Demography and epidemiology: Practical use of the Lexis diagram in the computer age.

or:

Who needs the Cox-model anyway?

Annual meeting of Finnish Statistical Society
23–24 May 2005
Revised December 2005.

Bendix Carstensen
Steno Diabetes Center, Gentofte, Denmark
& Department of Biostatistics, University of Copenhagen
bxc@steno.dk
www.biostat.ku.dk/~bxc

The contents of this paper was presented at the meeting of the Finnish Statistical Society in May 2005 in Oulu. The slides presented can be found on my homepage as
http://staff.pubhealth.ku.dk/~bxc/Talks/Oulu.pdf.

Contents

Abstract

This paper argues that the Cox-model is over-used, and that a more parsimonious approach to modelling of time-effects may have some advantages, particularly from a conceptual point of view. Taking a demographic point of view and visualizing follow-up data in a Lexis diagram moves the focus to time as a covariate, and naturally to the consideration of several possible time-scales.

1 Introduction

In the last 30 years, survival analysis has been virtually synonymous with application of the Cox-model. The common view of survival analysis (and teaching of it) from the Kaplan-Meier-estimator to the Cox-model is based on time as the response variable, incompletely observed (right-censored). This has automatically lent an aura of complexity to concepts such as time-dependent covariates, stratified analysis, delayed entry and time-varying coefficients. The aim of this paper is to show that it need no be so.

If survival studies is viewed in the light of the demographic tradition, the basic observation is not one time to event (or censoring) from each individual, but rather many small pieces of follow up from each individual. This makes concepts clearer as modelling of rates rather than time to response becomes the focus; the basic response is now a 0/1 outcome in each interval, albeit not independent anymore. Time is then correctly viewed as a covariate rather than a response. Time-dependent covariates will not have any special status relative to other covariates, except to the extent they are intermediate responses that one conditions on. Stratified analysis becomes a matter of interaction between time and a categorical covariate, and time-varying coefficients becomes interactions between time and a continuous covariate. Finally, the modelling tools needed reduces to Poisson regression.

The practical part of this change in modelling focus leads to splitting of the follow-up into many observations, and hence Poisson modelling of datasets with 10–50 times as many observations as persons. The lack of computing power has been a problem until recently, as has the absence of software for rational handling of this approach. There are now solutions available for at least Stata, SAS and **R**, which makes this approach widely accessible.

The only remaining advantage of the Cox-model is the ability to easily produce estimates of survival probabilities in (clinical) studies with a well-defined common entry time for all individuals. This can however also be produced from a Poisson model with not too complicated methods.

2 Time: Response or covariate?

One common exposition of survival analysis is as analysis of data (X, Z), where we only observe $\min(X, Z)$ and $\delta = 1\{Z < X\}$. This is an approach which takes the survival time X, as response variable, albeit not fully observed, limited by the censoring time, Z.

However from a life-table (demographical) point of view the survival time is better viewed as a covariate, and only differences (i.e. *risk time*) on (any) time-scale should be considered responses. In a life-table differences on the time-scale are accumulated as risk time whereas the position on the age-scale for these are used as a covariate classifying the table.

Consider a follow-up (survival) study where the follow-up time for each individual is divided into small intervals of equal length y, say, and each with an exit status recorded (this will be 0 for the vast majority of intervals and only 1 for the last interval for individuals experiencing an event)

Each small interval for an individual contributes an observation of what I will term an *empirical rate*, (d, y), where d is the number of events in the interval (0 or 1), and y is the length of the interval, i.e. the risk time. This definition is slightly different from the traditional as d/y (or $\sum d / \sum y$); it is designed to keep the entire information content in the demographic observation, even if the number of events is 0. This is in order to make it usable as a response variable in statistical analyses.

The *theoretical* rate of event occurrence is defined as a function, usually depending on some time-scale, t:

$$\lambda(t) = \lim_{h \searrow 0} \frac{\mathrm{P}\left\{\text{event in } (t, t+h) | \text{ at risk at time } t\right\}}{h}$$

The rate may depend on any number of covariates; incidentally on none at all. Note that in this formulation time(scale) t has the status of a covariate and risk time h the status of risk time, which is the difference between two points on the time-scale.

Likelihood for empirical rates

This definition can immediately be inverted to give the likelihood contribution from an observed empirical rate (d, y), for an interval with constant rate λ, namely the Bernoulli likelihood[1] with probability λy:

$$\begin{aligned} L\big(\lambda|(d,y)\big) &= (\lambda y)^d \times (1-\lambda y)^{1-d} = \left(\frac{\lambda y}{1-\lambda y}\right)^d (1-\lambda y) \\ \ell\big(\lambda|(d,y)\big) &= d \ln\left(\frac{\lambda y}{1-\lambda y}\right) + \ln(1-\lambda y) \approx d\ln(\lambda) + d\ln(y) - \lambda y \end{aligned}$$

where the term $d\ln(y)$ can be dispensed with because it does not depend on the parameter λ.

Observation of several independent empirical rates with the same theoretical rate parameter will give rise to a log likelihood that depends on the empirical rates only through $D = \sum d$ and $Y = \sum y$ of the form:

$$\ell(\lambda|(D,Y)) = D\ln(\lambda) - \lambda Y \tag{1}$$

[1] The random variables event (0/1) and follow-up time for each individual have in this formulation been transformed into a random number of 0/1 variables (of which at most the last can be 1). Hence the validity of the binomial argument, y is not a random quantity, but a fixed quantity.

which apart from a constant is the log likelihood for a Poisson variate D with mean λY.

The contributions to the likelihood from one individual will not be independent, but they will be conditionally independent — the total likelihood from one individual will be the product of conditional probabilities of the form:

$$\begin{aligned} \mathrm{P}\left\{\text{event in } (t_3,t_4)\middle|\ \text{alive at } t_3\right\} &\times \mathrm{P}\left\{\text{survive } (t_2,t_3)\middle|\ \text{alive at } t_2\right\} \\ &\times \mathrm{P}\left\{\text{survive } (t_1,t_2)\middle|\ \text{alive at } t_1\right\} \\ &\times \mathrm{P}\left\{\text{survive } (t_0,t_1)\middle|\ \text{alive at } t_0\right\} \end{aligned}$$

Hence the likelihood for a set of empirical rates *looks like* a likelihood for independent Poisson observations, but it is not, it is a product (and the log-likelihood a sum) of conditional probabilities.

Thus follow-up studies can be analysed in any desired detail using the Poisson likelihood for independent observations; it depends on how large intervals of constant rate one is prepared to accept. Of course the amount and spacing of events limits how detailed the rates can be modelled.

Note that it is only the likelihood that coincides with that of a Poisson model, not the distribution of the response variable (d, y), so only inference based on the likelihood is admissible. Any measures deriving from properties of the Poisson distribution as such are in principle irrelevant.

Analysis of a multiplicative model for the rate parameter λ amounts to fitting a Poisson model which is linear in $\log(\mu) = \log(\lambda Y) = \log(\lambda) + \log(Y)$. If $\log(\lambda)$ is to be the parameter the software must be able to handle the term $\log(Y)$ as a covariate with fixed coefficient equal to 0, a so called *offset*, see e.g. [5].

3 The Cox-likelihood as a profile likelihood

The Cox-model leaves the effect of one primary time-scale unspecified:

$$\lambda(t,x) = \lambda_0(t) \times \exp(\eta), \quad \eta = X\beta$$

Cox [2] devised the *partial* log-likelihood for the parameters $\beta = (\beta_1, \ldots, \beta_p)$ in the linear predictor $\eta_i = \beta_1 x_{1i} + \cdots + \beta_p x_{pi}$:

$$\ell(\beta) = \sum_{\text{death times}} \log\left(\frac{\mathrm{e}^{\eta_{\text{death}}}}{\sum_{i\in\mathcal{R}_t} \mathrm{e}^{\eta_i}}\right)$$

where $\mathcal{R}_t$ is the risk set at time t, i.e. the set of individuals at risk at time t.

Suppose the time-scale has been divided into small time intervals with at most one death in each, and that we in addition to the regression parameters describing the effect of covariates use one parameter per death time to describe the effect of time (i.e. the chosen time-scale). Thus the model with constant rates in each small interval is:

$$\log\big(\lambda(t,x_i)\big) = \log\big(\lambda_0(t)\big) + \beta_1 x_{1i} + \cdots + \beta_p x_{pi} = \alpha_t + \eta_i$$

Assume w.l.o.g. that the ys for all these empirical rates are 1. The log-likelihood contributions that contain information on a specific time-scale parameter α_t, relating to

time t say, will be contributions from the empirical rate $(d, y) = (1, 1)$ with the death at time t, and all the empirical rates $(d, y) = (0, 1)$ from all the other individuals that were at risk at time t. There is exactly one contribution from each individual at risk to this part of the log-likelihood:

$$\ell_t(\alpha_t, \beta) = \sum_{i \in \mathcal{R}_t} \left\{ d_i(\alpha_t + \eta_i) - \mathrm{e}^{\alpha_t + \eta_i} \right\} = \alpha_t + \eta_{\text{death}} - \mathrm{e}^{\alpha_t} \sum_{i \in \mathcal{R}_t} \mathrm{e}^{\eta_i}$$

where η_{death} is the linear predictor for the individual that died. The derivative w.r.t. α_t is:

$$\mathrm{D}_{\alpha_t} \ell(\alpha_t, \beta) = 1 - \mathrm{e}^{\alpha_t} \sum_{i \in \mathcal{R}_t} \mathrm{e}^{\eta_i} = 0 \quad \Leftrightarrow \quad \mathrm{e}^{\alpha_t} = \frac{1}{\sum_{i \in \mathcal{R}_t} \mathrm{e}^{\eta_i}}$$

If the estimate of e^{α_t} is fed back into the log-likelihood for α_t, we get the *profile likelihood* (with α_t "profiled out"):

$$\log \left(\frac{1}{\sum_{i \in \mathcal{R}_t} \mathrm{e}^{\eta_i}} \right) + \eta_{\text{death}} - 1 = \log \left(\frac{\mathrm{e}^{\eta_{\text{death}}}}{\sum_{i \in \mathcal{R}_t} \mathrm{e}^{\eta_i}} \right) - 1$$

which is the same as the contribution from time t to Cox's partial log-likelihood.

The Cox-model could therefore have been formulated as one where there was a separate time-scale parameter for each time-interval.

For those intervals on the time-scale where no deaths occur, the estimate of the α_t will be $-\infty$, and so these intervals will not contribute to the log-likelihood.

Hence, the Cox-model can be estimated by standard Poisson-regression-software by splitting the data finely and specifying the model as having one rate parameter per time interval. The results will be the same, also for the standard errors, because it is the same likelihood that is maximized. This is illustrated empirically in the first part of the program `lung-ex.R` listed in section 9

This is by no means a new discovery, already John Whitehead [4] pointed this possibility out in 1980. However, the computational capacity problems connected with this approach in the 1980s were too large for the method to catch on in practice.

4 Practical data processing

Implementation of the Poisson-approach in practice requires that follow-up for each individual is split in small pieces of follow-up along one or more time-scales. The relevant time-varying covariates should be computed for each interval and fixed covariates should be carried over to all intervals for a given individual.

Presently there are tools for this in:

Stata: The function `stsplit` (part of standard Stata), a descendant of `stlexis` written by Michael Hills & David Clayton.

SAS: A macro `%Lexis`, available at `http://wwww.biostat.ku.dk/~bxc/Lexis`, written by Bendix Carstensen.

R: A function `Lexis`, written by David Clayton, included in the package `Epi` which is available at the comprehensive **R** archive network, CRAN, `http://cran.r-project.org/`.

These tools expand a traditional survival dataset with one record per individual to one with several records per individual, one record per follow-up interval.

The split data makes a clear distinction between *risk time* which is the length of each interval and *time-scale* which is the value of the time-scale at (the beginning of) each interval, that be time since entry or current age of the individual.

This is apparent in analysis of tabulated data from cancer registries, where events from the register database and population figures from the statistical bureau are classified by age and and period of diagnosis. Here the population figures play the role of risk time and the classification by age and period the role of the time-scale(s). The separation of risk time and time-scale in this sense has a long standing tradition in epidemiology and demography.

In Poisson modelling the log-risk time is used as offset and the time is used as covariate. Thus Poisson modelling of follow-up data makes a clear distinction between risk time as the response variable and time scale(s) as covariate(s).

Estimation in the model

Once data has been split into little pieces of follow-up time, the effect of any time-scale can be estimated using parametric regression tools as for example splines. This will directly produce estimated baseline rates by using a tool to predict from a generalized linear model with a given set of covariates.

