

MT1007. Fisheries Assessment.

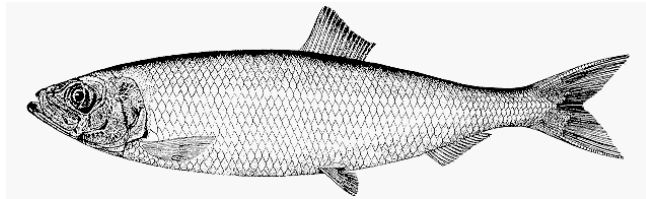
This case study will look at an example taken from the statistical research work here at St Andrews: the use of statistical modelling to estimate the sizes of fish stocks, in order to help manage commercial fisheries. We'll start with some background on fisheries and then look at ways of visualizing the fairly complicated datasets that arise in this sort of work. After that we'll see how some simple generalizations of the linear models that you have already met can be used to model the spatial distributions from fisheries survey data.

1 Fisheries

The total annual catch of wild fish stands at around 9.3×10^{10} kg (1999 figure from UN FAO), or around 1.3 kg of fish per month for every person on the planet (fish farming brings the total up to 12.2×10^{10} kg per year). Around 26 million people worldwide are engaged in fishing (FAO 1990 figure). Clearly fishing is an important source of food (especially protein) and an economically significant activity. Although the growth in fishing over the 20th century has been massive (around a threefold growth between 1948 and 1988, for example), fisheries have been an important food source and of great economic importance for a long time.

Herring and cod, for example, were of major trade importance in the middle ages, and have supported large fisheries ever since. They also supply two of the most spectacular examples of what can go wrong with a fishery.

1.1 The North Sea Herring Fishery



A herring.

Herring have been fished for in the north sea for a long time, often for substantial reward as this quote referring to the fishery around Shetland makes clear.

The Hollanders also repair to these isles [Shetland] in June... for their herring fishery... The fishing trade is very lucrative and enriching to those who closely follow it; it is commonly said that it is the fishing which first raised the Netherlands to that pitch of grandeur and wealth which now they have arrived at: hence some historians call the fishery the Gold-mine of Holland. — J. Brand, A brief description of Orkney, Zetland etc. Edinburgh, 1701.

In fact the herring trade had been important well before this period: for example the Hanseatic league¹ took a keen interest in the herring trade and in the 13th and 14th centuries held a monopoly over large parts of it. As another example, part of the obligations of the city of Norwich to the king consisted of supplying him with “six score fresh herrings in 25 pies” - this obligation seems to have extended back to William the Conqueror.

By the 19th century some fishermen were beginning to complain of diminishing returns. Specifically, the users of traditional “drift nets”, were complaining that their catches were being eroded by fishermen using new longline methods. The influential scientific philosopher T.H. Huxley was appointed to a British Fishing Commission in 1862 to look into the matter, but Huxley was a great believer in the indomitable force of nature and firmly believed that it was impossible to overfish. His commission rejected the fishermen’s concerns about overfishing as “unscientific”, adding that:

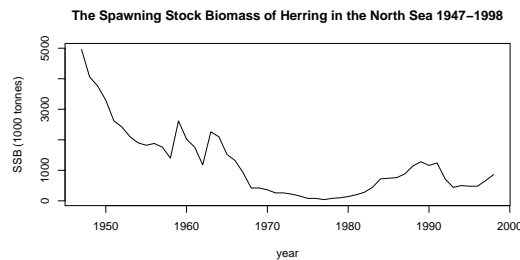
¹A trade federation of northern German towns formed in the 12th century - it operated by controlling the mouths of the major rivers, and thereby trade.

Fishermen, as a class, are exceedingly unobservant of anything about fish which is not absolutely forced upon them by their daily avocations.

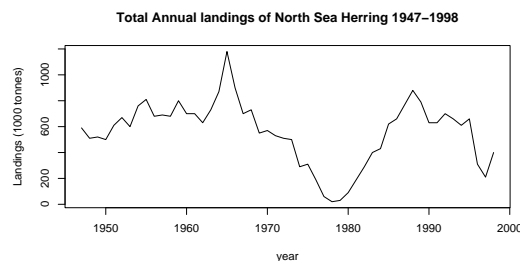
Huxley was appointed to three separate fisheries commissions in total, where his views were taken very seriously. At the 1883 International fisheries meeting in London, which was attended by most of the big fishing nations, Huxley gave an address explaining why fears of overfishing were unscientific and unfounded ...

Any tendency to over-fishing will meet with its natural check in the diminution of the supply, ... this check will always come into operation long before anything like permanent exhaustion has occurred.

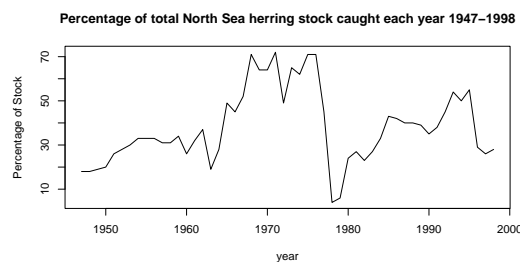
These views were not unusual, but here is what actually happened to the North Sea herring fishery last century (Spawning Stock Biomass is the weight of all herring that are old enough to spawn).



The low point of this graph in 1977 corresponds to a total weight of adult herring in the north sea of 53,000 tonnes: around 1% of what had been there in 1947. From 1947 to 1970 the average annual catch of Herring had been about 700,000 tonnes, at the low point there were less than 200,000 tonnes of herring (mature and immature) in the North sea. Clearly the herring stock had collapsed by 1977. That over-fishing played a major part in this collapse is illustrated by this plot of the total herring catch for each year from 1947–1998.



Notice how the amount taken actually increases while the stock is in decline until the mid 1960's when the stock decline gets so serious that landings start to drop too. Notice also how landings were allowed to creep up again in the 1980's despite only limited stock recovery. The fishing pressure to which the herring stock was subjected is made even clearer, by a plot of the percentage of the total stock (SSB+immatures) being taken each year (all these data are in **herring** in the **mt1007** R package):



To get a feel for the effects on the herring stock of mortality rates up at 70% , note that herring don't become sexually mature until 3-7 years of age. The chances of surviving 3 years if you stand a 0.7 chance of dying each year is only 0.3^3 , about 3%. The chances of surviving 7 years are 0.3^7 , or around 0.02%.

From 1977-1980 there was a complete moratorium on fishing for herring in the North Sea (the catches in that period are herring caught accidentally when fishing for something else). Economically this episode was extremely painful. The stock was being monitored by scientists throughout the period before the crash, but they were slow to pick up the drastic decline that was occurring. Part of the reason for this slowness related to the way in which the size of the stock was estimated, via a method called Virtual Population Analysis (VPA).

VPA consists of estimating the total landings of fish by fisherman each year, along with some information of the distribution of the ages of these fish. With enough years data of this sort, and some assumptions about natural death rates of fish, it is possible to work back in time, adding up the fishermen's catches to arrive at estimates of what the stock size must have been several years ago. Clearly this is not an approach well suited to assessing a rapidly changing situation. To manage fish stocks properly there's a need for rather more direct assessment methods, preferably with fewer assumptions (or at least different ones). The particular example covered later in this case study uses a direct method.

The other problem associated with the herring debacle was the political lag time. The scientists did eventually notice that something was wrong and were pressing for strong conservation measures by 1970, but the politicians did not act on this advice until the problem became unavoidable in 1977. Part of the issue here relates to the confidence that can be attached to estimates of stock size. Unless reasonably precise and reliable estimates of the scale of a problem are available then politicians are unlikely to act. This is where statistical modelling is important in fisheries assessment: it provides the basis for turning fisheries data into useful estimates of stock size.



