Active Session 1: Tuesday

For each "active session" you can choose whether you want work in the computer lab and implement spline methods and use them to analyse data or whether you want to so some more theoretical pen-and-paper work.

The sessions do not depend on each other, so you can mix and match.

Pen-and-paper tasks

1. In this task we consider 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 $\mathbf{B}^\top\mathbf{B} + \lambda\mathbf{D}^\top\mathbf{D}$?

2. In this task you will prove the optimality property of natural cubic splines.

   (a) We start with proving the lemma from the notes, that natural cubic splines are optimal interpolators in the sense that they minimise the integrated second derivative penalty.

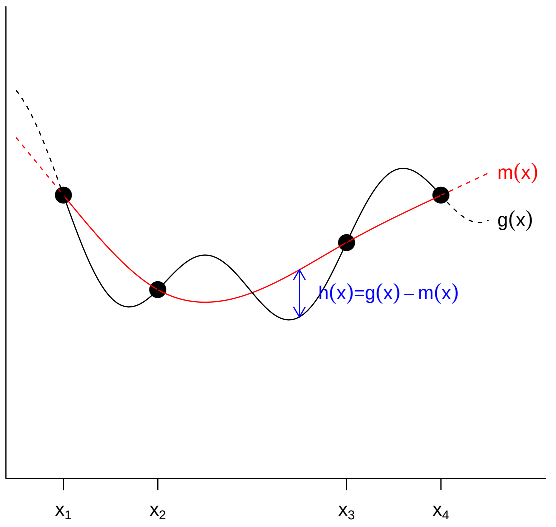
   Lemma. *Amongst all functions on $[a, b]$ which are twice continuously differentiable and which interpolate the set of points (x_i, y_i), a natural cubic spline with knots at the x_i yields the smallest roughness penalty*

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

   Let $m(\cdot)$ 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

$$h(x) = g(x) - m(x)$$

   the difference between the two functions.



i. Explain why $h(x_i) = 0$ for all observed x_i.

ii. Explain why $m''(a) = m''(b) = 0$.

iii. Use integration by parts to show that $\int_a^b m''(x)h''(x)\ dx = 0$.
Hint: $m(\cdot)$ is a piecewise cubic polynomial, so for each segment $m'''(x)$ is constant.

iv. Show that $\int_a^b g''(x)^2\ dx = \int_a^b h''(x)^2\ dx + \int_a^b m''(x)^2\ dx$.
Hint: Use $g(x) = m(x) + h(x)$, i.e. $g''(x) = m''(x) + h''(x)$, expand the square and use the result from part iii.

v. Use your result from part iv. to show that

$$\int_a^b g''(x)^2\ dx \geq \int_a^b m''(x)^2\ dx, \tag{1}$$

i.e. the roughness penalty is larger for $g(\cdot)$ that it would be for $m(\cdot)$.

vi. Show that equality in (1) only holds if $g(\cdot)$ is also a natural cubic spline on (a, b).

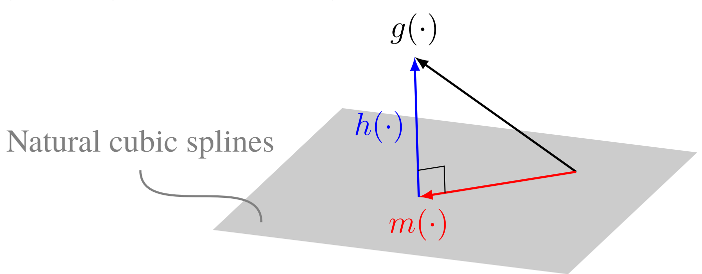
vii. The results above can be interpreted in terms of orthogonality, inner products and norms. Consider the inner product

$$\langle m, h \rangle = \int_a^b m''(x)h''(x)\ dx$$

and the associated norm

$$\|g\|^2 = \langle g, g \rangle = \int_a^b g''(x)^2\ dx.$$

Explain parts iii., iv. and v. in these terms.
Hint: You might find the figure below helpful.



(b) We will now use the lemma to show that the theorem.

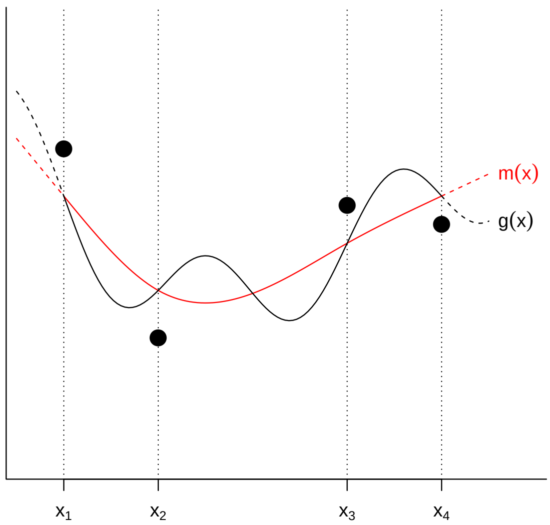
Theorem. *The minimiser of*

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

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

Assume that we have a candidate $g(\cdot)$ which we think is optimal. We will now construct a natural cubic spline which we can prove will do better.

Define a natural cubic spline $m(\cdot)$ which generates exactly the same predictions at the observed $x_1, \ldots, x_n$, i.e. $m(x_i) = g(x_i)$ for $i = 1, \ldots, n$.



i. Explain why $\sum_{i=1}^{n}(y_i - g(x_i))^2 = \sum_{i=1}^{n}(y_i - m(x_i))^2$, i.e. both functions fit the data equally well.

ii. We know from part (a) that, unless $g(\cdot)$ is itself a natural cubic spline, $\int_a^b g''(x)^2\,dx \geq \int_a^b m''(x)^2\,dx$. Use this to complete the proof of the theorem.

Computer tasks

1-D splines

In this section you can work with any of the following three datasets:

Divorces in the US

This data set (called `divorces`) contains the number of divorces per 10,000 women for most of the 20th century. it has the following columns:

`year`	Calendar year
`divorces`	Number of divorce per 1,000 women (aged 15+) in that year *(response)*

Radiocarbon dating.

This is the data (called `radiocarbon`) used in the lectures.

`cal.age`	True calendar age
`rc.age`	Age predicted from the radiocarbon dating process *(response)*

Great barrier reef.

This is a univariate version of the data (called `gbr`) used in the lectures. Only two columns are retained:

`longitude`	Longitude
`score1`	Principal component score summarising the catch *(response)*

The command

```
load(url("http://www.stats.gla.ac.uk/~adrian/apts/splines-1.RData"))
```

will make all three datasets available in your workspace.

Now use your chosen dataset to undertake the following tasks.

1. Explore the data set you have chosen graphically.
2. Launch the function `pspline.cartoon` with the name of your dataset as argument i.e.

   Divorces in the US. `pspline.cartoon(divorces)`

   Radiocarbon dating. `pspline.cartoon(radiocarbon)`

   Great barrier reef. `pspline.cartoon(gbr)`

   The function fits a P-spline model to the data and shows the fitted model together with the basis function. A P-spline uses a regular B-spline basis, but combines it with difference penalty for the coefficients (typically based on second-order differences), i.e. it minimises the penalised least squares criterion

$$\sum_{i=1}^{n}(y_i - m(x_i))^2 + \lambda \sum_{j=1}^{l+r-3} (\beta_j - 2\beta_{j+1} + \beta_{j+2})^2$$

Including the difference penalty allows using a "too large" number of basis functions, as the penalty parameter λ, rather than the number of basis functions r, now controls the level of smoothness and the bias-variance trade-off.

Experiment with the degree of the spline, the number of knots and the smoothing parameter λ.

3. Reproduce the results from `pspline.cartoon` by fitting a B-spline model (or even a P-spline model) from first principles in R.

   i. The function `bbase`, which constructs the B-spline basis, should be available in your workspace (supplied in `splines-1.RData`). Use this function to create the matrix $\mathbf{B}$ of basis functions.
   For example, you can create a B-spline basis of degree 3 with 10 knots using

   ```
   B <- bbase(x, deg=3, n.knots=10)
   ```

   Can you plot the basis functions?

   ii. Compute the (penalised) least-squares coefficients $\hat{\boldsymbol{\beta}}$.

   iii. Create a plot of the data together with the fitted spline function. Adjust the number of knots if necessary.

Additive and related models

In tackling the questions below, you may find it helpful to have a brief indication of the `gam` function in the `mgcv` package in `R`. This package, written by Simon Wood, is a powerful set of tools for fitting and analysing generalised additive models. The book, Wood (2006), referenced in the lecture notes, makes extensive reference to `R` and the `mgcv` package so this is a good way of learning both about additive models and how to use them in practice. The principal function is `gam` which is called, for example, as

```
model <- gam(y ~ s(x1) + s(x2) + s(x3))
```

where the vector of responses `y` is modelled through a set of smooth functions in `x1`, `x2` and `x3`. The `s(x1)` notation identifies that a smooth function should be fitted. Linear terms can be fitted in the usual way simply through the variable name. The syntax and operation of the `gam` function follows that of `lm` and `glm` as closely as possible.

In the exercises below, scripts are provided to get you started on each problem, and in some cases to lead you through. Of course, feel free to experiment. There should be someone to call on if you get stuck.

1. **The Clyde data** `clyde.R`

   In one of the lectures we looked at the relationship of DO to Temperature, Salinity and Year at a single sampling station in the River Clyde. Repeat this analysis at one or two other stations, to see whether the same features are apparent at these locations.

2. **The mackerel egg survey** `mackerel.R`

   This exercise also repeats one of the lecture examples. A multi-country survey of mackerel eggs was carried out in the Eastern Atlantic in 1992. The aim was to estimate the total biomass of spawning mackerel. However, a first step is to construct a model which describes the pattern of egg counts in the water samples collected.

   Use the script to try the following operations.

   (a) Plot the sampling region

   (b) Explore the (marginal) relationship between the egg density (log scale) and latitude, longitude, depth (log scale) and temperature.

(c) Fit and interpret a simple additive model

(d) Consider whether the Temperature variable is really needed.

Can you identify the preferred depth at which mackerel lay their eggs?

3. **SO_2 in the Czech Republic** `CZ03.R`

Measurements of SO_2 (on a log scale) from a monitoring station in the Czech Republic are available in this dataset.

(a) Begin by fitting an additive model which relates texttttSO2 to the covariates `Year` and `Week`.

(b) There is interest in whether having information on local meteorology will be important in estimating the effects of `Year` and `Week`. Add the variables `Rain`, `Temp`, `Humidity` and `Flow` as further flexible terms in the additive model and interpret what you see.

4. **Water quality in Loch Leven** `lochleven.R`

Loch Leven is situated in lowland Scotland in the Perth and Kinross area. It is the largest shallow lake in Great Britain with an area of 13.3km^2, mean depth 3.9m and a maximum depth 25.5m. There are approximately 150 variables measured at Loch Leven (chemical, physical, biological and meteorological) and most of this monitoring is carried out by the Centre for Ecology & Hydrology in Edinburgh.

