2},\dots,d_{pq})^\top$$

  $\mathbf{A}$ is a $q \times n$ matrix and $\mathbf{B}$ is a $p \times m$ matrix, thus $\mathbf{C}$ is a $m \times n$ matrix and $\mathbf{D}$ is a $p \times q$ matrix.

  In R this can be verified as follows.

```r
# Create example matrices
A <- matrix(rnorm(6), ncol=3)
B <- matrix(rnorm(8), ncol=2)
C <- matrix(rnorm(6), ncol=3)
c <- as.vector(C)
# Calculate d
kronecker(A,B)%*%c
# Calculate D
B%*%C%*%t(A)
```

Post-course assessment — Model answers

1. Mean: $\int \hat{f}(y) y dy = \frac{1}{n}\sum_{i=1}^n \int y \frac{1}{h} w\left(\frac{y-y_i}{h}\right) dy = \frac{1}{n}\sum_{i=1}^n y_i = \bar{y}$

   $\int \hat{f}(y) y^2 dy = \frac{1}{n}\sum_{i=1}^n \int y^2 \frac{1}{h} w\left(\frac{y-y_i}{h}\right) dy = \frac{1}{n}\sum_{i=1}^n (h^2 + y_i^2)$

   Variance: $h^2 + \frac{1}{n}\sum_{i=1}^n y_i^2 - \bar{y}^2 = h^2 + \frac{1}{n}\sum_{i=1}^n (y_i - \bar{y})^2$

   So, the mean of the distribution is the sample mean of the data, while the variance is the sample variance (using the divisor n) inflated by h^2. This 'variance inflation' quantifies the effect of smoothing.

2. (a) $\mathbf{S} = (s_{ij})$ with $s_{ij} = \frac{w(x_i - x_j; h)}{\sum_{k=1}^n w(x_k - x_i; h)}$.

   (b) *(parts (i) to (iv))*

```
# Simply call covariate x and response y
x <- radiocarbon$cal.age
y <- radiocarbon$rc.age

# Set bandwitdh
h <- 0.05

# Compute the smoothing matrix S
S <- dnorm(outer(x, x, ”–”),sd=h)
S <- S / rowSums(S)

# Plot data
plot(x, y, xlab=”Calibrated␣age”, ylab=”Radiocarbon␣age”)

# Compute fitted values and add fitted line
fitted <- S%*%y
lines(x, fitted)

# Effective degrees of freedom
sum(diag(S))
```

   Increasing h reduces the effective degrees of freedom and makes the regression function less flexible.

3. (a) We have that

$$
\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{\beta}} (\mathbf{y} - \mathbf{B}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{B}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^\top \mathbf{D}^\top \mathbf{D}\boldsymbol{\beta} \\
&= \frac{\partial}{\partial \boldsymbol{\beta}} \mathbf{y}^\top \mathbf{y} - 2\boldsymbol{\beta}^\top \mathbf{B}^\top \mathbf{y} + \boldsymbol{\beta}^\top \mathbf{B}^\top \mathbf{B}\boldsymbol{\beta} + \lambda \boldsymbol{\beta}^\top \mathbf{D}^\top \mathbf{D}\boldsymbol{\beta} \\
&= -2\mathbf{B}^\top \mathbf{y} + \mathbf{B}^\top \mathbf{B}\boldsymbol{\beta} + \lambda \mathbf{D}^\top \mathbf{D}\boldsymbol{\beta}
\end{aligned}
$$

   Setting this derivative to zero yields the equation

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

   which has

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

   as solution.

(b) Using the formula that $\begin{pmatrix}\mathbf{A}\\ \mathbf{B}\end{pmatrix}^\top \begin{pmatrix}\mathbf{C}\\ \mathbf{D}\end{pmatrix} = \mathbf{A}^\top\mathbf{C} + \mathbf{B}^\top\mathbf{D}$ we have that

$$\begin{aligned}
&(\mathbf{y} - \mathbf{B}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{B}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^\top\mathbf{D}^\top\mathbf{D}\boldsymbol{\beta}\\
&= (\mathbf{y} - \mathbf{B}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{B}\boldsymbol{\beta}) + (\mathbf{0} - \sqrt{\lambda}\mathbf{D}\boldsymbol{\beta})^\top(\mathbf{0} - \sqrt{\lambda}\mathbf{D}\boldsymbol{\beta})\\
&= \left(\begin{pmatrix}\mathbf{y}\\ \mathbf{0}\end{pmatrix} - \begin{pmatrix}\mathbf{B}\\ \sqrt{\lambda}\mathbf{D}\end{pmatrix}\boldsymbol{\beta}\right)^\top \left(\begin{pmatrix}\mathbf{y}\\ \mathbf{0}\end{pmatrix} - \begin{pmatrix}\mathbf{B}\\ \sqrt{\lambda}\mathbf{D}\end{pmatrix}\boldsymbol{\beta}\right)\\
&= \left\|\begin{pmatrix}\mathbf{y}\\ \mathbf{0}\end{pmatrix} - \begin{pmatrix}\mathbf{B}\\ \sqrt{\lambda}\mathbf{D}\end{pmatrix}\boldsymbol{\beta}\right\|^2
\end{aligned}$$

