

# The MortalitySmooth-Package: Additional Examples

Carlo Giovanni Camarda

30. October 2014

```
library(MortalitySmooth)

## Loading required package:  svcm
## Loading required package:  Matrix
## Loading required package:  splines
## Loading required package:  lattice
```

In this document, I collected additional examples which would have take too long for standard package examples. For more details on the methods behind, please check on:

```
citation("MortalitySmooth")
```

## 1 Examples in one-dimension

### 1.1 Including Extra-Poisson variation

This first example takes the documentation from `Mort1Dsmooth()` and shows the difference when an additional overdispersion parameter is included in the estimation procedure.

Directly from the examples in `Mort1Dsmooth()`, we select some data and then smooth them optimizing the smoothing parameter by a BIC:

```
years <- 1950:2006
death <- selectHMDdata("Japan", "Deaths", "Females",
                       ages = 80, years = years)
exposure <- selectHMDdata("Japan", "Exposures", "Females",
                          ages = 80, years = years)
fitBIC <- Mort1Dsmooth(x=years, y=death,
                      offset=log(exposure))
```

Then we check the presence of overdispersion by looking at:

```
fitBIC$psi2
## [1] 12.34
```

which is quite larger than 1 and therefore we should account for such extra-Poisson variation in searching the optimal smoothing parameter. This can be done by setting the argument `overdispersion` to `TRUE`:

```
fitBICover <- Mort1Dsmooth(x=years, y=death,
                          offset=log(exposure),
                          overdispersion=TRUE)
```

The difference in the selected  $\lambda$  gives an idea of the influence of such procedure:

```
fitBIC$lambda;fitBICover$lambda
## [1] 316.2
## [1] 3162
```

i.e. in presence of overdispersion we tend to get smoother outcomes. Let's see the difference in a plot (Figure 1):

```
plot(fitBICover)
lines(years, fitBIC$logmortality, col=4, lwd=2, lty=2)
```

## 1.2 Interpolating/Extrapolating using dummy data

Though it would be more efficient and elegant to interpolate and extrapolate fitted values by means of `predict()`, here we show how to do using solely the main function `Mort1Dsmooth()` and suitable dummy data.

Taking the previous datasets, we assume the absence of data between 1969 and 1989 by adding `NA` in the vector of exposures:

```
exposure.int <- exposure
exposure.int[20:40] <- NA
```

A first attempt would be to directly fit using the new exposures, but a clear error message will be produced.

The proper way is to set up a suitable vector of weights which will inform the main function that estimation between 1969 and 1989 have zero-weight:

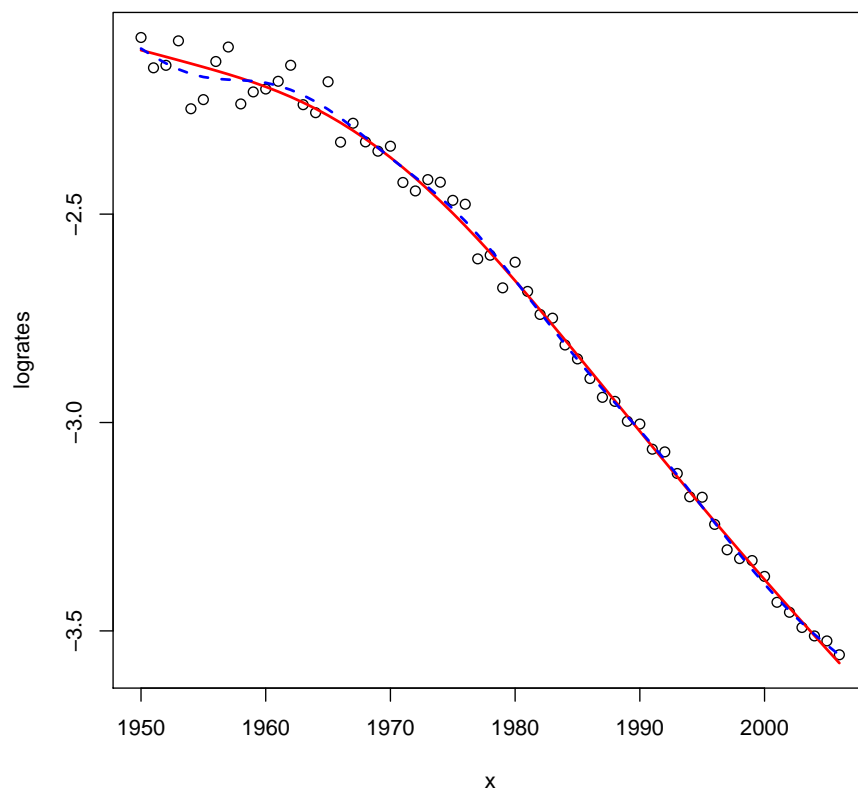


Figure 1: Actual and fitted log-mortality with and without accounting for extra-Poisson variation. Japanese women age 80 in years 1950-2006

```
w.int <- rep(1, length(years))
w.int[20:40] <- 0
fit1Dint <- Mort1Dsmooth(x=years, y=death,
                        offset=log(exposure.int), w=w.int)

## Warning: Interpolation and/or extrapolation is taking place
```

Note that the function will produce a warning inform the user about the interpolation.