One of the features of interest is the water quality and hence the relationship between chlorophyll$_a$ (as an indicator of water quality) and Soluble Reactive Phosphorus (a nutrient) is very important. This case study explores this relationship.

The data provided are the natural logarithm of the monthly means for chlorophyll$_a$ (`lchla`) and SRP (`lsrp`) from January 1988 to December 2007. Natural log transforms of the data are used to stabilize the variance and there are some missing values. Columns of data for year and month are also provided.

The script contains commands to investigate the following questions.

(a) Plot the data to examine the relationship between `lsrp` and `lchla`.

(b) Is the relationship between `lsrp` and `lchla` affected by month?

(c) A possible model is a *varying coefficient* model. This fits a linear regression between the two variables but allows the parameters of the regression to change smoothly with time of year. This can be fitted by smoothing over `month` and `lchla`, with a very large smoothing parameter for `month` to make this term linear.

Active Session 1: Tuesday – Model answers

Pen-and-paper tasks

1. (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

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

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.

2. In this task you will prove the optimality property of natural cubic splines.

(a) We start with proving the lemma from the notes, that natural cubic splines are optimal interpolators in the sense that they minimise the integrated second derivative penalty.

i. Both $m(\cdot)$ and $g(\cdot)$ interpolate the data, i.e. $y_i = m(x_i)$ and $y_i = g(x_i)$, and thus $h(x_i) = g(x_i) - m(x_i) = y_i - y_i = 0$ (for $i = 1, \dots, n$).

ii. $m(\cdot)$ is a natural cubic spline, so by definition $m''(a) = m''(b) = 0$.

iii. Using integration by parts we obtain that

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

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 $m'''(x)$ is constant.

iv. Using $g(x) = m(x) + h(x)$,

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

v. Thus,

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

vi. 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 $g(x) = m(x)$ for all $x \in [a, b]$.

vii. The result from above that

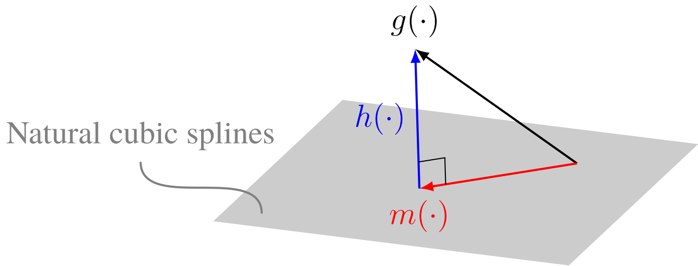
$$\langle m, h \rangle = \int_a^b m''(x)h''(x)\,dx = 0,$$

means that $m(\cdot)$ and the "residual" $h(\cdot)$ are orthogonal, so by the Pythagoraean theorem $\|g\|^2 = \|m\|^2 + \|h\|^2$, and thus

$$\|m\|^2 = \int_a^b m''(x)^2\,dx \leq \int_a^b g''(x)^2\,dx = \|g\|^2$$

(in a right traingle, any other side cannot be longer than the hypothenuse).

The space of all twice continuously differentiable functions is a vector space, with the space of natural cubic splines with a set of given knots being a subspace. $m(\cdot)$ is then the projection of $g(\cdot)$ into the space of natural cubic splines, and thus the norm ("length") of $m(\cdot)$ cannot be greater than the norm of $g(\cdot)$.



(b) i. As $m(\cdot)$ interpolates the fitted values $\hat{y}_i = g(x_i)$, we also have $\hat{y}_i = m(x_i)$, thus the sum of squared residuals is the same, i.e. $\sum_{i=1}^n (y_i - g(x_i))^2 = \sum_{i=1}^n (y_i - m(x_i))^2$.

ii. From part (a) $\int_a^b g''(x)^2 \, dx \geq \int_a^b m''(x)^2 \, dx$ and thus

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

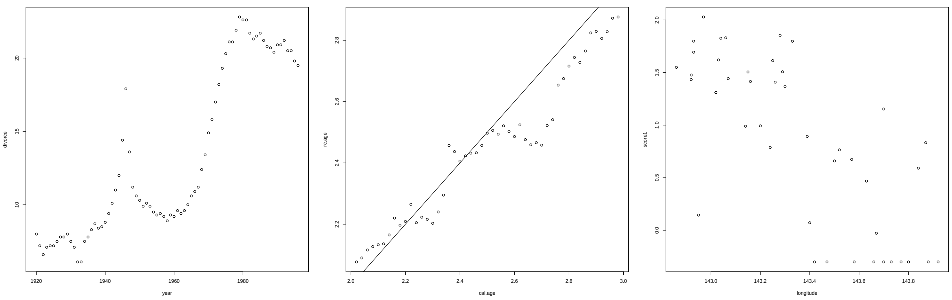
Computer tasks

Solutions to computer tasks

1-D splines

1. *Graphical exploration*. The plots for the three datasets are shown below.

```
plot(divorce ~ year, data = divorces)
plot(rc.age ~ cal.age, data = radiocarbon)
abline(0, 1)
plot(score1 ~ longitude, data = gbr)
```



Divorces show a large spike at the end of the First World War and then a large and sustained increase during the 1960's, when the divorce law was changed.

The radiocarbon data shows a pattern which is not too far from the line of equality (displayed on the plot) but with some marked fluctuations away from this. This is expected, due to the fact that the natural background radioactivity has fluctuated over time.

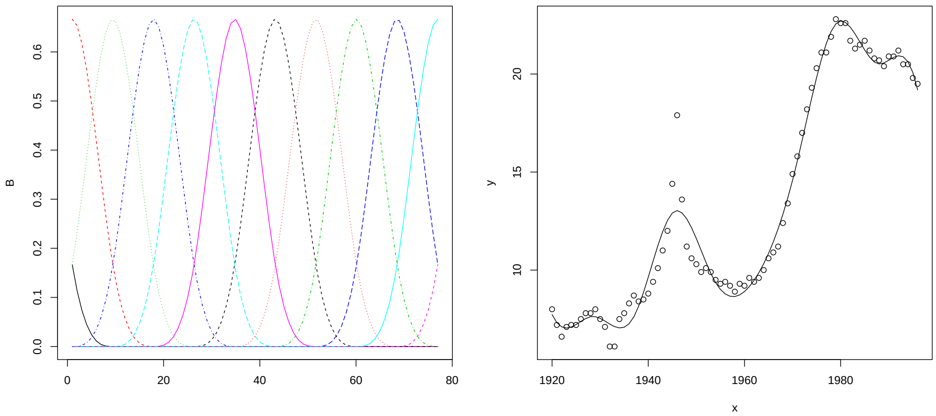
The Reef data have been discussed in lectures. There is a strong downward trend as longitude increases, but with a lot of variability in the measurements.

2. *pspline.cartoon*. Intuition on the effects of different parameters should be gained as participants play with the buttons and sliders.

3. *First principles*. The code below does this for the divorces data. This will work in the same way for the other datasets, simply by changing the construction of the `x` and `y` variables in the first two lines.

Firstly the 'design' matrix is computed and its columns are plotted to show the underlying b-spline basis. Then the weights are estimated using the formula discussed in lectures. The fitted values are then displayed.

```
y     <- divorces$divorce
x     <- divorces$year
B     <- bbase(x, deg=3, n.knots=10)
matplot(B, type = "l")

betahat <- c(solve(crossprod(B))%*% t(B) %*% y)
fv      <- c(B %*% betahat)
plot(x, y)
lines(x, fv)
```

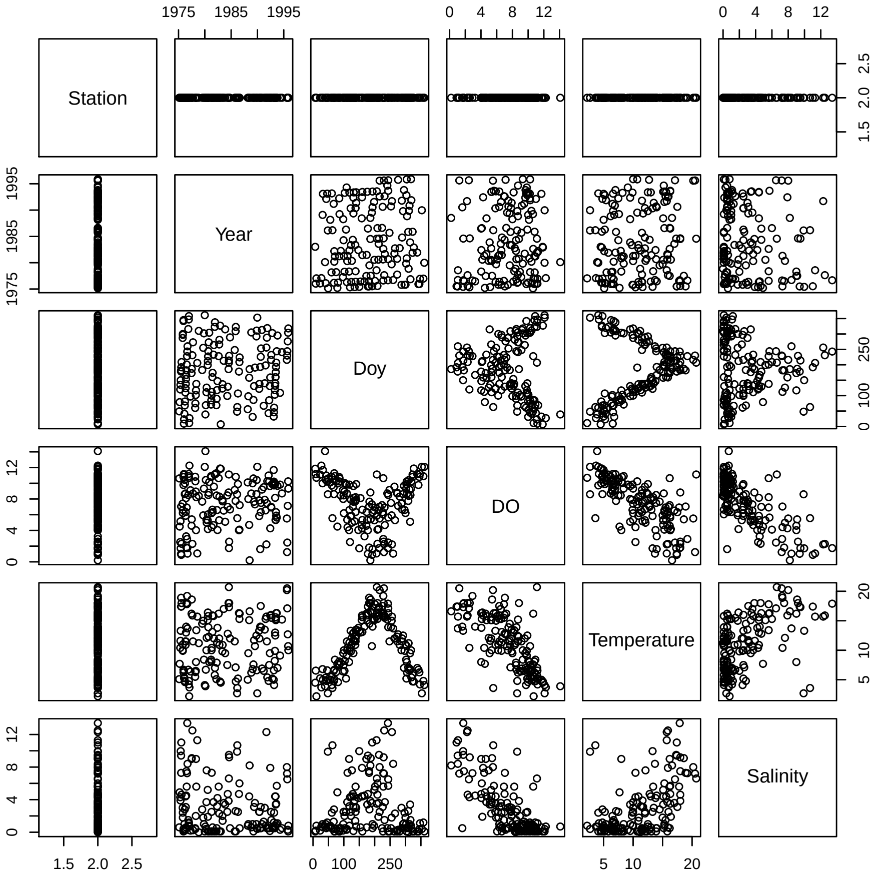


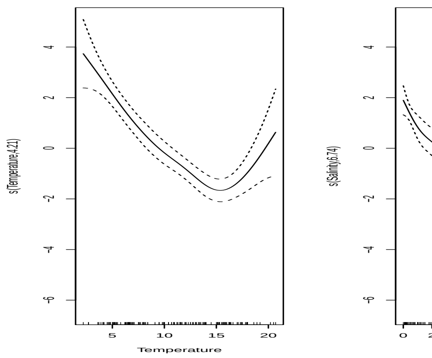
Additive and related models

1. **Clyde data**

   The data can be subsetted, plotted, and a model fitted, as shown below. Setting `Station` to some other even number up to 24 will change the station for which the data are selected. There is quite a lot to learn about how things change along the river by viewing the results of models for different stations.

```
load(url("http://www.stats.gla.ac.uk/~adrian/clyde.rda"))
clyde.subset <- subset(clyde, Station == 2)
pairs(clyde.subset)

library(mgcv)
model <- gam(DO ~ s(Temperature) + s(Salinity) + s(Year),
                      data = clyde.subset)
plot(model)
```