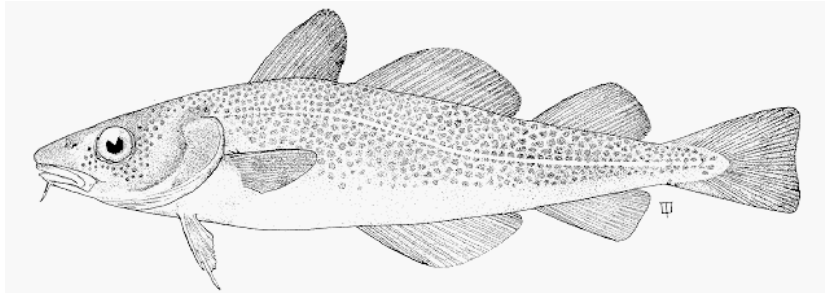
A "purse seiner" taking on board a catch of herring.

1.2 The Newfoundland Cod Fishery

Cod has been a staple food for thousands of years. Because of its very low fat content, Cod is easily preserved by drying, which was an advantage before re Fridgeration. There had been a flourishing trade in Icelandic dried cod for centuries when the Hanseatic league started to try and monopolise the trade in the 1400's. As part of this attempt they cut off supplies of cod to the Merchants of Bristol in 1475. In response, two Bristol merchants went into partnership to fund a voyage to look for new cod fishing grounds in the Atlantic. They were rather secretive about what they found, but didn't seem to have any problems with supplying cod thereafter. It's likely that they had found the cod fishing grounds off Newfoundland, which was named by Giovanni Caboto in 1497 on behalf of the English King. Raimondo di Soncino, Milan's envoy in London reported back on Caboto's return as follows:

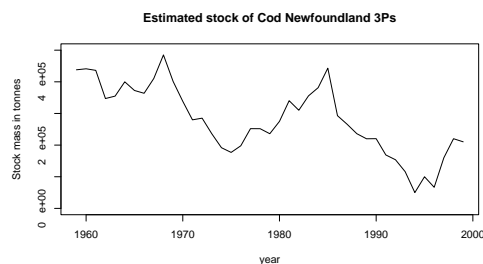
The sea there is swarming with fish which can be taken not only with a net but in baskets let down with a stone, so that it sinks in the water. I have heard this Messer Zoane state so much. These same English, his companions, say that they could bring so much fish that this Kingdom would have no further need of Iceland, from which there comes a very great quantity of the fish called stockfish.

The reports did not turn out to be excessively exaggerated, cod were indeed very abundant off Newfoundland, and a trade rapidly grew up with cod fishing stations established in many localities in New England (including Salem - later famous for its witch hunt).

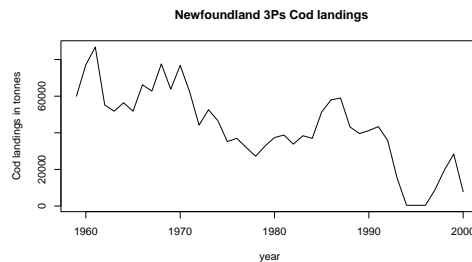


A cod.

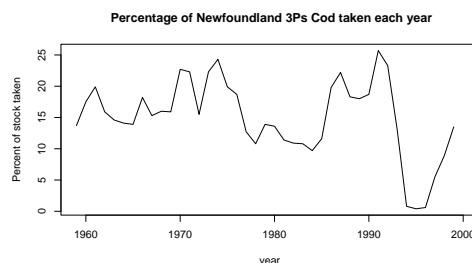
The huge productivity of this fishery lasted until late last century. The following plot from one typical area of the Newfoundland cod stock, shows how it ended.



The situation was not allowed to go quite as far as the North Sea herring: in August 1993 the Fishery was closed, leading to 30,000 layoffs in the fishing industry. It didn't reopen until May 1997. Here is what the landings record looked like for the same period (actually one year longer).



(Data for these plots are in `cod` in the `mt1007` R package). Again the graph of the percentage of the stock taken each year suggests that overfishing is the culprit - notice the way that the fishing mortality stays at a high level right through the stock decline that starts in 1981-82.



Cod reach reproductive age between 8 and 12, and again, at these levels of fishing mortality the number surviving to reach that age will not be high (although modern restrictions aimed at reducing the take of juvenile fish do have some positive impact on this).

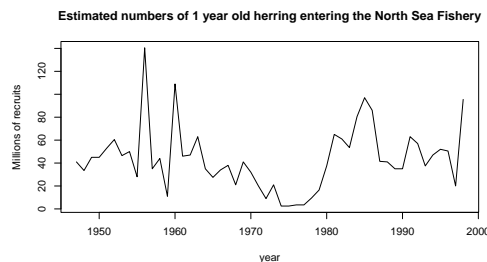
Again with Cod, we see some evidence for a failure to act soon enough, and this again emphasises the need for effective ways of reliably assessing stock size now (rather than some time in the past), if management is going to become responsive enough to prevent this sort of collapse, rather than trying to reverse it after it has happened.

1.3 Why aren't Cod and Herring extinct, and why is stock prediction hard?

Part of the answer to the extinction question probably lies in the fishing bans imposed for both stocks, once the stocks had effectively collapsed. Another part of the answer lies in the truly phenomenal number of eggs spawned by these fish. A female Cod spawns between 500,000 and 5 million eggs each year, depending on size. A female herring 20-40,000. This alone ought to provide some insurance against extinction. However, this reasoning is a little bit dubious: there is a large number of predators that a cod or herring have to escape between egg and adult, as is clear from the fact that the stocks do not grow several thousand fold each year.

The enormous death rates suffered by larval fish lead to some odd effects. Suppose that a fish has the same probability S of surviving each day for the first year of its life, and that each day is independent of all the others. It's probability of surviving the year will then be: S^{365} . It's not hard to see that for the herring, if just 2 eggs are to survive until they are one year old then $S^{365} = 0.0001 \Rightarrow S = 0.975$. Now imagine that the daily survival rate changes by just 1% of it's current value, i.e. it changes to $1.01S$. The survival over a full year will now be $1.01^{365}S^{365}$, that is $1.01^{365} = 37.8$ times its previous value. Decrease the daily survival by just 1% of it's previous value and the annual survival drops to about 2.5% of its previous value. Clearly a rather small change in daily survival rates can make for a rather large change in the number of eggs surviving to adulthood.

This effect may be part of the explanation for the variability of recruitment rates into the stock of fish more than a year old. For example, here is a plot of the recruitment rate for Herring:



Despite this variability in the data, and the forgoing argument about why it should be so, there persisted a hope for some time that it might be possible to predict future fish stocks using models relating recruitment rate to current spawning stock size. All attempts to do so have proved rather futile. To see why, you just need to plot the recruitment rate against the size of the spawning stock that produced those recruits:



... clearly SSB doesn't do a great job at predicting future recruitment.

1.4 How fisheries are managed

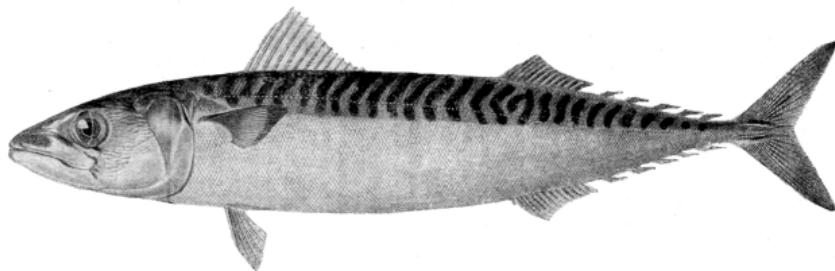
Most major fish stock are now “managed” and have been for some time. That this process has not prevented over-exploitation is clear from the collapses that occurred in the Herring and Cod fisheries: two major stocks. In excess of 60% of fish stocks are probably over-exploited. Within the EU, international working groups of scientists meet and pool as much data as possible in order to come up with an assessment of the size of the stock with which they are concerned. These assessments are used to provide recommendations to the politicians who ultimately make the decisions about how many fish should be taken in any given year. Once the decisions are made, quotas are set, which in theory limit the amount of fish that can be landed.