(c) The QR decomposition of $\tilde{\mathbf{B}} = \begin{pmatrix}\mathbf{B}\\ \sqrt{\lambda}\mathbf{D}\end{pmatrix}$ is $\tilde{\mathbf{B}} = \begin{pmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{pmatrix}\begin{pmatrix}\mathbf{R}\\ \mathbf{0}\end{pmatrix} = \mathbf{Q}_1\mathbf{R}$. $\mathbf{Q} = \begin{pmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{pmatrix}$ is an orthogonal matrix and $\mathbf{R}$ is upper triangular. With $\tilde{\mathbf{y}} = \begin{pmatrix}\mathbf{y}\\ \mathbf{0}\end{pmatrix}$ we have that

$$\begin{aligned}
\|\tilde{\mathbf{y}} - \tilde{\mathbf{B}}\boldsymbol{\beta}\|^2 = \|\tilde{\mathbf{y}} - \mathbf{Q}_1\mathbf{R}\boldsymbol{\beta}\|^2 &= \|\mathbf{Q}^\top(\tilde{\mathbf{y}} - \mathbf{Q}_1\mathbf{R}\boldsymbol{\beta}\|^2\\
&= \left\|\begin{pmatrix}\mathbf{Q}_1^\top\tilde{\mathbf{y}}\\ \mathbf{Q}_2^\top\tilde{\mathbf{y}}\end{pmatrix} - \begin{pmatrix}\mathbf{R}\boldsymbol{\beta}\\ 0\end{pmatrix}\right\|^2 = \|\mathbf{Q}_1^\top\mathbf{y} - \mathbf{R}\boldsymbol{\beta}\|^2 + \|\mathbf{Q}_2^\top\mathbf{y}\|^2
\end{aligned}$$

We have used that $\mathbf{Q}_1^\top\mathbf{Q}_1 = \mathbf{I}$ and $\mathbf{Q}_1^\top\mathbf{Q}_2 = \mathbf{0}$.

$\|\mathbf{Q}_2^\top\mathbf{y}\|^2$ does not depend on $\boldsymbol{\beta}$ and $\|\mathbf{Q}_1^\top\mathbf{y} - \mathbf{R}\boldsymbol{\beta}\|^2$ can (if $\tilde{\mathbf{B}}$ is of full column rank) be made exactly zero by solving $\mathbf{R}\boldsymbol{\beta} = \mathbf{Q}_1^\top\tilde{\mathbf{y}}$, which can, due to $\mathbf{R}$ being upper-triagonal, be performed very efficiently.

4. The following code should be sufficient.

```r
clyde.subset <- subset(clyde, Station == 10)
plot(clyde.subset)

library(mgcv)
model1 <- gam(DO ~ s(Doy, Year) + s(Salinity), data = clyde.subset)
model0 <- gam(DO ~ s(Doy) + s(Year) + s(Salinity), data = clyde.subset)
par(mfrow = c(2, 2))
plot(model1)
par(mfrow = c(1, 1))
anova(model0, model1, test = ”F”)
```

This shows some evidence of change in the seasonal effect over the years.

5. (a) The proof is by induction.

$\underline{r = 0}$: For $r = 0$ the formula comes down to

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

which is exactly how we have defined a B-spline basis of degree $r = 0$.

$\underline{r - 1 \rightsquigarrow r}$: Suppose we know that $B_j^{r-1}(x) = \sum_{i=0}^{r} \frac{\Delta_i^r}{(r-1)!\delta^{r-1}}(x - \kappa_{j-r+1+i})_+^{r-1}$ Using the recursive definition of B-splines of higher order we have that

$$\begin{aligned}
B_j^r(x) &= \frac{x-\kappa_{j-r}}{\kappa_j-\kappa_{j-r}} B_{j-1}^{r-1}(x) + \frac{\kappa_{j+1}-x}{\kappa_{j+1}-\kappa_{j+1-r}} B_j^{r-1}(x) \\
&= \frac{x-\kappa_{j-r}}{r\cdot\delta} \sum_{i=0}^{r} \frac{\Delta_i^r}{(r-1)!\delta^{r-1}} (x-\kappa_{j-r+i})_+^{r-1} - \frac{x-\kappa_{j+1}}{r\cdot\delta} \sum_{i=0}^{r} \frac{\Delta_i^r}{(r-1)!\delta^{r-1}} (x-\kappa_{j-r+1+i})_+^{r-1} \\
&= \frac{1}{r!\cdot\delta^r} \sum_{i=0}^{r} \Delta_i^r (x-\kappa_{j-r+i})_+^{r-1} \underbrace{(x-\kappa_{j-r})}_{=x-\kappa_{j-r+i}-i\delta} - \frac{1}{r!\cdot\delta^r} \sum_{i=0}^{r} \Delta_i^r (x-\kappa_{j-r+1+i})_+^{r-1} \underbrace{(x-\kappa_{j+1})}_{=x-\kappa_{j-r+1+i}+(r-i)\cdot\delta} \\
&= \frac{1}{r!\cdot\delta^r} \sum_{i=0}^{r} \Delta_i^r (x-\kappa_{j-r+i})_+^{r} - \frac{\delta}{r!\cdot\delta^r} \sum_{i=0}^{r} i\Delta_i^r (x-\kappa_{j-r+i})_+^{r-1} \\
&\qquad - \frac{1}{r!\cdot\delta^r} \underbrace{\sum_{i=0}^{r} \Delta_i^r (x-\kappa_{j-r+1+i})_+^{r}}_{=\sum_{i=1}^{r+1} \Delta_{i-1}^r (x-\kappa_{j-r+i})_+^r} - \frac{\delta}{r!\cdot\delta^r} \underbrace{\sum_{i=0}^{r} (r-i)\Delta_i^r (x-\kappa_{j-r+1+i})_+^{r-1}}_{=\sum_{i=1}^{r+1}(r-i+1)\Delta_{i-1}^r (x-\kappa_{j-r+i})_+^{r-1}} \\
&= \frac{1}{r!\cdot\delta^r} \sum_{i=0}^{r+1} \underbrace{(\Delta_i^r-\Delta_{i-1}^r)}_{=\Delta_i^{r+1}} (x-\kappa_{j-r+i})_+^r - \frac{\delta}{r!\cdot\delta^r} \sum_{i=1}^{r} \underbrace{(i\Delta_i^r-(r-i+1)\Delta_{i-1}^r)}_{=0} (x-\kappa_{j-r+i})_+^{r-1} \\
&= \sum_{i=0}^{r+1} \frac{\Delta_i^{r+1}}{r!\delta^r} (x-\kappa_{j-r+i})_+^r
\end{aligned}$$