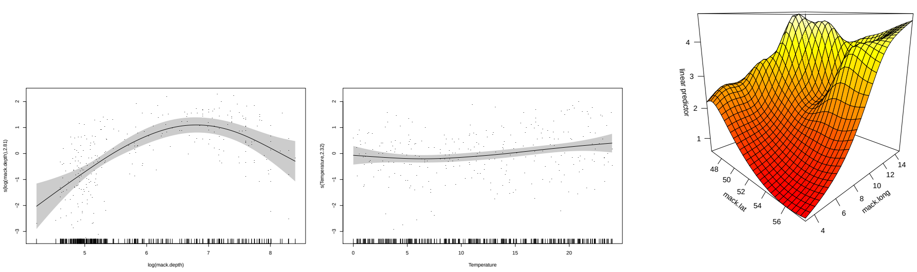




2. **The mackerel egg survey**

   A pairs plot bears a lot of inspection. The fitted modelshows a quadratic effect of log(depth), a small effect of temperature, and a complex effect of remaining spatial variation.

```
library(sm)
pairs(mackerel)

library(mgcv)
model1  <- gam(log(Density) ~ s(log(mack.depth)) + s(Temperature)
                         + s(mack.lat, mack.long), data = mackerel)
plot(model1, select = 1, se = TRUE, shade = TRUE, residuals = TRUE)
plot(model1, select = 2, se = TRUE, shade = TRUE, residuals = TRUE)
vis.gam(model1, c(”mack.lat”, ”mack.long”),
      ticktype = ”detailed”, theta = 45)

anova(model1)
```



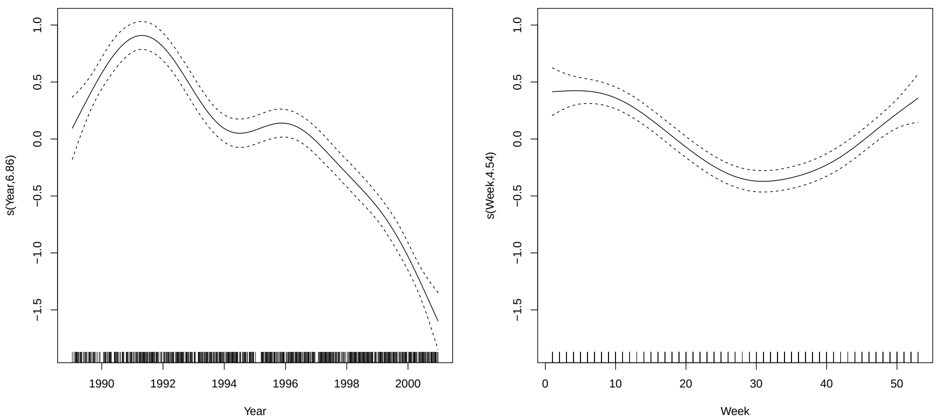

```
Approximate significance of smooth terms:
```

```
                         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
```

3. **SO_2 in the Czech Republic**

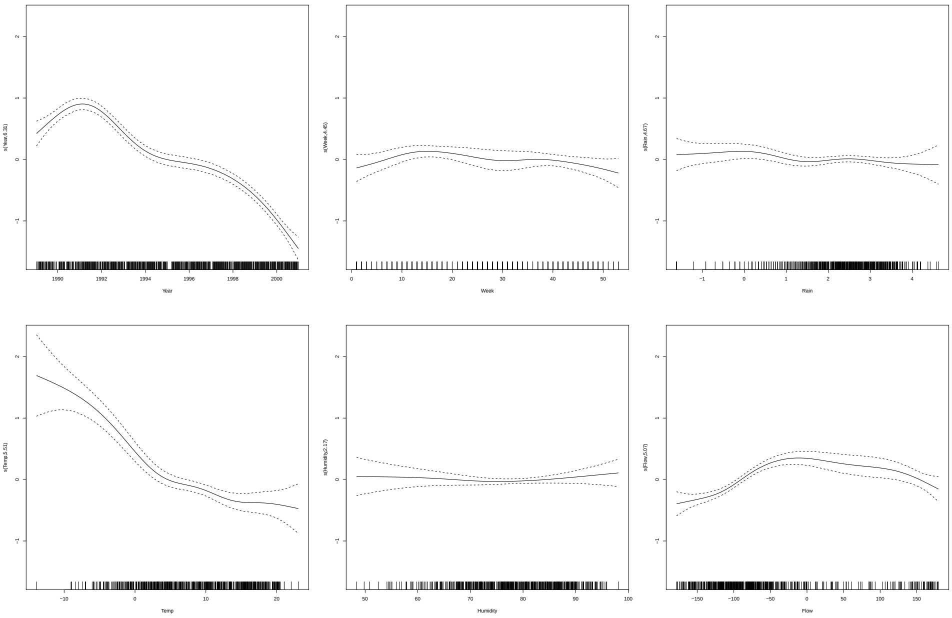
   The code in the script plots the data and fits a simple additive model. There is clear downward trend over the years and, as expected a strong seasonal effect.

```
CZ03 <- read.table(url("http://www.stats.gla.ac.uk/~adrian/apts/CZ03.dat"))
names(CZ03) <- c("SO2", "Year", "Week", "Rain", "Temp", "Humidity", "Flow")

library(mgcv)
model <- gam(SO2 ~ s(Year) + s(Week), data = CZ03)
plot(model)
```



If the model
`model1 <- gam(SO2 ∼ s(Year)+s(Week)+s(Rain)+s(Temp)+s(Humidity) + s(Flow), data = CZ03)`
is fitted the results are shown below. The estimate of trend over the years is very similar. The seasonal effect largely disappears, simply because temperature is now present and this has a seasonal pattern. The only other variable which shows some effect is `Flow`. An analysis of variance confirms this view.

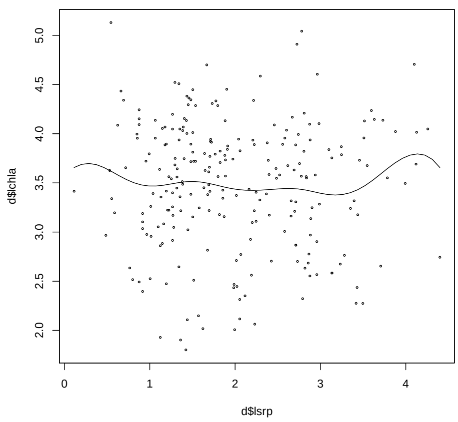
```
model1 <- gam(SO2 ~ s(Year) + s(Week) + s(Rain) + s(Temp) +
                      s(Humidity) + s(Flow), data = CZ03)
par(mfrow = c(2, 3))
plot(model1)
par(mfrow = c(1, 1))
anova(model1)
```



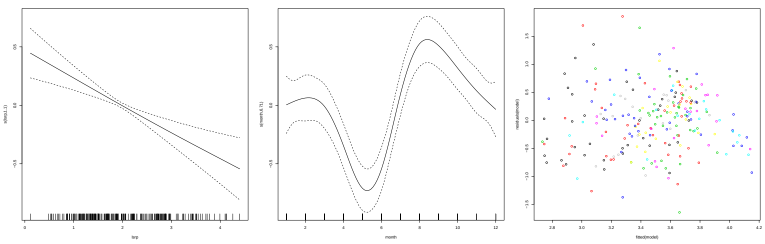

```
Approximate significance of smooth terms:
              edf Ref.df       F  p-value
s(Year)     6.305  7.445 136.774  <2e-16
s(Week)     4.450  5.567   2.333  0.0359
s(Rain)     4.666  5.733   1.145  0.2832
s(Temp)     5.511  6.717  19.970  <2e-16
s(Humidity) 2.172  2.784   0.944  0.4748
s(Flow)     5.072  6.178  16.661  <2e-16
```

4. **Water quality in Loch Leven**

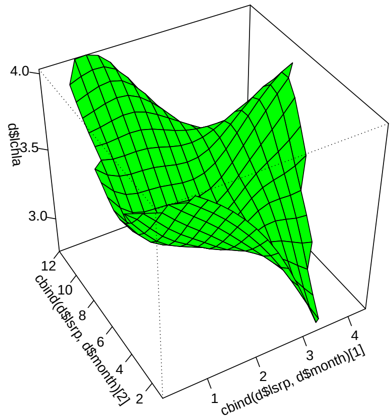
   After using the script to read the data, the suggested `sm.regression` command does not show any (marginal) relationship between `lsrp` and `lchla`.



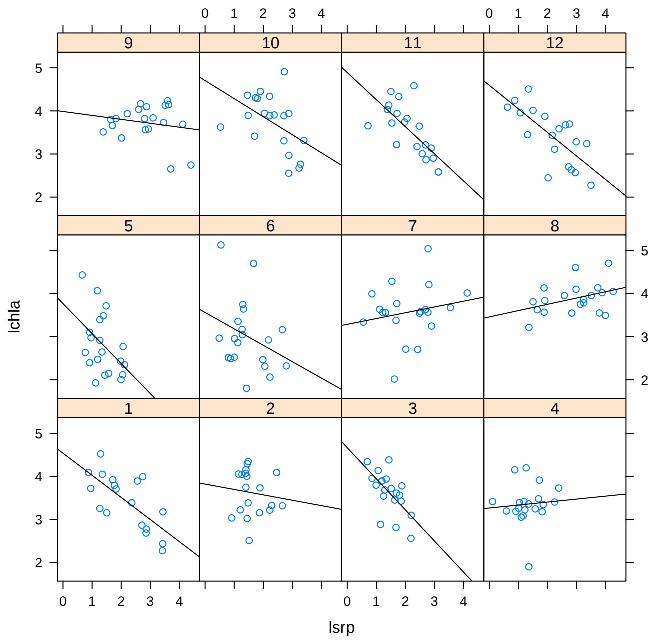
   When we include month and use an additive model, strong effects emerge.



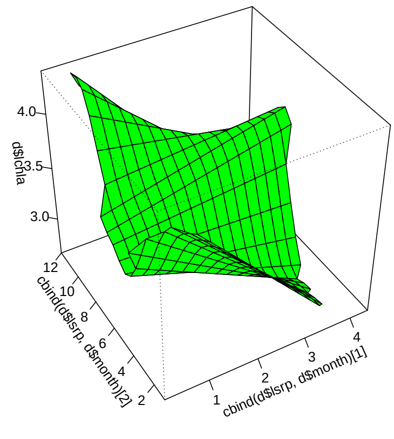
   This is confirmed by a surface plot, using `sm.regression`, with `lchla` as response adn the two covariates `lsrp` and `month`.



A little more insight comes from a lattice plot which separates out the months. This suggests a linear relationship whose slope changes with month. When we fit a linear model with `lsrp` as a covariate which interacts with `month` the residual plots suggest this fits well.



Use of `sm.regression` with a very large smoothing parameter for `lsrp` shows a model where the linear slope varies smoothly with month. This describes the data well and has a clear biological interpretation.