As mentioned above, a large part of the data that traditionally goes into the stock assessment process is catch data, which has to be obtained from the fishing boats. It is not easy to get the fishers to reveal potentially sensitive data to the fisheries scientists, and there are concerns that there are quite major biases in the data available. The catch data is then used to estimate what the stock must have been in the past, but inevitably the methods for doing this tend to give fairly reliable estimates of the stock size several years ago, and very unreliable estimates of the stock size now. Furthermore, the estimates are conditional on some very hard to test assumptions, in particular relating to the “natural” mortality that the fish suffer.

For these reasons it is extremely useful to have independent estimates of stock size, based on properly designed scientific surveys, rather than catch data. One method for making such estimates is known as the Daily Egg Production Method, and the rest of this case study will be concerned with applying this method to survey data for mackerel.

2 Mackerel and the DEPM

Fisheries managers would rather that the Mackerel:



...did not go the way of the Cod or Herring. As part of the management process for Mackerel they therefore attempt to estimate the spawning stock biomass of mackerel directly using the Daily Egg Production Method (DEPM).

The basic idea behind the DEPM is very simple: Conduct a survey which collects and counts mackerel eggs in order to estimate the total number of eggs that the mackerel spawn in a day over their whole spawning grounds. From this it is possible to work out the total weight of the fish that there must be to produce these eggs.

In practice, to go from number of eggs to weight of adult mackerel involves a second survey which measures the number of eggs released per day per unit body weight of female fish, the proportion of female fish spawning on any given day and the sex ratio of the fish.

The reason for not just surveying adults directly, is that it's extremely difficult to obtain a sample that can be turned into an estimate of mackerel per unit volume of water and hence scaled up to an estimate of the whole stock. This is because the adult mackerel are fast moving, occur in shoals, will tend to avoid nets etc. Eggs on the other hand are nice and passive.

2.1 Graphical investigation of the 1992 Mackerel Egg Survey

As mentioned in the previous section, part of the stock assessment process for mackerel consists of mackerel egg surveys, which are used to estimate the weight of parent fish there must have been to produce these eggs. The surveys consist of research boats going out to sea within a pre-defined survey area, and sampling eggs by pulling a net up through the water, from close to the sea bed to the surface. The material collected in these nets is carefully sorted to find the mackerel eggs, and these are then counted. The number of eggs counted are then converted to density of eggs produced per day, where “density” is defined as numbers per day per square metre of sea surface (irrespective of how deep the sea is at the sample locations).

Some statistical modelling is required to turn the egg density data into estimates of the total number of eggs produced over the whole survey area per day. In fact, to get reasonably precise estimates, it will be necessary to try and model the way in which egg density depends on location and other predictor variables. This can be done most effectively with a slight extension of the linear models that have already been covered. But before modelling the survey data it is important to take a good look at it, to get a feel for what is being modelled.

There is quite a lot of data associated with the mackerel egg surveys, so that finding useful ways of displaying it requires somewhat more advanced graphical methods than we’ve typically used so far in the course. Fortunately these are all easily available in R.

In this case study we’ll examine data from a 1992 mackerel egg survey: the data relating to this survey are in the `mt1007` package. For the moment you will need:

<code>mack</code>	The mackerel egg survey data.
<code>bath</code>	The sea-bed depth over $(-15^{\circ}\text{E}, -1^{\circ}\text{E}) \times (44^{\circ}\text{N}, 58^{\circ}\text{N})$
<code>coast</code>	The coast-line of Europe over the relevant area.
<code>cont.200m</code>	The location of the 200m sea bed depth contour.

All these data frames have their own help pages, and you can always remind yourself of what they contain using something like:

```
> names(bath)
```

Note that all location information in these data frames is given in degrees of longitude and latitude. The lon, lat co-ordinate system covers the globe. 0 longitude is the shortest curve running from pole to pole through Greenwich London (“1E” means one degree longitude east of Greenwich); 0 latitude is the equator (“50N” means 50 degrees of latitude north of the equator - the poles being at 90N and 90S).

The R graphics commands that you will need to use are:

<code>plot()</code>	Used for producing x, y plots.
<code>points()</code>	Used for adding more x, y , points to an existing plot.
<code>lines()</code>	Used to add lines to an existing plot.
<code>contour()</code>	Draw a “contour plot” of x, y, z data.
<code>image()</code>	Plot x, y, z data using colours to represent z .
<code>persp()</code>	Produce an image of a 3D representation of x, y, z data.

Don’t forget that you can use the helps system to find out about any of these. In addition you will need to use some graphics options:

<code>col</code>	Used to set plotting colours.
<code>cex</code>	Used to set the size of plotted points.
<code>pch</code>	Lets you change the default plotting symbol.
<code>xlab</code>	Defines the label for the x axis.
<code>ylab</code>	Defines the label for the y axis.
<code>main</code>	The title for the plot.

2.1.1 Where did the survey happen?

The first thing to do with the data is to look at where the survey took place. To do this load the `mt1007` package and then the survey boundary data frame:



```
> data(bound)
```

This dataframe contains a set of (`lon`,`lat`) points that can be joined up to define the boundary of the survey region. Plot these using:

```
> plot(bound$lon,bound$lat,type="l",xlab="longitude",ylab="latitude",col=6)
```

The resulting plot probably won't give you much feel for where the survey area is. The data frame `coast` contains longitudes and latitudes defining the coast of Europe over an appropriate area. Use the `lines()` function to add this coast to your plot, in a different colour to the survey boundary. Write the commands for doing this in this space:

Now you should have a better general idea of where the survey took place. It's time to look at where the individual net samples were taken. `lon` and `lat`, within data frame `mack` give these locations. Use the `points()` command to add these sampling locations to your plot, again using a different colour to the ones you've used so far. If the symbols used to plot the points are too large, redo the plots, adding the optional argument `cex=0.5` to `points()` when plotting the sample locations. Record the exact commands used to produce your final plot here:

Resize your graphics output window so that the UK is a sensible shape. To do this click on the  button at the top right of the window. Then resize the plot by dragging the  button at bottom right of the window.

2.1.2 How do the estimated egg densities vary with location?

The plot produced in the previous section shows you where the survey happened, but nothing about what it uncovered about egg densities. A useful way of extending the plot to show this information, is to make the size of plotting symbols for the sample locations proportional to egg density.

The optional argument `cex` to the `points()` function controls the size of the symbols used to plot the points. Usually this is set to a single number applying to all points, but you can also supply an array, giving a different size for each point. So, you can produce an array of plot symbol sizes that are proportional to the egg densities, and use this array for `cex`. It's best if `cex` values don't get bigger than 4 or 5, or smaller than 0.2 (if you want the points to appear at all). `mack` contains `egg.dens`, which gives the egg densities measured at each sampling station. Create an array of symbol sizes proportional to the egg densities:

```
> sysi<-mack$egg.dens/150+0.2
```

for example. Now use these symbol sizes to redo your plot from the last section, but this time make the size of the sampling station plotting symbols proportional to the egg density for that station. Record the commands used here:

You should now have a feel for where the high egg densities occur.

2.1.3 How does egg density depend on sea-bed depth?

The Fisheries biologists involved in the survey believe that mackerel like to congregate to spawn around the 200m sea bed depth contour. Data frame `cont.200m` contains a series of `lon`, `lat` points that can be joined up to draw the contour on your plot. Use `lines()` to add the 200m depth contour to your plot in a new colour. Record the command for doing this here:

Also record any observations about where the high egg densities tend to be in relation to the 200m contour:

It's quite instructive to look at the relationship between egg densities and sea bed depth more generally. The data list `bath` contains seabed depth (and elevations above sea-level recorded as negative sea-bed depths). `bath` contains a (169×169) matrix of sea bed depths averaged over 1/12th degree squares arranged on a regular grid: `b.depth` (it's a very bad idea to type `bath$b.depth`). `bath` also contains two arrays `lon` and `lat` (each of length 169) defining the locations of the centres of the grid squares. For example `bath$lon[4]`, `bath$lat[30]` is the location of the centre of the square whose average depth is `bath$b.depth[4,30]`.

