6. GENERATION OF PSEUDO-RANDOM DATA

In order to approve the computational procedures we have to check these procedures for generated data for different values for the parameters of the model. If we estimate them correctly we can use them for real data.

Let us suppose that there is no truncation and censoring. Let Z be a frailty variable. Let us assume that the base hazard is $\mu(x) = ae^{bx}$ and the individual hazard is $\mu(z \mid Z) = Z\mu(x)$. The conditional survival function is defined as

$$S(x \mid Z) = e^{-H(x|Z)} = Z\frac{a}{b}(e^{bx} - 1),$$

where $H(x \mid Z) = \int_0^x Z\mu(u)du$. To generate data we use a pseudo - random number generator for uniform distribution in (0, 1) which is build in MATLAB 6.0. Let y be a uniform distributed random variable, then using the inverse function of the survival function we have

$$x = \frac{1}{b}log(1 - Z\frac{b}{a}log(y))$$

and x is random value with survival function $S(x \mid Z)$. Using this relations we can summarise the algorithm:

1. Generate sets of independent Gamma distributed numbers Y_0, Y_1, Y_2 with proper parameters.
2. Find the sums $Z1 = Y_0 + Y_1$, $Z_2 = Y_0 + Y_2$.
3. Generate two independent sets of uniform distributed numbers y_1 and y_2.
4. Obtain the values

$$x1 = \frac{1}{b}log(1 - Z1\frac{b}{a}log(y1))$$

$$x2 = \frac{1}{b}log(1 - Z2\frac{b}{a}log(y2))$$

In this way all the shared frailty is summed in Y_0 and all the difference between the individuals is assumed to be represented by Y_1 and Y_2. Choosing the parameters of the generating procedure one can obtain different characteristic of the generated data. The estimated and observed univariate mortality are shown on fig. 1. on a sample of 5000 observations.

The main disadvantage of this procedure is that the mortality hazard is in a very simple form and this allows us to easily calculate the inverse function of the survival function. It is simple to extend the procedure and to use it for a more complicated hazard function. To do this it is necessary to invert the survival function. An easy algorithm for this can be summarized as follows: Let us suppose that y is real number between 0 and 1. Then

1. Generate a big set of numbers $X = \{xi, i = 1 \cdots N\}$ (uniform distributed in (x_0, x_1))
2. Find the values $S(x_i)$, $i = 1 \cdots N$