A simple illustration of this machinery, using **R** and the package `splines` would go as follows:

```
# Simple model modeling time-effects with splines
#
library( splines )
# The package for time-splitting
library( Epi )
# Split the data in pieces of length 0.5,
# producing new variables Enter, Exit, Fail, Time
spl.dat <- Lexis( entry=0, exit=time, fail=D,
                  breaks=seq( 0, 10, 0.5 ),
                  include=list( sex, expos), data=dfr )
# Then fit a Poisson model using these new variables
m1 <- glm( Fail ~ ns( Time ) + sex + expos + offset( log( Exit-Entry ) ),
           family=poisson, data=spl.dat )
```

Estimation of baseline hazard

Suppose $h(t)$ is a parametric function which is parametrized linearly by the parameters in τ, $h(t) = w\tau$ (w is a row vector, τ a column vector). The model can be formulated as:

$$\log\bigl(\lambda(t,x)\bigr) = h(t) + x\gamma = w\tau + x\gamma = (w\,x)\begin{pmatrix}\tau\\ \gamma\end{pmatrix}$$

Standard prediction machinery can be used to produce estimates of log-rates with standard errors for a set of values of t (and hence w), and some chosen values of the variables in x. This is a standard tool in any statistical package for analysis of generalized linear models. Rate estimates with. c.i.s are then derived by taking the exponential function of the estimates for the log-rates with confidence intervals.

Estimation of survival function

The survival function is connected to the rate function $\lambda(t)$ by:

$$S(t) = \exp\left(-\int_0^t \lambda(s)\ \mathrm{d}s\right)$$

In order to estimate this from a parametric model we need to derive the integral, i.e. a cumulative sum of predictions. If we want standard errors for this we must have not only standard errors for the λs, but the entire the variance-covariance matrix of estimated values of λ.

From a generalized linear model we can easily extract estimates for $\log\big(\lambda(t)\big)$ at any set of points. This is just a linear function of the parameters, and so the variance-covariance matrix of these can be computed from the variance-covariance matrix of the parameters.

A Taylor approximation of the variance-covariance matrix for $\lambda(t)$ can be obtained from this by using the derivative of the function that maps $\log(\lambda(t))$ to $\lambda(t)$. This is the coordinate-wise exponential function, so the matrix required is the diagonal matrix with entries $\lambda(t)$.

Finally the cumulative sum is obtained by multiplying with a matrix with 1s on and below the diagonal, so this matrix just needs to be pre and post-multiplied in order to produce the variance-covariance of the cumulative hazard at the prespecified points.

In technical terms, let $\hat{f}(t_i)$ be estimates for the log-rates for a certain set of covariate values (x) at points $t_i, i = 1, \ldots, I$, derived by:

$$\hat{f}(t_i) = \mathbf{B}\,\hat{\beta}$$

where $\beta = (\tau, \gamma)$ is the parameter vector in the model, including the parameters that describe the baseline hazard. Let the estimated variance-covariance matrix of $\hat{\beta}$ be $\hat{\Sigma}$.

Then the variance-covariance of $\hat{f}(t_i)$ is $\mathbf{B}\Sigma\mathbf{B}'$. The transformation to the rates is the coordinate wise exponential function so the derivative of this is the diagonal matrix with entries $\exp\big(\hat{f}(t_i)\big)$, so the variance-covariance matrix of the rates at the points t_i is (by the δ-method):

$$\mathrm{diag}(\mathrm{e}^{\hat{f}(t_i)})\,\mathbf{B}\,\hat{\Sigma}\,\mathbf{B}'\,\mathrm{diag}(\mathrm{e}^{\hat{f}(t_i)})'$$

Finally, the transformation to the cumulative hazard (assuming that all intervals have length ℓ) is by a matrix of the form

$$\mathbf{L} = \ell \times \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

so the (approximate) variance-covariance matrix for the cumulative hazard is:

$$\mathbf{L}\,\mathrm{diag}(\mathrm{e}^{\hat{f}(t_i)})\,\mathbf{B}\,\hat{\Sigma}\,\mathbf{B}'\,\mathrm{diag}(\mathrm{e}^{\hat{f}(t_i)})'\,\mathbf{L}'$$

Example: Mayo clinic lung cancer data

As an example, consider the Mayo clinic lung cancer survival data, provided in the **R**-library `survival`.

In the section 9 is a transcript of an **R**-program that fits both a classical Cox-model and a parametric model for the baseline using natural splines. Also the survival function with confidence intervals is estimated in both models, and the results are contrasted in figure 1.

Estimation of regression parameters

In the case where one parameter per time point is used the regression coefficients will be exactly the same in the Poisson model and in the Cox-model, because the partial likelihood is the profile likelihood from the Poisson model. The natural use of the Poisson-approach is however to avoid the overly detailed modelling of the underlying hazard inherent in the partial likelihood approach, by using a more parismonous representation of the baseline hazard.

However the modelling should not bee too parsimonous as is suggested by the following small simulation study: 100 survival datasets were simulated with 200 individuals in each, all with entry at time 0, and a model with baseline hazard as shown in the left panel of figure 2, censoring at time 10, and two independently distributed covariates, one standard normal with rate-ratio of 4 and the other log-normal with rate-ratio of 0.25.

The following three models were fitted to each dataset: 1) a standard Cox-model, 2) a Poisson model using natural splines with knots in 2,4,6,8 and boundary knots in 0 and 10 for the time effect (6 parameters for the baseline), and 3) a Poisson-model using a constant effect of time (1 parameter for the baseline).

The results for the first of the two regression parameters by the three approaches are shown in figure 2. If the baseline hazard is grossly misspecified we get a diluted effect of the covariate, but there is virtually no difference between the results obtained from modelling the baseline hazard by a very detailed step-function or by natural splines with just 6 parameters. As should be expected from standard statistical theory, the standard errors obtained by the Poisson-spline approach are on average very slightly smaller.

Thus the only reason to use the Cox-model is computational convenience and only in the case of a single relevant time-scale and if the survival function rather than the baseline hazard is of primary interest.

5 Multiple time-scales

The major advantage of separating time as response variable (differences on time-scales) from time as covariate (points on a time scale) is the possibility to accommodate effects

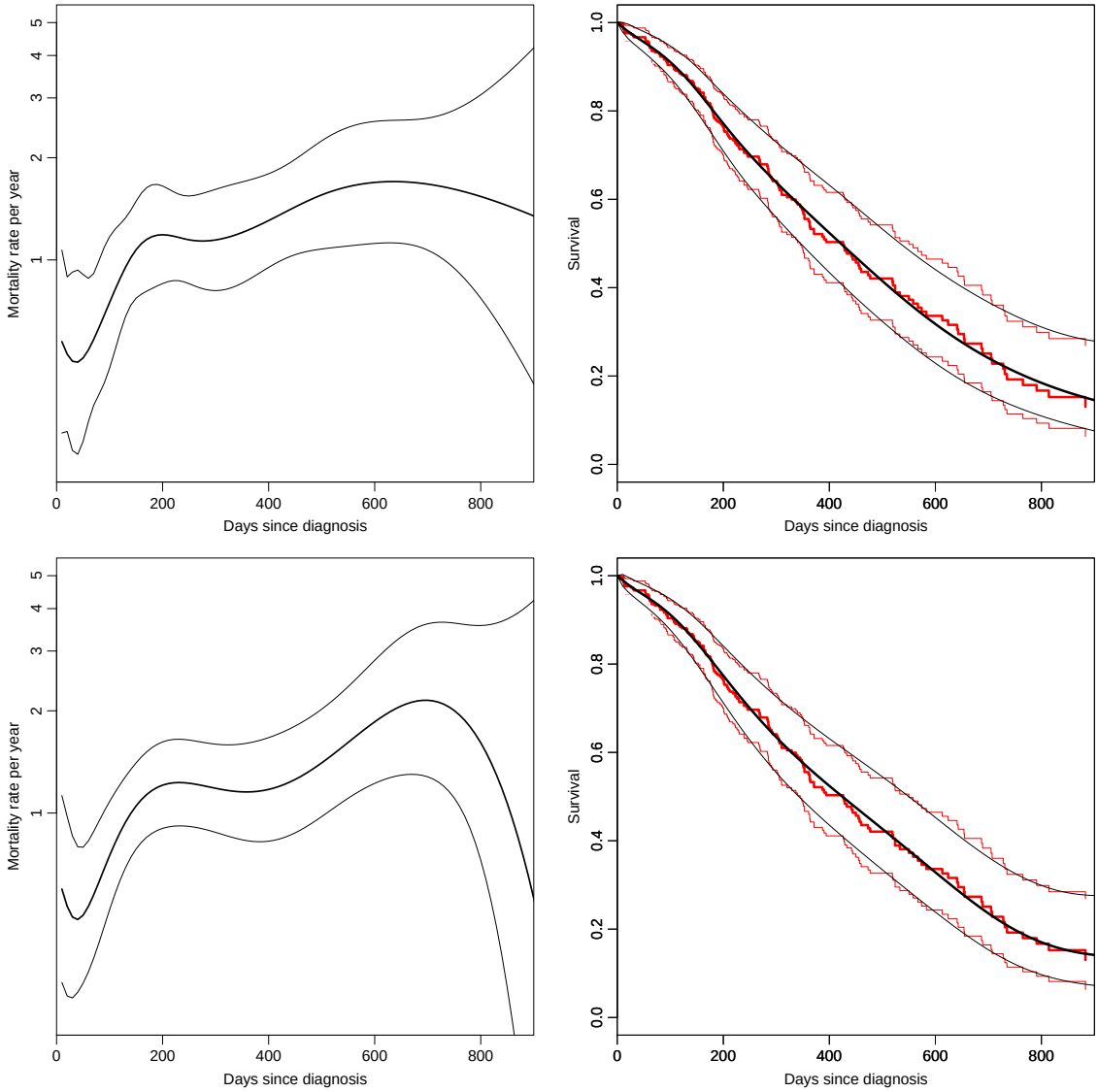

Figure 1: *Estimated underlying hazard (left) and comparison with survival function from the classical Cox-approach (Breslow-estimator). The top panels are produced using natural splines with knots at* $0, 25, 75, 150, 250, 500, 1000$ *and the bottom ones using fractional polynomials of* $t + 30$ *days, with powers* $1/3, 1/2, 0\ (= \log), 1, 1.5, 2$.

of multiple time-scales simultaneously.

Consider the classical situation of a mortality study which is modelled in a Cox-model with time since entry as the time-scale and age at entry as covariate. Using t for time since entry, a for current age and $e = a - t$ for age at entry, the model is:

$$\log\big(\lambda(a,t)\big) = f(t) + \beta e = \big(f(t) - \beta t\big) + \beta a$$

If the underlying log-hazard was assumed linear in time since entry or current age the two approaches would coincide, i.e. replacing age at entry with current age (as a time-dependent variable) would not change the model.