One way to look at the seabed depth data is to get R to produce a contour plot of it - basically to draw a map of the sea bed:

```
> contour(bath$lon,bath$lat,bath$b.depth)
```

According to this plot, at roughly what longitude and latitude is the sea-bed deepest?

Another useful way of looking at these data uses the R command `persp()`, which produces "3D" plots. Due to limitations on the Macs, you will need to issue the following commands before trying this out:

```
> dev.off()
> X11(colortype="gray")
```

For the most impressive plot try out the following:

```
> persp(bath$lon,bath$lat,-bath$b.depth,zlim=c(-6000,10000),
+ phi=35,theta=-35,border=NA,shade=0.99,box=FALSE)
```

The negative sign on the bottom depth ensures that the sea bed is lower than the land in this plot. `phi` and `theta` determine the angle from which you view the surface. `border=NA` prevents gridlines being drawn all over the surface, `box=FALSE` turns off the drawing of axes, and the `shade` option controls how diffuse the "lighting" of the surface is. If you have plenty of time, you might want to experiment with some of these options. Before leaving this plot, make sure that you have identified the British Isles and the edge of the continental shelf.

Finally there is a third way of looking at the sea bed depth data, in which the depth is represented on a colour scale. Since you had to turn colour off for the last plot, make sure it gets turned on again by closing the current black and white graphics window:

```
> dev.off()
```

To see the colour coded plotting in action, type:

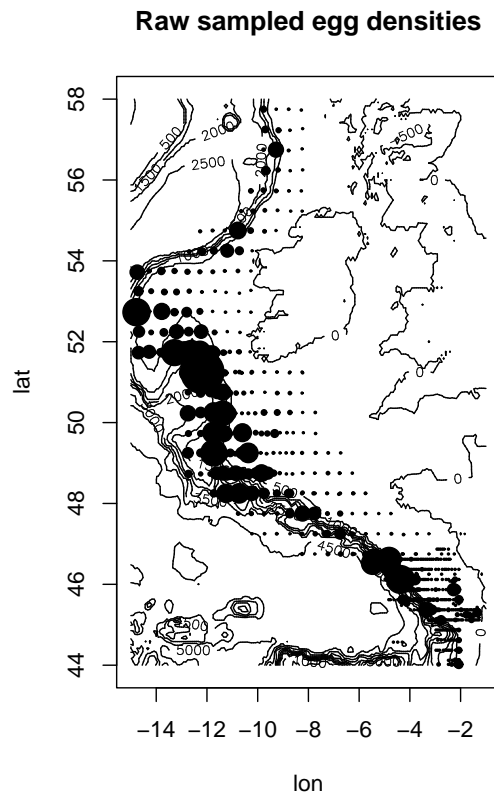
```
> image(bath$lon,bath$lat,bath$b.depth)
```

Identify which end of the colour range corresponds to shallow waters and which to deep:

Now use `lines()` to add in the coastline in a suitably contrasting colour (`col="magenta"` is rather fetching). Being quite careful in your choice of colours, overlay the locations of the sampling stations on this image, making the symbol sizes proportional to density, so that you can see as clearly as possible the relationship between egg density and sea bed depth. Add in the 200m contour line as well. Record the full set of commands for producing this plot here:

do the highest densities of mackerel seem to be around the 200m contour, in shallower water or in deeper water?

As a final exercise write out the commands that would have produced the following boringly black and white plot:



(Note that `pch=19` is the optional argument to `points()` that results in solid circles as symbols)

3 Models suitable for the mackerel egg data

The mackerel survey data that you investigated in the last section has to be turned into an estimate of the total number of eggs produced within the survey area (per day). This would be easy to do if we knew the number of eggs produced per day beneath each and every square metre of sea surface within the survey area: we could just add them up. But what we actually have are some scattered estimates of numbers per square metre per day (the `egg.dens` at the sample `lon`, `lat` positions). To estimate the egg production density beneath each and every square metre of the survey area requires a model, which can be fitted to the observed egg densities. What we need is a model of the basic form:

$$\text{observed egg density} = \text{true egg density} + \text{random variation}$$

Where typically, we may suspect that “true egg density” depends on all sorts of predictor variables, such as location, sea-bed depth, water temperature, etc.

An obvious place to start constructing such models would be to use the linear models that were covered in section 5 of the theory section of the notes. For example suppose that the variables concerned are:

- y_i Estimated number of mackerel eggs produced per day per m^2 of sea surface at sample station i .
- x_{1i} Degrees latitude for station i .
- x_{2i} Degrees of longitude for station i .
- x_{3i} Sea bed depth for station i .

and the x_j ’s are thought to determine the true egg density. We might try the linear model:

$$Y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \epsilon_i$$

where as usual Y_i is the random variable that y_i is an observation of, and the ϵ_i are i.i.d $N(0, \sigma^2)$ random variables. This model can be fitted in the usual way using least squares, and does an exceptionally poor job.

The reason for the inadequacy of the simple linear model is because that the relationship between each of the predictor variables (the x_{ji} ’s) and the response (the y_i ’s) is quite complicated, and not well captured by a straight line. We *could* attempt to fix this by, for example, adding quadratic terms into the linear model:

$$Y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \beta_4 x_{1i}^2 + \beta_5 x_{2i}^2 + \beta_6 x_{3i}^2 + \epsilon_i$$

but this doesn’t do so well either, perhaps we should add cubic terms? or terms like $x_{1i}x_{2i}$ etc? You can see that this approach is going to get quite difficult, since we have several predictor variables and no clear idea of how they should be related to the response variable.

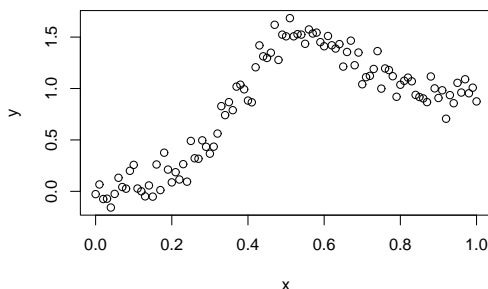
The way out of this impasse, is to step back from the details of the model for a moment and think about the problem in a more general way. All we really know about the mackerel example is that we expect that there should be some smooth curve relating the egg density to latitude, a smooth curve relating egg density to sea bed depth and so on. i.e. we expect true egg density to vary *smoothly* as the predictor variables change, we just don’t know the *form* that this smooth change will take. It would be nice if we could just write down the model for egg densities as something like:

$$y_i = \beta_0 + s_1(x_{1i}) + s_2(x_{2i}) + s_3(x_{3i}) + \epsilon_i \tag{1}$$

where the s_i ’s are just general smooth functions of the predictor variables, to be somehow chosen automatically as part of model fitting. It turns out that it possible to do exactly this.

3.1 Modelling with “smooths”

The basic ideas behind modelling with smooth functions can all be understood by considering a single predictor variable, and can be developed in a simple way from more familiar linear modelling. Consider trying to find a suitable model for these data (available in the `mt1007` package as `smooth.eg`):



The obvious model to try is something like:

$$y_i = s(x_i) + \epsilon_i$$

where s is some smooth function of x . One way of representing s is to use a polynomial (i.e. $\beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \dots$), but how many terms should be included? Looking at the plot of the data, it's clear that a straight line or quadratic won't do the job. A cubic might get somewhere close, but the flat sections at both ends of the plot won't be well matched by a cubic - perhaps a quartic or quintic, will do.