4.0
3.5
3.0
d$chla
12
10
8
6
4
2
cbind(d$lsrp, d$month)[2]
1
2
3
4
cbind(d$lsrp, d$month)[1]

Active Session 2: Wednesday

Pen-and-paper tasks

1. Consider the Bayesian linear model with

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

   Show that in this model

$$\mathbb{E}(\mathbf{y}) = \mathbf{0} \qquad\qquad \mathbf{Var}(\mathbf{y}) = \tau^2\mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I},$$

   and thus the marginal distribution of $\mathbf{y}$ is given by

$$\mathbf{y} \sim \mathsf{N}(\mathbf{0}, \tau^2\mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}),$$

   i.e. the Bayesian linear model is a special case of a Gaussian process with kernel/covariance matrix $\mathbf{K} = \tau^2\mathbf{X}\mathbf{X}^\top$.

2. Consider ridge regression, which minimises the objective function

$$\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|^2.$$

   In this task we will show that we can retrieve the predicted values from ridge regression also from a Gaussian process with the kernel/covariance from question 1.

   (a) Explain why the prediction at a newly observed $\mathbf{x}_0$ is given by $\hat{y}_0 = \mathbf{x}_0^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$.

   (b) Show that

$$(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})\mathbf{X}^\top = \mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})$$

   and deduce from it that

$$\mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})^{-1} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top.$$

   (c) Using the formula from part (b) show that an alternative formula for $\hat{y}_0$ is given by

$$\hat{y}_0 = \mathbf{x}_0^\top\mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})^{-1}\mathbf{y}.$$

   If $\mathbf{X}$ is a $n \times p$ matrix, then this formula is based on the inverse of a $n \times n$ matrix, whereas the formula from part (a) is based on the inverse of a $p \times p$ matrix.
   Also, the formula involving $\mathbf{X}\mathbf{X}^\top$ depends on the data only through inner products, which then allows us to apply the kernel trick to ridge regression.

   (d) Suppose we assume that the data come from a Gaussian process with $\mathbf{K} = \tau^2\mathbf{X}\mathbf{X}^\top$ and $\mathbf{k}_0 = \tau^2\mathbf{X}\mathbf{x}_0$. State the predictive mean for y_0 for this choice of kernel/covariance matrix and compare it to the result from part (c).

3. In this task you will show that the mean of the posterior predictive distribution of a Gaussian process can also be obtained without making any assumption of Gaussianity, just in the same way as the estimated regression coefficient in the normal linear model can also be found only resorting to the principle of least squares (and then the estimate can be shown to be best linear unbiased estimate).

   Like in the lectures we will consider simple case of a process with mean zero ("simple kriging").

   Consider a stochastic process for which observations $y_1 = y(\mathbf{x}_1)$, ..., $y_n = y(\mathbf{x}_n)$ at 'locations' $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are available.

   Assume that the process has zero mean, i.e. $\mathbb{E}(y_i) = 0$ and $\text{Cov}(y_i, y_j) = k(\mathbf{x}_i, \mathbf{x}_j)$ (for $i \neq j$) and $\text{Var}(y_i) = k(\mathbf{x}_i, \mathbf{x}_i) + \sigma^2$. Using the notation from the lectures,

$$\text{Var}(\mathbf{y}) = \begin{bmatrix} \text{Var}(y_1) & \ldots & \text{Cov}(y_1, y_n) \\ \vdots & \ddots & \vdots \\ \text{Cov}(y_1, y_n) & \ldots & \text{Var}(y_n) \end{bmatrix} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) + \sigma^2 & \ldots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \ldots & k(\mathbf{x}_n, \mathbf{x}_n) + \sigma^2 \end{bmatrix} = \mathbf{K} + \sigma^2\mathbf{I}.$$

   Suppose that we want to construct an estimate of $y_0 = y(\mathbf{x}_0)$ at a new location $\mathbf{x}_0$.

   Using the notation from the lectures,

$$\text{Cov}(\mathbf{y}, y_0) = \begin{bmatrix} \text{Cov}(y_1, y_0) \\ \vdots \\ \text{Cov}(y_n, y_0) \end{bmatrix} = \begin{bmatrix} k(x_1, x_0) \\ \vdots \\ k(x_n, x_0) \end{bmatrix} = \mathbf{k}_0$$

$$\text{Var}(y_0) = k(x_0, x_0) + \sigma^2 = k_{00} + \sigma^2$$

   We now want to construct a linear estimate of $y_0 = y(\mathbf{x}_0)$, i.e. we want to construct an estimate of the form

$$\hat{y}_0 = \sum_{i=1}^{n} w_i y_i$$

   In this task we will address the question what the optimal choice for $\mathbf{w} = \begin{bmatrix} w_1, \ldots, w_n \end{bmatrix}^\top$ would be.

   (a) Show that for any fixed choice of $\mathbf{w}$, $\mathbb{E}(\hat{y}_0) = 0$, thus $\hat{y}_0$ is an unbiased estimate of y_0.

   (b) Explain why

$$\text{Var}\left(\begin{bmatrix} \mathbf{y} \\ y_0 \end{bmatrix}\right) = \begin{bmatrix} \mathbf{K} + \sigma^2\mathbf{I} & \mathbf{k}_0 \\ \mathbf{k}_0^\top & k_{00} + \sigma^2 \end{bmatrix}.$$

   (c) Show that the variance of the predictive error is given by

$$\text{Var}(\hat{y}_0 - y_0) = \mathbf{w}^\top(\mathbf{K} + \sigma^2\mathbf{I})\mathbf{w} - 2\mathbf{k}_0^\top\mathbf{w} + k_{00} + \sigma^2.$$

   Hint: Write $\hat{y}_0 - y_0 = \begin{bmatrix} \mathbf{w} \\ -1 \end{bmatrix}^\top \begin{bmatrix} \mathbf{y} \\ y_0 \end{bmatrix}$ *and use the formula for the covariance of linear combinations.*

   (d) Show that the optimal choice of $\mathbf{w}$ minimising the variance of the predictive error is given by

$$\mathbf{w} = (\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{k}_0$$

   and thus the best linear (unbiased) prediction (BLUP) is given by

$$\hat{y}_0 = \mathbf{w}^\top\mathbf{y} = \mathbf{k}_0^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y},$$

   which is the mean of the posterior predictive distribution of a Gaussian process.
   Hint: Take the derivative of the expression from part (c) with respect to $\mathbf{w}$.

Computer tasks

Gaussian processes and Kriging

The command
`load(url("http://www.stats.gla.ac.uk/~levers/apts2.RData"))`
will make all datasets and illustrations need for this session available in your workspace.

1. (a) Launch the function `gp.prior.cartoon()`. It shows a plot with three panels. The first panel shows the corvariance function, the second shows the covariance/kernel matrix for a set of equally-spaced observations and the third panel shows a few draws from that prior.
   Explore what effect changing the different parameters of the Matérn kernel has.

   (b) Launch the function `gp.post.cartoon()`. It shows a plot of draws from the posterior distribution of a Gaussian process for a toy example using the chosen values of the hyperparameters of the Matèrn kernel.
   Explore what effect changing the different parameters of the Matérn kernel has.

2. In this task you will fit a Gaussian process to the univariate data from the first computer lab. For example, to use the radiocarbon data,

```
x <- radiocarbon$cal.age
y <- radiocarbon$rc.age
```

   For the sake of simplicity, we will work with a square exponential kernel

$$k(x_i, x_j) = \tau^2 \exp(-\rho \cdot (x_i - x_j)^2).$$

   (a) Plot the data.

   (b) You can create the matrix $\mathbf{K}$ using the code

```
tau2 <- 10
sigma2 <- 0.1
rho <- 30
K <- tau2 * exp(-rho*outer(x, x, "-")^2)
```

   The values of the hyperparameters are at this stage arbitrary, but will influence the quality of the fit.

   (c) The predictions at the observed values ("fitted valued") are given by

$$\hat{\mathbf{y}} = \mathbf{K}(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}.$$

   Calculate the predictions and add them to the plot from part (a). You might have to change the hyperparameters to obtain a good fit.

   (d) You can estimate the hyperparameters for example using the package `mlegp` (the hyperparameters are tuned as part of the fitting process).

```
library(mlegp)
fit <- mlegp(x , y, nugget=TRUE)
yhat <- predict(fit)
```

   Add the fitted values from `mlegp` to your plot.

3. In this task you will use the R package `geoR` to perform kriging, i.e. to fit a Gaussian process model to a spatial data set.

In this task we will use the data set `mackerel`. It records the abundance of mackerel eggs off the coast of north-western Europe, from a multi-country survey in 1992. In this task we will only use the following columns

`Density`	egg density
`mack.lat`	latitude of sampling position
`mack.long`	longitude of sampling position

The data can be loaded into R using the code

```
library(sm)
data(mackerel)
# Make the longitude go the right way round
mackerel <- transform(mackerel, mack.long=-mack.long)
```

(a) Plot the sampling locations.

(b) Augment your plot from part (a) so that it also shows the observed egg densities.

(c) To be able to use the data in `geoR` we need to convert the data set to an object of the class `geodata`.

```
library(geoR)
data <- as.geodata(mackerel[-c(214,224,228,231),
                            c("mack.long","mack.lat","Density")])
```

We have also deleted four records, such that we only have one record for each location.

(d) It is best to estimate the hyperparameters using (restricted) maximum likelihood or using a Bayesian approach. The REML estimation can be performed using

```
reml.est <- likfit(data, ini.cov.pars=c(10,2), lik.method = "RML")
```

Historically, parameters of the covariance function (kernel) were estimated by constructing an empirical variogram and then fitting the covariance function to that variogram. This is best avoided, but for illustration the code below plots the empirical variogram and overlays the covariance function with the parameters estimated in part (d).

```
# Estimate variogram empirically
vario <- variog(data)
plot(vario)
# Plot variogram corresponding to parameter estimated from above
lines(reml.est, col="blue")
```

(e) With the hyperparameters estimated, we can now perform kriging. We first need to create a grid on which we want to predict.

```
# Create grid of locations for predictions
long=seq(min(mackerel$mack.long), max(mackerel$mack.long), len=50)
lat=seq(min(mackerel$mack.lat),max(mackerel$mack.lat), len=50)
new.locations <- expand.grid(long=long, lat=lat)
# Perform the kriging
kc <- krige.conv(data, loc = new.locations,
                 krige = krige.control(obj.m = reml.est))
# Show image of estimated densities
image(kc, loc = new.locations)
```

(f) Can you repeat the analysis, but now using local smoothing and/or splines rather than kriging for the estimation of the spatial effects?