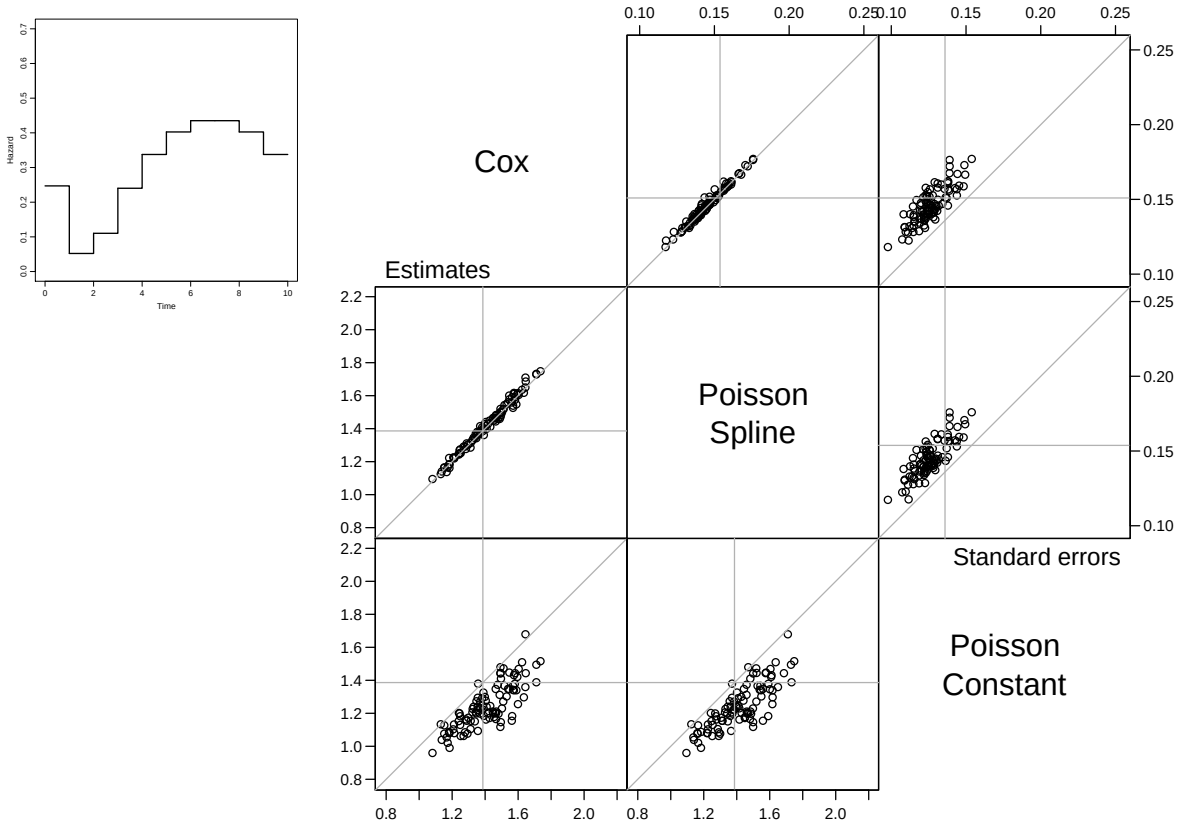

Figure 2: *Underlying hazard in the simulation study (upper left corner) and parameter estimates (lower left) and estimated standard errors (upper right) for the first parameter for three different approaches to modelling. Each point represents results from one simulated dataset. The vertical and horizontal lines in the plots for the estimates are the parameter values used in the simulations, whereas the vertical and horizontal lines in the plots of standard errors are at the empirical standard errors of the parameter estimates.*

However, if arbitrary effects of the three variables t, a and e are allowed we have a genuine extension of the model. Moreover, we will have a model with the same kind of identification problem as is seen in age-period-cohort models. If parametric functions of the three variables are included in the model, i.e.:

$$\log\big(\lambda(a,t)\big) = f(t) + g(a) + h(e)$$

There will be three quantities that can be arbitrarily moved between the three functions, so they can be replaced by three others with the same sum:

$$\begin{aligned}\tilde{f}(t) &= f(a) - \mu_a - \mu_e + \gamma t\\ \tilde{g}(a) &= g(p) + \mu_a - \gamma a\\ \tilde{h}(e) &= h(c) + \mu_a + \gamma e\end{aligned}$$

because $t - a + e = 0$.

Th phrase "age (at entry) is controlled for" in a Cox model often means that the (log)linear effect of age (at entry or current) is included along with an arbitrary effect of time since entry. No consideration neither qualitatively nor quantitatively has been given to wheather a non-linear effect of age at entry and/or current age would be reasonable.

If "controlling for age at entry" is done by entering age at entry as a categorized covariate or parametric function of some sort, the effect of *current* age is still assumed only linear on the log-rate scale. Introducing a non-linear effect for the last time-scale will open the well known can of worms: The age-period-cohort parametrization problems, as mentioned above.

Example: Renal failure data.

As an example we shall consider an extension of the analysis of time to death among diabetic patients with nephrotic range albuminuria (NRA), i.e. a U-albumin excretion exceeding 300 mg/24h, as reported in [3].

The data base for this analysis is illustrated in the Lexis diagrams in figure 3. The results from the paper's table 3 are shown here in table 1.

There is clearly a linear effect of age at entry (or current age), but the analysis presented does not address the question of whether there are non-linear effects of the two time-scales.

Table 1: *Results from a traditional Cox-analysis with remission as time-dependent covariate, as reported in Hovind et al. (2004), table 4.*

			Remission
	Total	Yes	No
No. patients	125	32	93
No. events	77	8	69
Follow-up time (years)	1084.7	259.9	824.8
Cox-model:			
time-scale:	Time since nephrotic range albuminuria (NRA)		
Entry:	2.5 years of GFR-measurements after NRA		
Outcome:	ESRD or Death		
Estimates:	RR	95% c.i.	p
Fixed covariates:			
Sex (F vs. M):	0.92	(0.53,1.57)	0.740
Age at NRA (per 10 years):	1.42	(1.08,1.87)	0.011
Time-dependent covariate:			
Obtained remission:	0.28	(0.13,0.59)	0.001

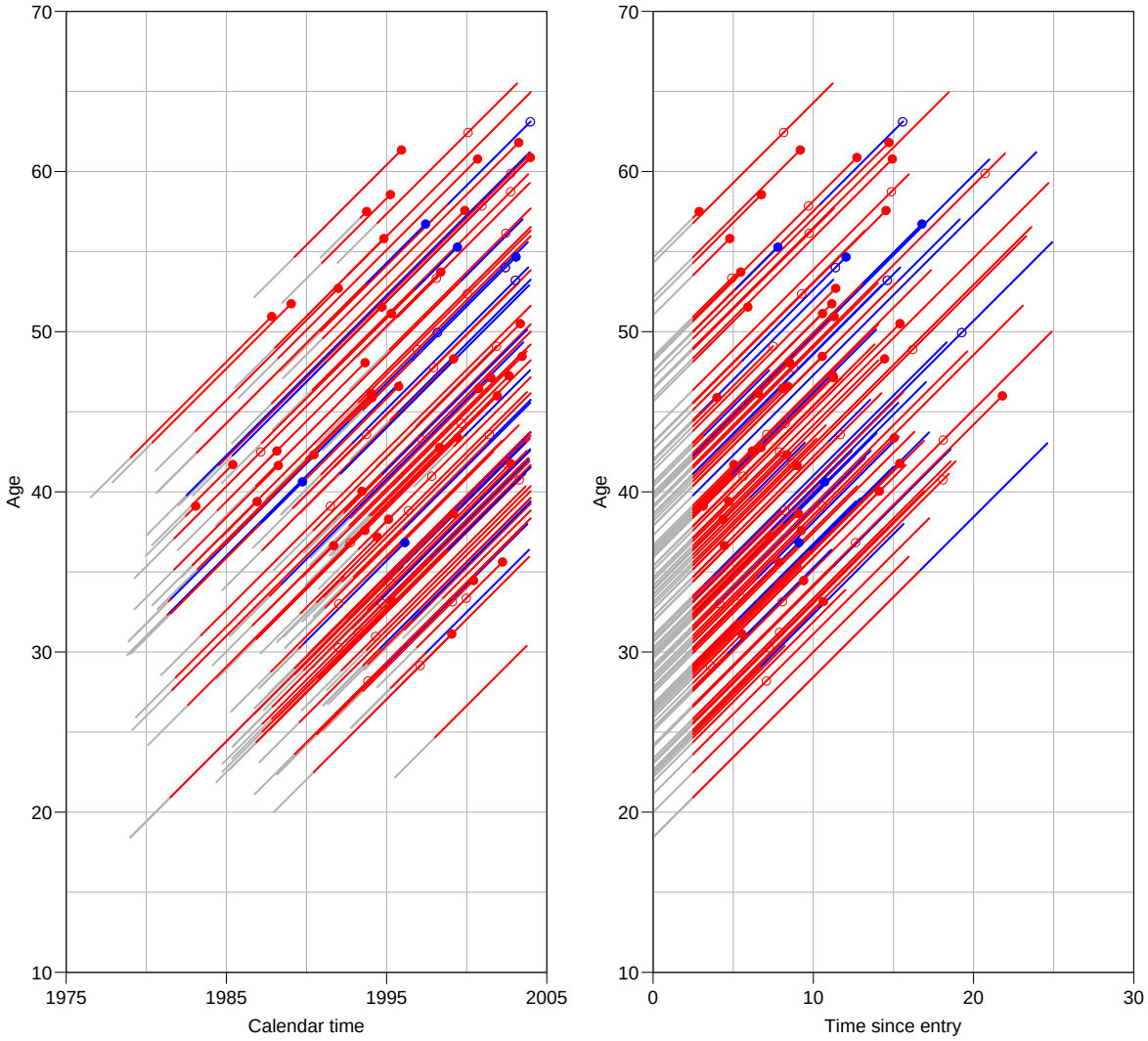

Figure 3: *Trajectories of the 126 patients in the study. The light gray lines is the part of the follow-up not counted, the 2.5 years of GFR-measurements required to enter. The red parts are follow up without remission and the blue are follow up after remission.*
The left panel is age-calendar time, whereas the right panel is the age-time on study, which is the basis for the analysis here.

The layout of the dataset

The dataset includes 96 patients out of which 29 have a remission in the course of the study. Since we have a time-dependent variable, "remission", the individuals that experience a remission are represented in the dataset with two records. The dataset therefore includes $96 + 29 = 125$ records, the first few are shown here:

```
   ptnr sex hba1c     diag    entry     exit fail    birth rem event eventdat
1    17   1 10.46 1993.514 1996.014 1997.095    1 1967.942   0     2 1997.095
2    26   2 10.60 1987.036 1989.536 1989.815    0 1959.304   0     1 1996.137
3    26   2 10.60 1987.036 1989.815 1996.137    1 1959.304   1     1 1996.137
```

```
4    27   2  8.00 1985.347 1987.847 1993.240    1 1962.012   0     3 1993.240
5    33   1  9.64 1992.744 1995.244 1995.718    0 1950.746   0     0 2003.994
6    33   1  9.64 1992.744 1995.718 2003.994    0 1950.746   1     0 2003.994
```

Patient 26 has a remission at 1989.8 and thus contributes two records, the latter with `rem`=1.

Splitting the data into intervals of 3 months length will produce a dataset with 4426 observations. Here the 125 records are further split by time since entry, so the first few records in the split dataset are:

```
   Expand    Entry     Exit Fail Time ptnr sex     diag    birth rem
1       1 1996.014 1996.264    0 2.50   17   1 1993.514 1967.942   0
2       1 1996.264 1996.514    0 2.75   17   1 1993.514 1967.942   0
3       1 1996.514 1996.764    0 3.00   17   1 1993.514 1967.942   0
4       1 1996.764 1997.014    0 3.25   17   1 1993.514 1967.942   0
5       1 1997.014 1997.095    1 3.50   17   1 1993.514 1967.942   0
6       2 1989.536 1989.786    0 2.50   26   2 1987.036 1959.304   0
7       2 1989.786 1989.815    0 2.75   26   2 1987.036 1959.304   0
8       3 1989.815 1990.036    0 2.75   26   2 1987.036 1959.304   1
9       3 1990.036 1990.286    0 3.00   26   2 1987.036 1959.304   1
10      3 1990.286 1990.536    0 3.25   26   2 1987.036 1959.304   1
...
```

Note that patient 26 has the first record split in two, and the second in many pieces, and that the initial two records have `rem`=0 while the other records originating from the second have `rem`=1.

Once we have the time-scale "time since entry" in the variable `Time`, as well as the variable `diag` (date of diagnosis of NRA) and `birth` (date of birth), all other time-scales can be constructed. Any time-scale in this context is but a covariate which changes by 3 months (0.25) from one interval to the next. Hence the assumption is that the rates are constant in 3-month intervals, but that the size of these change according to a smooth function.

The effect of such time-scales are then modelled using standard tools as natural splines, step functions or other parametric functions in a Poisson model, with $\log(Y) = \log(\texttt{Exit} - \texttt{Entry})$ as offset variable.

6 The illness-death model

The example above, one categorical time dependent variable is modelled as a partial analysis of an illness-death model as seen in figure 6, by ignoring (i.e. not modelling) the remission rate, λ. The analysis presented assumes that the two rates μ_{NRA} and μ_{rem} only differ by a constant, i.e. they depend in the same way on time since entry into the study, age at entry (or current age) and sex. In particular it is assumed that time of remission and time since remission does not have any impact on μ_{rem}.

The likelihood for (complete) observation in the illness-death model is just a product of the likelihoods for each transition, where other transitions out of a box are considered as censorings.

