Who needs the Cox model anyway?

SDCC
August 2019
http://bendixcarstensen.com/WntCma.pdf
Version 7

Compiled Saturday 3rd August, 2019, 21:41
from: /home/bendix/teach/AdvCoh/art/WntCma/WntCma.tex

Bendix Carstensen Steno Diabetes Center Copenhagen, Gentofte, Denmark
& Department of Biostatistics, University of Copenhagen
bcar0029@regionh.dk b@bxc.dk
http://BendixCarstensen.com

Contents

Chapter 1

Theory

1.1 Introduction

The purpose of this note is to give an overview of the relationship between the Cox-model and the corresponding Poisson model(s). The first chapter lays out the theory establishing the equality of the Cox-model and a particular Poisson model. The second chapter demonstrates this equality and shows sane alternatives to the Cox model, through worked examples. In section 2.6 is a condensed overview of R-code needed to estimate and report results from a model with smooth baseline hazard(s).

It should be noted that the observation that the Cox-model is equivalent to a specific Poisson model is by no means new; it was already pointed out in 1976 by Theodore Holford in theory [3] and in practice by John Whitehead in 1980 [4].

I am grateful to Paul Dickman and Lars Diaz for critical remarks that improved the note.

1.2 History and current status

In the last 40–50 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 a certain aura of complexity to concepts such as time-dependent covariates, stratified analysis, delayed entry and time-varying coefficients.

More unfortunate, however, is that the use of this particular technique for survival analysis has become a dominant tool in epidemiology too, largely restricting models for occurrence rates to models with only one time scale, and effectively concealing a vital part of the determinant — the baseline hazard, from the researchers.

1.2.1 Overview

If survival studies is viewed in the light of the demographic tradition, the basic observation is not one time to event (or censoring) for each individual, but rather many small pieces of follow up from each individual. This makes concepts clearer as modeling 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, but with a likelihood which is a product across intervals.

In this set up, time(scale) is then correctly viewed as a covariate rather than a response, and risk time (exposure time) a part of the response. From a practical point of view time-dependent covariates will not have any special status relative to other covariates. 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 modeling tools needed reduces to Poisson regression (and ultimately logistic regression) — standard generalized linear models.

The Cox-model may actually be viewed as a special case of a Poisson model where the detail in modeling of the time covariate has been taken *ad absurdum*, namely with one parameter per failure time. The main advantage of the demographic view is therefore that researchers will be forced to explicitly consider which time-scale(s) to use and to what degree of detail it is relevant to model interactions between time scales and other covariates.

Contrary to this, Poisson modeling of disease rates and follow-up studies in epidemiology has traditionally (and until 1990 for good computational reasons) been restricted to analysis of tables where rates have been assumed constant over fairly broad time-spans, typically 5 years, as most methods have been developed in cancer epidemiology, where 5 years is considered a short age-span. This approach is essentially one where initial tabulation of data unnecessarily limits the flexibility of modeling (and discards information). If follow-up time both in survival and cohort studies are considered in small intervals, the smoothing of rates can be done with standard regression tools in Poisson modeling. The practical implementation of this type of modeling requires a splitting of the follow-up in many small intervals, and hence Poisson modeling of datasets with many records, each representing a small piece of the follow-up time for a person.

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, and using a single timescale. This can however also be produced from a model using a smooth parametric form for the occurrence rates.

1.3 Time: Response or covariate?

Both, actually.

A common exposition of survival analysis is one which takes the survival time X, as response variable, albeit not fully observed, but limited by the censoring time, Z. Thus data are taken as (X, Z), where we only observe the time $\min(X, Z)$ and the event indicator $\delta = 1\{X < Z\}$.

However from a life-table (demographic) point of view the survival time is better viewed as a covariate, and only differences (*risk* time) should be considered responses. In a life-table, differences on the time scale are accumulated as risk time whereas the *position* on the time scale (age) 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 (bivariate) response variable in all situations.

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

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

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 h has the status of risk time, namely the difference between two points on the timescale (in this case $t+h$ and t).

1.3.1 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:

$$L\big(\lambda|(d,y)\big) = (\lambda y)^d \times (1-\lambda y)^{1-d} = \left(\frac{\lambda y}{1-\lambda y}\right)^d (1-\lambda y)$$

$$\log(L) = \ell\big(\lambda|(d,y)\big) = d\log\left(\frac{\lambda y}{1-\lambda y}\right) + \log(1-\lambda y) \approx d\log(\lambda) + d\log(y) - \lambda y$$

where the term $d\log(y)$ can be dispensed with because it does not depend on the parameter λ. The result is an expression which is also the 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 followed over the intervals delimited by t_0, t_1, t_2, t_3, t_4 will be the product of conditional probabilities of the form:

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

Hence the likelihood for a set of empirical rates *looks like* a likelihood for independent Poisson observations, but the observations are not independent, even if the likelihood is a product (of conditional probabilities).

Thus follow-up studies can be analyzed using the Poisson likelihood in any desired detail; 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 for independent variates, not the distribution of the response variable (d, y) — remember that there is not a one-to-one correspondence between models and likelihoods; two different models

[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, in this context y is not a random quantity, but a fixed quantity.

may have identical likelihoods. Hence only inference based on the likelihood is admissible. Any measures deriving from properties of the Poisson distribution as such are in principle irrelevant.

1.4 The Cox-likelihood as a profile likelihood

The Cox model [2] specifies the intensity (rate, λ) as a function of time (t) and the covariates $(x_1, \dots x_p)$ through the linear predictor $\eta_i = \beta_1 x_{1i} + \cdots + \beta_p x_{pi}$ as:

$$\lambda(t, x_i) = \lambda_0(t) \exp(\eta_i)$$

leaving the baseline hazard λ_0 completely unspecified.

Cox devised the *partial* (log-)likelihood for the parameters $\beta = (\beta_1, \dots, \beta_p)$ in the linear predictor

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

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 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 timescale). Thus the model with constant rates in each small interval can be written:

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

using $\alpha_t = \log\big(\lambda_0(t)\big)$ Assume w.l.o.g. the y for these empirical rates are 1. The log-likelihood contributions that contain information on a specific time-scale parameter α_t, relating to a particular time t, will be contributions from the empirical rate $(d, y) = (1, 1)$ with the death at time t, and the empirical rates $(d, y) = (0, 1)$ from all other individuals at risk at time t.

Note that there is exactly one contribution from each individual at risk at t 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 at t. For those intervals on the time-scale where no deaths occur, the estimate of the α_t will be $-\infty$[2], and so these intervals will not contribute to the log-likelihood.

The derivative w.r.t. α_t is:

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

If this 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"):

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

[2] This is because the term $\alpha_t + \eta_{\text{death}}$ vanishes if all $d_i = 0$, and the last term is maximal if $\mathrm{e}^{\alpha_t} = 0 \Leftrightarrow \alpha_t = -\infty$

which is the same as the contribution from time t to Cox's partial likelihood (except for the -1). Thus we may estimate the regression parameters from the Cox model by standard Poisson-regression software by splitting the data finely and specifying the model as having one rate parameter per time interval.

The Cox model could therefore have been formulated as model with a baseline rate modeled by a timescale parameter for each time recorded. This is an exchangeable model for the baseline rate parameters, thus using neither the ordering nor the absolute scaling of the times. The results for the regression parameters will be the same, also for the standard errors. This is illustrated in section 2.1, where fully parametric alternatives to the Cox model is described too.

1.5 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 (at least the following) tools for this in:

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

SAS: A macro `%Lexis`, available at http://BendixCarstensen.com/Lexis, written by Bendix Carstensen. Another macro `%pyrsstep` is by Klaus Rostgaard [6] https://sourceforge.net/p/pyrsstep/wiki/Home/.

R: Function `survSplit` from the `survival` package does the job. The `Epi` package has a function `splitLexis` that does this for `Lexis` objects [5, 1], and in the `popEpi` package there is a faster `data.table` based version, `splitMulti`, which also has a more friendly syntax.

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. In the following we shall restrict attention to the `Lexis` tools in R. A demonstration in Stata by Paul Dickman can be found in http://pauldickman.com/software/stata/compare-cox-poisson/.

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 timescale at (the beginning of) each interval, be that time since entry, current age or calendar time.

In the Poisson modeling, the event is the response, the log-risk time is used as offset and the time scale is used as covariate. Thus Poisson modeling of follow-up data makes a clear distinction between risk time as the response variable and time scale(s) as covariate(s), but it treats the the two components of the response (d, y) differently. A recent addition to the `Epi` package is the family `poisreg`[3], which uses a more intuitive specification of the response as a two-column vector of events and person-years — the empirical rates.