Replacing the “smooth function” s with a quintic, the model becomes:

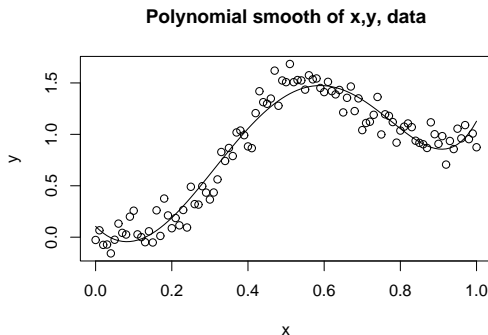
$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \beta_5 x_i^5 + \epsilon_i$$

which can easily be fitted to the data using `lm()` in R . First create arrays containing x_i^2 , x_i^3 etc. and then fit the model:

```
> x2<-x^2;x3<-x^3;x4<-x^4;x5<-x^5
> m1<-lm(y~x+x2+x3+x4+x5)
```

Here is what the fitted model (i.e. the estimated $s(x)$) looks like, overlaid on the data:

```
> plot(x,y,main="Polynomial smooth of x,y, data")
> lines(x,fitted(m1))
```



This fit doesn't look too bad, but unfortunately the approach has some quite bad problems. These are related to the very close correlation between the different terms in the polynomial. For example if we look at the correlation between the x_i^4 values and x_i^5 values for the example data, it is very high indeed:

```
> cor(x4,x5)
[1] 0.9949854
```

As we saw in section 5.3 of the theory section, quite modest correlations can make linear models a bit difficult to interpret: correlations of 0.995 cause real headaches. To see one of the problems the summary of `m1` can be consulted:

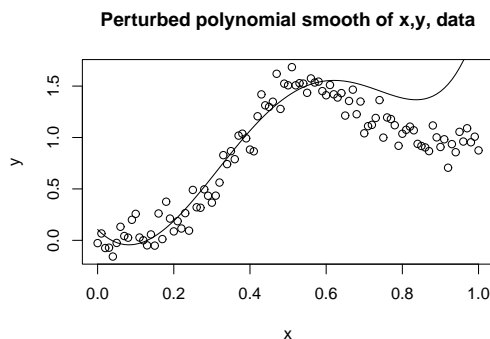
```
> summary(m1)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.1053	0.0745	1.414	0.16073
x	-3.7914	1.5315	-2.476	0.01507 *
x2	25.9022	9.6097	2.695	0.00831 **
x3	-20.6281	24.5010	-0.842	0.40194
x4	-22.9652	27.0626	-0.849	0.39824
x5	22.5051	10.7698	2.090	0.03932 *

...notice how large the standard errors of the estimates are in relation to the size of the estimates. Now let's see what happens to the fitted smooth when just one of the estimated parameters is changed by what should be a statistically insignificant amount. Let's increase β_5 by less than a tenth of its standard error, by adding 1 to its value (changing it from 22.5051 to 23.5051). This is equivalent to adding $1 \times x_i^5$ to each of the fitted values, so it's easy to see what happens to the smooth:

```
> plot(x,y,main="Perturbed polynomial smooth of x,y, data")
> lines(x,fitted(m1)+x5)
```



So, changing one of the parameters of the fitted smooth, by what appeared to be a completely insignificant amount has completely spoiled the fit of the model. This is typical of the sort of problem that arises when the predictor variables are so highly correlated. In fact, if we wanted to use a very flexible smooth curve for these data and use a polynomial with terms up to x_i^{13} , then the problems get so bad that it is actually impossible for R to estimate β_{13} .

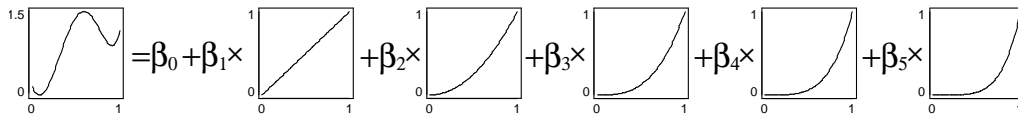
Does this mean that modelling with general “smooth curves” is impossible. Happily not. The problem is quite easy to solve by looking graphically at what is really going on when we represent a smooth using a polynomial, and then modifying the approach a little, to get something that works better.

3.1.1 Splines

Consider the equation for the polynomial smooth that was just fitted to the example data:

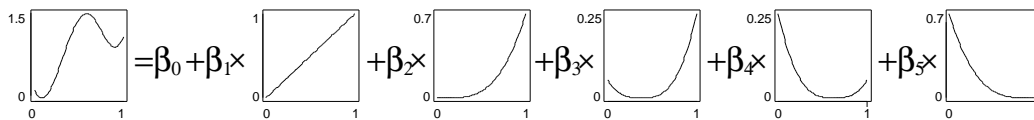
$$s(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4 + \beta_5 x^5$$

If you were to draw this equation (as it was fitted to the data), it would look like this:



(remember that the estimated β_i values are 0.105, -3.79, 25.9, -20.6, -22.7 and 22.5). i.e. the smooth is made up of “a parameter” + “another parameter” \times “a straight line” + “yet another parameter” \times “a quadratic curve”, and so on. The complicated smooth curve is just a weighted sum of extremely simple curves, where the weights are the model parameters.

The problems with the polynomial smooth arises because the simple curves being added up to make it are so very similar to each other. The question then arises: could a better set of simple curves be found which could be added up to make the smooth? The answer is yes. It turns out that there exists a set of simple curves of this sort with very good properties. A pictorial representation of a smooth made up from these curves (and having the same number of parameters as the polynomial we’ve looked at) is:



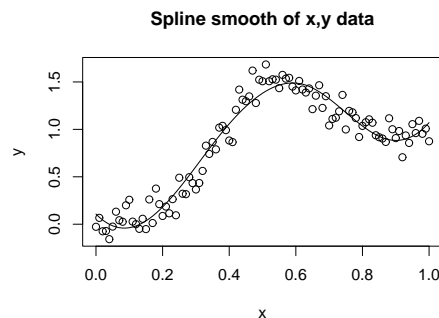
(where the β_i ’s will of course have different values to the β_i ’s of the polynomial). Now the curves multiplying $\beta_2 \dots \beta_5$ all have exactly the same shape, but they are centred at different places on the x axis. As a result they have *much* lower correlations with each other than the curves that make up the polynomial. The equation for the smooth shown above is:

$$s(x) = \beta_0 + \beta_1 x + \beta_2 |x - 0.125|^3 + \beta_3 |x - 0.375|^3 + \beta_4 |x - 0.625|^3 + \beta_5 |x - 0.875|^3$$

and it is an example of a “cubic spline”. The values 0.125, 0.375, 0.625 and 0.875 are chosen to be nicely spread out along the x axis. If we wanted a more complicated smooth, we’d just add up more smooth functions, and have them spaced out closer together. To fit the spline to data all we have to do is to create some new predictor variables, having the values $|x_i - 0.125|^3$, $|x_i - 0.375|^3$ etc. and then use `lm()` in the usual manner. For example:

```
> s1<-abs(x-0.125)^3
> s2<-abs(x-0.375)^3
> s3<-abs(x-0.625)^3
> s4<-abs(x-0.875)^3
> m2<-lm(y~x+s1+s2+s3+s4)
```

Here is what the resulting smooth fit looks like:

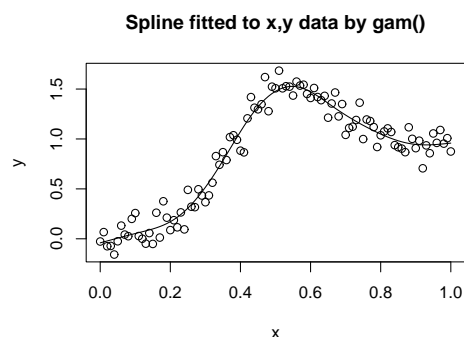


A check of the correlations between the new predictor variables reveals that the largest correlation is now 0.95, with most of the correlations much smaller in magnitude than that. As a result the fit is much more robust to changes in the parameters, and it is *possible* to estimate as many parameters of the model as you have data (although probably not statistically *sensible*).

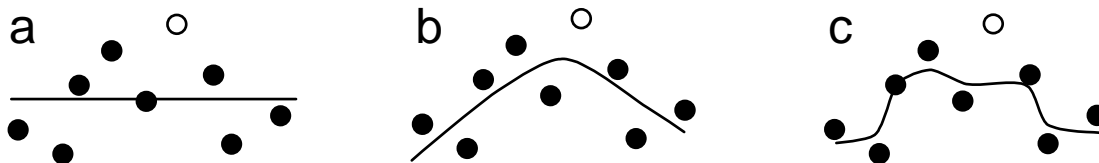
The “spline” approach to estimating smooth relationships between response and predictor variables is very popular in modern statistical modelling. This is partly for the reasons covered above, and partly because there is some quite wonderful mathematics that goes with splines, which suggests that it would be hard to come up with a better way of representing smooths.

Needless to say, for practical modelling (as opposed to teaching), it is never necessary to construct splines in the somewhat tedious manner given above. All the necessary calculations are built into R, using function `gam` in a package called `mgcv`. Fitting a spline to the example data is very easy:

```
> library(mgcv)
This is mgcv 0.5.0
> m3<-gam(y~s(x))
> plot(x,y,main="Spline fitted to x,y data by gam()")
> lines(x,fitted(m3))
```



Note that `m3` actually does a slightly better job than any of our previous attempts. This is because `gam()` actually chooses how wiggly the smooth should be as well as fitting it (this is basically equivalent to choosing how many parameters it should have - more parameters making for a more wiggly curve). The way it does it is based on the following idea. Suppose you left a datapoint, \circ , out of the model fit and just fitted the model to the remaining data, \bullet . Now consider how well the model would fit the missing data point, \circ , under different scenarios:



In scenario **a**, we fit a very smooth model with very few parameters² — it doesn’t fit any of the data very closely and doesn’t do any better on the missing point. In scenario **c**, a very wiggly model with lots of parameters is fitted — this seems to be fitting the “noise” in the data as well as the systematic trend: as a result it wiggles about a lot and is a long way from the point \circ . The intermediate scenario **b** is better: here a moderately smooth curve has been fitted that seems to capture the trend quite well, while smoothing out the noise — with this optimum amount of wiggleness the model does the best job of getting close to the point \circ .

Choosing the model that does best at matching points in the data set that it was not actually fitted to, is called “cross validation”. In practice each datapoint is left out in turn and the average squared difference between missing data and model is calculated for each candidate model. The model that

²just 2 actually, it’s a straight line!

minimises this quantity if doing best at predicting the data it was not fitted to — and is therefore selected. The different candidate models will differ in how wiggly they can be (i.e. they'll basically have different numbers of parameters).

There are various ingenious ways in which the cross validation calculations can be improved and speeded up. They will not be covered here. All that matters is that you have some idea of how the smoothness of the model is being chosen.

3.1.2 A model suitable for mackerel eggs

Having seen how to construct smooth terms, we're now in a position to use model (1). The s_j terms in:

$$y_i = \beta_0 + s_1(x_{1i}) + s_2(x_{2i}) + s_3(x_{3i}) + \epsilon_i$$

can each be represented using a spline, and you won't be surprised to learn that the model can be fitted using `gam()`. The command for fitting the above would be something like:

```
> sm<-gam(y~s(x1)+s(x2)+s(x3),data=a.name)
```

Where `data=a.name` tells R to look in the data frame called `a.name` in order to find variables `y`, `x1` etc. If you don't supply a data frame name then R will search among your current objects and attached data frames for the variables — often this is ok, but it can cause problems if you already have an object with the same name as a variable stored in a data frame.

The name `gam`, comes from the fact that models made up of sums of smooths are known as **G**eneralized **A**dditive **M**odels, or GAMs. To see how to use these models let's return to the real mackerel egg data.

4 Modelling mackerel eggs using GAMs

The aim of this section is for you to produce a model capable of giving an estimate of the mackerel egg production density any where in the 1992 mackerel survey area, based on the data in `mack`. The models to use are the slight generalisations of linear models, known as GAMs, which allow you to represent the dependence of the response variable on each predictor variable using flexible smooth functions.

The first thing to do is to load packages `mgcv` and `mt1007` and the data frame `mack` into R. It's worth attaching `mack` as well. The response will be `egg.dens`: the number of eggs produced per day per square metre of sea surface, estimated from net samples. `mack` also contains a large number of possible predictor variables which might be used to help model egg density. Typing `?mack` will give a fair amount of information about them all, while `names(mack)` will list the names of everything available in the data frame.

From your exploratory analysis of the data, it was clear that densities varied with location, and seemed to depend on sea-bed depth: hence `lon`, `lat` and `b.depth` should probably be included in the model to start with. The biologists believe that the mackerel like to spawn near the 200m sea bed contour, so it seems sensible to include a predictor variable based on distance to that contour: `c.dist` measures exactly that for each sampling station. There is also reason to believe that temperature might be influential: `temp.surf` is the sea surface temperature, and should probably be included initially as well. (You could try including `salinity` of the sea water as well, but if so, you'll need to replace the `NA`'s with something usable, like the mean salinity.)

So it would be sensible to start by fitting a GAM to the egg density data using `lon`, `lat`, `b.depth`, `c.dist` and `temp.surf` as predictor variables (to ensure that it will be straightforward to predict from your model, it is best to use the variable names given, rather than re-assigning the data to new variables with names like `y`, `x1` etc.) . Do this, calling the resulting fitted model object `m1`, and record the command you used here:

As with all modelling the first thing we need to do is to check the residual plots. Since GAMs are basically linear models you would like the assumptions of equal variance and independence to be met (at

least approximately), and it would be quite nice if the residuals were not too far from Normal. Check the residuals using:

```
> plot(fitted(m1), resid(m1))
> qqnorm(resid(m1))
```

Sketch the patterns you see here, and comment on them:

To improve the residual plots it will probably be necessary to transform the egg densities. Try egg density raised to the power 0.2, 0.3, 0.4 and 0.5 and look at the residual plots for each (you might want to divide up the graphics window in some way to aid the comparison). Note that you can specify the transformation directly on the l.h.s. of the `gam()` model formula e.g. `egg.dens^0.5~s(lat)+...`. None of the suggested transformations will achieve perfect results, but select the one that seems least bad to you. Record your selection and reasons here:

Before going any further you should plot the fitted values from the model in the original survey area. First create an array containing the fitted values back transformed to the original measurement scale: the raw fitted values from your model will be fitted egg densities raised to the power you chose: you need to turn them back into fitted egg densities. For example if you chose to raise egg densities to the power 0.5 then you would need to raise the fitted values to the power 2, to “undo” the original transformation. Write the command for getting these “untransformed” fitted values here:

Now divide your graphics window into 2 columns using:

```
> par(mfrow=c(1,2))
```

Then plot the raw data, the coastline and the survey boundary on the left hand plot, in the same way that you did in section 2. On the right hand plot do the same, but use the fitted egg densities rather than the raw densities. In both cases use colour to help distinguish the different features of the plots. Record the commands used here:

4.1 Examining the gam fitted model object

In this section it will be assumed that your fitted GAM model is called `m1` - if it isn't then substitute the name of your fitted model everywhere `m1` appears.

There are two things that you should always do with a fitted `gam` object. The first is to type its name, so that some default information about it is printed. For example, here's what I get for a fit to the untransformed data (you should get something different to this, since you are using transformed data):

```
> m1

Family: gaussian
Link function: identity
```

```
Formula:
egg.dens ~ s(lon) + s(lat) + s(b.depth) + s(c.dist) + s(temp.surf)
```

```
Estimated degrees of freedom:
 3.786038 6.045426 6.337431 1.000024 1.000003   total = 19.16892
```