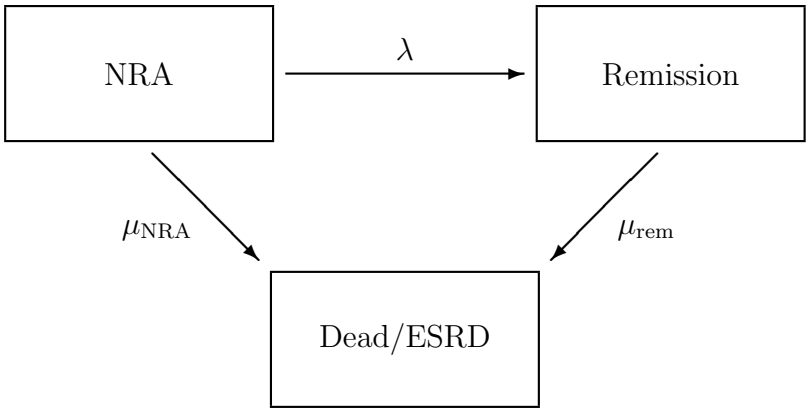

Figure 4: *States and transition rates used in the illness-death-model: μ_{NRA} is the mortality/ESRD rate for patients without remission, λ is the remission rate and μ_{rem} is the mortality/ESRD rate for patients who at some point have had a remission.*

Therefore a Cox-model can easily fit models for the three rates simultaneously with any degree of similarity of covariate effects between the transitions, by simply combining datasets for all the transitions. The traditional ("Andersen-Gill") approach to inclusion of (non-deterministic) time-dependent variables is precisely to include the records for the [NRA→Death] and [Remission→Death] transitions in the dataset. For those with a remission we will have both records, otherwise only the record relating to the [NRA→Death] transition.

However, the drawback of the Cox-modelling approach is that transitions that are modelled simultaneously require the *same* baseline time-scale[2]. Rates where effects of covariates are required (i.e. assumed) to be identical must be included in a common model for a combined dataset.

The most severe drawback of Cox-modelling is the problems arising when needing to deal with multiple time-scales. In an illness-death model as for example the one shown in figure 6 it will be natural to have both time since entry into the study and time since (first) remission as time-scales. Several time-scales can only be incorporated in a Cox-modelling approach if all other time-scales than the baseline are included as time dependent covariates.

However, it seems more reasonable to be able to include the effects of any kind of time-scale in the same fashion. If follow-up and events for each transition are split in little pieces, then modelling of time-scale effects is merely a question of including a suitable parametric function of each of the time-scales in a Poisson model — all time-scales will be available as covariates in the split dataset.

[2]Strictly speaking, only time-scales modelled the same way. In principle one could use time since entry for μ_{NRA} and λ, and time since remission for μ_{rem}, but that would hardly make any clinical sense

7 Stratified models

A stratified Cox-model is a model where the underlying hazard is allowed to differ between strata, i.e. the effect of time varies across the levels of a categorical variable.

Thus this is merely an interaction between time and a categorical variable. If a spline basis has been chosen as model for the time variable, a model with separate baseline hazards for each level of a factor F this is easily modelled by saying:

```
# Simple model with splines modeling time-effects, proportional hazards
#
m1 <- glm( D ~ -1 + ns( t, knots=i.kn, Boundary.knots=b.kn ) + sex + expos +
               offset( log( Y ) ), family=poisson )

# Model with stratified baseline, non-proportional hazards
#
m1 <- glm( D ~ - 1 + ns( t, knots=i.kn, Boundary.knots=b.kn ):F + sex + expos +
               offset( log( Y ) ), family=poisson )
```

The underlying hazards in the points `tp` can then be extracted by multiplying the coefficients from each of the F-levels by the matrix:

```
bs( tp, knots=i.kn, Boundary.knots=b.kn )
```

The period method for survival analysis

Brenner *et al.* [1], suggests a method of adjusting survival for cancer patients by using a so called period analysis. The idea is to use only the survival experience for a fairly recent calendar period, basically because cancer patient mortality in remote time-periods is considered irrelevant. This is best illustrated with reference to a Lexis diagram of cancer patient follow up as shown in figure 5.

Brenner *et al.* propose to restrict analysis to the most recent period and then report results by survival curves, based on the mortality observed in this period only. One way of seeing this proposal is a re-invention of the cross-sectional life-table, it is but a special case of interaction between current date and time since diagnosis. If an interaction model is fitted and only the estimates from the last period are given, then we have the period analysis. Instead one could report separate (cross-sectional) survival curves for each period, and assess whether there is an interaction with calendar time, and if there is report it. Period analysis reports only the last set of parameters, because it is considered the *clinically* most relevant, but apparently without assessing whether this is relevant or not.

8 So who needs the Cox-model?

Since everything which is possible using the Cox-model can be done using the Poisson modelling of split data, there is no loss of capability by switching to Poisson modelling.

The Cox-model is computationally vastly more efficient, and it is easier to produce a survival curve by standard software, which is relevant in clinical studies. The computational efficiency of the Cox-model is presumably one of the reasons for its

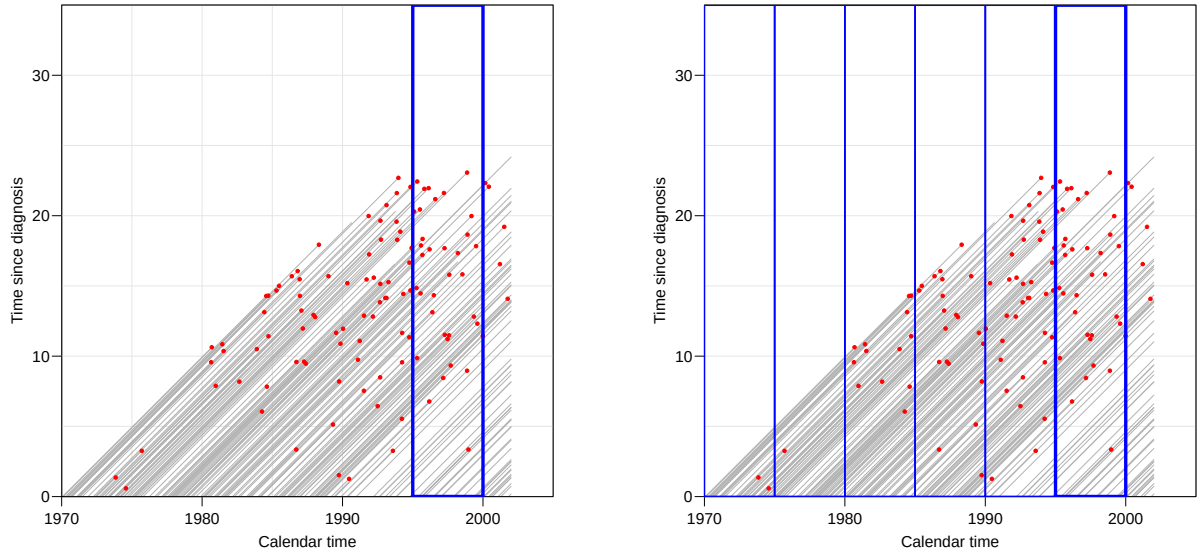

Figure 5: *Lexis diagrams illustrating follow-up of cancer patients. Period analysis suggest to use only the follow-up in the box on the left panel. A stratified model or interaction model assumes different mortality pattern by time since diagnosis in each of the boxes in the right panel. In this case we would even have time-varying strata!*

success. In the 1980s it was effectively the only possibility; the use of Poisson models for datasets with 10000+ observations was out of the question in practical analyses due to lack of computing power. An entire generation of statisticians have learnt to think in terms of one time scale and was never exposed to the demographic way of looking at individual-based follow-up studies. The demographic thinking was restricted to tabulated data from cancer registries and the like, and the intimate connection between the two is still not widely recognized, particularly in the epidemiological community.

A drawback of the Cox-model is the overly detailed modelling of survival curves that may lead to over-interpretation of little humps and notches on the survival curve.

When stratification or time-dependent variables are involved, the facilities in the standard Cox-analysis programs limits the ways in which the desired interactions can be modelled, and moreover distracts the user from realizing that other interactions between covariates may be of interest.

Thus it seems that the Cox-model is useful in the following cases:

- Clinical follow-up studies with only one relevant timescale and the focus on the effect of other covariates than time.
- Studies where a pathologically detailed modelling of one time-scale is desired.
- Studies analysed on computing equipment pre-1985.

In other settings it seems preferable to split time and use the Poisson approach, because it:

- Clarifies the distinction between (risk) time as response variable and time(scales) as covariates.

- Enables smoothing of the effects of time-scales using standard regression tools. In particular it allows more credible estimates of survival functions in the simple case with only time since entry as time-scale.
- Enables sensible modelling of interactions between time-scales and other variable (and between time-scales).

As the necessary computing power and software is available, the computational problems encountered previously are now non-existent.

However, the user-interface to the Poisson modelling is more complex than that offered by standard packages for the Cox-model. This is partly because the Poisson approach requires a more explicit specification of time-scales and how to model them. The latter should however be considered an advantage from a scientific point of view, because it explicitly requires the researcher to make informed choices about which time-scales are relevant, how to model the effect of them, and in particular how to report these effects.

References

[1] H Brenner, O Gefeller, and T Hakulinen. Period analysis for 'up-to-date' cancer survival data: theory, empirical evaluation, computational realisation and applications. *Eur J Cancer*, 40(3):326–335, Feb 2004.

[2] DR Cox. Regression and life-tables (with discussion). *J. Roy. Statist. Soc B*, 34:187–220, 1972.

[3] Peter Hovind, Lise Tarnow, Peter Rossing, Bendix Carstensen, and Hans-Henrik Parving. Improved survival in patients obtaining remission of nephrotic range albuminuria in diabetic nephropathy. *Kidney Int*, 66(3):1180–1186, Sep 2004.

[4] Whitehead J. Fitting Cox's regression model to survival data using GLIM. *Applied Statistics*, 29(3):268–275, 1980.

[5] P McCullagh and JA Nelder. *Generalized Linear Models*. Chapman & Hall, 2nd edition edition, 1988.

9 Programs used

lung-ex.R

Program that takes the Mayo clinic lung cancer survival data, and analyses it both by the Cox-model and by the Poisson-model for split follow-up data.

Illustrates that the Cox-likelihood and the Poisson-approach gives *exactly* the same results whereas the approaches smoothing the underlying hazard gives almost the same.