(b) We create a covariate vector with values ranging from 0 to 1.

```
x <- seq(0, 1, len=100)
n.knots <- 6
delta <- (max(x)-min(x))/(n.knots-1)
```

Degree $r = 0$:

```
# Create knots
knots <- seq(min(x), max(x), delta)
# Create truncated polynomials
B <- outer(x, knots, function(x,y) pmax(x-y,0))>0
# Create B-spline matrix
B <- B[,-ncol(B)] - B[,-1]
# Plot basis functions
matplot(x, B, type=”l”)
```

Degree $r = 1$:

```
# Create knots
knots <- seq(min(x)-delta, max(x)+delta, delta)
# Create truncated polynomials
B <- outer(x, knots, function(x,y) pmax(x-y,0))
# Create B-spline matrix
B <- (B[,1:(ncol(B)-2)] - 2*B[,2:(ncol(B)-1)] + B[, 3:ncol(B)])/delta
# Plot basis functions
matplot(x, B, type=”l”)
```

Degree $r = 2$:

```
# Create knots
knots <- seq(min(x)-2*delta, max(x)+2*delta, delta)
```

```
# Create truncated polynomials
B <- outer(x, knots, function(x,y) pmax(x-y,0))^2
# Create B-spline matrix
B <- (B[,1:(ncol(B)-3)] - 3*B[,2:(ncol(B)-2)] + 3*B[,3:(ncol(B)-1)] - B[, 4:ncol(B)])
      / (2*delta^2)
# Plot basis functions
matplot(x, B, type=”l”)
```

Any degree r:

```
# Set example degree
r <- 3
# Create knots
knots <- seq(min(x)-r*delta, max(x)+r*delta, delta)
# Create truncated polynomials
B <- outer(x, knots, function(x,y) pmax(x-y,0))^r
# Create B-spline matrix
B <- B%*%t(diff(diag(ncol(B)), diff=r+1))/(delta^r*factorial(r))
# Plot basis functions
matplot(x, B, type=”l”)
```

6. (a) The proof is by induction.

$r = 1$: For $r = 1$ we have that

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

thus

$$\begin{aligned} \frac{\partial}{\partial x} B_j^r(x) &= \begin{cases} \frac{1}{\kappa_j-\kappa_{j-1}} & \text{for } \kappa_{j-1} \leq x < \kappa_j \\ \frac{-1}{\kappa_{j+1}-\kappa_j} & \text{for } \kappa_j \leq x < \kappa_{j+1} \\ 0 & \text{otherwise,} \end{cases} \\ &= \frac{1}{\kappa_j - \kappa_{j-1}} B_{j-1}^{r-1}(x) - \frac{1}{\kappa_{j+1} - \kappa_j} B_j^{r-1}(x) \end{aligned}$$

$r - 1 \rightsquigarrow r$: Suppose we know that

$$\frac{\partial}{\partial x} B_j^{r-1}(x) = -\frac{r-1}{\kappa_j - \kappa_{j-r+1}} B_{j-1}^{r-2}(x) + \frac{r-1}{\kappa_{j+1} - \kappa_{j-r+2}} B_j^{r-2}(x).$$

We will take the derivative of $B_j^r(x)$.

$$\begin{aligned} \frac{\partial}{\partial x} B_j^r(x) = {} & \frac{1}{\kappa_j - \kappa_{j-r}} B_{j-1}^{r-1}(x) + \frac{x - \kappa_{j-r}}{\kappa_j - \kappa_{j-r}} \frac{\partial}{\partial x} B_{j-1}^{r-1}(x) \\ & - \frac{1}{\kappa_{j+1} - \kappa_{j+1-r}} B_j^{r-1}(x) + \frac{\kappa_{j+1} - x}{\kappa_{j+1} - \kappa_{j+1-r}} \frac{\partial}{\partial x} B_j^{r-1}(x) \end{aligned} \tag{1}$$

We will now rewrite the right-most term in each of the last two lines of the above equation.

$$\begin{aligned}
\frac{x-\kappa_{j-r}}{\kappa_j-\kappa_{j-r}}\frac{\partial}{\partial x}B_{j-1}^{r-1}(x) &= \frac{x-\kappa_{j-r}}{\kappa_j-\kappa_{j-r}}\left(\frac{r-1}{\kappa_{j-1}-\kappa_{j-r}}B_{j-2}^{r-2}(x)-\frac{r-1}{\kappa_j-\kappa_{j+1-r}}B_{j-1}^{r-2}(x)\right)\\
&= \frac{r-1}{\kappa_j-\kappa_{j-r}}\left(\underbrace{\frac{x-\kappa_{j-r}}{\kappa_{j-1}-\kappa_{j-r}}B_{j-2}^{r-2}(x)+\frac{\kappa_j-x}{\kappa_j-\kappa_{j+1-r}}B_{j-1}^{r-2}(x)}_{=-B_{j-1}^{r-1}(x)}\right.\\
&\qquad\left.\underbrace{-\frac{\kappa_j-x}{\kappa_j-\kappa_{j+1-r}}B_{j-1}^{r-2}(x)-\frac{x-\kappa_{j-r}}{\kappa_j-\kappa_{j+1-r}}B_{j-1}^{r-2}(x)}_{=-\frac{\kappa_j-\kappa_{j-r}}{\kappa_j-\kappa_{j+1-r}}B_{j-1}^{r-2}(x)}\right)\\
&= \frac{r-1}{\kappa_j-\kappa_{j-r}}B_{j-1}^{r-1}(x)-\frac{r-1}{\kappa_j-\kappa_{j+1-r}}B_{j-1}^{r-2}(x)
\end{aligned}$$