Active Session 2: Wednesday – Model answers

```
## Error in eval(expr, envir, enclos): object 'opts_chunk' not found
```

Pen-and-paper tasks

1. Using the law of the iterated conditional expectation and variance,

$$\begin{aligned}
\mathbb{E}(\mathbf{y}) &= \mathbb{E}_{\boldsymbol\beta}\left(\mathbb{E}_{\mathbf{y}|\boldsymbol\beta}(\mathbf{y})\right) = \mathbb{E}_{\boldsymbol\beta}(\mathbf{X}\boldsymbol\beta) = \mathbf{X}\mathbb{E}_{\boldsymbol\beta}(\boldsymbol\beta) = \mathbf{0} \\
\mathbf{Var}(\mathbf{y}) &= \mathbf{Var}_{\boldsymbol\beta}\left(\mathbb{E}_{\mathbf{y}|\boldsymbol\beta}(\mathbf{y})\right) + \mathbb{E}_{\boldsymbol\beta}\left(\mathbf{Var}_{\mathbf{y}|\boldsymbol\beta}(\mathbf{y})\right) \\
&= \mathbf{Var}_{\boldsymbol\beta}(\mathbf{X}\boldsymbol\beta) + \mathbb{E}_{\boldsymbol\beta}\left(\sigma^2\mathbf{I}\right) = \mathbf{X}\mathbf{Var}_{\boldsymbol\beta}(\boldsymbol\beta)\mathbf{X}^\top + \sigma^2\mathbf{I} \\
&= \tau^2\mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}
\end{aligned}$$

2. (a) In ridge regression,

$$\begin{aligned}
\hat{\boldsymbol\beta} &= (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y} \\
\hat{y}_0 &= \mathbf{x}_0^\top\hat{\boldsymbol\beta} = \mathbf{x}_0^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}
\end{aligned}$$

   (b) Using

$$(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})\mathbf{X}^\top = \mathbf{X}^\top\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{X}^\top = \mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})$$

   and multiplying from the left by $(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}$ and from the right by $(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})^{-1}$ we obtain

$$\mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})^{-1} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top$$

   (c) From part (a),

$$\begin{aligned}
\hat{y}_0 &= \mathbf{x}_0^\top\underbrace{(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top}_{=\mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top+\lambda\mathbf{I})^{-1}}\mathbf{y} \\
&= \mathbf{x}_0^\top\mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})^{-1}\mathbf{y}
\end{aligned}$$

   (d) The predictive mean is given by

$$\begin{aligned}
\mathbf{k}_0^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y} &= \tau^2\mathbf{x}_0^\top\mathbf{X}^\top(\tau^2\mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I})^{-1}\mathbf{y} \\
&= \mathbf{x}_0^\top\mathbf{X}^\top\left(\mathbf{X}\mathbf{X}^\top + \frac{\sigma^2}{\tau^2}\mathbf{I}\right)^{-1}\mathbf{y},
\end{aligned}$$

   which is, if we set $\lambda = \frac{\sigma^2}{\tau^2}$, the same as the expression obtained in the previous part.

3. (a) For any fixed choice of $\mathbf{w}$,

$$\mathbb{E}(\hat{y}_0) = \mathbb{E}\left(\sum_{i=1}^n w_i y_i\right) = \sum_{i=1}^n w_i\underbrace{\mathbb{E}(y_i)}_{=0} = 0$$

(b) We have

$$\operatorname{Var}\left(\begin{bmatrix}\mathbf{y}\\ y_0\end{bmatrix}\right) = \begin{bmatrix}\operatorname{Var}(\mathbf{y}) & \operatorname{Cov}(\mathbf{y}, y_0)\\ \operatorname{Cov}(y_0, \mathbf{y}) & \operatorname{Var}(y_0)\end{bmatrix} = \begin{bmatrix}\mathbf{K}+\sigma^2\mathbf{I} & \mathbf{k}_0\\ \mathbf{k}_0^\top & k_{00}+\sigma^2\end{bmatrix}.$$

(c) We have

$$\begin{aligned}\operatorname{Var}(\hat{y}_0 - y_0) = \operatorname{Var}\left(\begin{bmatrix}\mathbf{w}\\ -1\end{bmatrix}\begin{bmatrix}\mathbf{y}\\ y_0\end{bmatrix}\right) &= \begin{bmatrix}\mathbf{w}\\ -1\end{bmatrix}^\top \operatorname{Var}\left(\begin{bmatrix}\mathbf{y}\\ y_0\end{bmatrix}\right)\begin{bmatrix}\mathbf{w}\\ -1\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{w}\\ -1\end{bmatrix}^\top \begin{bmatrix}\operatorname{Var}(\mathbf{y}) & \operatorname{Cov}(\mathbf{y}, y_0)\\ \operatorname{Cov}(y_0, \mathbf{y}0) & \operatorname{Var}(y_0)\end{bmatrix}\begin{bmatrix}\mathbf{w}\\ -1\end{bmatrix}\\ &= \mathbf{w}^\top(\mathbf{K}+\sigma^2\mathbf{I})\mathbf{w} - 2\mathbf{k}_0^\top\mathbf{w} + k_{00} + \sigma^2\end{aligned}$$

(d) We first find the derivative of this variance with respect to $\mathbf{w}$ and set it to $\mathbf{0}$.

$$\frac{\partial}{\partial \mathbf{w}}\mathbf{w}^\top(\mathbf{K}+\sigma^2\mathbf{I})\mathbf{w} - 2\mathbf{k}_0^\top\mathbf{w} + k_{00} + \sigma^2 = 2(\mathbf{K}+\sigma^2\mathbf{I})\mathbf{w} - 2\mathbf{k}_0 = \mathbf{0},$$

which gives

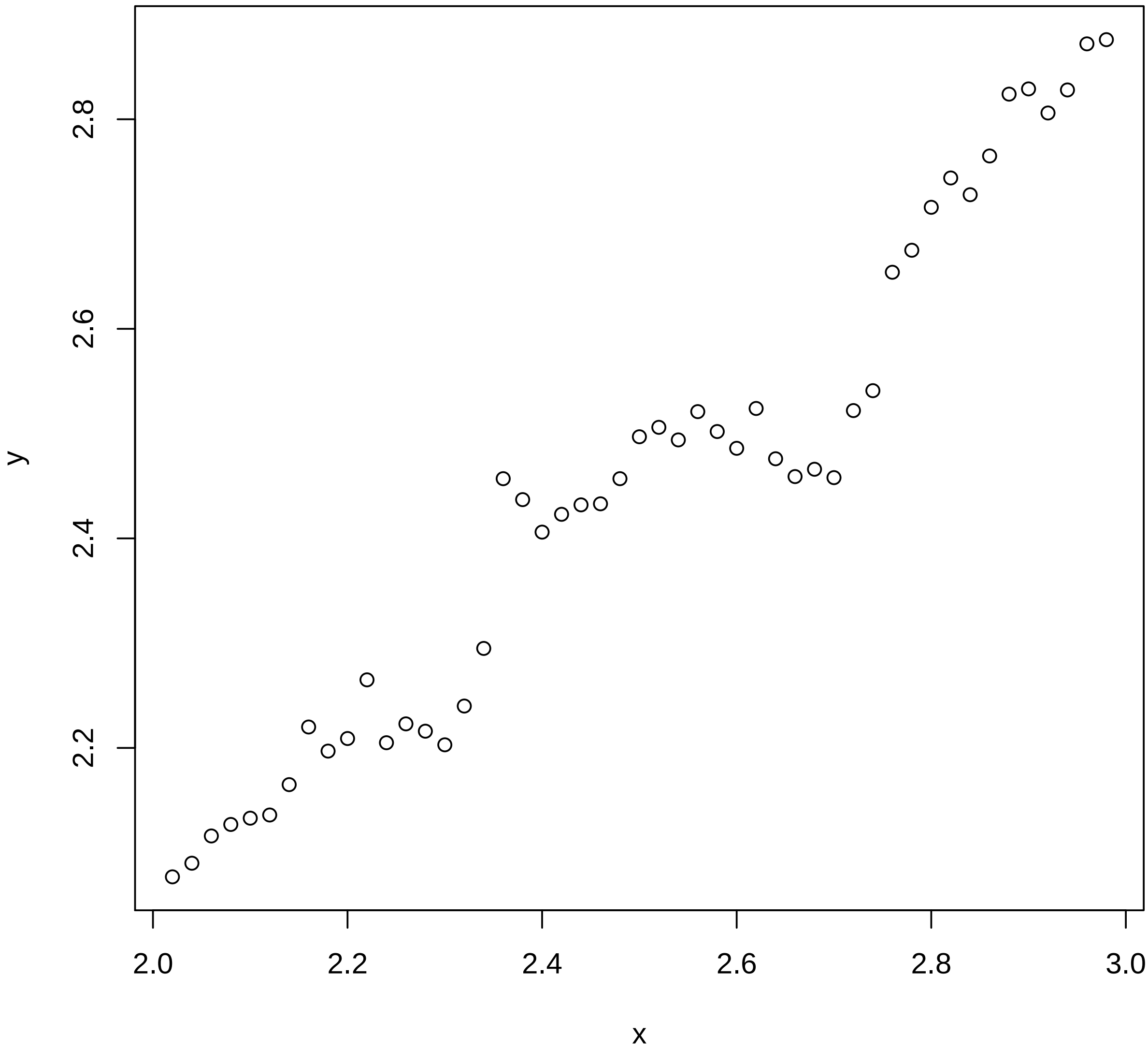
$$\begin{aligned}2(\mathbf{K}+\sigma^2\mathbf{I})\mathbf{w} &= 2\mathbf{k}_0\\ \mathbf{w} &= (\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{k}_0\end{aligned}$$

and thus

$$\hat{y}_0 = \mathbf{w}^\top\mathbf{y} = \mathbf{k}_0^\top(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{y}.$$

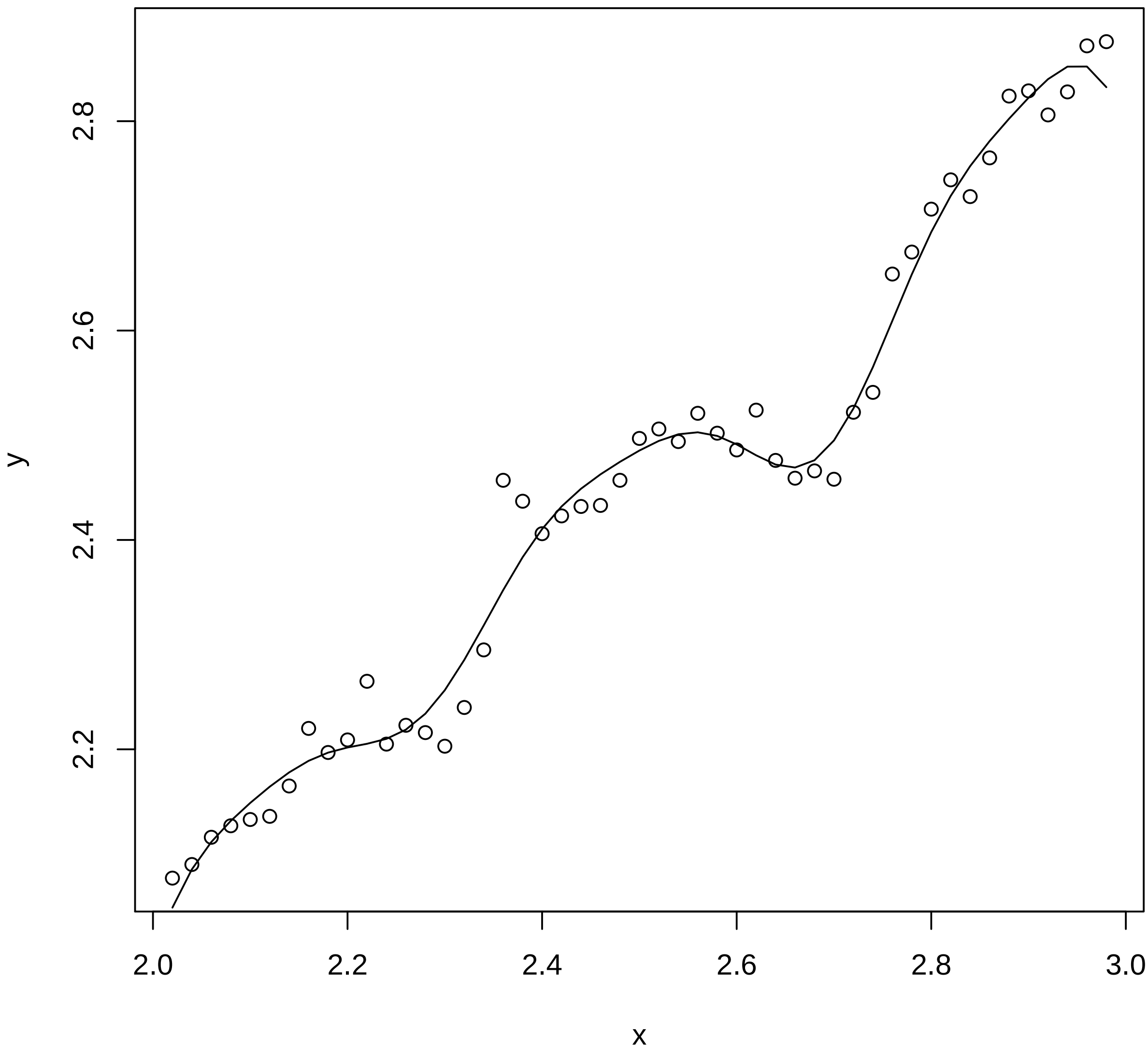
Computer tasks

1. …
2. (a)
```
x <- radiocarbon$cal.age
y <- radiocarbon$rc.age
plot(x, y)
```



(b)
```
tau2 <- 10
sigma2 <- 0.1
rho <- 30
K <- tau2 * exp(-rho*abs(outer(x, x, "-"))^2)
```

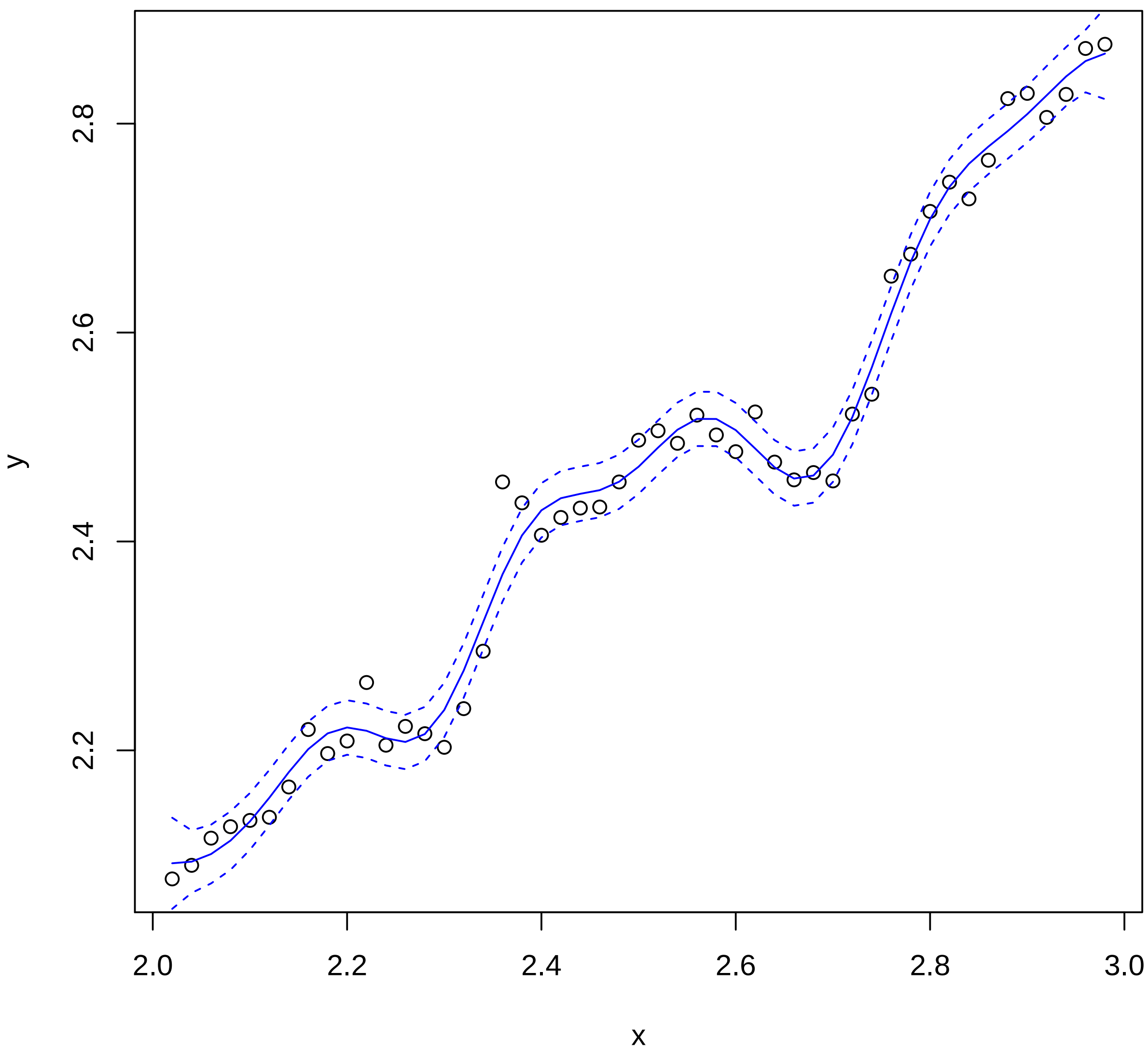
(c)
```
yhat <- K%*%solve(K+sigma2*diag(nrow(K)), y)
plot(x, y)
lines(x, yhat)
```



(d)
```
library(mlegp)
fit <- mlegp(x , y, nugget=TRUE)

##
## ========== FITTING GP # 1 ==============================
## intial_scaled nugget is 18.128134
## running simplex # 1...
## ...done
## ...simplex #1 complete, loglike = 78.349945 (convergence)
## running simplex # 2...
## ...done
## ...simplex #2 complete, loglike = 78.349945 (convergence)
## running simplex # 3...
## ...done
## ...simplex #3 complete, loglike = 78.349945 (convergence)
## running simplex # 4...
## ...done
## ...simplex #4 complete, loglike = 78.349945 (convergence)
## running simplex # 5...
```

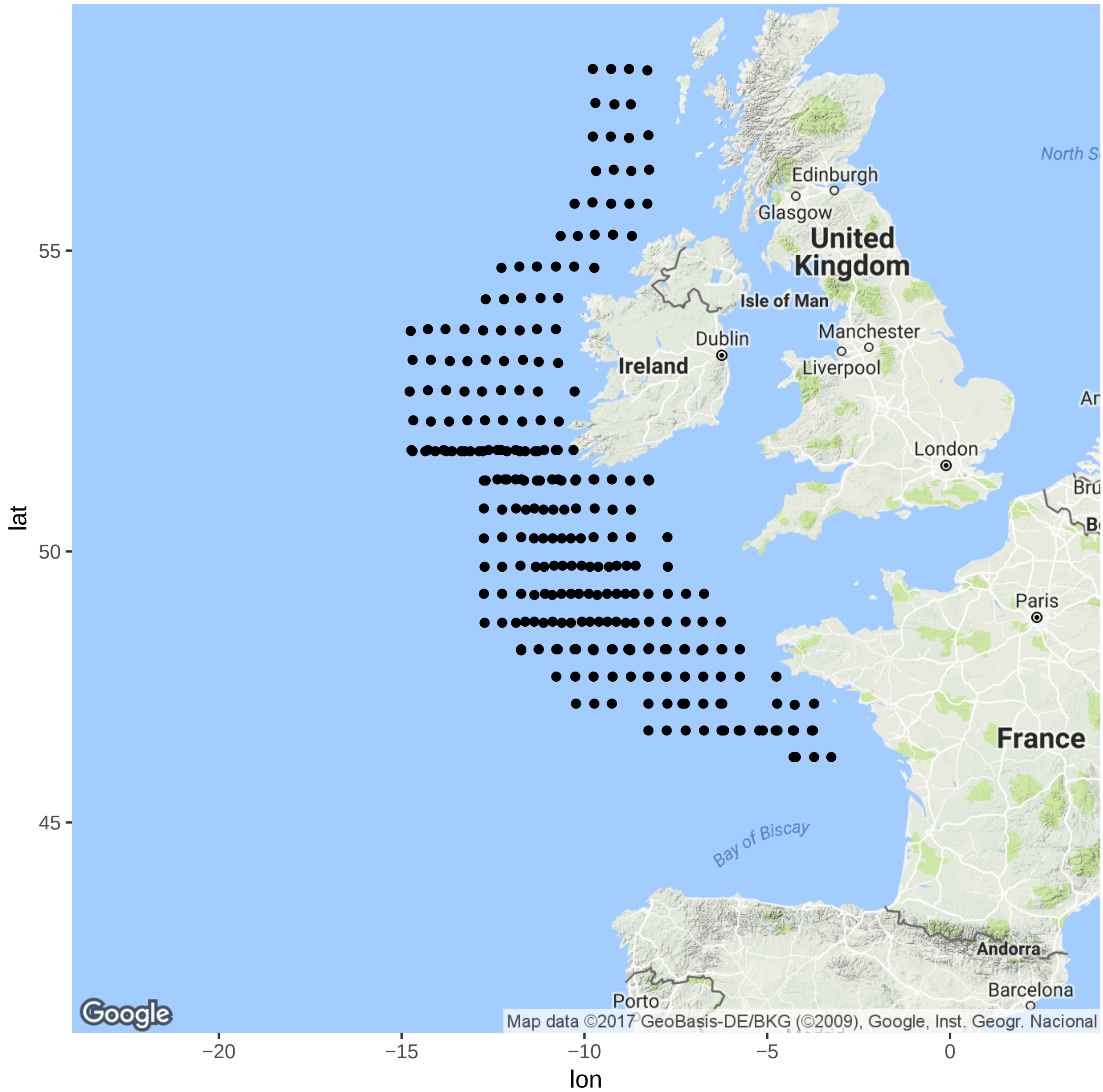
```
## ...done
## ...simplex #5 complete, loglike = 78.349945 (convergence)
##
## using L-BFGS method from simplex #4...
##  iteration: 1,loglike = 78.349945
## ...L-BFGS method complete
##
## Maximum likelihood estimates found, log like =  78.349945
## addNuggets...
## creating gp object......done
```

```
yhat <- predict(fit)
plot(x, y)
lines(x, yhat, col="blue")
# You can add confidence bands using ...
yhat <- predict(fit, se.fit=TRUE)
lines(x, yhat$fit+1.96*yhat$se.fit, col="blue", lty=2)
lines(x, yhat$fit-1.96*yhat$se.fit, col="blue", lty=2)
```

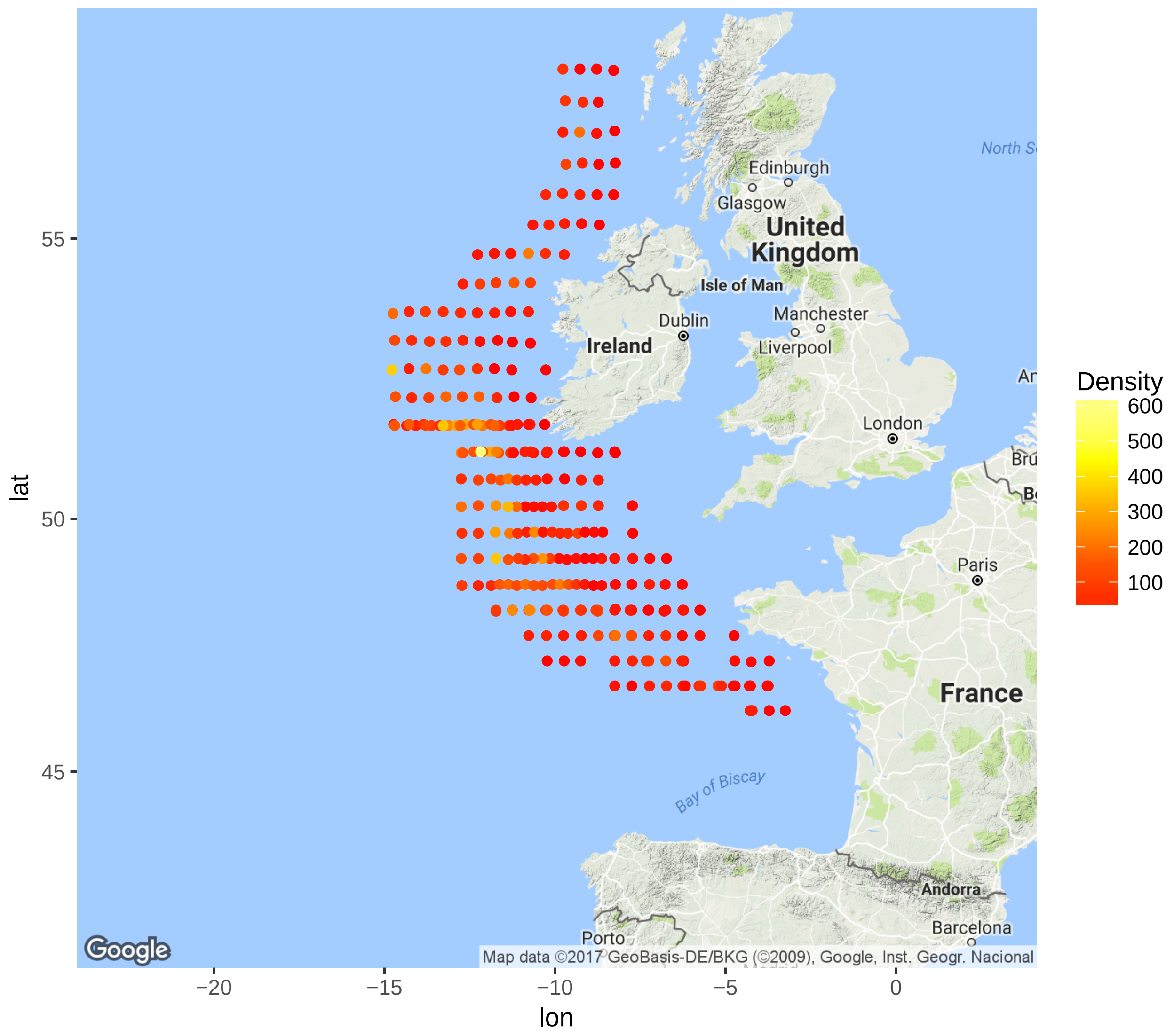


3. (a)
```r
library(sm)
## Package 'sm', version 2.2-5.4: type help(sm) for summary information
data(mackerel)
# Make the longitude go the right way round
mackerel <- transform(mackerel, mack.long=-mack.long)
```

(b)
```r
library(ggmap)
## Loading required package: ggplot2
map <- get_map(location=c(lon=mean(mackerel$mack.long),
                          lat=mean(mackerel$mack.lat)), zoom=5)
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=50.590681,-
9.977993&zoom=5&size=640x640&scale=2&maptype=terrain&language=en-EN&sensor=false
ggmap(map) + geom_point(aes(x=mack.long, y=mack.lat),
                        data=mackerel)
```



(c)
```r
ggmap(map) + geom_point(aes(x=mack.long, y=mack.lat, colour=Density),
                        data=mackerel) +
    scale_color_gradientn(colours = heat.colors(5))
```



(d)
```r
library(geoR)

## ---------------------------------------------------------
##  Analysis of Geostatistical Data
##  For an Introduction to geoR go to http://www.leg.ufpr.br/geoR
##  geoR version 1.7-5.2 (built on 2016-05-02) is now loaded
## ---------------------------------------------------------

data <- as.geodata(mackerel[-c(214,224,228,231),
                   c("mack.long","mack.lat","Density")])
```

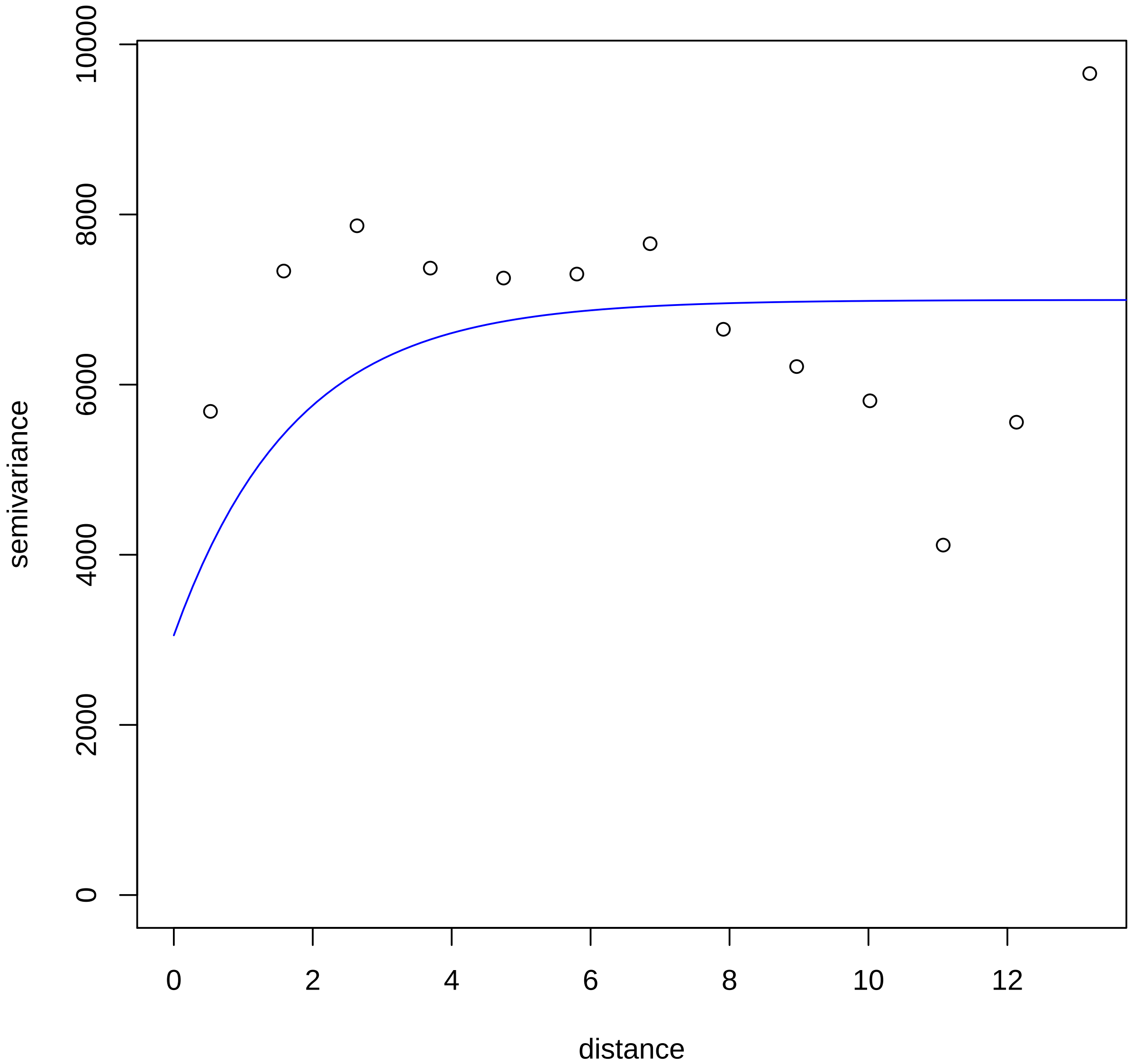
(e)
```r
reml.est <- likfit(data, ini.cov.pars=c(10,2), lik.method = "RML")

## ---------------------------------------------------------------
## likfit: likelihood maximisation using the function optim.
## likfit: Use control() to pass additional
```

```
##           arguments for the maximisation function.
##          For further details see documentation for optim.
## likfit: It is highly advisable to run this function several
##          times with different initial values for the parameters.
## likfit: WARNING: This step can be time demanding!
## ---------------------------------------------------------------
## likfit: end of numerical maximisation.
```

```
vario <- variog(data)
```

```
## variog: computing omnidirectional variogram
```

```
plot(vario)
lines(reml.est, col="blue")
```