Also illustrates how to compute the survival function with confidence limits using the Poisson-approach with smooth underlying hazard.

```
 R 1.9.0
 ---------------------------------------------
 Program:  lung-ex.R
  Folder:  C:\Bendix\Artikler\WntCma\R
 Started:  tirsdag 10. august 2004, 14:29:38
 ---------------------------------------------
> library( splines )
> library( Lexis )

Attaching package 'Lexis':


The following object(s) are masked from package:Useful :

 ci.lin steplines


The following object(s) are masked from package:Epi :

 interp lines.est nice plot.est points.est print.floated ROC ROC.tic steplines tabplot twoby2 weeks

> library( survival )
> data( lung )
> str( lung )
`data.frame': 228 obs. of  10 variables:
 $ inst     : num  3 3 3 5 1 12 7 11 1 7 ...
 $ time     : num   306  455 1010  210  883 ...
 $ status   : num  2 2 1 2 2 1 2 2 2 2 ...
 $ age      : num  74 68 56 57 60 74 68 71 53 61 ...
 $ sex      : num  1 1 1 1 1 1 2 2 1 1 ...
 $ ph.ecog  : num  1 0 0 1 0 1 2 2 1 2 ...
 $ ph.karno : num  90 90 90 90 100 50 70 60 70 70 ...
 $ pat.karno: num  100 90 90 60 90 80 60 80 80 70 ...
 $ meal.cal : num  1175 1225   NA 1150   NA ...
 $ wt.loss  : num  NA 15 15 11 0 0 10 1 16 34 ...
>
> table( lung$status )

  1   2
 63 165
> table( table( lung$time ) )

  1   2   3
146  38   2
>
> system.time(
+ c.res <- coxph( Surv( time, status==2 ) ~ age + factor( sex ),
+                 method="breslow", eps=10^-8, iter.max=25, data=lung )
+             )
[1] 0.01 0.00 0.02   NA   NA
> summary( c.res )
Call:
coxph(formula = Surv(time, status == 2) ~ age + factor(sex),
    data = lung, method = "breslow", eps = 10^-8, iter.max = 25)

  n= 228

               coef exp(coef) se(coef)     z      p
age           0.017     1.017  0.00922  1.84 0.0650
factor(sex)2 -0.513     0.599  0.16746 -3.06 0.0022

             exp(coef) exp(-coef) lower .95 upper .95
age              1.017      0.983     0.999     1.036
factor(sex)2     0.599      1.670     0.431     0.832

Rsquare= 0.06   (max possible= 0.999 )
```

```
Likelihood ratio test= 14.1  on 2 df,   p=0.000874
Wald test            = 13.4  on 2 df,   p=0.00121
Score (logrank) test = 13.7  on 2 df,   p=0.00107

>
> # Cut data at all entry and exit times to form another data frame
> #
> dx <- Lexis( exit=time, fail=(status==2), breaks=sort( c(0,unique( time )) ),
+              include=list( age, sex ), data=lung )
> str( dx )
`data.frame': 20022 obs. of  7 variables:
 $ Expand: num  1 1 1 1 1 1 1 1 1 1 ...
 $ Entry : num  2.85e-07 5.00e+00 1.10e+01 1.20e+01 1.30e+01 ...
 $ Exit  : num   5 11 12 13 15 ...
 $ Fail  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Time  : num  0 5 11 12 13 15 26 30 31 53 ...
 $ age   : num  74 74 74 74 74 74 74 74 74 74 ...
 $ sex   : num  1 1 1 1 1 1 1 1 1 1 ...
>
> # Fit the detailed Posisson model
> #
> system.time(
+ p.res <- glm( Fail ~ factor(Time) - 1 + age + factor( sex ) + offset( log(Exit-Entry) ),
+               family=poisson, data=dx, eps=10^-8, maxit=25 )
+             )
[1] 63.50  2.30 71.45    NA    NA
>
> # Define internal and boundary knots for the spline basis and fit the
> # spline model
> #
> i.kn <- c(25,75,150,250,500)
> b.kn <- c(0,1000)
> system.time(
+ s.res <- glm( Fail ~ ns( Time, knots=i.kn, Bo=b.kn, intercept=F )
+                      + age + factor( sex ) + offset( log(Exit-Entry) ),
+               family=poisson, data=dx, eps=10^-8, maxit=25 )
+             )
[1] 2.55 0.14 2.92   NA   NA
>
> # Function to define fractional polynomials and degress to use
> fp <- function( x, deg ) outer( x, deg, FUN=function( x, y ) ifelse( y==0, log(x), x^y ) )
> f.dg <- c(1/3,1/2,0,1,1.5,2)
> f.of <- 30
> system.time(
+ f.res <- glm( Fail ~ fp( Time + f.of, deg=f.dg ) +
+                      + age + factor( sex ) + offset( log(Exit-Entry) ),
+               family=poisson, data=dx, eps=10^-8, maxit=25 )
+             )
[1] 2.32 0.02 2.53   NA   NA
>
> # Show only the relevant estimates and compare them
> #
> ( cr <- ci.lin( c.res )[,1:2] )
                 Estimate       StdErr
age           0.01701289 0.009221954
factor(sex)2 -0.51256479 0.167462063
> ( pr <- ci.lin( p.res, subset=length( coef( p.res ) )-1:0 )[,1:2] )
                 Estimate       StdErr
age           0.01701289 0.009221954
factor(sex)2 -0.51256480 0.167462060
> ( sr <- ci.lin( s.res, subset=length( coef( s.res ) )-1:0 )[,1:2] )
                 Estimate       StdErr
age           0.01636881 0.009204915
factor(sex)2 -0.51200146 0.167452705
> ( fr <- ci.lin( f.res, subset=length( coef( f.res ) )-1:0 )[,1:2] )
                 Estimate     StdErr
age           0.01689966 0.00921589
factor(sex)2 -0.51427541 0.16736859
>
> all <- cbind(
+        rbind( cr[1,], pr[1,], sr[1,], fr[1,], (cr/pr)[1,], (cr/sr)[1,], (cr/fr)[1,] ),
+        rbind( cr[2,], pr[2,], sr[2,], fr[2,], (cr/pr)[2,], (cr/sr)[2,], (cr/fr)[2,] ) )
> rownames( all ) <- c("Cox","Poisson","Spline","Fract","C/P","C/S","C/F")
> colnames( all ) <- paste( rep( rownames( pr ), rep( 2, 2 ) ),
+                           rep( c("Est","SE"), 2 ) )
> print( round( all, 5 ) )
       age Est  age SE factor(sex)2 Est factor(sex)2 SE
Cox     0.01701 0.00922         -0.51256         0.16746
Poisson 0.01701 0.00922         -0.51256         0.16746
Spline  0.01637 0.00920         -0.51200         0.16745
Fract   0.01690 0.00922         -0.51428         0.16737
C/P     1.00000 1.00000          1.00000         1.00000
C/S     1.03935 1.00185          1.00110         1.00006
C/F     1.00670 1.00066          0.99667         1.00056
>
```

```
> # Compute the estimated cumulative itensity over 10-day periods
> # for 60 year old men, and then the survival function
> new <- data.frame( Time=1:100*10, Entry=rep(0,100), Exit=rep(10,100),
+                    sex=rep(1,100), age=rep(60,100) )
>
> # This is the simple way to get the survival function, but the standar
> # erros does not come with it this way.
> s.pr <- predict( s.res, newdata=new, type="link", se.fit=T )
> s.surv <- exp( -cumsum( c(0,exp( s.pr$fit )) ) )
>
> # In order to get the predictions from the spline model we need to
> # devise the right contrast matrix because we need the covaiance
> # between the pointestimates for log-incidence rates.
>
> # Points where we evaulate the rates
> tt <- 0:100*10
> # The explicit model matrix
> cm <- cbind( rep( 1, 101 ),
+              ns( tt, knots=i.kn, Bo=b.kn, intercept=F ),
+              rep( 60, 101 ),
+              rep( 1, 101 ) )
> # Coefficients from the model
> s.cf <- coef( s.res )[1:9]
> s.vc <- vcov( s.res )[1:9,1:9]
> # Estimates and variance-covariance for the log-rates at times tt
> s.es.lrate <- cm %*% s.cf
> s.vc.lrate <- cm %*% s.vc %*% t(cm)
> # The variance-covariance of rates ( exp( log-rates ) ), using a first
> # order Taylor-approximation:
> s.es.rates <- exp( s.es.lrate )
> s.vc.rates <- diag( s.es.rates[,1] ) %*% s.vc.lrate %*% diag( s.es.rates[,1] )
> # Finally the transformation to cumulative rates:
> cum.mat <- 10 * ( 1 - upper.tri( diag( 101 ) ) )
> cum.mat[1:10,1:10]
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]   10    0    0    0    0    0    0    0    0     0
 [2,]   10   10    0    0    0    0    0    0    0     0
 [3,]   10   10   10    0    0    0    0    0    0     0
 [4,]   10   10   10   10    0    0    0    0    0     0
 [5,]   10   10   10   10   10    0    0    0    0     0
 [6,]   10   10   10   10   10   10    0    0    0     0
 [7,]   10   10   10   10   10   10   10    0    0     0
 [8,]   10   10   10   10   10   10   10   10    0     0
 [9,]   10   10   10   10   10   10   10   10   10     0
[10,]   10   10   10   10   10   10   10   10   10    10
> s.es.crate <- cum.mat %*% s.es.rates
> s.vc.crate <- cum.mat %*% s.vc.rates %*% t( cum.mat )
> s.surv <- exp( - cbind( s.es.crate, sqrt( diag( s.vc.crate ) ) ) %*% ci.mat() )
> # Finally adjustmen of the times to match the end of the intervals
> tt <- c( tt, 1010 )
> s.surv <- rbind( c(1,1,1), s.surv )
>
> # Now for all the plots:
> # First the estimated rates:
> plt( "Lung-rate-s" )
> par( mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> matplot( 1:100*10, exp( cbind( s.pr$fit, s.pr$se.fit ) %*% ci.mat() ) * 36.525,
+          type="l", lwd=c(2,1,1), lty=1, col="black", log="y",
+          xlim=c(0,900), xaxs="i", ylim=c(1/4,5),
+          xlab="Days since diagnosis",
+          ylab="Mortality rate per year")
>
> # Then the survival curves by the two methods
> plt( "Lung-surv-s" )
> par( mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> sf <-  survfit( c.res, newdata=data.frame( sex=factor(1:2), age=60:61 ) )
> plot( sf, lwd=1, col=c("white","red"), conf.int=T,
+           xlab="Days since diagnosis",
+           ylab="Survival", xlim=c(0,900),xaxs="i")
> par( new=T)
> plot( sf, lwd=3, col=c("white","red"), conf.int=F, xlim=c(0,900),xaxs="i")
> matlines( tt, s.surv, lwd=c(3,1,1), col="black", lty=1 )
>
> #---------------------------------------------------------------------
> # Repeat the exercise for the fractional polynomials
>
> # This is the simple way to get the survival function, but the standar
> # erros does not come with it this way.
> f.pr <- predict( f.res, newdata=new, type="link", se.fit=T )
> f.surv <- exp( -cumsum( c(0,exp( f.pr$fit )) ) )
>
> # In order to get the predictions from the spline model we need to
> # devise the right contrast matrix because we need the covaiance
> # between the pointestimates for log-incidence rates.
>
```

```
> # Points where we evaulate the rates
> tt <- 0:100*10
> # The explicit model matrix
> cm <- cbind( rep( 1, 101 ),
+              fp( tt + f.of, deg=f.dg ),
+             rep( 60, 101 ),
+             rep( 1, 101 ) )
> # Coefficients from the model
> f.cf <- coef( f.res )[1:9]
> f.vc <- vcov( f.res )[1:9,1:9]
> # Estimates and variance-covariance for the log-rates at times tt
> f.es.lrate <- cm %*% f.cf
> f.vc.lrate <- cm %*% f.vc %*% t(cm)
> # The variance-covariance of rates ( exp( log-rates ) ), using a first
> # order Taylor-approximation:
> f.es.rates <- exp( f.es.lrate )
> f.vc.rates <- diag( f.es.rates[,1] ) %*% f.vc.lrate %*% diag( f.es.rates[,1] )
> # Finally the transformation to cumulative rates:
> cum.mat <- 10 * ( 1 - upper.tri( diag( 101 ) ) )
> cum.mat[1:10,1:10]
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]   10    0    0    0    0    0    0    0    0     0
 [2,]   10   10    0    0    0    0    0    0    0     0
 [3,]   10   10   10    0    0    0    0    0    0     0
 [4,]   10   10   10   10    0    0    0    0    0     0
 [5,]   10   10   10   10   10    0    0    0    0     0
 [6,]   10   10   10   10   10   10    0    0    0     0
 [7,]   10   10   10   10   10   10   10    0    0     0
 [8,]   10   10   10   10   10   10   10   10    0     0
 [9,]   10   10   10   10   10   10   10   10   10     0
[10,]   10   10   10   10   10   10   10   10   10    10
> f.es.crate <- cum.mat %*% f.es.rates
> f.vc.crate <- cum.mat %*% f.vc.rates %*% t( cum.mat )
> f.surv <- exp( - cbind( f.es.crate, sqrt( diag( f.vc.crate ) ) ) %*% ci.mat() )
> # Finally adjustmen of the times to match the end of the intervals
> tt <- c( tt, 1010 )
> f.surv <- rbind( c(1,1,1), f.surv )
>
> # Now for all the plots:
> # First the estimated rates:
> plt( "Lung-rate-f" )
> par( mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> matplot( 1:100*10, exp( cbind( f.pr$fit, f.pr$se.fit ) %*% ci.mat() ) * 36.525,
+          type="l", lwd=c(2,1,1), lty=1, col="black", log="y",
+          xlim=c(0,900), xaxs="i", ylim=c(1/4,5),
+          xlab="Days since diagnosis",
+          ylab="Mortality rate per year" )
>
> # Then the survival curves by the two methods
> plt( "Lung-surv-f" )
> par( mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> sf <-  survfit( c.res, newdata=data.frame( sex=factor(1:2), age=60:61 ) )
> plot( sf, lwd=1, col=c("white","red"), conf.int=T, xlim=c(0,900), xaxs="i",
+           xlab="Days since diagnosis",
+           ylab="Survival")
> par( new=T)
> plot( sf, lwd=3, col=c("white","red"), conf.int=F, xlim=c(0,900), xaxs="i" )
> matlines( tt, f.surv, lwd=c(3,1,1), col="black", lty=1 )
>
 ---------------------------------------------
 Program: lung-ex.R
  Folder: C:\Bendix\Artikler\WntCma\R
   Ended: tirsdag 10. august 2004, 14:30:59
 Elapsed: 00:01:21
 ---------------------------------------------
```