A plot will illustrate the effect of interpolating data (Figure 2):

```
plot(fit1Dint)
```

In the next example we illustrate how to extrapolate mortality data directly by `Mort1Dsmooth()` and with the means of proper weights and augmented data. Also in this case it would be simpler to use `predict()`, but in the following it will be easier to grasp the concept behind the procedure.

First we augment the vector of data:

```
years.ext <- seq(years[1], years[length(years)]+20)
exposure.ext <- c(exposure, rep(NA, 20))
death.ext <- c(death, rep(NA, 20))
```

Then we create a vector of weights in which we have ones in the observed years and zeros in the forecast ones:

```
w.ext <- c(rep(1, length(years)), rep(0, 20))
```

Then we can easily fit the model with these new data using the optimized smoothing parameter from `fit1Dover`:

```
fit1Dext <- Mort1Dsmooth(x=years.ext, y=death.ext,
                        offset=log(exposure.ext), w=w.ext,
                        method=3, lambda=fitBICover$lambda)

## Warning: Interpolation and/or extrapolation is taking place

plot(fit1Dext)
```

Figure 3 shows the outcomes.

## 2 Examples in two-dimensions

### 2.1 Including Extra-Poisson variation

In the next example we will show the effect of including an overdispersion parameter in the searching of the optimal smoothing parameters in a 2D setting.

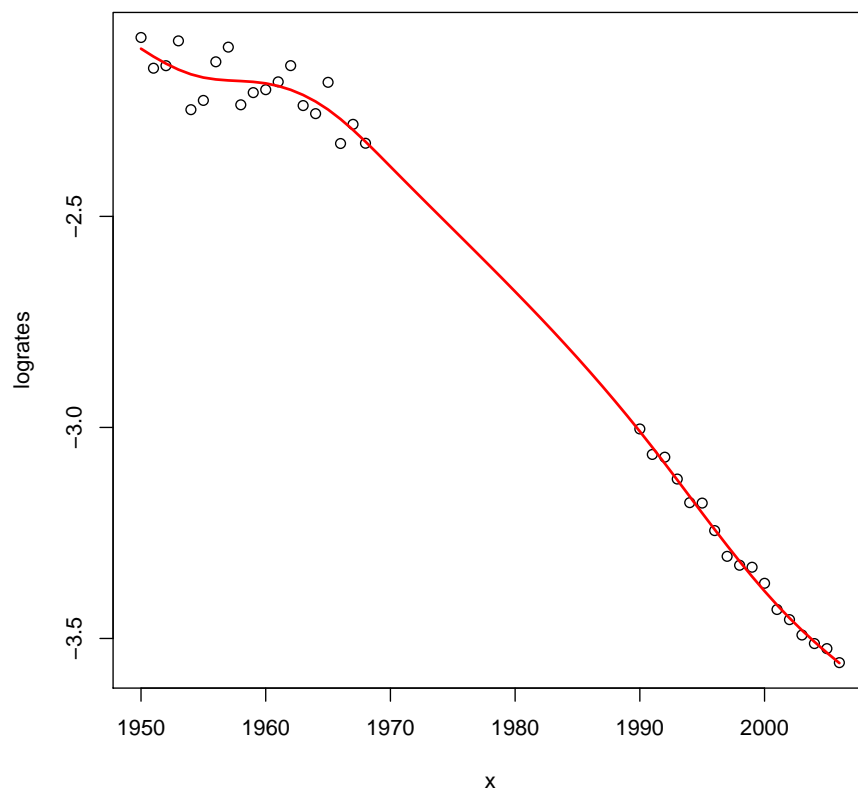


Figure 2: Actual and fitted log-mortality with interpolation between 1969 and 1989. Japanese women age 80 in years 1950-2006.