[3]This means that in a `glm` or `gam` model you can specify `family=poisreg`, and then use `cbind(d,y)` as response, with no need for an offset.

1.5.1 Estimation of baseline hazard

Once data has been split in little pieces of follow-up time, the effect of any time scale (as defined at the start of each interval) can be estimated using parametric regression tools such as splines. This will directly produce estimated baseline rates by using standard prediction machinery for generalized linear models with a given set of covariates.

Suppose $h(t)$ is a smooth function of time which is parametrized linearly by the parameters in γ, $h(t) = w'\gamma$ (w and γ are column vectors). The Cox (proportional hazards) model with a smooth baseline hazard can then be formulated as:

$$\log\big(\lambda(t,x)\big) = h(t) + x'\beta = w'\gamma + x'\beta = (w\,x)'\begin{pmatrix}\gamma\\ \beta\end{pmatrix}$$

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 capable of fitting generalized linear models. Rate estimates with confidence intervals are then derived by taking the exponential function of the estimates for the log-rates with confidence intervals.

In the `Epi` package this is handled by the `ci.pred` function that produces predicted rates for a specified set of prediction points.

1.5.2 Estimation of survival function

The survival function is a simple, albeit non-linear, function of the rates:

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

In order to estimate this from a parametric model for the log-rates we need to derive the integral, i.e. a cumulative sum of predictions on the rate scale. 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 $\log\big(\lambda(t)\big)$ to $\lambda(t)$. This is the coordinate-wise exponential function, so the derivative matrix is the diagonal matrix with entries $\lambda(t)$ (formally, $\mathrm{e}^{\log(\lambda(t))}$).

The cumulative sum is obtained by multiplying with a matrix with 1s on and below the diagonal and 0s above, 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 we 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:

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

where $\zeta = (\gamma, \beta)$ is the parameter vector in the model, including the parameters that describe the baseline hazard, and $\mathbf{B}$ is a matrix with I rows, each row corresponding to a time point t_i.

Now let the estimated variance-covariance matrix of ζ be Σ. 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, approximately):

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

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

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

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

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

However, this formula for the variance of the cumulative hazard does not guarantee that the lower bound of the confidence interval for the cumulative hazard is larger than 0. But this can be fixed by computing confidence intervals for the log-cumulative hazard using the δ-method (1^{st}-order Taylor approximation), and back-transforming to the rate scale.

These calculations are implemented in the `Epi` package function `ci.cum`, which requires (at least) 3 objects as arguments: 1) a model object representing a multiplicative model for occurrence rates, 2) a prediction data frame which will produce rate-estimates from the model at a set of equidistant times since some origin, and 3) a scalar representing the distance between the prediction times (in the units in which the person-years was supplied to the model). The function also has a facility for computing the confidence limits on the log-cumulative hazard scale and back transforming to ensure positive lower confidence bounds for the integrated hazard.

Once we have estimated the cumulative hazard function as a function of time we can transform it to the survival function by the exponential. This is implemented in the function `ci.surv` that returns the survival function based on a parametric model.

Chapter 2

Examples

This demonstration uses R, but a demonstration of basic aspects treated here using Stata (by Paul Dickman) can be found in
http://pauldickman.com/software/stata/compare-cox-poisson/.

2.1 Equality of Cox and Poisson modeling: The lung cancer example

In this section we use the lung cancer example data from the `survival` package to illustrate that the results from a Cox model actually are identical to results from (quite) a(n absurd) Poisson model. Moreover we also illustrate two ways to use a parametrically smoothed version of the linear predictor in the Poisson model to obtain a sane estimate for the baseline hazard.

First we load the relevant packages:

```
> library( Epi )
> library( popEpi )
> library( survival )
> library( mgcv )
> print( sessionInfo(), l=F )
R version 3.6.0 (2019-04-26)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 14.04.6 LTS

Matrix products: default
BLAS:   /usr/lib/openblas-base/libopenblas.so.0
LAPACK: /usr/lib/lapack/liblapack.so.3.0

attached base packages:
[1] utils     datasets  graphics  grDevices stats     methods   base

other attached packages:
[1] mgcv_1.8-28       nlme_3.1-139      survival_2.44-1.1 popEpi_0.4.4
[5] Epi_2.37

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.0         lattice_0.20-38    zoo_1.8-4          MASS_7.3-51.1
 [5] grid_3.6.0         plyr_1.8.4         etm_1.0.4          data.table_1.12.0
 [9] Matrix_1.2-17      splines_3.6.0      tools_3.6.0        cmprsk_2.2-7
[13] numDeriv_2016.8-1 parallel_3.6.0     compiler_3.6.0
```

```
> data( lung )
> lung[1:5,]
  inst time status age sex ph.ecog ph.karno pat.karno meal.cal wt.loss
1    3  306      2  74   1       1       90       100     1175      NA
2    3  455      2  68   1       0       90        90     1225      15
3    3 1010      1  56   1       0       90        90       NA      15
4    5  210      2  57   1       1       90        60     1150      11
5    1  883      2  60   1       0      100        90       NA       0
```

Convert sex to a factor:

```
> lung$sex <- factor( lung$sex, labels=c("M","F") )
```

How many distinct event times do we have?

```
> addmargins( table( table( lung$time ) ) )
  1   2   3 Sum
146  38   2 186
```

To avoid tied event times we add a small random quantity to each time:

```
> set.seed(1952)
> lung$time <- lung$time + round(runif(nrow(lung),-3,3),2)
> table( table(lung$time) )
  1
228
```

First we fit a traditional Cox-model for the Mayo Clinic lung cancer data as supplied:

```
> m0.cox <- coxph( Surv( time, status==2 ) ~ age + sex, data=lung )
> summary( m0.cox )
Call:
coxph(formula = Surv(time, status == 2) ~ age + sex, data = lung)

  n= 228, number of events= 165

          coef exp(coef)  se(coef)      z Pr(>|z|)
age   0.017053  1.017199  0.009218  1.850  0.06432
sexF -0.520328  0.594326  0.167512 -3.106  0.00189

     exp(coef) exp(-coef) lower .95 upper .95
age     1.0172     0.9831     0.999    1.0357
sexF    0.5943     1.6826     0.428    0.8253

Concordance= 0.603  (se = 0.025 )
Likelihood ratio test= 14.42  on 2 df,   p=7e-04
Wald test            = 13.75  on 2 df,   p=0.001
Score (logrank) test = 14.01  on 2 df,   p=9e-04
```

This analysis shows that the mortality increases 1.7% per year of age at diagnosis, and that women have some 40% lower mortality than men.

Now we create a Lexis object from the dataset `lung`, to represent the follow-up time and events:

```
> Lung <- Lexis( exit = list( tfe=time ),
+         exit.status = factor( status, labels=c("Alive","Dead") ),
+                data = lung )
```

```
NOTE: entry.status has been set to "Alive" for all.
NOTE: entry is assumed to be 0 on the tfe timescale.
> summary( Lung )
Transitions:
     To
From    Alive Dead  Records:  Events: Risk time:  Persons:
  Alive    63  165       228      165    69631.6       228
> save( Lung, file='lungLx.Rda' )
```

Split data in small intervals, defined by all recorded event and censoring times. Person 32 has only 9 records so he is used for illustration of the structure of a time-split `Lexis` object:

```
> Lung.s <- splitMulti( Lung, tfe=c(0,sort(unique(Lung$time))) )
> summary( Lung.s )
Transitions:
     To
From    Alive Dead  Records:  Events: Risk time:  Persons:
  Alive 25941  165     26106      165    69631.6       228
> Lung.s[lex.id==32,1:10]
   lex.id   tfe lex.dur lex.Cst lex.Xst inst  time status age sex
1:     32  0.00    7.67   Alive   Alive    1 23.89      2  73   M
2:     32  7.67    1.88   Alive   Alive    1 23.89      2  73   M
3:     32  9.55    0.23   Alive   Alive    1 23.89      2  73   M
4:     32  9.78    0.57   Alive   Alive    1 23.89      2  73   M
5:     32 10.35    2.25   Alive   Alive    1 23.89      2  73   M
6:     32 12.60    0.45   Alive   Alive    1 23.89      2  73   M
7:     32 13.05    2.43   Alive   Alive    1 23.89      2  73   M
8:     32 15.48    0.51   Alive   Alive    1 23.89      2  73   M
9:     32 15.99    7.90   Alive    Dead    1 23.89      2  73   M
```