Cox-sim-c.R

Program that simulates survival datasets and anlyses them by three different models, and produces plots to compare them.

```
 R 1.9.0
 ---------------------------------------------
 Program:  Cox-sim-c.R
  Folder:  C:\Bendix\Artikler\WntCma\R
 Started:  mandag 02. august 2004, 14:53:35
 ---------------------------------------------
> library( survival )
> library( splines )
> library( Lexis )
```

```
Attaching package 'Lexis':


The following object(s) are masked from package:Useful :

 ci.lin steplines


The following object(s) are masked from package:Epi :

 interp lines.est nice plot.est points.est print.floated ROC ROC.tic steplines tabplot twoby2 weeks

>
> Sim.const <-
+ function( fu = 10,
+            N = 200,
+          haz = 1/fu,
+           bi = log( 2 ),
+           bo = -log( 2 ) )
+ {
+ # Function to sumulate N data points from a Cox-model with constant
+ # baseline hazard haz and two continuos covariates with a normal
+ # (ping) resp. log-normal (pong) distribution with RRs of bi and bo,
+ # respectively. Times are censores at time fu.
+
+ # Simulate covariates and compute RR
+ ping <- rnorm( N )
+ pong <- exp( rnorm( N ) )
+ RR   <- exp( ping * bi + pong * bo )
+ # Simulate the survival times
+ time <- -log( runif( N ) ) / ( haz * RR )
+ # Which ones were deaths
+ fail <- as.numeric( time < fu )
+ # Censor the times
+ time <- pmin( fu, time )
+ # Return data
+ data.frame( time=time, fail=fail, ping=ping, pong=pong )
+ }
>
>
> Sim.haz <-
+ function( hz = rep( 1/10, 10 ),
+           ni = length( hz ),
+            N = 200,
+           bi = log( 2 ),
+           bo = -log( 2 ) )
+ {
+ # Function to sumulate N data points from a Cox-model with
+ # baseline hazard hz (given as hazards in intervals of length 1) and
+ # two continuos covariates with a normal (ping) resp. log-normal
+ # (pong) distribution with RRs of bi and bo, respectively. Times are
+ # censored at time length(hz).
+
+ # First simulate covariates and RR
+ ping <- rnorm( N )
+ pong <- exp( rnorm( N ) )
+ RR   <- exp( ping * bi + pong * bo )
+
+ # Useful function
+ sint <-
+ function( x, y )
+ {
+ # Function to sumulate length(x) survival times with a cumulative
+ # hazard x*y
+ pmin( 1, -log( runif( x ) ) / ( x * y ) )
+ }
+
+ # Simulate for each person (i.e. for each RR) the survival time in
+ # each of the intervals of length 1 (i.e. for each hz)
+ si   <- outer( RR, hz, sint )
+ # Then multiply together as long as we had a survival time exceeding 1
+ # (i.e. put 0s from the first death time less than 1)
+ ok   <- t( apply( ( si >= 1 ), 1, cumprod ) )
+ # Then compute the simulated survival time
+ time <- apply( cbind( si[,1], si[,2:ni] * ok[,1:(ni-1)] ), 1, sum )
+ fail <- as.numeric( time < ni )
+ data.frame( time=time, fail=fail, ping=ping, pong=pong )
+ }
>
> # Define the baseline hazard function
> x <- 1:10
> # haz <- abs( (x-2.3)*(x-12.7) )
> haz <- rep( 1, 10 )
> haz <- 3 * haz / sum( haz )
```

```
>
> plt( "haz-c" )
> par( mar=c(3,3,1,1), mgp=c(3,1,0)/1.5 )
> plot( c(0,x), haz[c(1:10,10)], type="s",
+       ylim=c(0,0.7), lwd=3, xlab="Time", ylab="Hazard" )
>
> Sim.haz( hz = haz, N=20 )
         time fail        ping      pong
1  10.00000000    0 -2.36311150 1.7970969
2  10.00000000    0 -1.23325934 2.1082490
3   0.81456026    1  0.83391569 0.8713003
4  10.00000000    0  0.05474751 2.0151734
5   1.96800055    1  1.73760142 1.3494013
6   0.11602405    1  1.64865673 0.8118940
7   4.50134461    1  0.33638008 0.6004780
8   0.06443735    1  0.22310678 1.2611728
9   0.94988150    1  2.08837050 0.3567075
10  3.73566349    1  0.77143651 0.3887416
11 10.00000000    0 -3.19908283 0.3413883
12  1.54290184    1  1.66640833 0.3189720
13  8.37089382    1  1.22283849 1.0488130
14  0.32142153    1  2.45633169 0.1730299
15  1.46654335    1  0.19312621 0.4166438
16 10.00000000    0 -0.61977668 2.6851711
17 10.00000000    0 -0.70285032 0.8428157
18  1.66896876    1  0.62371878 0.3818595
19  6.84896792    1  0.97933566 2.1948316
20  0.75159817    1  0.16429308 2.1726228
>
> si <- Sim.haz( hz = haz )
> cx <- coxph( Surv( time, fail ) ~ ping + pong, data = si )
>
> # Number of datasets to simulate
> n.sim <- 100
> # Array to collect results: for each dataset, type of model and
> # regression parameter we collect estimates and standard errror.
> res <- array( NA, dim=c(n.sim,3,2,2),
+                   dimnames = list( sim = 1:n.sim,
+                                    model = c("cx","ps","pl"),
+                                      par = names( coef( cx ) ),
+                                     what = c("Estimate","StdErr") ) )
> pct.cens <- numeric( n.sim )
>
> system.time(
+ for( i in 1:n.sim )
+ {
+ si <- Sim.haz( hz=haz, bi=log(4), bo=-log(4) )
+ cx <- coxph( Surv( time, fail ) ~ ping + pong, data = si )
+ ps <- glm( Fail ~ ping + pong + ns( Time, knots=seq(2,8,2), Bo=c(0,10) ) +
+                   offset( log( Exit-Entry ) ),
+            family=poisson, eps=10e-8,
+            data=Lexis( exit=time, fail=fail,
+                        breaks=seq(0,10,0.25),
+                        data=si, include=list( ping, pong ) ) )
+ pl <- glm( fail ~ ping + pong + offset( log( time ) ),
+            family=poisson, eps=10e-8, data=si )
+ pct.cens[i] <- mean( 1-si$fail )
+ res[i,"cx",,] <- ci.lin( cx )[,1:2]
+ res[i,"ps",,] <- ci.lin( ps, subset=2:3 )[,1:2]
+ res[i,"pl",,] <- ci.lin( pl, subset=2:3 )[,1:2]
+ }
+ )
[1] 126.62   0.29 138.45     NA     NA
There were 32 warnings (use warnings() to see them)
>
> save( res, haz, file="../data/sim.cox-c.Rdata" )
>
> quantile( pct.cens )
    0%    25%    50%    75%   100%
0.4550 0.5000 0.5225 0.5500 0.6150
> ftable( as.table( apply( res, 2:4, mean     ) ), row.vars=2:1 )
           what   Estimate     StdErr
par  model
ping cx          1.3991046  0.1444929
     ps          1.4124915  0.1439325
     pl          1.3975393  0.1247730
pong cx         -1.4118479  0.1888851
     ps         -1.4238519  0.1888828
     pl         -1.4093158  0.1775310
> ftable( as.table( apply( res, 2:4, sd       ) ), row.vars=2:1 )
           what    Estimate      StdErr
par  model
ping cx         0.133164100 0.010333584
     ps         0.132485457 0.010187820
     pl         0.117803505 0.009650702
```

```
pong cx         0.177357215 0.018393848
     ps         0.182237184 0.018573384
     pl         0.172392458 0.016801579
> ftable( as.table( apply( res, 2:4, quantile ) ), row.vars=4:2 )
                           0%        25%        50%        75%       100%
what     par  model
Estimate ping cx      1.1336794  1.3060459  1.3979503  1.4742146  1.7866718
              ps      1.1281195  1.3254112  1.4143950  1.4949118  1.7947056
              pl      1.0354242  1.3354874  1.3974619  1.4878014  1.6971339
         pong cx     -1.9389250 -1.5160139 -1.3968602 -1.2903697 -1.0350829
              ps     -1.9527297 -1.5228097 -1.4020293 -1.2963813 -1.0404059
              pl     -1.8955430 -1.5360770 -1.4034182 -1.2884372 -1.0518593
StdErr   ping cx      0.1180908  0.1374335  0.1432891  0.1520282  0.1672459
              ps      0.1181869  0.1365541  0.1432674  0.1509713  0.1659881
              pl      0.1007091  0.1182328  0.1255017  0.1303183  0.1439568
         pong cx      0.1508149  0.1750003  0.1903099  0.2012243  0.2279669
              ps      0.1498236  0.1740911  0.1907421  0.2009992  0.2271104
              pl      0.1432627  0.1650412  0.1789040  0.1907630  0.2177002
>
> ( lse <- range( res[,,"StdErr"] ) )
[1] 0.1007091 0.2279669
> ( leo <- -( lei <- range( c( range( res[,,"ping","Estimate"] ),
+                              range(-res[,,"pong","Estimate"] ) ) ) ) )
[1] -1.035083 -1.952730
>
> #-----------------------------------------------------------------------
> plt( "ping-c" )
> par( mfrow=c(3,3), mar=c(0,0,0,0), oma=c(3,3,3,3), las=1 )
> #
> plot( NA, type="n", xlim=0:1, ylim=0:1, xaxt="n", yaxt="n", bty="n" )
> text( 0.5, 0.5, "Cox", cex=2, font=2 )
> text( 0, 0, "Estimates", cex=1.5, adj=c(0,0) )
> #
> plot( res[,"ps","ping","StdErr"], res[,"cx","ping","StdErr"],
+       xaxt="n", yaxt="n", xlim=lse, ylim=lse, pch=1 )
> box()
> abline( 0, 1, v=sd( res[,"ps","ping","Estimate"] ),
+               h=sd( res[,"cx","ping","Estimate"] ), col=gray(0.7) )
> axis( side=3 )
> #
> plot( res[,"pl","ping","StdErr"], res[,"cx","ping","StdErr"],
+       xaxt="n", yaxt="n", xlim=lse, ylim=lse, pch=1 )
> box()
> abline( 0, 1, v=sd( res[,"pl","ping","Estimate"] ),
+               h=sd( res[,"cx","ping","Estimate"] ), col=gray(0.7) )
> axis( side=3)
> axis( side=4)
> #
> plot( res[,"cx","ping","Estimate"], res[,"ps","ping","Estimate"],
+       xaxt="n", yaxt="n", xlim=lei, ylim=lei, pch=1 )
> box()
> abline( 0, 1, h=log(4), v=log(4), col=gray(0.7) )
> axis( side=2)
> #
> plot( NA, type="n", xlim=0:1, ylim=0:1, xaxt="n", yaxt="n", bty="n" )
> text( 0.5, 0.5, "Poisson\n\nSpline", cex=2, font=2 )
> #
> plot( res[,"pl","ping","StdErr"], res[,"ps","ping","StdErr"],
+       xaxt="n", yaxt="n", xlim=lse, ylim=lse, pch=1 )
> box()
> abline( 0, 1, v=sd( res[,"pl","ping","Estimate"] ),
+               h=sd( res[,"ps","ping","Estimate"] ), col=gray(0.7) )
> axis( side=4 )
> #
> plot( res[,"cx","ping","Estimate"], res[,"pl","ping","Estimate"],
+       xaxt="n", yaxt="n", xlim=lei, ylim=lei, pch=1 )
> box()
> abline( 0, 1, h=log(4), v=log(4), col=gray(0.7) )
> axis( side=1)
> axis( side=2)
> #
> plot( res[,"ps","ping","Estimate"], res[,"pl","ping","Estimate"],
+       xaxt="n", yaxt="n", xlim=lei, ylim=lei, pch=1 )
> box()
> abline( 0, 1, h=log(4), v=log(4), col=gray(0.7) )
> axis( side=1)
> #
> plot( NA, type="n", xlim=0:1, ylim=0:1, xaxt="n", yaxt="n", bty="n" )
> text( 0.5, 0.5, "Poisson\n\nConstant", cex=2, font=2 )
> text( 1, 1, "Standard errors", cex=1.5, adj=c(1,1) )
>
> #-----------------------------------------------------------------------
> plt( "pong-c" )
> par( mfrow=c(3,3), mar=c(0,0,0,0), oma=c(3,3,3,3), las=1 )
>
```

```
> # 1st row
> plot( NA, type="n", xlim=0:1, ylim=0:1, xaxt="n", yaxt="n", bty="n" )
> text( 0.5, 0.5, "Cox", cex=2, font=2 )
> text( 0, 0, "Estimates", cex=1.5, adj=c(0,0) )
>
> plot( res[,"ps","pong","StdErr"], res[,"cx","pong","StdErr"],
+       xaxt="n", yaxt="n", xlim=lse, ylim=lse, pch=1 )
> box()
> abline( 0, 1, v=sd( res[,"ps","pong","Estimate"] ),
+               h=sd( res[,"cx","pong","Estimate"] ), col=gray(0.7) )
> axis( side=3 )
>
> plot( res[,"pl","pong","StdErr"], res[,"cx","pong","StdErr"],
+       xaxt="n", yaxt="n", xlim=lse, ylim=lse, pch=1 )
> box()
> abline( 0, 1, v=sd( res[,"pl","pong","Estimate"] ),
+               h=sd( res[,"cx","pong","Estimate"] ), col=gray(0.7) )
> axis( side=3)
> axis( side=4)
>
> # 2nd row
> plot( res[,"cx","pong","Estimate"], res[,"ps","pong","Estimate"],
+       xaxt="n", yaxt="n", xlim=leo, ylim=leo, pch=1 )
> box()
> abline( 0, 1, h=-log(4), v=-log(4), col=gray(0.7) )
> axis( side=2)
>
> plot( NA, type="n", xlim=0:1, ylim=0:1, xaxt="n", yaxt="n", bty="n" )
> text( 0.5, 0.5, "Poisson\n\nSpline", cex=2, font=2 )
>
> plot( res[,"pl","pong","StdErr"], res[,"ps","pong","StdErr"],
+       xaxt="n", yaxt="n", xlim=lse, ylim=lse, pch=1 )
> box()
> abline( 0, 1, v=sd( res[,"pl","pong","Estimate"] ),
+               h=sd( res[,"ps","pong","Estimate"] ), col=gray(0.7) )
> axis( side=4 )
>
> # 3rd row
> plot( res[,"cx","pong","Estimate"], res[,"pl","pong","Estimate"],
+       xaxt="n", yaxt="n", xlim=leo, ylim=leo, pch=1 )
> box()
> abline( 0, 1, h=-log(4), v=-log(4), col=gray(0.7) )
> axis( side=1)
> axis( side=2)
>
> plot( res[,"ps","pong","Estimate"], res[,"pl","pong","Estimate"],
+       xaxt="n", yaxt="n", xlim=leo, ylim=leo, pch=1 )
> box()
> abline( 0, 1, h=-log(4), v=-log(4), col=gray(0.7) )
> axis( side=1)
>
> plot( NA, type="n", xlim=0:1, ylim=0:1, xaxt="n", yaxt="n", bty="n" )
> text( 0.5, 0.5, "Poisson\n\nConstant", cex=2, font=2 )
> text( 1, 1, "Standard errors", cex=1.5, adj=c(1,1) )
>
 ------------------------------------------
 Program: Cox-sim-c.R
  Folder: C:\Bendix\Artikler\WntCma\R
   Ended: mandag 02. august 2004, 14:55:55
 Elapsed: 00:02:20
 ------------------------------------------
```

Renal-ex.R

Program that reads the renal failure data and performs both the reported Cox-analysis as well as the corresponding Poisson analysis after splitting data. Also checks if 1) there are non-linear effects of age at entry or current age, and if there is any interaction between remission and time since entry (called either time-varying coefficients, or stratified analysis).

```
 R 2.2.0
 ------------------------------------------
 Program:  Renal-ex.R
  Folder:  C:\Bendix\Artikler\WntCma\R
 Started:  torsdag 08. december 2005, 17:41:40
 ------------------------------------------
> library( survival )
```

```
Loading required package: splines
> library( splines )
> library( Epi )
>
> load( file="c:/Bendix/Steno/PHov/nefro/data/ESRD.Rdata" )
> names( tot )[grep("diag",names(tot))] <- "ddate"
> str( tot )
`data.frame': 154 obs. of  11 variables:
 $ ptnr    : num  17 26 26 27 33 33 42 42 46 47 ...
 $ sex     : num  1 2 2 2 1 1 2 2 2 1 ...
 $ hba1c   : num  10.46 10.60 10.60  8.00  9.64 ...
 $ ddate   : num  1994 1987 1987 1985 1993 ...
 $ entry   : num  1996 1990 1990 1988 1995 ...
 $ exit    : num  1997 1990 1996 1993 1996 ...
 $ fail    : int  1 0 1 1 0 0 0 0 1 1 ...
 $ birth   : num  1968 1959 1959 1962 1951 ...
 $ rem     : num  0 0 1 0 0 1 0 1 0 0 ...
 $ event   : num  2 1 1 3 0 0 0 0 2 1 ...
 $ eventdat: num  1997 1996 1996 1993 2004 ...
> tot[1:10,]
   ptnr sex hba1c    ddate    entry     exit fail    birth rem event eventdat
1    17   1 10.46 1993.513 1996.013 1997.094    1 1967.944   0     2 1997.094
2    26   2 10.60 1987.035 1989.535 1989.814    0 1959.306   0     1 1996.136
3    26   2 10.60 1987.035 1989.814 1996.136    1 1959.306   1     1 1996.136
4    27   2  8.00 1985.346 1987.846 1993.239    1 1962.014   0     3 1993.239
5    33   1  9.64 1992.743 1995.243 1995.717    0 1950.747   0     0 2003.993
6    33   1  9.64 1992.743 1995.717 2003.993    0 1950.747   1     0 2003.993
7    42   2    NA 1985.384 1987.884 1996.650    0 1961.296   0     0 2003.955
8    42   2    NA 1985.384 1996.650 2003.955    0 1961.296   1     0 2003.955
9    46   2    NA 1980.919 1983.419 1991.484    1 1952.374   0     2 1991.484
10   47   1  9.30 1984.404 1986.904 1993.650    1 1956.064   0     1 1993.650
> # There is a bug in the coding of date of birth
> tot$birth <- tot$birth - 100 * ( tot$birth > 2000 )
>
> # Fit the Cox-models as reported in the paper
> # Date of birth as explanatory variable
> mb <- coxph( Surv( entry-ddate, exit-ddate, fail ) ~
+              sex + I((birth-1955)/10) + rem, data=tot )
> summary( mb )
Call:
coxph(formula = Surv(entry - ddate, exit - ddate, fail) ~ sex +
    I((birth - 1955)/10) + rem, data = tot)

  n= 154
                       coef exp(coef) se(coef)      z      p
sex                  -0.0432     0.958    0.275 -0.157 0.8800
I((birth - 1955)/10) -0.3776     0.686    0.130 -2.908 0.0036
rem                  -1.2593     0.284    0.385 -3.275 0.0011

                     exp(coef) exp(-coef) lower .95 upper .95
sex                      0.958       1.04     0.559     1.642
I((birth - 1955)/10)     0.686       1.46     0.531     0.884
rem                      0.284       3.52     0.134     0.603

Rsquare= 0.146   (max possible= 0.984 )
Likelihood ratio test= 24.4  on 3 df,   p=2.10e-05
Wald test            = 20.4  on 3 df,   p=0.000138
Score (logrank) test = 22.2  on 3 df,   p=5.89e-05

>
> # Age at entry as explanatory variable
> ma <- coxph( Surv( entry-ddate, exit-ddate, fail ) ~
+              sex + I((ddate-birth-50)/10) + rem, data=tot )
> summary( ma )
Call:
coxph(formula = Surv(entry - ddate, exit - ddate, fail) ~ sex +
    I((ddate - birth - 50)/10) + rem, data = tot)

  n= 154
                           coef exp(coef) se(coef)      z       p
sex                      -0.0553     0.946    0.275 -0.201 0.84000
I((ddate - birth - 50)/10)  0.5219     1.685    0.137  3.822 0.00013
rem                      -1.2624     0.283    0.385 -3.280 0.00100

                         exp(coef) exp(-coef) lower .95 upper .95
sex                          0.946      1.057     0.552     1.622
I((ddate - birth - 50)/10)     1.685      0.593     1.290     2.202
rem                            0.283      3.534     0.133     0.602

Rsquare= 0.179   (max possible= 0.984 )
Likelihood ratio test= 30.3  on 3 df,   p=1.19e-06
Wald test            = 27.1  on 3 df,   p=5.68e-06
Score (logrank) test = 29.4  on 3 df,   p=1.84e-06

>
```

```
> # Age at entry as explanatory variable plus hba1c as explantory
> # variable
> mh <- coxph( Surv( entry-ddate, exit-ddate, fail ) ~
+              sex + I((ddate-birth-50)/10) + hba1c + rem, data=tot )
> summary( mh )
Call:
coxph(formula = Surv(entry - ddate, exit - ddate, fail) ~ sex +
    I((ddate - birth - 50)/10) + hba1c + rem, data = tot)

  n=119 (35 observations deleted due to missing)
                            coef exp(coef) se(coef)      z      p
sex                       -0.250     0.778   0.3607 -0.694 0.4900
I((ddate - birth - 50)/10)  0.516     1.676   0.1573  3.282 0.0010
hba1c                       0.261     1.299   0.0879  2.975 0.0029
rem                       -1.210     0.298   0.5323 -2.273 0.0230

                          exp(coef) exp(-coef) lower .95 upper .95
sex                           0.778      1.285     0.384     1.578
I((ddate - birth - 50)/10)    1.676      0.597     1.231     2.281
hba1c                         1.299      0.770     1.093     1.543
rem                           0.298      3.353     0.105     0.847

Rsquare= 0.25   (max possible= 0.975 )
Likelihood ratio test= 34.2  on 4 df,   p=6.7e-07
Wald test            = 32.6  on 4 df,   p=1.41e-06
Score (logrank) test = 34.9  on 4 df,   p=4.97e-07

>
> attach( tot )
>
> # Plot the Lexis ddaterams
>
> plt( "ESRD-p-a", height=11, width=6, pointsize=18 )
> par( mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> Lexis.diagram( age=c(10,70), date=c(1975,2005) )
> segments( ddate, ddate-birth, entry, entry-birth, lwd=2, col=gray(0.7) )
> segments( entry, entry-birth, exit, exit-birth, lwd=2, col=c("red","blue")[rem+1] )
> points( eventdat, eventdat-birth, pch=c(NA,1)[(event>1)+1], col=c("red","blue")[rem+1] )
> points( exit, exit-birth, pch=c(NA,16)[fail+1], col=c("red","blue")[rem+1] )
> box()
>
> plt( "ESRD-t-a", height=11, width=6, pointsize=18 )
> par( mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> Lexis.diagram( age=c(10,70), date=c(0,30), dlab="Time since entry" )
> segments( ddate-ddate, ddate-birth, entry-ddate, entry-birth, lwd=2, col=gray(0.7) )
> segments( entry-ddate, entry-birth, exit-ddate, exit-birth, lwd=2, col=c("red","blue")[rem+1] )
> points( eventdat-ddate, eventdat-birth, pch=c(NA,1)[(event>1)+1], col=c("red","blue")[rem+1] )
> points( exit-ddate, exit-birth, pch=c(NA,16)[fail+1], col=c("red","blue")[rem+1] )
> box()
>
> plt( "ESRD-p-t", height=11, width=6, pointsize=18 )
> par( mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> Lexis.diagram( age=c(0,60), date=c(1975,2005), alab="Time since diagnosis" )
> segments(  ddate,  ddate-ddate, entry, entry-ddate, lwd=2, col=gray(0.7) )
> segments( entry, entry-ddate,  exit , exit-ddate, lwd=2, col=c("red","blue")[rem+1] )
> points( eventdat, eventdat-ddate, pch=c(NA,1)[(event>1)+1], col=c("red","blue")[rem+1] )
> points( exit, exit-ddate, pch=c(NA,16)[fail+1], col=c("red","blue")[rem+1] )
> box()
>
> # Split the follow-up time i 3-month intervals after time since entry
> spl <- Lexis( entry = entry,
+                exit = exit,
+                fail = fail,
+              origin = ddate,
+              breaks = seq(2.5,25,0.25),
+             include = list(ptnr,sex,ddate,birth,rem),
+                data = tot )

The following object(s) are masked from tot :

 birth ddate entry event eventdat exit fail hba1c ptnr rem sex

> str( spl )
`data.frame': 4426 obs. of  10 variables:
 $ Expand: num  1 1 1 1 1 2 2 3 3 3 ...
 $ Entry : num  1996 1996 1997 1997 1997 1997 ...
 $ Exit  : num  1996 1997 1997 1997 1997 ...
 $ Fail  : num  0 0 0 0 1 0 0 0 0 0 ...
 $ Time  : num  2.5 2.75 3 3.25 3.5 2.5 2.75 2.75 3 3.25 ...
 $ ptnr  : num  17 17 17 17 17 26 26 26 26 26 ...
 $ sex   : num  1 1 1 1 1 2 2 2 2 2 ...
 $ ddate : num  1994 1994 1994 1994 1994 ...
 $ birth : num  1968 1968 1968 1968 1968 ...
 $ rem   : num  0 0 0 0 0 0 0 1 1 1 ...
> spl[1:10,]
```

```
   Expand    Entry     Exit Fail Time ptnr sex    ddate    birth rem
1       1 1996.013 1996.263    0 2.50   17   1 1993.513 1967.944   0
2       1 1996.263 1996.513    0 2.75   17   1 1993.513 1967.944   0
3       1 1996.513 1996.763    0 3.00   17   1 1993.513 1967.944   0
4       1 1996.763 1997.013    0 3.25   17   1 1993.513 1967.944   0
5       1 1997.013 1997.094    1 3.50   17   1 1993.513 1967.944   0
6       2 1989.535 1989.785    0 2.50   26   2 1987.035 1959.306   0
7       2 1989.785 1989.814    0 2.75   26   2 1987.035 1959.306   0
8       3 1989.814 1990.035    0 2.75   26   2 1987.035 1959.306   1
9       3 1990.035 1990.285    0 3.00   26   2 1987.035 1959.306   1
10      3 1990.285 1990.535    0 3.25   26   2 1987.035 1959.306   1
> # Define relevant timing variables
> spl$age.in  <- spl$ddate - spl$birth
> spl$age.cur <- spl$age.in + spl$Time
> spl$Y <- spl$Exit - spl$Entry
>
> # Poisson model that resembles the Cox-model with
> # ma <- coxph( Surv( entry-ddate, exit-ddate, fail ) ~
> #            sex + I((ddate-birth-50)/10) + rem, data=tot )
> pa <- glm( Fail ~ ns( Time, kn=c(6,10,14), Bo=c(2.5,20) ) +
+                   sex + I((age.in-50)/10) + rem +
+                   offset( log( Y ) ),
+            family=poisson, data=spl )
> summary( pa )

Call:
glm(formula = Fail ~ ns(Time, kn = c(6, 10, 14), Bo = c(2.5,
    20)) + sex + I((age.in - 50)/10) + rem + offset(log(Y)),
    family = poisson, data = spl)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.4157  -0.2149  -0.1618  -0.1124   3.5342

Coefficients:
                                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                                     -3.32793    0.67638  -4.920 8.64e-07
ns(Time, kn = c(6, 10, 14), Bo = c(2.5, 20))1  1.94131    0.59924   3.240 0.001197
ns(Time, kn = c(6, 10, 14), Bo = c(2.5, 20))2  1.67060    0.61828   2.702 0.006892
ns(Time, kn = c(6, 10, 14), Bo = c(2.5, 20))3  4.42812    1.38326   3.201 0.001368
ns(Time, kn = c(6, 10, 14), Bo = c(2.5, 20))4  1.41944    0.57593   2.465 0.013717
sex                                             -0.07126    0.27467  -0.259 0.795286
I((age.in - 50)/10)                              0.53021    0.13679   3.876 0.000106
rem                                             -1.27862    0.38517  -3.320 0.000902

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 746.99  on 4425  degrees of freedom
Residual deviance: 701.86  on 4418  degrees of freedom
AIC: 871.86

Number of Fisher Scoring iterations: 7

> # Show only the relevant regression parameters
> round( ci.lin( pa, subset=length(coef(pa))-2:0, E=T )[,5:7], 3 )
                    exp(Est.)  2.5% 97.5%
sex                     0.931 0.544 1.595
I((age.in - 50)/10)     1.699 1.300 2.222
rem                     0.278 0.131 0.592
> # - and then the ones from the Cox-model
> round( exp( ci.lin( ma )[,-(2:4)] ), 3 )
                           Estimate  2.5% 97.5%
sex                           0.946 0.552 1.622
I((ddate - birth - 50)/10)    1.685 1.290 2.202
rem                           0.283 0.133 0.602
>
> # But we are using a spline form of the time since entry
> # and a linear form of age at entry / current age.
> # We could enter either or both of these in non-linear form.
>
> # First all three (t: time, a: current age, e: entry age)
> p.tae <- glm( Fail ~ ns( Time   , kn=c(6,10,14), Bo=c(2.5,20) ) +
+                      ns( age.in , kn=c(35,45,55), Bo=c(20,65) ) +
+                      ns( age.cur, kn=c(35,45,55), Bo=c(20,65) ) +
+                      sex + rem +
+                      offset( log( Y ) ),
+               family=poisson, data=spl )
> summary( p.tae )

Call:
glm(formula = Fail ~ ns(Time, kn = c(6, 10, 14), Bo = c(2.5,
    20)) + ns(age.in, kn = c(35, 45, 55), Bo = c(20, 65)) + ns(age.cur,
    kn = c(35, 45, 55), Bo = c(20, 65)) + sex + rem + offset(log(Y)),
    family = poisson, data = spl)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.5578  -0.2193  -0.1567  -0.1142   3.5351

Coefficients: (1 not defined because of singularities)
                                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                                      -4.78877    1.76205  -2.718 0.006573
ns(Time, kn = c(6, 10, 14), Bo = c(2.5, 20))1     2.16107    0.62954   3.433 0.000597
ns(Time, kn = c(6, 10, 14), Bo = c(2.5, 20))2     1.84368    0.63497   2.904 0.003690
ns(Time, kn = c(6, 10, 14), Bo = c(2.5, 20))3     4.63042    1.40163   3.304 0.000955
ns(Time, kn = c(6, 10, 14), Bo = c(2.5, 20))4     1.55921    0.68703   2.269 0.023239
ns(age.in, kn = c(35, 45, 55), Bo = c(20, 65))1   1.30910    0.88453   1.480 0.138873
ns(age.in, kn = c(35, 45, 55), Bo = c(20, 65))2   2.66273    1.29501   2.056 0.039768
ns(age.in, kn = c(35, 45, 55), Bo = c(20, 65))3   4.69927    2.04627   2.297 0.021647
ns(age.in, kn = c(35, 45, 55), Bo = c(20, 65))4   2.41245    3.18608   0.757 0.448940
ns(age.cur, kn = c(35, 45, 55), Bo = c(20, 65))1 -0.83482    1.64040  -0.509 0.610814
ns(age.cur, kn = c(35, 45, 55), Bo = c(20, 65))2 -0.87304    1.47137  -0.593 0.552943
ns(age.cur, kn = c(35, 45, 55), Bo = c(20, 65))3 -1.26074    3.46039  -0.364 0.715608
ns(age.cur, kn = c(35, 45, 55), Bo = c(20, 65))4       NA         NA      NA       NA
sex                                              -0.02424    0.27839  -0.087 0.930612
rem                                              -1.27360    0.38701  -3.291 0.000999

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 746.99  on 4425  degrees of freedom
Residual deviance: 699.02  on 4412  degrees of freedom
AIC: 881.02

Number of Fisher Scoring iterations: 7

>
> # Then test for the non-linear components of each effect
> p.ta <- update( p.tae, . ~ . - ns( age.in , kn=c(35,45,55), Bo=c(20,65)  ) )
> p.te <- update( p.tae, . ~ . - ns( age.cur, kn=c(35,45,55), Bo=c(20,65)  ) )
> p.ae <- update( p.tae, . ~ . - ns( Time   , kn=c(6,10,14) , Bo=c(2.5,20) ) )
>
> anova( p.tae, p.ta,
+        p.tae, p.te,
+        p.tae, p.ae,
+        test="Chisq" )
Analysis of Deviance Table

Model 1: Fail ~ ns(Time, kn = c(6, 10, 14), Bo = c(2.5, 20)) + ns(age.in,
    kn = c(35, 45, 55), Bo = c(20, 65)) + ns(age.cur, kn = c(35,
    45, 55), Bo = c(20, 65)) + sex + rem + offset(log(Y))
Model 2: Fail ~ ns(Time, kn = c(6, 10, 14), Bo = c(2.5, 20)) + ns(age.cur,
    kn = c(35, 45, 55), Bo = c(20, 65)) + sex + rem + offset(log(Y))
Model 3: Fail ~ ns(Time, kn = c(6, 10, 14), Bo = c(2.5, 20)) + ns(age.in,
    kn = c(35, 45, 55), Bo = c(20, 65)) + ns(age.cur, kn = c(35,
    45, 55), Bo = c(20, 65)) + sex + rem + offset(log(Y))
Model 4: Fail ~ ns(Time, kn = c(6, 10, 14), Bo = c(2.5, 20)) + ns(age.in,
    kn = c(35, 45, 55), Bo = c(20, 65)) + sex + rem + offset(log(Y))
Model 5: Fail ~ ns(Time, kn = c(6, 10, 14), Bo = c(2.5, 20)) + ns(age.in,
    kn = c(35, 45, 55), Bo = c(20, 65)) + ns(age.cur, kn = c(35,
    45, 55), Bo = c(20, 65)) + sex + rem + offset(log(Y))
Model 6: Fail ~ ns(age.in, kn = c(35, 45, 55), Bo = c(20, 65)) + ns(age.cur,
    kn = c(35, 45, 55), Bo = c(20, 65)) + sex + rem + offset(log(Y))
  Resid. Df Resid. Dev   Df Deviance P(>|Chi|)
1      4412     699.02
2      4415     700.33   -3    -1.31      0.73
3      4412     699.02    3     1.31      0.73
4      4415     699.58   -3    -0.55      0.91
5      4412     699.02    3     0.55      0.91
6      4415     707.09   -3    -8.07      0.04
>
> # Extract the estimates of remission
> rem.res <- rbind(
+   ci.lin( ma   , subset="rem" ),
+   ci.lin( pa   , subset="rem" ),
+   ci.lin( p.tae, subset="rem" ),
+   ci.lin( p.ta , subset="rem" ),
+   ci.lin( p.te , subset="rem" ),
+   ci.lin( p.ae , subset="rem" ) )
> rownames( rem.res ) <- c( "Cox",
+                           "p.t",
+                           "p.tae",
+                           "p.ta",
+                           "p.te",
+                           "p.ae" )
> round( exp( rem.res[,-(2:4)] ), 3 )
      Estimate  2.5% 97.5%
Cox      0.283 0.133 0.602
p.t      0.278 0.131 0.592
p.tae    0.280 0.131 0.597
p.ta     0.284 0.133 0.607
```

```
p.te     0.277 0.130 0.590
p.ae     0.271 0.126 0.580
>
> plt( "ESRD-rates" )
> par( mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> # Estimates of the estimated occurrence rates in the model with
> # time since entry as the underlying timescale and age at entry
> # as a linear effect:
> nd <- data.frame( Time = rep( seq(2.5,20,,100), 2),
+                     sex = rep( 1, 200 ),
+                  age.in = rep( 50, 200),
+                     rem = rep( 0:1, each=100 ),
+                       Y = rep( 1000, 200 ) )
> pr.pa <- predict( pa, newdata=nd, se.fit=T )
> pr.ci <- cbind( pr.pa$fit, pr.pa$se.fit ) %*% ci.mat()
> pr.ci <- cbind( pr.ci[1:100,], pr.ci[101:200,] )
> matplot( nd$Time[1:100], exp( pr.ci ), type="l",
+          log="y", xlab="Time since diagnosis (years)",
+          ylim=c(5,2000), ylab="Rate of ESRD/death per 1000 person-years",
+          lwd=rep(c(3,1,1),2), lty=1, col=rep(c("black","red"),each=3) )
> text( cnr(2,98), "50 year old male", adj=c(0,1) )
>
> Dr <- (spl$Exit-spl$ddate)[spl$Fail==1 & spl$rem==1]
> Dn <- (spl$Exit-spl$ddate)[spl$Fail==1 & spl$rem==0]
> points( Dn, rep(5,length(Dn)) )
> points( Dr, rep(5,length(Dr)), pch=16, col="red" )
>
> plt( "ESRD-rates-i" )
> par( mar=c(3,3,1,1), mgp=c(3,1,0)/1.6 )
> matplot( nd$Time[1:100], exp( pr.ci ), type="l",
+          log="y", xlab="Time since diagnosis (years)",
+          ylim=c(5,2000), ylab="Rate of ESRD/death per 1000 person-years",
+          lwd=rep(c(3,1,1),2), lty=1, col=rep(c("black","red"),each=3) )
> text( cnr(2,98), "50 year old male", adj=c(0,1) )
>
> # Now a model with interaction (timevarying coefficients)
> pai <- update( pa, . ~ . + rem:ns( Time, kn=c(6,10,14), Bo=c(2.5,20) ) )
Warning message:
fitted rates numerically 0 occurred in: glm.fit(x = X, y = Y, weights = weights, start = start, etastart = etastart,
>
> pr.pai <- predict( pai, newdata=nd, se.fit=T )
> pri.ci <- cbind( pr.pai$fit, pr.pai$se.fit ) %*% ci.mat()
> pri.ci <- cbind( pri.ci[1:100,], pri.ci[101:200,] )
> matlines( nd$Time[1:100], exp( pri.ci ), type="l",
+           lwd=rep(c(3,1,1),2), lty=2, col=rep(c("black","red"),each=3) )
> points( Dn, rep(5,length(Dn)) )
> points( Dr, rep(5,length(Dr)), pch=16, col="red" )
>
>
 ------------------------------------------
 Program: Renal-ex.R
  Folder: C:\Bendix\Artikler\WntCma\R
   Ended: torsdag 08. december 2005, 17:41:51
 Elapsed: 00:00:11
 ------------------------------------------
```