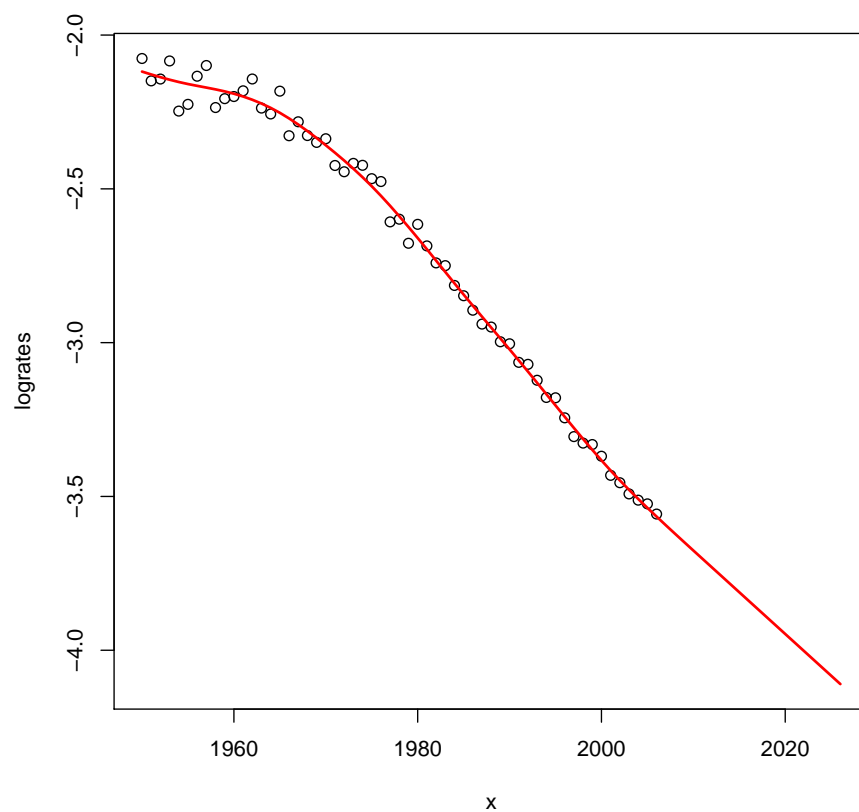


Figure 3: Actual, fitted and forecast log-mortality. Japanese women age 80 in observed years 1950-2006 and forecast years 2007-2026.

First we take data and default fit from the documentation in `Mort2Dsmooth()`:

```
ages <- 50:100
years <- 1950:2006
death <- selectHMDdata("Sweden", "Deaths", "Females",
                       ages = ages, years = years)
exposure <- selectHMDdata("Sweden", "Exposures", "Females",
                          ages = ages, years = years)
fitBIC <- Mort2Dsmooth(x=ages, y=years, Z=death,
                      offset=log(exposure))
```

Then we check the estimated overdispersion parameter:

```
fitBIC$psi2
## [1] 1.309
```

which seems larger than 1. Therefore we set the argument `overdispersion` equal to `TRUE`:

```
fitBICover <- Mort2Dsmooth(x=ages, y=years, Z=death,
                           offset=log(exposure),
                           overdispersion=TRUE)
```

The difference in the smoothing parameters is clear especially in the year direction:

```
fitBIC$lambda
## [1] 1000.0 316.2

fitBICover$lambda
## [1] 316.2 3162.3
```

## 2.2 Interpolating/Extrapolating using dummy data

Here we follow the same procedure presented in the one-dimensional setting, generalizing over a second domain.

Using the previous data, we assume the absence of data between age 69 and 79 by inserting `NA` in the matrix of exposures. Then we create a suitable matrix of weights and we use the main function `Mort2Dsmooth()`, plotting the fitted logrates:

```

new.exposure <- exposure
new.exposure[20:30, ] <- 0
W <- matrix(1, length(ages), length(years))
W[20:30, ] <- 0
fit2Dint <- Mort2Dsmooth(x=ages, y=years, Z=death,
                        offset=log(new.exposure), W=W)

## Warning: Interpolation and/or extrapolation is taking place
plot(fit2Dint)

```

Outcomes are given in Figure 4.

An additional example in 2D assume the lost of 90% of the mortality data and the interpolation over age and time of the complete mortality surface.

First we create a matrix of weights all equal to 1 in which we randomly assign zero values to about 90% of the cells. Then we just add this matrix in the argument W within the main function `Mort2Dsmooth()`. Outcomes are presented in Figure 5.

```

W <- matrix(1, length(ages), length(years))
set.seed(3)
zeros <- sample(x=1:prod(dim(W)),
               size=round(prod(dim(W))/100*90),
               replace=FALSE)
W[zeros] <- 0
fit2Dint10 <- Mort2Dsmooth(x=ages, y=years, Z=death,
                          offset=log(exposure), W=W)

## Warning: Interpolation and/or extrapolation is taking place
plot(fit2Dint10)

```

Likewise in 1D, we extrapolate our data by augmenting the data (matrices now) over the year domain and creating a suitable matrix of weights with zeros in the forecast years.

```

years.new <- 1950:2025
M <- matrix(0, nrow=length(ages),
           ncol=length(years.new)-length(years))
death.new <- cbind(death, M)
exposure.new <- cbind(exposure, M)
W <- matrix(1, length(ages), length(years))
W.new <- cbind(W, M)
fit2Dext <- Mort2Dsmooth(x=ages, y=years.new, Z=death.new,
                        offset=log(exposure.new), W=W.new)

## Warning: Interpolation and/or extrapolation is taking place

```



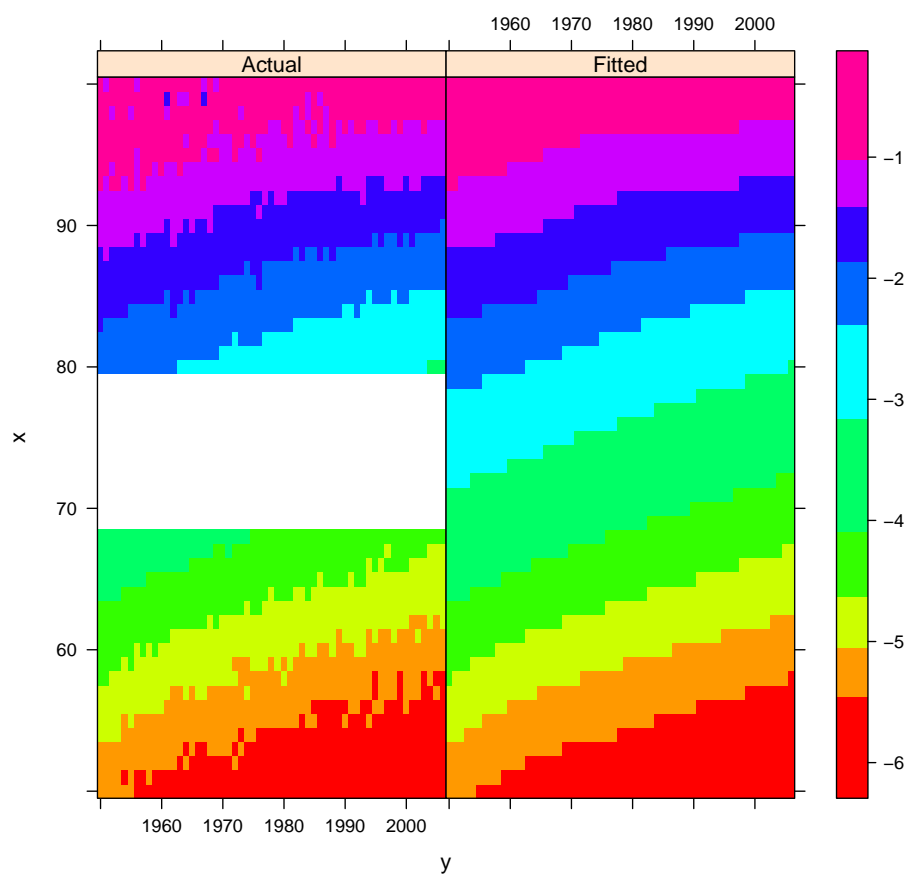


Figure 4: Actual and fitted log-mortality with interpolation between age 69 and 89. Swedish women from 50 to 100 in years 1950-2006.

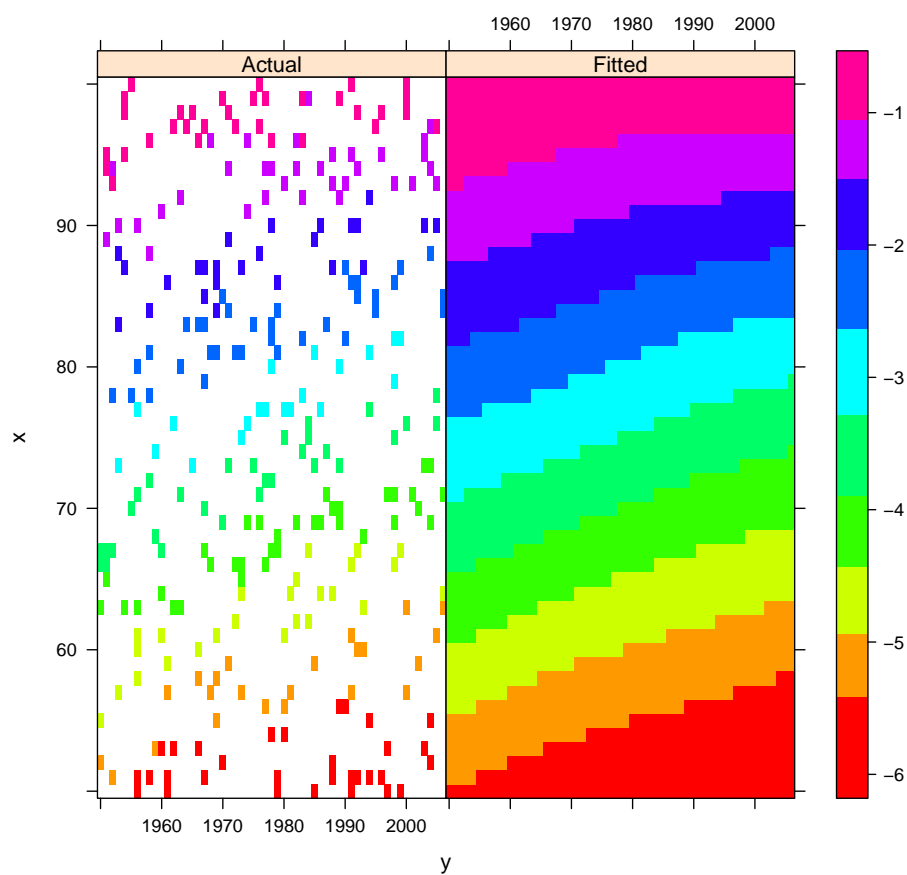


Figure 5: Actual and fitted log-mortality using only random 10% of the available information. Swedish women from 50 to 100 in years 1950-2006.

```
plot(fit2Dext)
```

Outcomes are shown in Figure 6.

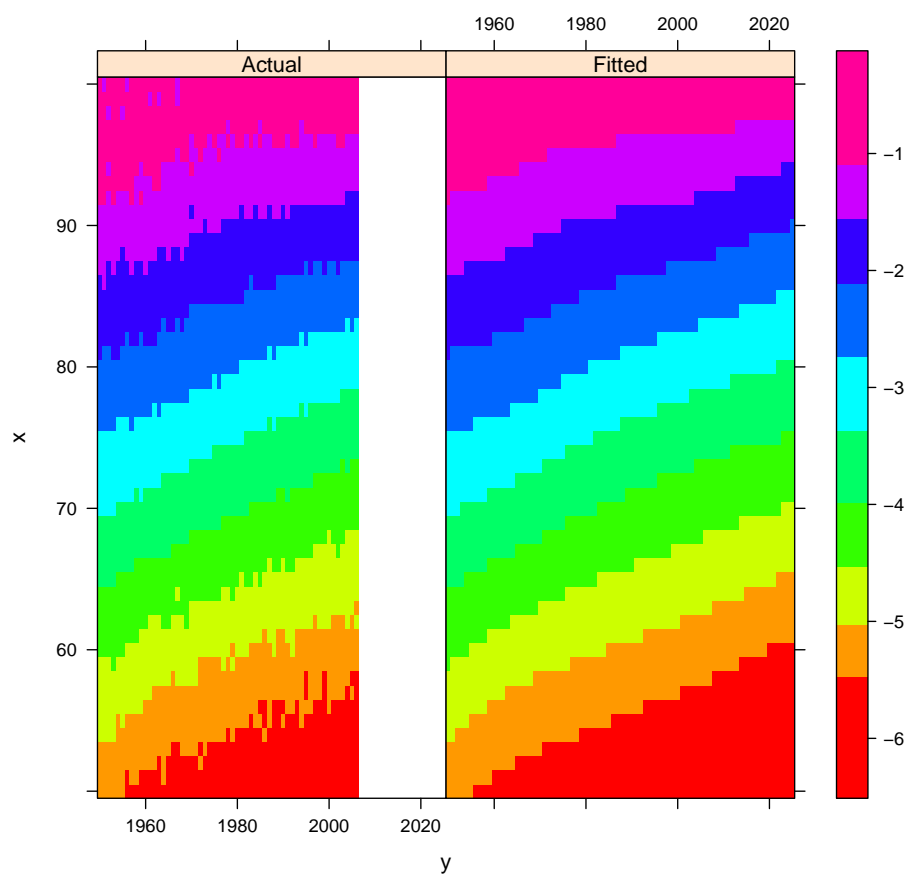


Figure 6: Actual, fitted and forecast log-mortality. Swedish women from 50 to 100 in observed years 1950-2006 and forecast years 2007-2025.