We then fit the Cox model to the `Lexis` data set as well as the time-split `Lexis` data set; note the code is exactly the same, only the `data=` argument differs:

```
>  mL.cox <- coxph( Surv( tfe, tfe+lex.dur, lex.Xst=="Dead" ) ~ age + sex,
+                   eps=10^-11, iter.max=25, data=Lung )
> mLs.cox <- coxph( Surv( tfe, tfe+lex.dur, lex.Xst=="Dead" ) ~ age + sex,
+                   eps=10^-11, iter.max=25, data=Lung.s )
> round( cbind( ci.exp(m0.cox), ci.exp(mL.cox), ci.exp(mLs.cox) ), 6 )
     exp(Est.)     2.5%    97.5% exp(Est.)     2.5%    97.5% exp(Est.)     2.5%    97.5%
age   1.017199 0.998987 1.035743  1.017199 0.998987 1.035743  1.017199 0.998987 1.035743
sexF  0.594326 0.427994 0.825298  0.594326 0.427994 0.825298  0.594326 0.427994 0.825298
```

We see we get the same results from the three different sets of data — they contain exactly the same amount of information.

Now we fit the corresponding Poisson model with factor modeling of the time scale — note that we use the `poisreg` family where we enter events and person-years as a 2 column matrix:

```
> nlevels( factor( Lung.s$tfe ) )
[1] 228
> system.time(
+ mLs.pois.fc <- glm( cbind(lex.Xst=="Dead",lex.dur) ~ 0 + factor(tfe) + age + sex,
+                     family=poisreg, data=Lung.s ) )
```

```
   user  system elapsed
 23.745  35.722  18.622
> length( coef(mLs.pois.fc) )
[1] 230
> cbind( ci.exp(mLs.cox), ci.exp( mLs.pois.fc, subset=c("age","sex") ) )
     exp(Est.)      2.5%     97.5% exp(Est.)      2.5%     97.5%
age  1.0171990 0.9989867 1.0357433 1.0171990 0.9989867 1.0357433
sexF 0.5943256 0.4279945 0.8252978 0.5943256 0.4279945 0.8252978
```

In accordance with the mathematical derivations in the previous chapter, we see that the estimates of the regression coefficients are exactly the same from the Cox model and the Poisson model. The latter has an extra 228 parameters estimated, which is what causes the very long estimation time.

2.1.1 Parametric baseline

To allow for a more realistic model for the baseline rate we now define knots for a spline basis and fit the model with natural splines for the baseline effect of `tfe`. The knots we use are just taken out of thin air:

```
> t.kn <- c(0,25,100,500,1000)
> system.time(
+ mLs.pois.sp <- glm( cbind(lex.Xst=="Dead",lex.dur) ~ Ns(tfe,knots=t.kn) + age + sex,
+                      family=poisreg, data=Lung.s ) )
   user  system elapsed
  0.377   0.450   0.244
```

We also fit the model with a penalized spline model for the effect of `tfe` using `gam` from the `mgcv` package:

```
> system.time(
+ mLs.pois.ps <- gam( cbind(lex.Xst=="Dead",lex.dur) ~ s(tfe) + age + sex,
+                      family=poisreg, data=Lung.s ) )
   user  system elapsed
  2.048   2.755   1.295
> summary( mLs.pois.ps )
Family: poisson
Link function: log

Formula:
cbind(lex.Xst == "Dead", lex.dur) ~ s(tfe) + age + sex

Parametric coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.945162   0.594632 -11.680  < 2e-16
age          0.016287   0.009199   1.770  0.07665
sexF        -0.507182   0.167311  -3.031  0.00243

Approximate significance of smooth terms:
         edf Ref.df Chi.sq  p-value
s(tfe) 2.118  2.671  17.78 0.000583

R-sq.(adj) =  1.92e-05   Deviance explained =  1.7%
UBRE = -0.93029  Scale est. = 1         n = 26106
```

We see that the effective d.f. for the time scale effect (`tfe`) is about 2, so some indication that the arbitrary spline may be over-modeling data.

Finally we make an overall comparison of estimates of age and sex effects from the different approaches:

```
> ests <-
+ rbind( ci.exp(m0.cox),
+         ci.exp(mLs.cox),
+         ci.exp(mLs.pois.fc,subset=c("age","sex")),
+         ci.exp(mLs.pois.sp,subset=c("age","sex")),
+         ci.exp(mLs.pois.ps,subset=c("age","sex")) )
> cmp <- cbind( ests[c(1,3,5,7,9)  ,],
+               ests[c(1,3,5,7,9)+1,] )
> rownames( cmp ) <-
+     c("Cox","Cox-split","Poisson-factor","Poisson-spline","Poisson-penSpl")
> colnames( cmp )[c(1,4)] <- c("age","sex")
> round( cmp,5 )
                   age    2.5%   97.5%     sex    2.5%   97.5%
Cox            1.01720 0.99899 1.03574 0.59433 0.42799 0.82530
Cox-split      1.01720 0.99899 1.03574 0.59433 0.42799 0.82530
Poisson-factor 1.01720 0.99899 1.03574 0.59433 0.42799 0.82530
Poisson-spline 1.01620 0.99805 1.03468 0.59933 0.43163 0.83219
Poisson-penSpl 1.01642 0.99826 1.03491 0.60219 0.43383 0.83589
```

We see that even if the factor model, and by that token also the Cox-model, seem pretty far fetched in their (lack of) assumptions, there is minimal difference to the regression parameter estimates from the models with more realistic assumptions for the baseline rates. So using the Cox model is not likely to produce estimates of regression parameters that are off.

2.1.2 Rates, cumulative rates and survival

Parametric models

Now we compute the estimated rates and cumulative rates over 10-day periods for 60 year old men, and then the survival function at these points.

In order to get the predictions from the spline model we specify a prediction data frame, where we predict rates at equidistant points, using `ci.pred` for the rates. Since we used the `poisreg` family, the predicted rates are by definition per one unit of `lex.dur` (the second column in the response), which in our case is days, so we multiply by 365.25 to get rates per 1 PY.

When we compute the cumulative rates, we must also supply the interval length (distance between values of `tfe` in the prediction data frame):

```
> # prediction data frame with midpoints of 10-day intervals and age and sex
> nd <- data.frame( tfe=seq(5,995,10), age=60, sex="M" )
> #
> # the rates from the spline model (per 1 year)
> lambda <- ci.pred( mLs.pois.sp, nd )*365.25
> # the cumulative rates
> Lambda <- ci.cum ( mLs.pois.sp, nd, int=10 )
> # the survival function
> survP  <- ci.surv( mLs.pois.sp, nd, int=10 )
> #
> # same same for the penalized spline model
```

```
> lambdap <- ci.pred( mLs.pois.ps, nd )*365.25
> Lambdap <- ci.cum ( mLs.pois.ps, nd, intl=10 )
> survPp  <- ci.surv( mLs.pois.ps, nd, intl=10 )
```

So now we have the incidence rates per 1 PY as well as cumulative incidence rates and the corresponding survival function(s) based both on natural splines and a penalized likelihood via `gam`.

Natural spline vs. penalized splines

We can now compare the two smoothing approaches for the baseline hazard:

```
> par( mfrow=c(1,2), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6, bty="n", las=1, lend="butt" )
> matshade( nd$tfe, cbind(lambda,lambdap), plot=TRUE,
+           col=c("blue","red"), lwd=3, lty=c("solid","21"),
+           xlim=c(0,900), xaxs="i", ylim=c(1/5,20), log="y",
+           xlab="Days since diagnosis",
+           ylab="Mortality rate per 1 year")
> matshade( nd$tfe-5, cbind(survP,survPp), plot=TRUE,
+           col=c("blue","red"), lwd=3,, lty=c("solid","21"),
+           xlim=c(0,900), xaxs="i", yaxs="i", ylim=0:1,
+           xlab="Days since diagnosis",
+           ylab="Survival probability")
```

From figure 2.1 we see that there is only slight difference between the two parametric approaches; the penalized splines (red broken curve) smooths a bit more than the natural splines with arbitrarily chosen knots. When transformed to the survival scale, the two approaches are practically indistinguishable.