$$\begin{aligned}
\frac{\kappa_{j+1}-x}{\kappa_{j+1}-\kappa_{j+1-r}}\frac{\partial}{\partial x}B_{j-1}^{r-1}(x) &= \frac{\kappa_{j+1}-x}{\kappa_{j+1}-\kappa_{j+1-r}}\left(\frac{r-1}{\kappa_j-\kappa_{j+1-r}}B_{j-1}^{r-2}(x)-\frac{r-1}{\kappa_{j+1}-\kappa_{j+2-r}}B_j^{r-2}(x)\right)\\
&= \frac{r-1}{\kappa_{j+1}-\kappa_{j+1-r}}\left(\underbrace{\frac{\kappa_{j+1}-x}{\kappa_j-\kappa_{j+1-r}}B_{j-1}^{r-2}(x)+\frac{x-\kappa_{j-r+1}}{\kappa_j-\kappa_{j-r+1}}B_{j-1}^{r-2}(x)}_{=\frac{\kappa_{j+1}-\kappa_{j-r+1}}{\kappa_j-\kappa_{j-r+1}}B_{j-1}^{r-2}(x)}\right.\\
&\qquad\left.\underbrace{-\frac{x-\kappa_{j-r+1}}{\kappa_j-\kappa_{j-r+1}}B_{j-1}^{r-2}(x)-\frac{\kappa_{j+1}-x}{\kappa_{j+1}-\kappa_{j+2-r}}B_j^{r-2}(x)}_{=-B_j^{r-1}(x)}\right)\\
&= \frac{r-1}{\kappa_j-\kappa_{j-r+1}}B_{j-1}^{r-2}(x)-\frac{r-1}{\kappa_{j+1}-\kappa_{j+1-r}}B_j^{r-1}(x)
\end{aligned}$$

In both cases the second and third term inserted in the second step cancel each other out (second term is minus the third term). Plugging these two results into equation (1) yields

$$\begin{aligned}
\frac{\partial}{\partial x}B_j^r(x) &= \frac{1}{\kappa_j-\kappa_{j-r}}B_{j-1}^{r-1}(x)+\frac{r-1}{\kappa_j-\kappa_{j-r}}B_{j-1}^{r-1}(x)-\frac{r-1}{\kappa_j-\kappa_{j+1-r}}B_{j-1}^{r-2}(x)\\
&\quad-\frac{1}{\kappa_{j+1}-\kappa_{j+1-r}}B_j^{r-1}(x)+\frac{r-1}{\kappa_j-\kappa_{j-r+1}}B_{j-1}^{r-2}(x)-\frac{r-1}{\kappa_{j+1}-\kappa_{j+1-r}}B_j^{r-1}(x)\\
&= \frac{r}{\kappa_j-\kappa_{j-r}}B_{j-1}^{r-1}(x)-\frac{r}{\kappa_{j+1}-\kappa_{j+1-r}}B_j^{r-1}(x)
\end{aligned}$$

The proof in the special case of equally-spaced knots is much simpler and does not need to be performed using induction.

$$\begin{aligned}
\frac{\partial}{\partial x} B_j^r(x) &= \frac{\partial}{\partial x} \sum_{i=0}^{r+1} \frac{\Delta_i^{r+1}}{r!\delta^r} (x-\kappa_{j-r+i})_+^r \\
&= \sum_{i=0}^{r+1} \frac{\overbrace{\Delta_i^{r+1}}^{\Delta_i^r - \Delta_{i-1}^r}}{(r-1)!\delta^r} (x-\kappa_{j-r+i})_+^{r-1} \\
&= \frac{1}{\delta} \sum_{i=0}^{r} \frac{\Delta_i^r}{(r-1)!\delta^{r-1}} (x-\kappa_{j-r+i})_+^{r-1} - \frac{1}{\delta} \sum_{i=1}^{r+1} \frac{\Delta_{i-1}^r}{(r-1)!\delta^{r-1}} (x-\kappa_{j-r+i})_+^{r-1} \\
&= \frac{1}{\delta} \underbrace{\sum_{i=0}^{r} \frac{\Delta_i^r}{(r-1)!\delta^{r-1}} (x-\kappa_{j-1-r+1+i})_+^{r-1}}_{=B_{j-1}^{r-1}(x)} - \frac{1}{\delta} \underbrace{\sum_{i=0}^{r} \frac{\Delta_i^r}{(r-1)!\delta^{r-1}} (x-\kappa_{j-r+1+i})_+^{r-1}}_{=B_j^{r-1}(x)}
\end{aligned}$$

(b) Using the derivatives of the basis functions we have just derived

$$\begin{aligned}
\frac{\partial}{\partial x} m(x) &= \sum_{j=1}^{l+r-1} \frac{\partial}{\partial x} B_j^r(x) \beta_j \\
&= \sum_{j=1}^{l+r-1} (B_{j-1}^{r-1}(x) - B_j^{r-1}(x)) \frac{\beta_j}{\delta} \\
&= \sum_{j=1}^{l+r-2} B_j^{r-1}(x) \frac{\beta_{j+1} - \beta_j}{\delta}
\end{aligned}$$

In the last step we have exploited that within the range of the data $B_0^{r-1}(x) = B_r^{l+r-1}(x) = 0$.

(c) *(parts (i) to (iii))*

```
# Set x to covariate and y to response
x <- follicle$age
y <- follicle$log.count

# Create basis functions
B <- bbase(x, deg=3, n.knots=4)
# Estimate coefficients
beta <- qr.coef(qr(B), y)

# Plot fitted function
plot(x, y, xlab=”Age”, ylab=”log(Count)”)
xx<-seq(min(x), max(x), len=1e3)
BB <- bbase(xx, deg=3, n.knots=4)
lines(xx, BB%*%beta)

# Estimated derivative
BBd <-bbase(xx, deg=2, n.knots=4)
delta <- (max(x)-min(x))/(4-1)
gamma <- diff(beta)/delta
deriv <- BBd%*%gamma
plot(xx, deriv, type=”l”, xlab=”Age”, ylab=”Derivative”)
```

The estimated derivative is highly sensitive to the choice of number of basis functions.

Just like in the classical linear model we have that

$$\text{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2 * (\mathbf{B}'\mathbf{B})^{-1}$$

The estimated derivative is $\mathbf{B}^{r-1}\hat{\gamma} = \frac{1}{\delta}\mathbf{B}^{r-1}\mathbf{D}_1\hat{\boldsymbol{\beta}}$ where $\mathbf{D}_1 = \begin{pmatrix} 1 & -1 & \dots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 1 & -1 \end{pmatrix}$ is the first order differencing matrix and $\mathbf{B}^{r-1}$ is the corresponding B-spline basis matrix of degree $r-1$. Thus the variance of the estimated derivative is

$$\text{Var}(\frac{1}{\delta}\mathbf{B}^{-1}\mathbf{D}_1\hat{\boldsymbol{\beta}}) = \frac{\sigma^2}{\delta^2}\mathbf{B}^{r-1}\mathbf{D}_1(\mathbf{B}'\mathbf{B})^{-1}\mathbf{D}_1^\top\mathbf{B}^{r-1\top}$$

The square roots of the diagonal elements of the matrix are the standard error of the derivatives.

```
# Estimate residual sd
sigma.hat <- sqrt(1/(length(y)-ncol(B)) * sum((B%*%beta - y)^2))

# Compute standard error of derivatives
D <- diff(diag(ncol(B)))
sds <- sigma.hat/delta * sqrt(diag(BBd%*%D%*%solve(crossprod(B))%*%t(D)%*%t(BBd)))

# Plot confidence bands
lines(xx, deriv + qnorm(0.975) * sds, lty=2)
lines(xx, deriv - qnorm(0.975) * sds, lty=2)

# Add horizontal line at 0
abline(h=0, lty=3)
```

(d) The basis functions of a truncated power basis of degree r with l knots is

$$1, x, \dots, x^{r-1}, (x-\kappa_1)_+^r, \dots, (x-\kappa_{l-1})^r$$

Taking derivatives yields

$$0, 1, \dots, (r-1)x^{r-2}, r(x-\kappa_1)_+^{r-1}, \dots, r(x-\kappa_{l-1})^{r-1}$$

Thus the derivative of the regression function can be computed as

$$m'(\mathbf{x}) = \mathbf{B}^{-1}\boldsymbol{\gamma}$$

where $\mathbf{B}^{r-1}$ is the matrix of truncated power functions of degree $r-1$ and $\boldsymbol{\gamma} = (\beta_1, 2\beta_2, \dots, (r-1)\beta_{r-1}, r\beta_r, \dots, r\beta_{l+r-1})^\top$.

```
# Create basis functions (using tbase function from practical 1)
B <- tbase(x, deg=3, n.knots=4)
# Estimate coefficients
beta <- qr.coef(qr(B), y)

# Plot fitted function
plot(x, y, xlab=”Age”, ylab=”log(Count)”)
xx<-seq(min(x), max(x), len=1e3)
BB <- tbase(xx, deg=3, n.knots=4)
lines(xx, BB%*%beta)

# Estimated derivative
BBd <-tbase(xx, deg=2, n.knots=4)
gamma <- beta[-1]*c(1:2, rep(3, 4-1))
plot(xx, BBd%*%gamma, type=”l”, xlab=”Age”, ylab=”Derivative”)
```

(e) For the local smoother we have that

$$\begin{aligned}\frac{\partial}{\partial x} m(x) &= \frac{\partial}{\partial x} \frac{\sum_{k=1}^{n} w(x_k - x; h) y_k}{\sum_{k=1}^{n} w(x_k - x; h)} \\ &= \frac{\left(\sum_{k=1}^{n} w'(x_k - x; h) y_k\right)\left(\sum_{k=1}^{n} w(x_k - x; h)\right) - \left(\sum_{k=1}^{n} w(x_k - x; h) y_k\right)\left(\sum_{k=1}^{n} w'(x_k - x; h)\right)}{\left(\sum_{k=1}^{n} w(x_k - x; h)\right)^2}\end{aligned}$$

For a Gaussian kernel we have that

$$w(t;h) = \frac{1}{\sqrt{2 * \pi h}} \exp(-t^2/2h^2) \qquad w'(t;h) = -\frac{t}{\sqrt{2 * \pi h^3}} \exp(-t^2/2h^2) = -\frac{t}{h^2} w(t;h),$$

```r
# Set bandwitdh
h <- 5

# Compute the (predictive) smoothing matrix S
S <- dnorm(outer(xx, x, ”–”),sd=h)
S <- S / rowSums(S)

# Plot data
plot(x, y, xlab=”Age”, ylab=”log(Count)”)
lines(xx, S%*%y)

# Calculate derivative
D <- outer(xx, x, ”–”)
T <- dnorm(D,sd=h)
plot(xx, -1/h^2 * ((T*D)%*%y * rowSums(T) - T%*%y * rowSums(T*D))
                   / rowSums(T)^2, type=”l”)
```

7. (a) In question 6 we have shown that

$$\frac{\partial}{\partial x} m(x) = \sum_{j=1}^{l+r-2} B_j^{r-1}(x) \frac{\beta_{j+1} - \beta_j}{\delta}.$$

All basis functions of the B-spline basis are non-negative, thus the derivative is non-negative if $\boldsymbol{\beta}_{j+1} \geq \beta_j$.

(b) *(parts (i) to (iii))*
We first define a function which performs the monotonicity-constrained regression.

```r
monoreg <- function(B, y, lambda) {
    # Set initial difference matrix
    D1 <- diff(diag(ncol(B)), diff=1)
    # Set initial beta
    beta <- rep(0, ncol(B))
    for (h in 1:100) {
       # Compute delta and difference matrix
        delta <- beta[-1]<beta[-length(beta)]
        D <- delta*D1
        # Store old beta
        old.beta <- beta
        # Update beta (using QR decomposition)
        B.work <- rbind(B, sqrt(lambda)*D)
        y.work <- c(y, rep(0, nrow(D)))
        beta <- qr.coef(qr(B.work), y.work)
        if (all(abs(beta-old.beta) < 1e-10*abs(old.beta)))
            break
    }
    beta
}
```

We first a non-shape constrained B-spline with a (too) large number of knots.

```
# Simply call covariate x and response y
x <- radiocarbon$cal.age
y <- radiocarbon$rc.age

# Construct basis
B <- bbase(x, n.knots=20)

# B spline solution
beta <- qr.coef(qr(B), y)

# Plot the data
plot(rc.age~cal.age, data=radiocarbon)

# Create design matrix for predictions on a regular grid (makes plot prettier)
xx <- seq(min(x), max(x), len=1000)
BB <- bbase(xx, n.knots=20)

# Add fitted function to plot
lines(xx, BB%*%beta)
```

Now we fit the monotonicity-constrained spline using the function `monoreg` we have written above.

```
# Compute betas
beta <- monoreg(B, y, lambda=100)

# Add line to the plot
lines(xx, BB%*%beta, col=”red”)
```

(c) We use the simple trick of considering $-\mathbf{y}$, in which case we need to estimate a non-decreasing function, which is what we have done in part (b).

```
# Set x to covariate and y to -response
x <- follicle$age
y <- -follicle$log.count

# Construct basis functions and compute coefficient estimates
B <- bbase(x, n.knots=10)
beta <- monoreg(B, y, lambda=100)

# Plot data and fitted function
plot(x, -y, xlab=”Age”, ylab=”log(Count)”)
xx<-seq(min(x), max(x), len=1e3)
BB <- bbase(xx, deg=3, n.knots=10)
lines(xx, -BB%*%beta, col=”red”)

# For comparison add non-monotonicity-constrained fit
beta <- qr.coef(qr(B), y)
lines(xx, -BB%*%beta)
```

8. The solution sketched here, and in the script below, uses the gam function from the mgcv package, with its default settings. This means that smoothing parameters are selected automatically. This captures the main features of the data, but there would be scope for more careful analysis, to account for the effects of smoothing parameter selection, if students chose to do that.

   (a) A pairs plot shows some potential (but noisy) relationships between logSO2 and other variables, with a clear downward trend over the years. Temperature is strongly related to the week of the year and to humidity.

   (b) An additive model, using the default setting in the gam function from mgcv, shows strong effects of Year and Temp, a modest effect of Flow, and little effects from the other three variables. Week remains significant in the model which omits Rain and Flow, so we might reasonably stick with this model.

   (c) Plots of the Year term from the model selected above and from the model which uses only Year and Week are quite similar, except that the 'shoulder' in the estimate from the latter model is rather sharper. We need to be a little careful here, because the gam function is automatically choosing the degree of smoothing in each model. It might be better to fix the degree of smoothing to be the same in both cases. However, broadly speaking, the overall trend with Year is quite similar. There may therefore be relatively little added value in the meteorological variables. (Notice, though, that the standard errors are larger in the simpler model, because the meteorological variables have explained more of the variation in the data.)

   (d) The acf and pacf functions show some evidence (not surprisingly) of serial correlation. There are various options for modification here, one of which is to explore the use of the gamm function. In terms of effects, the estimates may change relatively little but the standard errors should increase. The effects here are sufficiently strong that this may not change the conclusions much, although the evidence for a Week effect may be further weakened (if you will excuse the pun).

```r
pairs(SO2station)

model1 <- gam(logSO2 ~ s(Year) + s(Week) + s(Rain) +
                       s(Temp) + s(Humidity) + s(Flow),
                       data = SO2station)
plot(model1, se = TRUE, pages = 1)
anova(model1)
model2 <- gam(logSO2 ~ s(Year) + s(Week) + s(Temp) + s(Flow),
                       data = SO2station)
plot(model2, se = TRUE, pages = 1)
anova(model2)

model0 <- gam(logSO2 ~ s(Year) + s(Week), data = SO2station)
par(mfrow = c(1, 2))
plot(model1, se = TRUE, select = 1, ylim = c(-1.9, 1.1))
plot(model0, se = TRUE, select = 1, ylim = c(-1.9, 1.1))
par(mfrow = c(1, 1))

acf(residuals(model1))
pacf(residuals(model1))
```

9. (a) We continue to use the double-indexing used in $\hat{\mathbf{y}}$. Then $\mathbf{S} = (s_{ij;kl})$ with

$$s_{ij;kl} = \hat{y}_{ij} = \hat{m}(x_i, z_j) = \frac{w(x_k - x_i; h)w(z_l - z_j; h)}{\sum_{\kappa=1}^{m}\sum_{\lambda=1}^{n} w(x_\kappa - x_i; h)w(z_\lambda - z_j; h)}$$
$$= \frac{w(x_k - x_i; h)}{\sum_{\kappa=1}^{m} w(x_\kappa - x_i; h)} \frac{w(z_l - z_j; h)}{\sum_{\lambda=1}^{n} w(z_\lambda - z_j; h)}.$$

(b) This follows directly from part (a) and the definition of the Kronecker product. The first fraction in part (a) comes from the matrix $\mathbf{S}_1$ and the second fraction comes from $\mathbf{S}_2$.

(c) This follows directly from the last property mentioned on page 8 of the assignment sheet.
The key advantage of this representation is that much smaller matrices are used, so computations are much more efficient and much larger data sets can be considered.

(d) We can use the following R code.

```
# Set bandwidth
h <- 1

# Construct S1
S1 <- dnorm(outer(us.easting, us.easting, ”–”),sd=h)
S1 <- S1 / rowSums(S1)

# Construct S2
S2 <- dnorm(outer(us.northing, us.northing, ”–”),sd=h)
S2 <- S2 / rowSums(S2)

# Compute fitted values
fitted.rain <-  S1%*%us.rain%*%t(S2)

# Plot results
library(maps)
m <- map(”state”, mar=rep(0.5,4))
image(us.easting-360, us.northing, fitted.rain, add=TRUE)
lines(m)
```

10. (a) We have that

$$
\begin{aligned}
\mathbf{B} &= \begin{pmatrix}
B_{11}(x_1,z_1) & \dots & B_{l_1+r-1,1}(x_1,z_1) & B_{12}(x_1,z_1) & \dots & B_{l_1+r-1,2}(x_1,z_1) & \dots & B_{l_1+r-1,l+2+r-1}(x_1,z_1)\\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots\\
B_{11}(x_m,z_1) & \dots & B_{l_1+r-1,1}(x_m,z_1) & B_{12}(x_m,z_1) & \dots & B_{l_1+r-1,2}(x_m,z_1) & \dots & B_{l_1+r-1,l+2+r-1}(x_m,z_1)\\
B_{11}(x_1,z_2) & \dots & B_{l_1+r-1,1}(x_1,z_2) & B_{12}(x_1,z_2) & \dots & B_{l_1+r-1,2}(x_1,z_2) & \dots & B_{l_1+r-1,l+2+r-1}(x_1,z_2)\\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots\\
B_{11}(x_m,z_n) & \dots & B_{l_1+r-1,1}(x_m,z_n) & B_{12}(x_m,z_n) & \dots & B_{l_1+r-1,2}(x_m,z_n) & \dots & B_{l_1+r-1,l+2+r-1}(x_m,z_n)
\end{pmatrix}\\
&= \begin{pmatrix}
B_1^{(1)}(x_1)B_1^{(2)}(z_1) & \dots & B_{l_1+r-1}^{(1)}(x_1)B_1^{(2)}(z_1) & B_1^{(1)}(x_1)B_2^{(2)}(z_1) & \dots & B_{l_1+r-1}^{(1)}(x_1)B_2^{(2)}(z_1) & \dots & B_{l_1+r-1}^{(1)}(x_1)B_{l_2+r-1}^{(2)}(z_1)\\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots\\
B_1^{(1)}(x_m)B_1^{(2)}(z_1) & \dots & B_{l_1+r-1}^{(1)}(x_m)B_1^{(2)}(z_1) & B_1^{(1)}(x_m)B_2^{(2)}(z_1) & \dots & B_{l_1+r-1}^{(1)}(x_m)B_2^{(2)}(z_1) & \dots & B_{l_1+r-1}^{(1)}(x_m)B_{l_2+r-1}^{(2)}(z_1)\\
B_1^{(1)}(x_1)B_1^{(2)}(z_2) & \dots & B_{l_1+r-1}^{(1)}(x_1)B_1^{(2)}(z_2) & B_1^{(1)}(x_1)B_2^{(2)}(z_2) & \dots & B_{l_1+r-1}^{(1)}(x_1)B_2^{(2)}(z_2) & \dots & B_{l_1+r-1}^{(1)}(x_1)B_{l_2+r-1}^{(2)}(z_2)\\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots\\
B_1^{(1)}(x_m)B_1^{(2)}(z_n) & \dots & B_{l_1+r-1}^{(1)}(x_m)B_1^{(2)}(z_n) & B_1^{(1)}(x_m)B_2^{(2)}(z_n) & \dots & B_{l_1+r-1}^{(1)}(x_m)B_2^{(2)}(z_n) & \dots & B_{l_1+r-1}^{(1)}(x_m)B_{l_2+r-1}^{(2)}(z_n)
\end{pmatrix}\\
&= \begin{pmatrix}
B_1^{(2)}(z_1) & \dots & B_{l_2+r-1}^{(2)}(z_1)\\
\vdots & \ddots & \vdots\\
B_1^{(2)}(z_n) & \dots & B_{l_2+r-1}^{(2)}(z_n)
\end{pmatrix} \otimes \begin{pmatrix}
B_1^{(1)}(x_1) & \dots & B_{l_1+r-1}^{(1)}(x_1)\\
\vdots & \ddots & \vdots\\
B_1^{(1)}(x_m) & \dots & B_{l_1+r-1}^{(1)}(x_m)
\end{pmatrix}\\
&= \mathbf{S}^{(2)} \otimes \mathbf{S}^{(1)}
\end{aligned}
$$

(b) Using that $\mathbf{B} = \mathbf{B}^{(2)} \otimes \mathbf{B}_1$ we have that

$$
\begin{aligned}
\mathbf{S} &= \mathbf{B}(\mathbf{B}^\top\mathbf{B})^{-1}\mathbf{B}^\top\\
&= (\mathbf{B}^{(2)} \otimes \mathbf{B}^{(1)}) \underbrace{\underbrace{\left((\mathbf{B}^{(2)} \otimes \mathbf{B}^{(1)})^\top(\mathbf{B}^{(2)} \otimes \mathbf{B}^{(1)})\right)^{-1}}_{=\left((\mathbf{B}^{(2)\top}\otimes\mathbf{B}^{(1)\top})(\mathbf{B}^{(2)}\otimes\mathbf{B}^{(1)})\right)^{-1}=(\mathbf{B}^{(2)\top}\mathbf{B}^{(2)})^{-1}\otimes(\mathbf{B}^{(1)\top}\mathbf{B}^{(1)})^{-1}} (\mathbf{B}^{(2)} \otimes \mathbf{B}^{(1)})^\top}_{=((\mathbf{B}^{(2)\top}\mathbf{B}^{(2)})^{-1}\mathbf{B}^{(2)})\otimes((\mathbf{B}^{(1)\top}\mathbf{B}^{(1)})^{-1}\mathbf{B}^{(1)})}\\
&= (\mathbf{B}^{(2)}(\mathbf{B}^{(2)\top}\mathbf{B}^{(2)})^{-1}\mathbf{B}^{(2)}) \otimes (\mathbf{B}^{(1)}(\mathbf{B}^{(1)\top}\mathbf{B}^{(1)})^{-1}\mathbf{B}^{(1)}) = \mathbf{S}^{(2)} \otimes \mathbf{S}^{(1)}
\end{aligned}
$$

The result follows from the last property mentioned on page 8 of the assignment sheet.

(c) For P-splines we have, using $\mathbf{D} = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ 0 & 1 & -1 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & \dots & 1 & -1 \end{pmatrix}$

$$
\begin{aligned}
\mathbf{S} &= \mathbf{B}(\mathbf{B}^\top\mathbf{B} + \lambda\mathbf{D}^\top\mathbf{D})^{-1}\mathbf{B}^\top \\
&= \mathbf{B}(\mathbf{B}^\top\mathbf{B} + \lambda\mathbf{D}^\top\mathbf{D})^{-1}\mathbf{B}^\top \\
\mathbf{S}^{(2)} \otimes \mathbf{S}^{(1)} &= (\mathbf{B}^{(2)}(\mathbf{B}^{(2)\top}\mathbf{B}^{(2)} + \lambda\mathbf{D}^{(2)\top}\mathbf{D}^{(2)})^{-1}\mathbf{B}^{(2)\top}) \otimes (\mathbf{B}^{(1)}(\mathbf{B}^{(1)\top}\mathbf{B}^{(1)} + \lambda\mathbf{D}^{(1)\top}\mathbf{D}^{(1)})^{-1}\mathbf{B}^{(1)\top}) \\
&= (\mathbf{B}^{(2)} \otimes \mathbf{B}^{(1)}) \Big((\mathbf{B}^{(2)\top}\mathbf{B}^{(2)} + \lambda\mathbf{D}^{(2)\top}\mathbf{D}^{(2)}) \otimes (\mathbf{B}^{(1)\top}\mathbf{B}^{(1)} + \lambda\mathbf{D}^{(1)\top}\mathbf{D}^{(1)})\Big)^{-1} (\mathbf{B}^{(2)} \otimes \mathbf{B}^{(1)})^\top \\
&= (\mathbf{B}^{(2)} \otimes \mathbf{B}^{(1)}) \big((\mathbf{B}^{(2)} \otimes \mathbf{B}^{(1)})^\top(\mathbf{B}^{(2)} \otimes \mathbf{B}^{(1)}) + \lambda(\mathbf{D}^{(2)} \otimes \mathbf{B}^{(1)})^\top(\mathbf{D}^{(2)} \otimes \mathbf{B}^{(1)}) \\
&\qquad \lambda(\mathbf{B}^{(2)} \otimes \mathbf{D}^{(1)})^\top(\mathbf{B}^{(2)} \otimes \mathbf{D}^{(1)}) + \lambda^2(\mathbf{D}^{(2)} \otimes \mathbf{D}^{(1)})^\top(\mathbf{D}^{(2)} \otimes \mathbf{D}^{(1)})\big)^{-1} (\mathbf{B}^{(2)} \otimes \mathbf{B}^{(1)})^\top \\
&= \mathbf{B}^\top \big(\mathbf{B}^\top\mathbf{B} + \lambda(\mathbf{D}^{(2)} \otimes \mathbf{B}^{(1)})^\top(\mathbf{D}^{(2)} \otimes \mathbf{B}^{(1)}) + \lambda(\mathbf{B}^{(2)} \otimes \mathbf{D}^{(1)})^\top(\mathbf{B}^{(2)} \otimes \mathbf{D}^{(1)}) \\
&\qquad + \lambda^2(\mathbf{D}^{(2)} \otimes \mathbf{D}^{(1)})^\top(\mathbf{D}^{(2)} \otimes \mathbf{D}^{(1)}))\big)^{-1} \mathbf{B}^\top
\end{aligned}
$$

Thus the two approaches are not the same, as they correspond to different penalties.

(d) We can use the following R code.

```r
# Construct S1
B1 <- bbase(us.easting, n.knots=16)
S1 <- B1%*%solve(crossprod(B1))%*%t(B1)

# Construct S2
B2 <- bbase(us.northing, n.knots=4)
S2 <- B2%*%solve(crossprod(B2))%*%t(B2)


# Compute fitted values
fitted.rain <-  S1%*%us.rain%*%t(S2)

# Plot results
library(maps)
m <- map(”state”, mar=rep(0.5,4))
image(us.easting-360, us.northing, fitted.rain, add=TRUE)
lines(m)
```