```
GCV score: 2960.627
```

Family: `gaussian`, indicates that you are assuming that the data are normally distributed (the Normal is also known as the Gaussian distribution), `gam()` can deal with other distributions too, but not in this course. For the purposes of this course **Link function:** `identity` can be ignored. The output then reminds you of the model formula that was used to create the fitted model object: quite useful if you have created several different model objects.

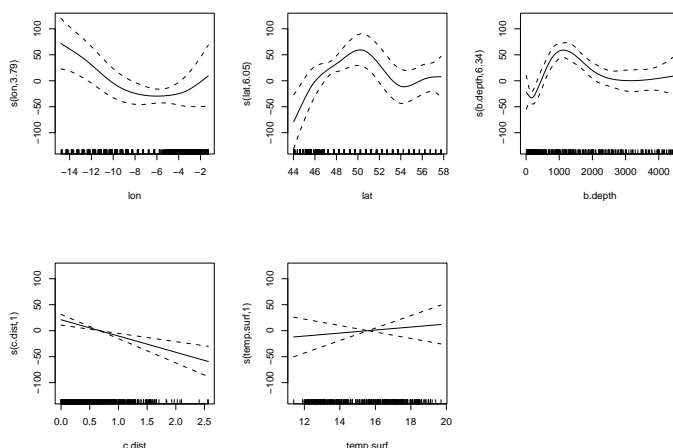
The output, **estimated degrees of freedom**, gives the estimated number of parameters for each smooth term in turn (these have been obtained using cross validation). For example, the smooth for `lon` is estimated to require 3.78 parameters. Exactly how we go about giving the smooth for `lon` the equivalent of 3.78 parameters (rather than 3 or 4) is something that you *really* don't want to know right now! Notice that the smooths for `c.dist` and `temp.surf` have very close to one parameter each - `gam` has decided that these are basically straight line terms.

The final bit of output is the **GCV score** (the cross validation score) — the “estimated degrees of freedom” for each smooth have been chosen to minimise this. Low values of the score are a good thing. A model with a low **GCV score** will do well at predicting the values of datapoints that it was not actually fitted to: this is a nice property as it suggests that the model ought to be quite good at predicting in locations where we don't have direct egg density measurements. Note that the absolute value of the GCV score is not really interpretable, but you can usefully compare scores of different models.

Now have a look at the estimated smooth functions by typing:

```
> plot(m1,pages=1)
```

(You'll probably want to resize your plot window to make this plot look reasonable.) As result you'll get something like this:



(Although, I've again used the wrong model, having left the egg densities untransformed.) There is one plot for each smooth term in your model: in each, the smooth relating to a predictor variable is plotted against that predictor variable. The y axis of each plot is labelled to indicate which predictor variable the smooth is a function of, and how many effective parameters have been estimated for that term. The x axis relates to the predictor variable itself. Along the x axis of each plot is a series of dashes, indicating the observed values of the predictor variables. The solid curve in each plot is the estimated smooth function of the predictor variable, while the dashed lines provide a 95% confidence region for the

smooth curve. Notice that all the smooths have an average value of zero: this is because each is set up to give the alteration of the mean of the response caused by its predictor variable.

From examining the plot for **your** fitted model, would you conclude that any of the predictor variables play no significant part in the model? If so, which variable is not needed? Give your answer and explain it here:

Fit a new model removing the term that you don't think is doing anything useful. Check whether the GCV score has gone down as a result. A substantial decrease is good evidence that you were right to remove the term: the model does a better job without it. A very small change up or down is more ambiguous, but you probably don't gain much by leaving the term in. (Note that you should only ever remove one model term at a time — usually the one that you judge to be least significant.)

Check the plots for the new model, to see if any further terms should be deleted, and to check that the residuals haven't got substantially worse. Write down your final selected model here:

4.2 Predicting over the whole survey area

Once you have a fitted model, you can use it to estimate the thing that is really needed for this survey — the egg production densities over the whole survey area. In general, to predict the egg density using your fitted model you need to know the values of the predictor variables at the location at which you would like a prediction. As an example, suppose that your modelled egg densities depend on longitude, latitude, sea bed depth and distance from the 200m contour, and that you would like a predicted egg density at -2E, 44 $\frac{1}{12}$ N. At that location the sea bed depth is 206 metres and the distance to the 200m contour is 0.0295³. So, if $\hat{s}_i()$ denotes the estimate for the i^{th} smooth then the predicted density \hat{y} will be given by :

$$\hat{y}^{0.4} = \beta_0 + \hat{s}_1(-2) + \hat{s}_2(44.0833) + \hat{s}_3(206) + \hat{s}_4(0.0295)$$

In this case the r.h.s. turns out to be 0.466, corresponding to a predicted production density \hat{y} of $0.466^{2.5} = 0.148m^{-2}d^{-1}$.

To obtain model predictions from the fitted GAM in R involves using the function `predict.gam()`. This function takes two arguments: your fitted GAM object (e.g. `m1`) and a data frame supplying the values of the predictor variables at the locations where predictions are required. `predict.gam()` returns an array containing the predictions at each of these locations.

Data frame `mackp` is designed for predicting over the whole mackerel survey area. The survey area has been divided up into 9757 rectangles, each 1/12 degree latitude by 1/12 degree longitude. Within `mackp`: arrays `lon` and `lat` provide the position of the centre of each rectangle; array `b.depth` provides the average sea bed depth for each rectangle and array `c.dist` provides the distance from the centre of each rectangle to the 200m depth contour.

To get predicted egg densities for every one of these grid squares, just type something like:

```
> data(mackp)
> pred.dens<-predict.gam(m1,mackp) # substitute your model object for m1
```

You'll find that here and there negative densities are predicted — reset these to zero using:

```
> pred.dens[pred.dens<0]<-0
```

Now back transform the predicted densities, in order to undo the transformation that you made when fitting.

³Somewhat odd units are used here, which treat degrees of long and lat as measuring distances

The obvious next step is to plot the model predictions using an `image` plot or similar. But there is a problem, because we only have predictions over the survey area, which is an irregular shape, unlike the nice rectangular region that `image()` requires. In fact it isn't too hard to work out how to copy the predictions that you have made into a rectangular grid suitable for plotting by `image()`: `mackp$area.index` is specially designed to help you do this.

First create arrays of longitude values and latitude values (each of length 169):

```
lon<-seq(-15,-1,1/12);lat<-seq(44,58,1/12)
```

and then create an array into which your predicted egg densities will be copied, but fill the array with NA's for the moment:

```
> zz<-array(NA,169^2)
```

Now the elements of your predicted value array `pred.dens` can be copied into `zz` in such a way that we'll be able to turn `zz` into a matrix for plotting purposes:

```
> zz[mackp$area.index]<-pred.dens
> zz<-matrix(zz,169,169) # converting zz to matrix
```

Now you can produce a suitable `image` plot:

```
image(lon,lat,zz)
```

Add the survey boundary and coast to this plot, and then overlay the raw data, in order to assess how well the model seems to be doing. Are there any obvious features of the data that seem to be missed? Put your answer here.

You might want to have a look at `contour` plots and/or `persp` plots of the fitted model as well.

4.3 How many mackerel eggs were produced per day over the survey area?

You are now in a position to estimate the total number of eggs produced per day over the whole survey area. To do this you simply estimate the total number of eggs in each 1/12 degree "prediction rectangle" and then sum over all the rectangles in the survey area.

To estimate the total number of eggs produced per day in a rectangle, you need to multiply the predicted egg production density per square metre (`pred.dens`) for each rectangle by the area of the rectangle in square metres. The north-south "height" of a 1/12th degree rectangle is always 9265.9 metres, but its east-west "width" depends on its latitude and is given by $9265.9 \cos(\text{lat} \times \pi/180)$, where π is in radians (and available in R as `pi`). So, to get to the total egg production work through the following steps:

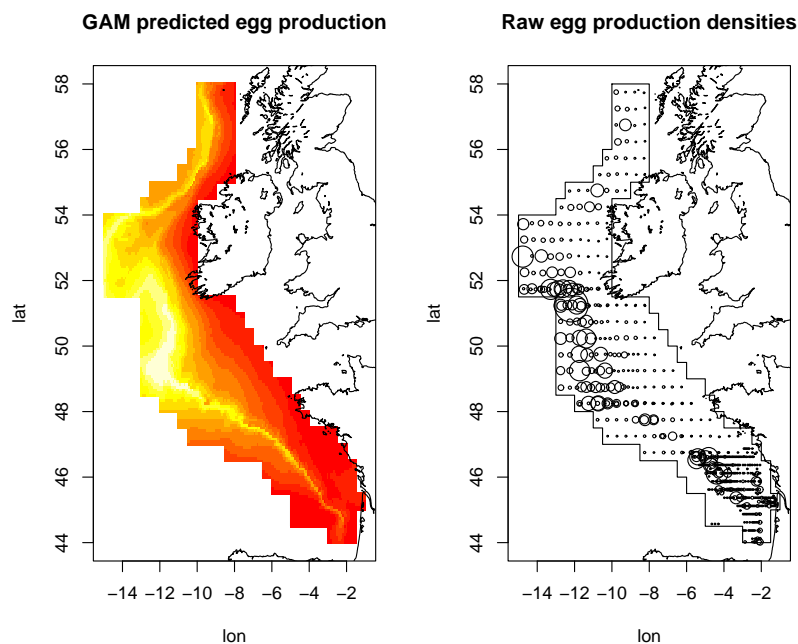
1. Make sure that your predicted densities are egg densities and not egg densities to some power (guess who took an hour to figure out that he hadn't done this!).
2. Using `mackp$lat` and the information on the prediction rectangle dimensions, given above, create an array containing the area of each prediction rectangle in m^2 . Write down the command here:
3. Create an array containing the estimated total eggs produced per day for every rectangle in the prediction grid. Write down the command here:

- Obtain an estimate of the total number of eggs produced per day over the whole survey area. Write the command and the answer here:

Congratulations! The difficult part of the estimation process is now done, all that remains is to convert your estimated egg production rate into the total number of adults needed to produce these eggs.

4.4 From eggs to adult mackerel

You have now modelled the mackerel egg density data, to obtain something like this:



and from the model you have an estimate of the total daily production of mackerel eggs in the survey area. This needs to be turned into an estimate of the weight of adult fish.

To do this you will need some information gathered from surveys of adult fish, and provided in data frames `mack.fish` and `mack.epsf` (`epsf` stands for **e**ggs **p**er **s**pawn**i**ng **f**emale). The data in these data frames were gathered by fishing for adults from the same boats that were used for the egg surveys. There are basically 3 pieces of information that you need to extract from the adult data:

- On average, how many eggs does a spawning female produce in a day per gramme of her bodyweight?
- What proportion of females spawn on any one day?
- What proportion of fish are female?

Armed with these estimates you'll be able to convert total eggs to total weight of spawning females, total weight of spawning females to total weight of all females, and total weight of females to total weight of all fish (assuming males and females weigh the same on average).

4.4.1 Eggs per gramme of spawning female

The data frame `mack.epsf` contains data on a large number of female mackerel taken from the fishing hauls. These were fish for which biologists were able to tell that they were about to spawn, because their ovaries contained eggs that were swollen with water, which only happens just before spawning.

Each row of the data frame corresponds to one fish. The biologists first weighed the fish (`wt.fish`) and then removed the fish's ovaries and weighed them (`wt.ovary`). Next a sample of the ovary was taken and weighed (`wt.samp`) and finally all the eggs that were hydrated and hence about to be spawned were counted in the sample (using a microscope) — `n.heggs`.

To estimate the total number of eggs that each fish would have spawned you have to scale the sample up to the whole ovary, by dividing the number of hydrated eggs (`n.heggs`) by the sample weight and multiplying by the ovary weight. Create an array containing the estimated total eggs that would have been spawned for each fish. Record the command here:

Estimating the number of eggs per gramme of female is now easy. Sum up the total number of eggs that would have been produced by all the sampled fish, and then divide by the sum of the weights of all the sampled fish. Write down the R command for doing this, and record the result:

Now estimate the total weight of female fish that spawn in a day, from the above result and your total egg production estimate. Write the answer here:

4.4.2 Proportion of females that spawn on any one day

To address the next estimation problem you will need data frame `mack.fish`. This contains data from the same fishing hauls as `fish.epsf`, but relates to different samples of fish from those hauls. Each row of the data frame corresponds to one fishing haul.

It is possible to examine the ovaries of female mackerel for the presence of what biologists term “migrating nuclei” — this allows you to tell whether or not a fish will spawn within the next 24 hours. From each fishing haul a number of females were sampled at random, and their ovaries were examined for evidence that they were about to spawn. `n.ov` gives the number of ovaries sampled in each haul, while `n.sp.ov` is the number that would have spawned in the next 24 hours.

To estimate the proportion of females that spawn on one day, divide the sum of all the “about to spawn” ovaries by the sum of all the ovaries examined. Write the command for doing this here, and record the estimated proportion of females that spawn on any one day.

Hence estimate the total mass of females in the survey area:

4.4.3 What proportion of the fish are female

For each fishing haul recorded in `mack.fish`, another sample of fish was taken for sexing. `n.fish` records the total number sampled for this purpose, while `n.female` records the number that were female. Add up the total number of fish sampled for sexing and the total number that turned out to be female, and hence estimate the proportion of mackerel that were female:

Finally, obtain an estimate of the total weight of adult mackerel in the spawning area. The stock

assessment!

If we were to continue further with this analysis, we could estimate confidence intervals for this quantity, and then use it to make recommendations about fishing quotas, but instead we'll stop, since you have now covered all the main concepts involved in estimating the mackerel stock size.

5 Conclusions

Hopefully you've learned several things from this case study: firstly, how statistical modelling is an integral part of solving this fairly important applied problem; secondly how computers can be used to help visualize data and models; and thirdly how simple linear modelling can be extended to produce quite powerful and general GAM methods.

The GAM methods you've met here are currently being used for the analysis of egg production method surveys, and the estimates from them are used to help to try and manage fish stocks. The hope is that by providing prompt and precise stock size assessments these methods will help avoid more disasters like Cod and Herring. Time will tell.

References

The fisheries information in this case study has been gleaned from a number of sources:

- Borchers, D.L., S.T. Buckland, I.G. Priede and S. Ahmadi (1997) Improving the precision of the daily egg production method using generalized additive models. *Canadian Journal of Fisheries and Aquatic Science* 54:2727-2742.
- DFO (2000) Subdivision 3Ps cod. DFO Stock Status Report A2-02(2000)
- Hilborn, R. and Walters, C.J. (1992) *Quantitative Fisheries Stock Assessment*. Chapman and Hall
- Kurlansky, M. (1997) *Cod*. Penguin books.
- Muus, B.J. and P. Dahlstrøm (1974) *Collins Guide to the Sea Fishes of Britain and North Western Europe*. Collins
- Nichols, J. (1999) *Saving the North Sea Herring*. CEFAS handout.
- www.cefas.co.uk - the UK fisheries labs.
- www.dfo-mpo.gc.ca - the Canadian fisheries labs.
- www.fao.org - the united nations food and agriculture organisation.

The mackerel data were provided by: A. Eltink, P. Lucio, J. Massé, L. Motos, J. Molloy, J. Nichols, C. Porteiro and M. Walsh.

Thanks to David Borchers for providing the mackerel data and supplementary data in a readily manageable form, for explaining it and for other help and discussion relating to this case study.

The theory of GAMs as used here is based on:

- Green, P.J. and B.W. Silverman (1994) *Nonparametric Regression and Generalized Linear Models*. Chapman & Hall
- Hastie T.J. and R.J. Tibshirani (1990) *Generalized Additive Models*. Chapman & Hall
- Wahba, G. (1990) *Spline Models for Observational Data*. SIAM, Philadelphia.
- Wood, S.N. (2000) Modelling and smoothing parameter estimation with multiple quadratic penalties. *Journal of the Royal Statistical Society (B)* 62(2): 413-428.

I would also like to thank the following:

