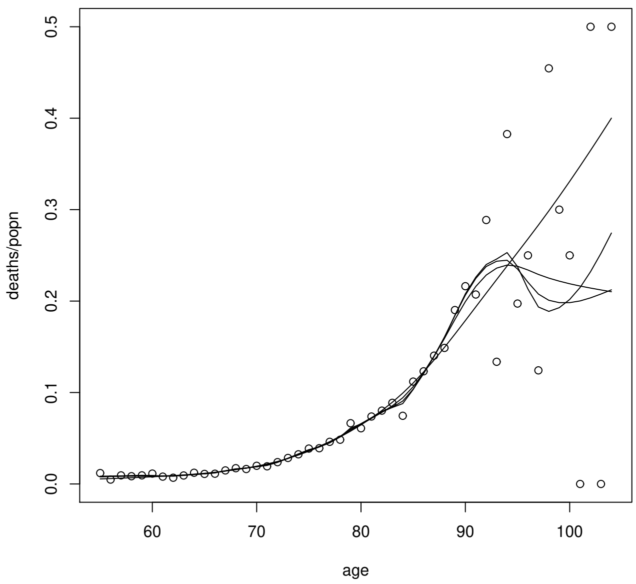
AST Nonparametric regression 2001-2

Using cubic smoothing splines within an iterative loop

Fitting a simple nonparametric generalised model

Let's enter some data – a mortality table for retired US white females – and plot crude death rates against age.

```
> age
 [1]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73
[20]  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92
[39]  93  94  95  96  97  98  99 100 101 102 103 104
> popn
 [1]    84   418  1066  2483  3721  5460  6231  8061  9487 10770 24267 26791
[13] 29174 28476 25840 23916 21412 20116 18876 17461 15012 11871 10002  8949
[25]  7751  6140  4718  3791  2806  2240  1715  1388   898   578   510   430
[37]   362   291   232   196   147   100   161    11    10     8     5     4
[49]     2     2
> deaths
 [1]   1   2  10  21  35  62  50  55  88 132 267 300 432 491 422 475 413 480 537
[20] 566 581 464 461 433 515 374 348 304 249 167 192 171 126  86  97  93  75  84
[39]  31  75  29  25  20   5   3   2   0   2   0   1
> plot(age,deaths/popn)
```



Now let's define a function for the nonparametric logistic regression, following the algebra in the lectures, calling `smooth.spline` in an iterative loop to implement the Fisher scoring method.

```
> nplogit
function(y,n,t,df,tol=1e-006)
{
```

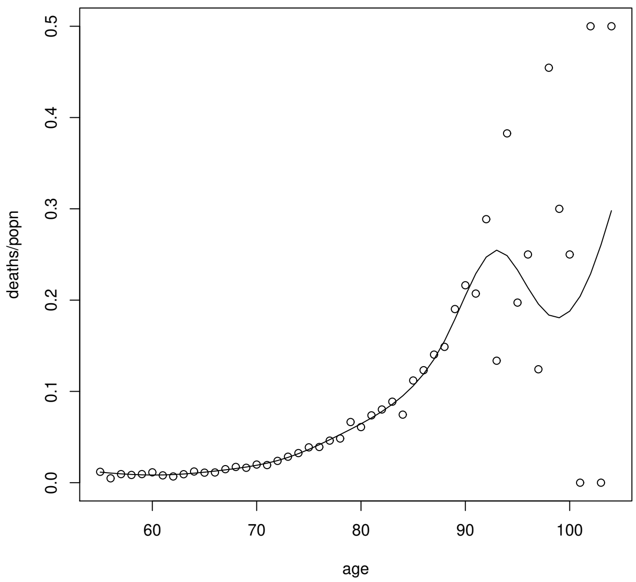
```
        gt_log((y+0.5)/(n-y+0.5))
        repeat{
                p_1/(1+exp(-gt))
                gtnew_smooth.spline(t,gt+(y-n*p)/(n*p*(1-p)),
                        n*p*(1-p),df=df)$y
                if(sum((gt-gtnew)^2)<tol) break
                gt_gtnew
                }
        p
}
```

Now call it for 5, 10, 15 and 20 degrees of freedom and plot the fitted curves – the smoothed death rates.

```
> library(modreg)
> for(df in c(5,10,15,20))
+ {p_nplogit(deaths,popn,age,df)
+ lines(age,p)
+ }
```

A similar analysis can be done by the function `gam` in the `mgcv` library, although this automatically provides the GCV degree of smoothing, and uses a penalised regression spline with (by default) 10 knots.

```
> library(mgcv)
> deathpopn_cbind(deaths,popn-deaths)
> fit_gam(deathpopn~s(age),binomial)
> lines(age,fitted(fit))
```



To find out more about how to use all these functions, use the **R** `?` or `help()` functions, as usual. Similar code works on **S-plus**, but the `library` commands are not then needed.