Comparison with the Cox model

The Breslow-estimator of the survival curve from the corresponding Cox-model for a male aged 60 is obtained from the `m0.cox` object:

```
> sf <-  survfit( m0.cox, newdata=data.frame(sex="M",age=60) )
```

We can extract the baseline rates from the Poisson version of the Cox model as well. Since `lex.dur` is supplied in units of days, to `mLs.pois.fc`, the predicted rates from using `ci.exp` will be in events per day, hence we rescale to events per year. We extract the times from the names of the parameters:

```
> ( nc <- length( coef(mLs.pois.fc) ) )
[1] 230
> br <- ci.exp( mLs.pois.fc, ctr.mat=cbind(diag(nc-2),60,0) )*365.25
> bt <- as.numeric( gsub( "factor\\(tfe)", "", names(coef(mLs.pois.fc))[1:(nc-2)] ) )
> head( cbind(bt,br) )
        bt exp(Est.)       2.5%     97.5%
[1,]  0.00 0.2346923 0.03294122  1.672084
[2,]  7.67 0.9605587 0.13482838  6.843315
[3,]  9.55 7.9011216 1.10905786 56.288968
[4,]  9.78 3.2110276 0.45074761 22.874660
[5,] 10.35 0.8180795 0.11483643  5.827890
[6,] 12.60 4.1167380 0.57788931 29.326606
```

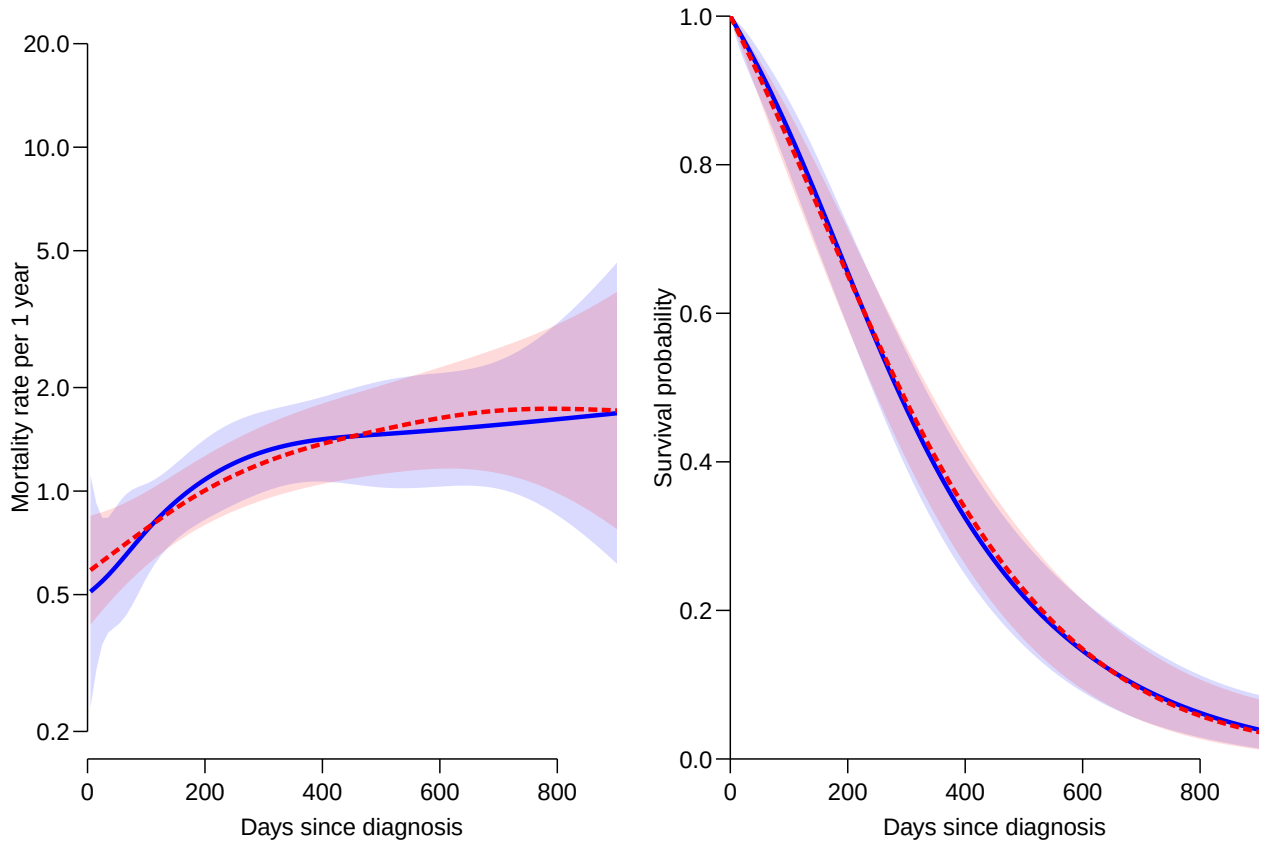

Figure 2.1: *Left panel: Estimated mortality rates for a 60 year old man by Poisson models; blue is the* `glm` *model using a natural spline with pre-chosen knots, red is the* `gam` *model with penalization. Right panel: The resulting survival curves. Shaded areas indicate 95% confidence intervals.* `./lung-rtsurv-sm`

Now we have the predicted rates in intervals between the times observed; the Poisson version of the Cox model implicitly assumes that event rates are constant within intervals between times. Since the deaths occur at the *end* of the intervals, and intervals are named by their *left* endpoint, plotting of the rates must use `type="s"`, which creates steps between successive points where the curve first moves horizontally, then vertically.

```
> par( mfrow=c(1,2), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6, bty="n", las=1, lend="butt" )
> plot( NA, xlim=c(0,900), xaxs="i", ylim=c(1/5,20), log="y",
+           xlab="Days since diagnosis",
+           ylab="Mortality rate per 1 year" )
> lines( bt, br[,1], type="s", col=gray(0.6) )
> matshade( nd$tfe, cbind(lambda,lambdap), # plot=TRUE,
+           col=c("blue","red"), lwd=3, lty=c("solid","21") )
> matshade( nd$tfe, cbind(survP[,-4],survPp[,-4]), plot=TRUE,
+           col=c("blue","red"), lwd=3,, lty=c("solid","21"),
+           xlim=c(0,900), xaxs="i", yaxs="i", ylim=0:1,
+           xlab="Days since diagnosis",
+           ylab="Survival probability")
> lines(sf,lwd=1,lty=c(1,1))
> lines(sf,lwd=2,conf.int=FALSE)
```

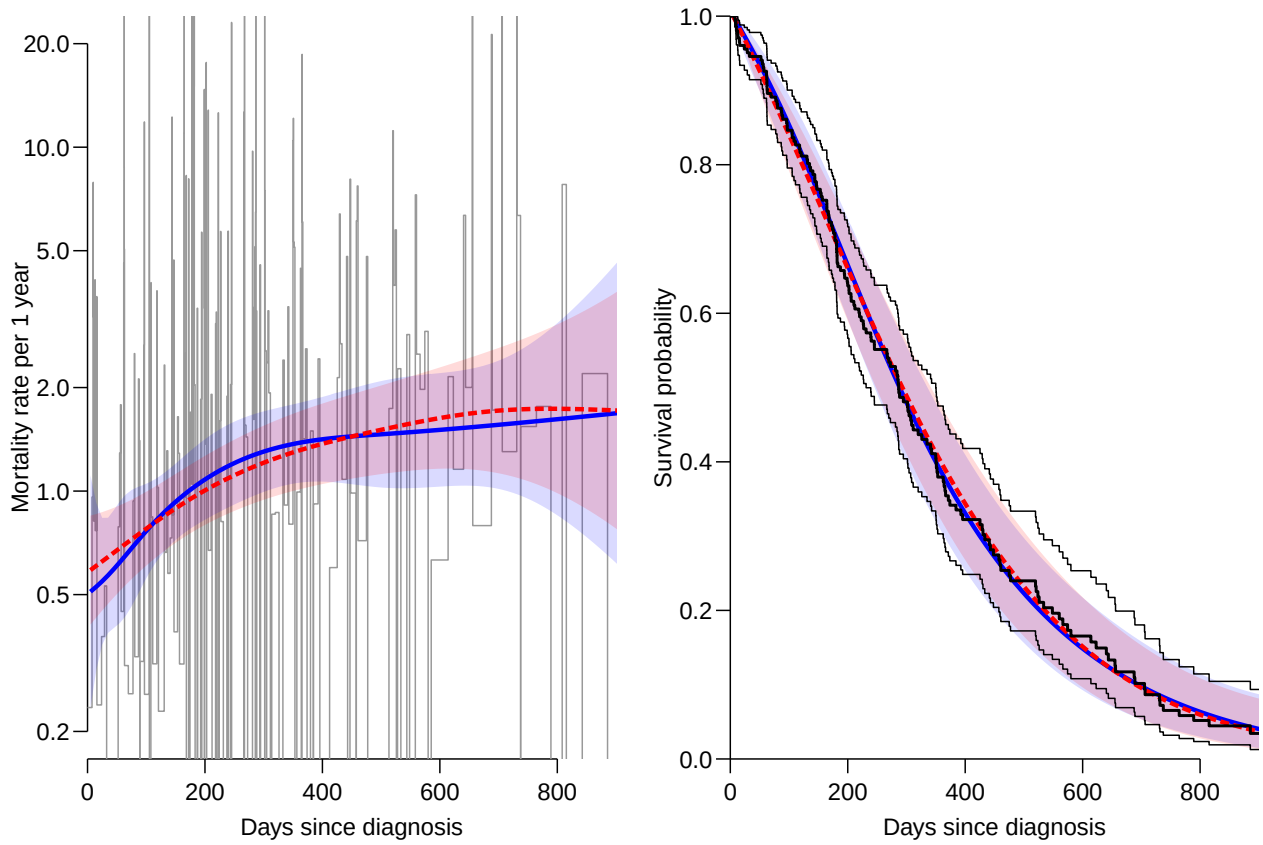

Figure 2.2: *Left panel: Estimated mortality rates by Poisson models; blue is the* `glm` *model using a natural spline with pre-chosen knots, the red is the* `gam` *model with penalization, and the thin gray line indicate the estimated baseline hazard from the Poisson (Cox) model with one parameter per event/censoring time. Right panel: The resulting survival curves, over-laid in black with the Breslow-estimator of the survival curve. Shaded areas and thin lines indicate 95% confidence intervals.* `./lung-rtsurv-cp`

Figure 2.2 has the Cox-model estimates overlaid; strictly speaking the baseline hazard is not really part of the Cox-model, the underlying hazard comes from the corresponding Poisson model, the survival curve is the Breslow estimator. There is a faint indication that the parametric curves produces slightly narrower confidence bands for the survival probabilities than the Breslow-estimator.

2.1.3 Practical time splitting

In practical applications the splitting of time need not be at the times of events and censorings; this was only done above to demonstrate the connection between the Cox model and the Poisson model.

The assumption behind the Poisson approach is essentially only the assumption that a model with constant rates in each small interval gives an adequate description of data. So in practice we would split data in small equidistant intervals. In the lung cancer dataset there are 165 deaths and the total observation period is some 1000 days, some 2.8 years, so we split the follow-up in intervals of 20 days:

```
> sL <- splitMulti( Lung, tfe=seq(0,1200,40) )
> summary( Lung.s )
Transitions:
     To
From    Alive Dead  Records:  Events: Risk time:  Persons:
  Alive 25941  165     26106      165    69631.6       228
> summary( sL )
Transitions:
     To
From    Alive Dead  Records:  Events: Risk time:  Persons:
  Alive  1696  165      1861      165    69631.6       228
```

so we have much fewer records but the same number of events and person-time. Person 32 now only have 2 records:

```
> sL[lex.id==32,1:10]
   lex.id tfe lex.dur lex.Cst lex.Xst inst  time status age sex
1:     32   0   23.89   Alive    Dead    1 23.89      2  73   M
```

We can then compare with the estimates from the parametric models `mLs.pois.sp`, if we instead use the equidistantly cut dataset:

```
> mLs.pois.se <- update( mLs.pois.sp, data=sL )
> round( cbind( ci.exp(mLs.pois.sp),
+               ci.exp(mLs.pois.se),
+               ci.exp(mLs.pois.sp)/
+               ci.exp(mLs.pois.se) ), 3 )
                       exp(Est.)  2.5%  97.5% exp(Est.)  2.5%  97.5% exp(Est.)  2.5%
(Intercept)                0.001 0.000  0.002     0.001 0.000  0.002     0.975 0.816
Ns(tfe, knots = t.kn)1     2.744 1.027  7.329     2.655 1.171  6.018     1.033 0.877
Ns(tfe, knots = t.kn)2     2.775 0.892  8.634     2.573 0.955  6.933     1.079 0.934
Ns(tfe, knots = t.kn)3     3.657 0.516 25.947     4.099 1.038 16.179     0.892 0.496
Ns(tfe, knots = t.kn)4     3.251 0.707 14.958     3.299 0.757 14.368     0.985 0.933
age                        1.016 0.998  1.035     1.016 0.998  1.035     1.000 1.000
sexF                       0.599 0.432  0.832     0.601 0.433  0.834     0.998 0.998
                       97.5%
(Intercept)            1.165
Ns(tfe, knots = t.kn)1 1.218
Ns(tfe, knots = t.kn)2 1.245
Ns(tfe, knots = t.kn)3 1.604
Ns(tfe, knots = t.kn)4 1.041
age                    1.000
sexF                   0.998
> mLs.pois.pe <- update( mLs.pois.ps, data=sL )
> round( cbind( ci.exp(mLs.pois.ps),
+               ci.exp(mLs.pois.pe),
+               ci.exp(mLs.pois.ps)/
+               ci.exp(mLs.pois.pe) ), 3 )
            exp(Est.)  2.5% 97.5% exp(Est.)  2.5% 97.5% exp(Est.)  2.5% 97.5%
(Intercept)     0.001 0.000 0.003     0.001 0.000 0.003     0.967 0.964 0.969
age             1.016 0.998 1.035     1.016 0.998 1.035     1.000 1.000 1.000
sexF            0.602 0.434 0.836     0.603 0.435 0.838     0.998 0.998 0.998
s(tfe).1        1.155 0.900 1.482     1.356 0.654 2.813     0.852 1.377 0.527
s(tfe).2        1.247 0.615 2.531     1.430 0.538 3.798     0.872 1.142 0.666
s(tfe).3        0.897 0.676 1.191     0.811 0.510 1.289     1.107 1.326 0.924
```

```
s(tfe).4        0.886 0.579 1.354     1.175 0.695 1.984     0.754 0.833 0.683
s(tfe).5        0.887 0.647 1.215     0.839 0.559 1.259     1.057 1.158 0.965
s(tfe).6        0.886 0.609 1.290     0.841 0.546 1.296     1.053 1.115 0.995
s(tfe).7        1.123 0.822 1.534     0.843 0.574 1.237     1.332 1.432 1.240
s(tfe).8        1.786 0.352 9.056     1.827 0.399 8.376     0.977 0.884 1.081
s(tfe).9        1.192 0.878 1.618     1.314 0.841 2.052     0.907 1.043 0.789
```

We see only minor differences in the estimated values of the regression parameters (age and sex), while it appears that the spline parameters are somewhat different. This does however not translate to any relevant differences in the estimated curves:

```
> par( mfrow=c(1,2), mar=c(3,3,1,1), mgp=c(3,1,0)/1.6, bty="n", las=1, lend="butt" )
> plot( NA, xlim=c(0,900), xaxs="i", ylim=c(1/5,10), log="y",
+           xlab="Days since diagnosis",
+           ylab="Mortality rate per 1 year" )
> matshade( nd$tfe, cbind( ci.pred(mLs.pois.sp,nd),
+                          ci.pred(mLs.pois.se,nd) )*365.25,
+           lwd=2, col=c('blue','black'), log="y", alpha=0.07 )
> plot( NA, xlim=c(0,900), xaxs="i", ylim=c(1/5,10), log="y",
+           xlab="Days since diagnosis",
+           ylab="Mortality rate per 1 year" )
> matshade( nd$tfe, cbind( ci.pred(mLs.pois.ps,nd),
+                          ci.pred(mLs.pois.pe,nd) )*365.25,
+           lwd=2, col=c('blue','black'), log="y", alpha=0.07 )
```

Thus from figure 2.3 it appears that the splitting of the follow-up time in 20-day intervals is sufficient to render the estimation of the baseline hazard reliable.

2.2 Stratified models

A stratified Cox-model is a model where the underlying hazard is allowed to differ in shape between strata, i.e. between levels of a categorical variable. "non-proportional hazards" is the common phrase used for this, but it is merely an interaction between the time scale and a categorical variable.

For illustration we use the lung cancer example again:

```
> summary( sL )
Transitions:
     To
From    Alive Dead  Records:  Events: Risk time:  Persons:
  Alive  3441  165      3606      165    69631.6       228
```

For the modeling of the baseline rate (timescale `tfe`) we define the knots and fit a natural spline, one with main effect of sex, the other with an interaction. Note that there is no requirement that the time-part of the interaction is parametrized in the same way as the main effect. The model `m3` below uses a simpler time-effect in the interaction:

```
> kn <- c(0,50,150,450)
> m1 <- glm( cbind(lex.Xst=="Dead",lex.dur) ~ Ns(tfe,knots=kn) + sex + age,
+            family=poisreg, data=sL )
> m2 <- glm( cbind(lex.Xst=="Dead",lex.dur) ~ Ns(tfe,knots=kn) * sex + age,
+            family=poisreg, data=sL )
> m3 <- update( m1, . ~ . + Ns(tfe,knots=kn[1:3]):sex )
> m4 <- update( m1, . ~ . +  I(tfe):sex )
```

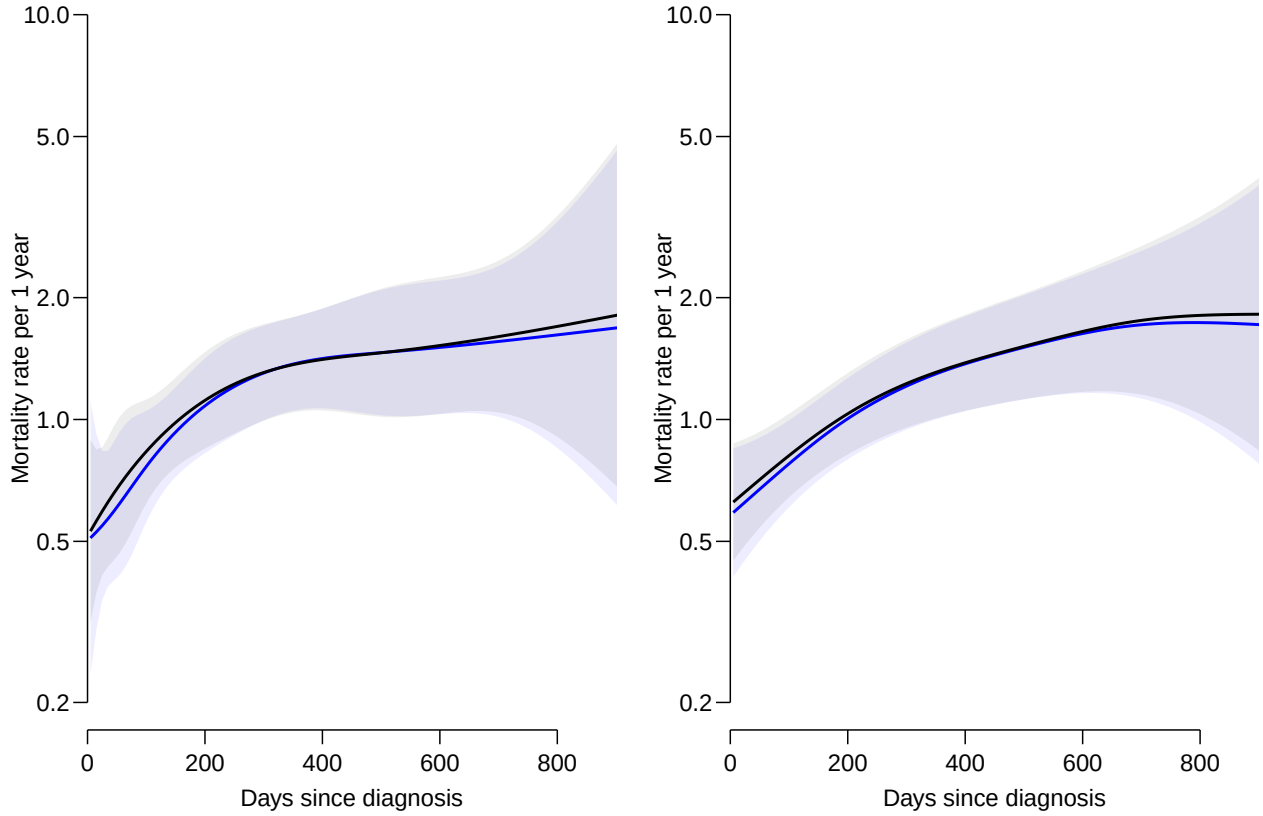

Figure 2.3: *Comparing the same model (left panel:* `glm` *model with natural spline, right panel:* `gam` *model) fitted to data split at all 228 recorded event and censoring times (26106 records) (blue), and fitted to a data set only cut every 40 days (1861 records) (black).* `./lung-spcmp`

Note that we are explicitly using a subset of the knots to define the lower-order interaction; if we used fewer but *different* knots we would get a true extension of the main effects as well.

```
> anova( m1, m4, m3, m1, m2, test="Chisq" )
Analysis of Deviance Table

Model 1: cbind(lex.Xst == "Dead", lex.dur) ~ Ns(tfe, knots = kn) + sex +
    age
Model 2: cbind(lex.Xst == "Dead", lex.dur) ~ Ns(tfe, knots = kn) + sex +
    age + sex:I(tfe)
Model 3: cbind(lex.Xst == "Dead", lex.dur) ~ Ns(tfe, knots = kn) + sex +
    age + sex:Ns(tfe, knots = kn[1:3])
Model 4: cbind(lex.Xst == "Dead", lex.dur) ~ Ns(tfe, knots = kn) + sex +
    age
Model 5: cbind(lex.Xst == "Dead", lex.dur) ~ Ns(tfe, knots = kn) * sex +
    age
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      3600     1352.4
2      3599     1349.1  1   3.3389  0.06766
3      3598     1348.2  1   0.8696  0.35107
4      3600     1352.4 -2  -4.2085  0.12194
5      3597     1347.7  3   4.7175  0.19369
```

There is no significant interaction here, but the 1 df. linear interaction is close. We also see

that the difference in deviance to the 2 and 3 df. interactions are not very big.

Thus the non-significance of the interaction with 3 df. is a reflection that the interaction may be included with too many degrees of freedom, so be careful with richly parametrized interactions, they may be swamped with too many degrees of freedom. If we are looking for an interaction in the first place we will of course want to inspect the *shape* of the interaction. That is the two fitted baseline rates, as well as their ratio, under the two different types of interaction models.

We are using `ci.pred` to extract the estimated rates for men and women respectively using the prediction data frames `nm` and `nf`. If we supply the two prediction data frames in a list to `ci.exp`, we will get the ratio of the predictions from the first to those from the second:

```
> par( mfrow=c(1,2), mar=c(3,3,1,3), mgp=c(3,1,0)/1.6, las=1 )
> nm <- data.frame( tfe=seq(0,1000,10), age=65, sex="M" )
> nf <- data.frame( tfe=seq(0,1000,10), age=65, sex="F" )
> plot( NA, xlim=c(0,900), xaxs="i", ylim=c(1/100,5), log="y",
+           xlab="Days since diagnosis",
+           ylab="Mortality rate per 1 year" )
> matshade( nm$tfe, cbind( ci.pred(m2,nm)*365.25,
+                          ci.pred(m2,nf)*365.25,
+                          ci.exp (m2,list(nm,nf))/20 ),
+           lwd=2, col=c('blue','red','black') )
> abline(h=1/20,lty=3)
> axis( side=4, at=c(2,5,10,15,20)/200, labels=c(2,5,10,15,20)/10 )
> axis( side=4, at=c(2:9)/200, labels=NA, tcl=-0.3 )
> plot( NA, xlim=c(0,900), xaxs="i", ylim=c(1/100,5), log="y",
+           xlab="Days since diagnosis",
+           ylab="Mortality rate per 1 year" )
> matshade( nm$tfe, cbind( ci.pred(m4,nm)*365.25,
+                          ci.pred(m4,nf)*365.25,
+                          ci.exp (m4,list(nm,nf))/20 ),
+           lwd=2, col=c('blue','red','black') )
> abline(h=1/20,lty=3)
> axis( side=4, at=c(2,5,10,15,20)/200, labels=c(2,5,10,15,20)/10 )
> axis( side=4, at=c(2:9)/200, labels=NA, tcl=-0.3 )
```

From figure 2.4, we see a clear tendency that the mortality among men is higher during the first year or so after diagnosis.

For illustration we repeat the same exercise with the `gam` machinery. The interaction specification `s(tfe,by=sex)` does not contain the main effect of sex, so this must be maintained in the interaction model. As above we also include a 1 df. interaction with time:

```
> p1 <- gam( cbind(lex.Xst=="Dead",lex.dur) ~ s(tfe)        + sex + age,
+            family=poisreg, data=sL )
> p2 <- gam( cbind(lex.Xst=="Dead",lex.dur) ~ s(tfe,by=sex) + sex + age,
+            family=poisreg, data=sL )
> p4 <- update( p1, . ~ . +  I(tfe):sex )
> anova( p4, p1, p2, test="Chisq" )
Analysis of Deviance Table

Model 1: cbind(lex.Xst == "Dead", lex.dur) ~ s(tfe) + sex + age + sex:I(tfe)
Model 2: cbind(lex.Xst == "Dead", lex.dur) ~ s(tfe) + sex + age
Model 3: cbind(lex.Xst == "Dead", lex.dur) ~ s(tfe, by = sex) + sex +
    age
  Resid. Df Resid. Dev       Df Deviance Pr(>Chi)
1    3599.3     1350.0
```

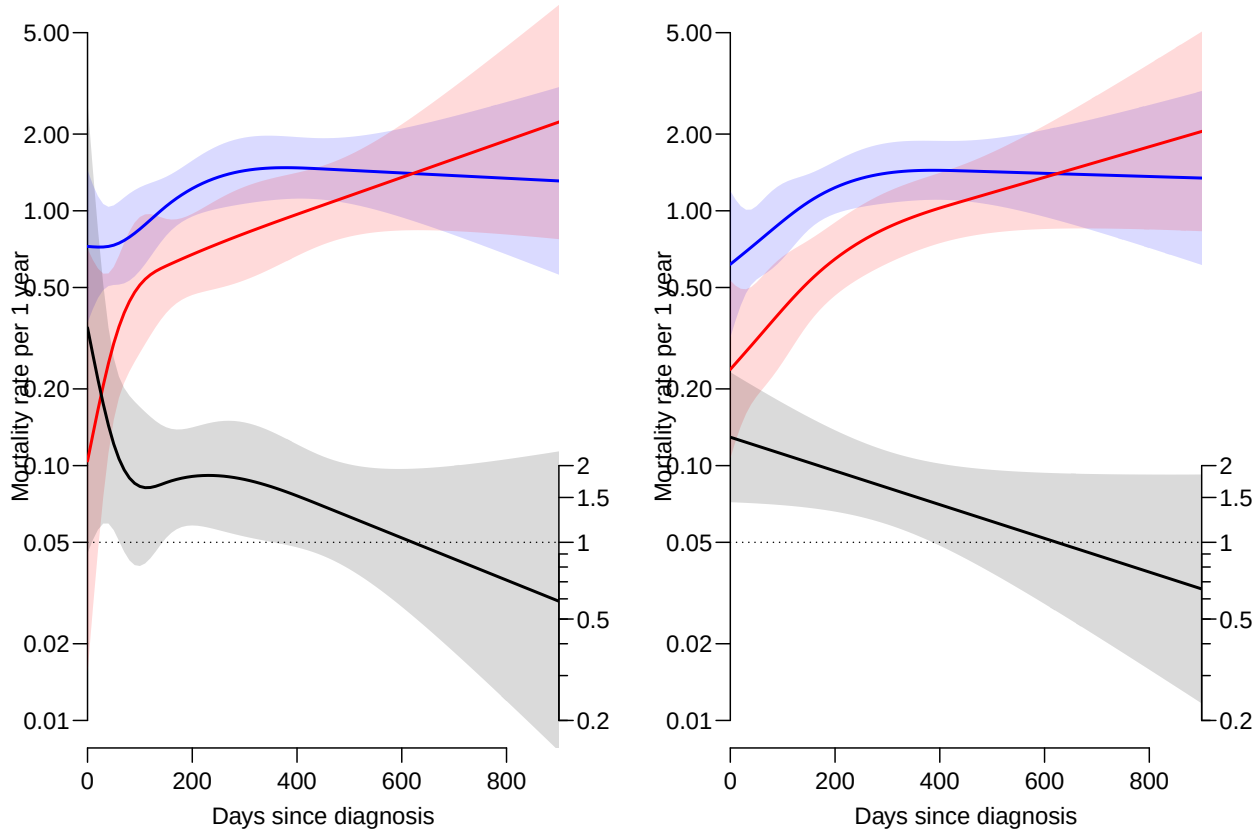

Figure 2.4: *Baseline rates for 65 year old men (blue) resp. women (red), and the rate-ratio between these (black). The leftmost panel uses the same set of knots for the main effect and the interaction, the rightmost a more parsimonious interaction specification.*
From a fanatic 5% significance point of view the gray curves are not *different from a horizontal line, but the p-value for this hypothesis is some 6% in the right one.* `./strat-prcmp`

```
2    3600.4     1353.2 -1.04720  -3.2140  0.07792
3    3599.9     1351.9  0.47267   1.3043  0.11052

> par( mfrow=c(1,2), mar=c(3,3,1,3), mgp=c(3,1,0)/1.6, las=1 )
> plot( NA, xlim=c(0,900), xaxs="i", ylim=c(1/100,5), log="y",
+           xlab="Days since diagnosis",
+           ylab="Mortality rate per 1 year" )
> matshade( nm$tfe, cbind( ci.pred(p2,nm)*365.25,
+                          ci.pred(p2,nf)*365.25,
+                          ci.exp (p2,ctr.mat=list(nm,nf))/20 ),
+           lwd=2, col=c('blue','red','black') )
> abline(h=1/20,lty=3)
> axis( side=4, at=c(2,5,10,15,20)/200, labels=c(2,5,10,15,20)/10 )
> axis( side=4, at=c(2:9)/200, labels=NA, tcl=-0.3 )
> plot( NA, xlim=c(0,900), xaxs="i", ylim=c(1/100,5), log="y",
+           Xlab="Days since diagnosis",
+           ylab="Mortality rate per 1 year" )
> matshade( nm$tfe, cbind( ci.pred(p4,nm)*365.25,
+                          ci.pred(p4,nf)*365.25,
+                          ci.exp (p4,ctr.mat=list(nm,nf))/20 ),
+           lwd=2, col=c('blue','red','black') )
> abline(h=1/20,lty=3)
```

```
> axis( side=4, at=c(2,5,10,15,20)/200, labels=c(2,5,10,15,20)/10 )
> axis( side=4, at=c(2:9)/200, labels=NA, tcl=-0.3 )
```