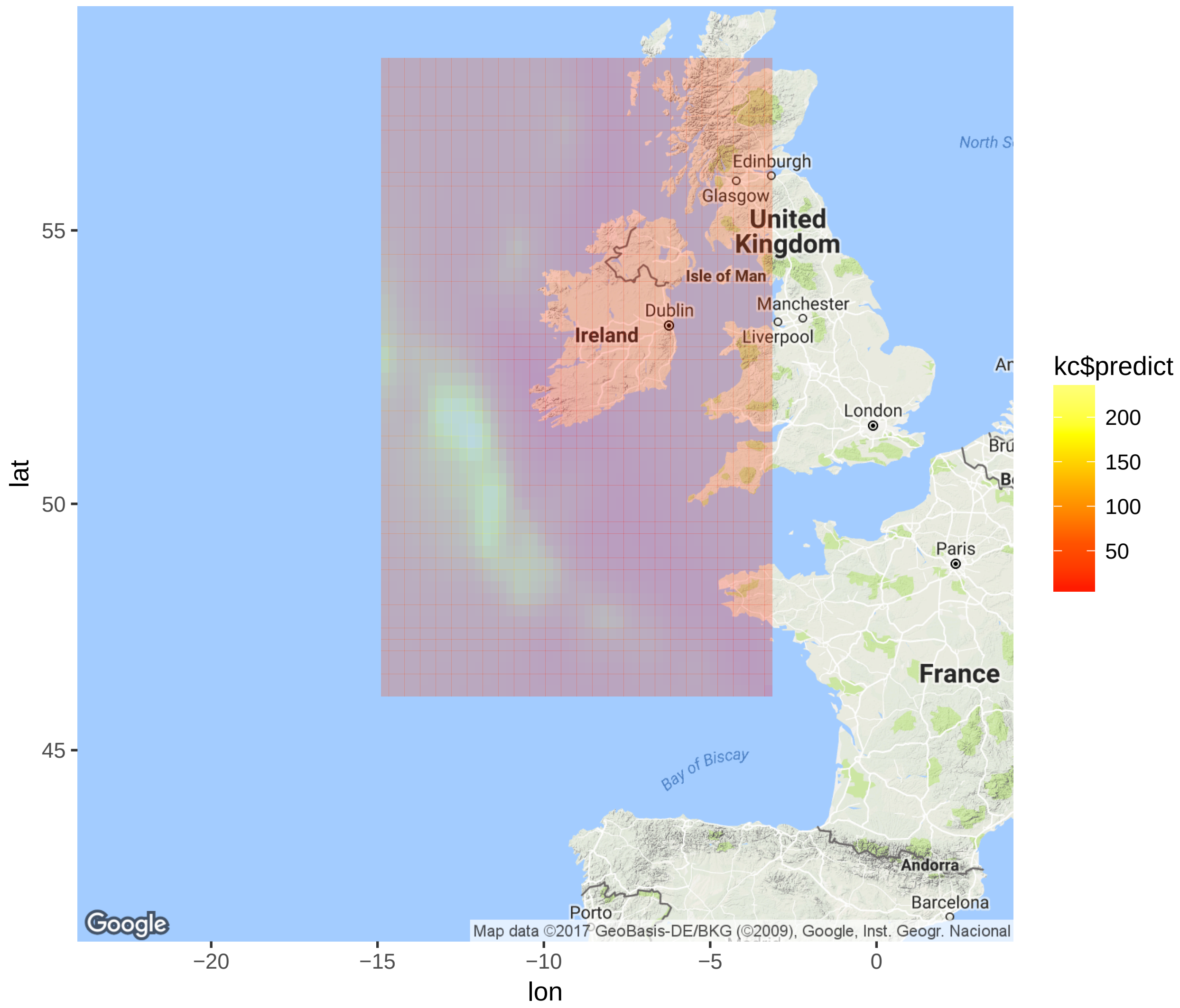


(f)
```
long=seq(min(mackerel$mack.long), max(mackerel$mack.long), len=50)
lat=seq(min(mackerel$mack.lat),max(mackerel$mack.lat), len=50)
new.locations <- expand.grid(mack.long=long, mack.lat=lat)
# Perform the kriging
kc <- krige.conv(data, loc = new.locations,
```

```r
                        krige = krige.control(obj.m = reml.est))
```

```
## krige.conv: model with constant mean
## krige.conv: Kriging performed using global neighbourhood
```

```r
# Show image of estimated densities
 ggmap(map) + geom_tile(aes(x=mack.long, y=mack.lat, fill=kc$predict),
                        data=new.locations, alpha=0.25) +
     scale_fill_gradientn(colours = heat.colors(5))
```

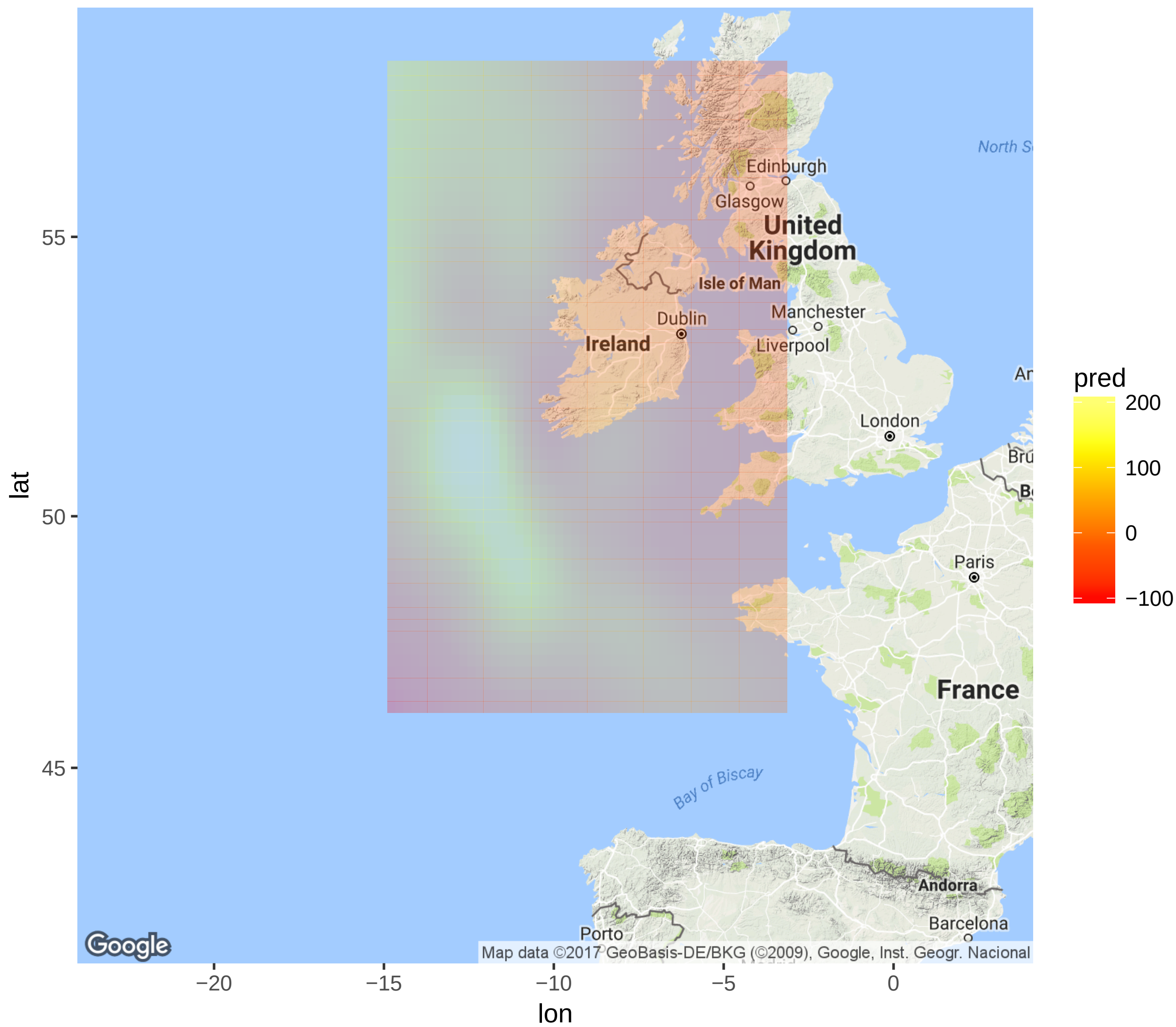


(g)
```r
library(mgcv)
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.8-18. For overview type 'help("mgcv-package")'.
```

```r
model <- gam(Density~s(mack.long, mack.lat), data=mackerel)
pred <- predict(model, new.locations)
ggmap(map) + geom_tile(aes(x=mack.long, y=mack.lat, fill=pred),
                        data=new.locations, alpha=0.25) +
     scale_fill_gradientn(colours = heat.colors(5))
```



pred
200
100
0
−100
lat
55
50
45
lon
−20
−15
−10
−5
0
Edinburgh
Glasgow
United Kingdom
Isle of Man
Manchester
Liverpool
Dublin
Ireland
London
Paris
France
Bay of Biscay
Andorra
Barcelona
Porto
Map data ©2017 GeoBasis-DE/BKG (©2009), Google, Inst. Geogr. Nacional

Active Session 3: Thursday

There are no pen-and-paper tasks for this session.

Computer tasks

The code for this session is available at

`http://www.stats.gla.ac.uk/~adrian/apts/case-studies.R`

1. **Simulating Gaussian processes**

   Some insight on the nature of Gaussian processes can be gained by generating simulations with different parameter settings. The function `rp.geosim()` in the `rpanel` package for `R` can do this for you, using buttons and sliders.

   (a) Try changing the range parameter in particular.

   (b) Set up a sampling grid and add a nugget effect. Now remove the surface from the display, leaving only the sampled points. Look at the sample variogram and see to what extent this is able to identify the features of the underlying process.

Spatiotemporal models

2. **Simulating spatiotemporal Gaussian processes**

   If the file `spatiotemporal.R` is sourced, this will launch a display of simulated spatiotemporal data. Use the slides to see the persistence over time of the random spatial patterns this produces.

3. **SO_2 over Europe**

   In the 1970's and 1980's there was considerable concern about SO2 air pollution. This was emitted by power stations and other installations and the material rises high in the atmosphere and can travel long distances, causing pollution problems in neighbouring countries. The `SO2` dataset documents values of SO_2, on a log scale, from monitoring stations across Europe from 1990 to 2001. The aim of the monitoring stations was to assess whether increasing European regulatory control of SO_2 emissions was effective.

   The data were collected through the *European monitoring and evaluation programme* (EMEP) and they are available at `www.emep.int`. The data recorded here have been organised into a convenient form for analysis. The data file consists of six variables:

`site`	a site code for the monitoring station
`longitude`	longitude of the monitoring station
`latitude`	latitude of the monitoring station
`year`	year of measurement
`month`	month of measurement
`logSO2`	SO2 measurement on a log scale

   Here are some things for you to consider.

(a) The script gives commands which organise and plot the data over space and time. See whether you can identify spatial and temporal patterns from this.

(b) One of the roles of a model is to clarify the nature and size of different effects. The script shows how to fit and plot a model which is additive in space and time. What do the results show?

(c) A more realistic model would allow space-time interaction. A command is given to fit this (which may take a little time). Use the earlier code to plot this new model and consider the difference it makes.

Analysis of these data is reported in *Spatiotemporal smoothing and sulphur dioxide trends over Europe*, A. W. Bowman, M. Giannitrapani and E. M. Scott; *Applied Statistics*, 58 (2009), 737–752.

Non-Gaussian models

4. **The mackerel egg survey (again)**

We considered this dataset in an earlier session, where an additive model was used to describe the pattern of egg counts in the water samples collected. However, the data collected by Spanish vessels differs from that of most other countries, in that it is largely in the form of presence or absence of eggs in each water sample. The script will lead you through some plots of this. Also use the script to try the following operations.

(a) Explore the (marginal) effect of depth and temperature. In particular, is there any difference from the earlier dataset in the indication of preferred depth at which mackerel lay their eggs?

(b) Fit and interpret an additive model