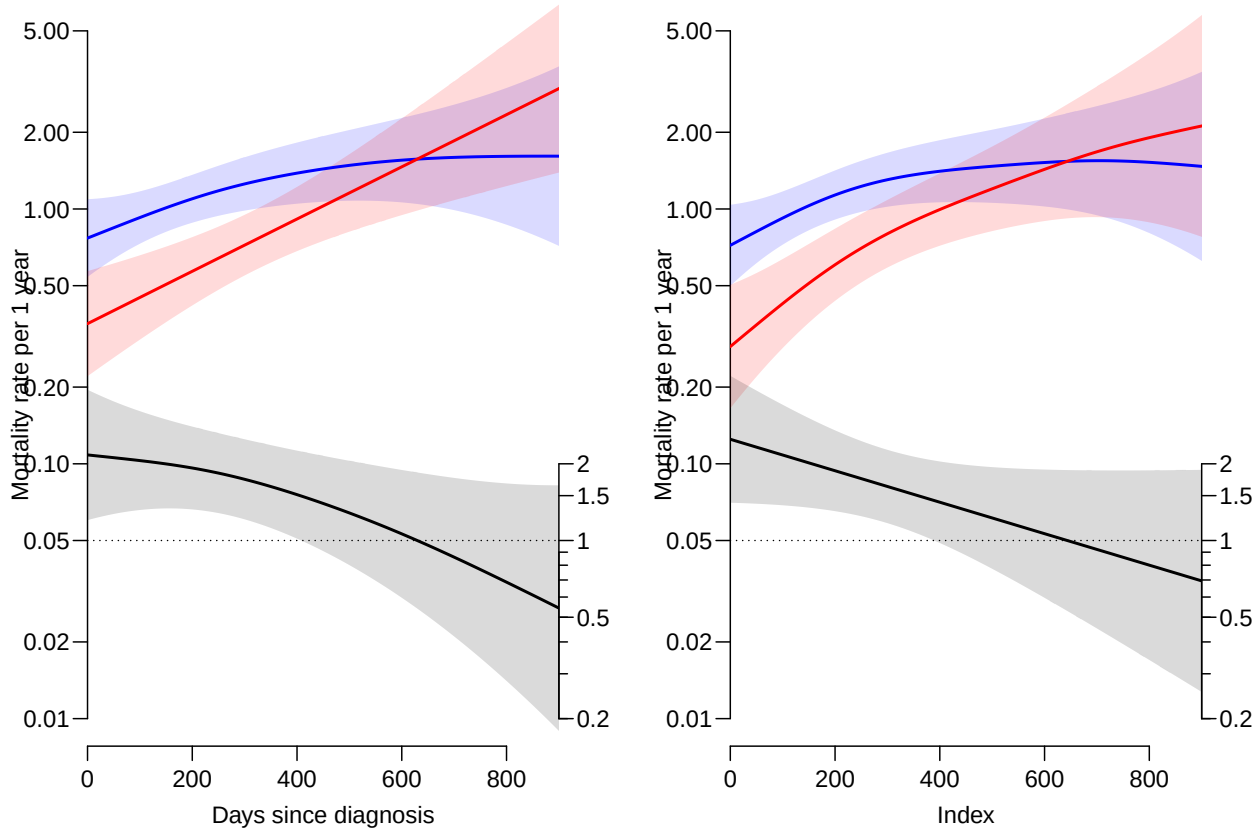

Figure 2.5: *Estimated rates and rate-ratio by the* `gam` *fitting machinery. The rightmost plot is with an (almost significant) linear sex by time interaction.* `./strat-pnsh`

From figure 2.5 we see the same overall tendency, but substantially more smoothed. But with this type of analysis we have a more firm evidence that male mortality actually is higher in the first year or so, despite the p-value above 10%.

The formal test of whether the black lines in figures 2.4 and 2.5 are horizontal may be too unspecific in this context, inspection of the shape of the interaction may reveal features of interest that are swamped in degrees of freedom in the test.

2.3 Time-varying coefficients

When it is suspected that effects of a quantitative variable is not constant along some time scale, it has been proposed to allow the coefficient of a variable to vary by time:

$$\lambda_i(t) = \lambda_0(t) \exp(\beta(t) x_i + \cdots)$$

Recalling that a time scale is a covariate, this is merely an interaction between the covariate and time, which is restricted by letting the x-effect be linear for any fixed value of time.

The substantial reason for this particular choice of form of interaction is slightly opaque. Given that one variable (time) in a Cox model is meticulously modelled it seems strange to

insist on a conditionally linear effect of x. It would seem to be more intuitive to explore more parsimonious parametrizations of interactions that were more directly addressing biologically meaningful deviations from the log-linear additivity of the effects.

There is however a tradition in epidemiological analysis of trends in rates to summarize calendar time trends separately in each age-group by computing the average trend within each age class (age-specific secular trend). The continuous time version of this is precisely a varying coefficients model where the effect of calendar time is taken as linear at each age. This would correspond to adding an interaction between x and some grouping of time. Again this approach can be taken *ad absurdum* with increasingly fine groupings of time until we end up with the Cox-model formulation of the problem.

But when the main effect of time is modelled by a spline or any other smooth function, implemented as columns of the model matrix in the Poisson regression model, we can estimate time-varying coefficients by adding the same columns multiplied by x to the model matrix. The coefficients of these will then be the ones that determine the (time-varying) effect of the covariate x.

A simple illustration of this using the lung cancer example again: We use the same dataset as before, but now we have the interaction with the quantitative variable `age`. Note that we include the intercept in the spline basis used for the interaction, in order to accommodate the main effect of age.

```
> kn <- c(0,50,150,450)
> m1 <- glm( cbind(lex.Xst=="Dead",lex.dur) ~ Ns(tfe,knots=kn) + age + sex,
+            family=poisreg, data=sL )
> mv <- update( m1, . ~ . + Ns(tfe,knots=kn,i=T):age )
> anova( m1, mv, test="Chisq" )
Analysis of Deviance Table

Model 1: cbind(lex.Xst == "Dead", lex.dur) ~ Ns(tfe, knots = kn) + age +
    sex
Model 2: cbind(lex.Xst == "Dead", lex.dur) ~ Ns(tfe, knots = kn) + age +
    sex + age:Ns(tfe, knots = kn, i = T)
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1      3600     1352.4
2      3597     1334.2  3   18.184 0.0004031
```

Here we see that there actually is a massive interaction — the age-effects does vary considerably by time. But the test give no clue as to how.

The parameters of interest are those from the second `Ns` term in the model, but of course taken out as a curve. We can extract the age-effect as a difference between two predictions, namely the rate-ratio between two persons, say 5 years apart in age:

```
> nx <- data.frame( tfe=seq(0,1000,10), age=55, sex="M" )
> nr <- data.frame( tfe=seq(0,1000,10), age=50, sex="M" )
> matshade( nx$tfe, psl<-ci.exp( mv, list(nx,nr) ),
+           plot=TRUE, lwd=3, log="y",
+           xlab="Time since diagnosis (days)",
+           ylab="RR per 5 years of age at diagnosis" )
> abline( h=1, lty=3 )
> wh <- which(as.logical(abs(diff(psl[,1]>1))))
> psl[sort(c(wh,wh+1)),]
   exp(Est.)      2.5%    97.5%
8  1.0122808 0.8422378 1.216655
9  0.9593764 0.7998789 1.150678
```

```
19 0.9950550 0.8883973 1.114518
20 1.0084351 0.8986241 1.131665
> abline( h=1, v=kk<-nx$tfe[wh]+5 )
```

Since the effect of age is linear given any value of the time scale (`tfe`), the extracted effect would have been the same for any two ages 5 years apart.

From the figure **??** we see that the age at diagnosis matters a lot for the mortality the first few months after diagnosis, but after about 3 months there is no effect.

However, this is not the usual way to show an interaction; an interaction between two quantitative variables is best shown as a curve with the effect of one of the variables conditional on a specific value of the other, or conditional on a sequence of values of the other. Thus in this case we would show the mortality rates as a function of `tfe` — time since diagnosis for different values of age (at diagnosis). So we make predictions of mortality as a function of time since diagnosis for ages at diagnosis 40,45,...,75.

```
> par( mfrow=c(1,2), mar=c(3,3,0.1,0.1), mgp=c(3,1,0)/1.6, bty="n", las=1 )
> # The time-varying coefficient
> matshade( nx$tfe, psl,
+           plot=TRUE, lwd=3, log="y",
+           xlab="Time since diagnosis (days)",
+           ylab="RR per 5 years of age at diagnosis" )
> abline( h=1, lty=3 )
> pra <- NULL
> for( aa in seq(40,70,5) )
+      pra <- cbind( pra, ci.pred( mv, transform( nx, age=aa ) ) )
> matplot( nx$tfe, pra[,0:6*3+1]*1000, col=gray((7:1+4)/13),
+          type="l", lwd=2, lty=1, log="y", ylim=c(0.1,10), alphs=0.02,
+          xlab="Time since diagnosis (days)",
+          ylab="Mortality per 1000 PY" )
> abline( v=kk, lty=3 )
```

Figure 2.6 is an illustration of how the model imposes quite unrealistic assumptions on the shape of the interaction. A more realistic interaction would spend more more d.f.. on the age-dimension:

```
> mw <- update( m1, . ~ Ns(tfe,knots=kn)*Ns(age,knots=5:7*10) + sex )
> anova( mw, m1, mv, test="Chisq" )
Analysis of Deviance Table

Model 1: cbind(lex.Xst == "Dead", lex.dur) ~ Ns(tfe, knots = kn) + Ns(age,
    knots = 5:7 * 10) + sex + Ns(tfe, knots = kn):Ns(age, knots = 5:7 *
    10)
Model 2: cbind(lex.Xst == "Dead", lex.dur) ~ Ns(tfe, knots = kn) + age +
    sex
Model 3: cbind(lex.Xst == "Dead", lex.dur) ~ Ns(tfe, knots = kn) + age +
    sex + age:Ns(tfe, knots = kn, i = T)
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1      3593     1328.8
2      3600     1352.4 -7  -23.656 0.0013095
3      3597     1334.2  3   18.184 0.0004031


> pra <- NULL
> for( aa in seq(40,70,5) )
+      pra <- cbind( pra, ci.pred( mw, transform( nx, age=aa ) ) )
```

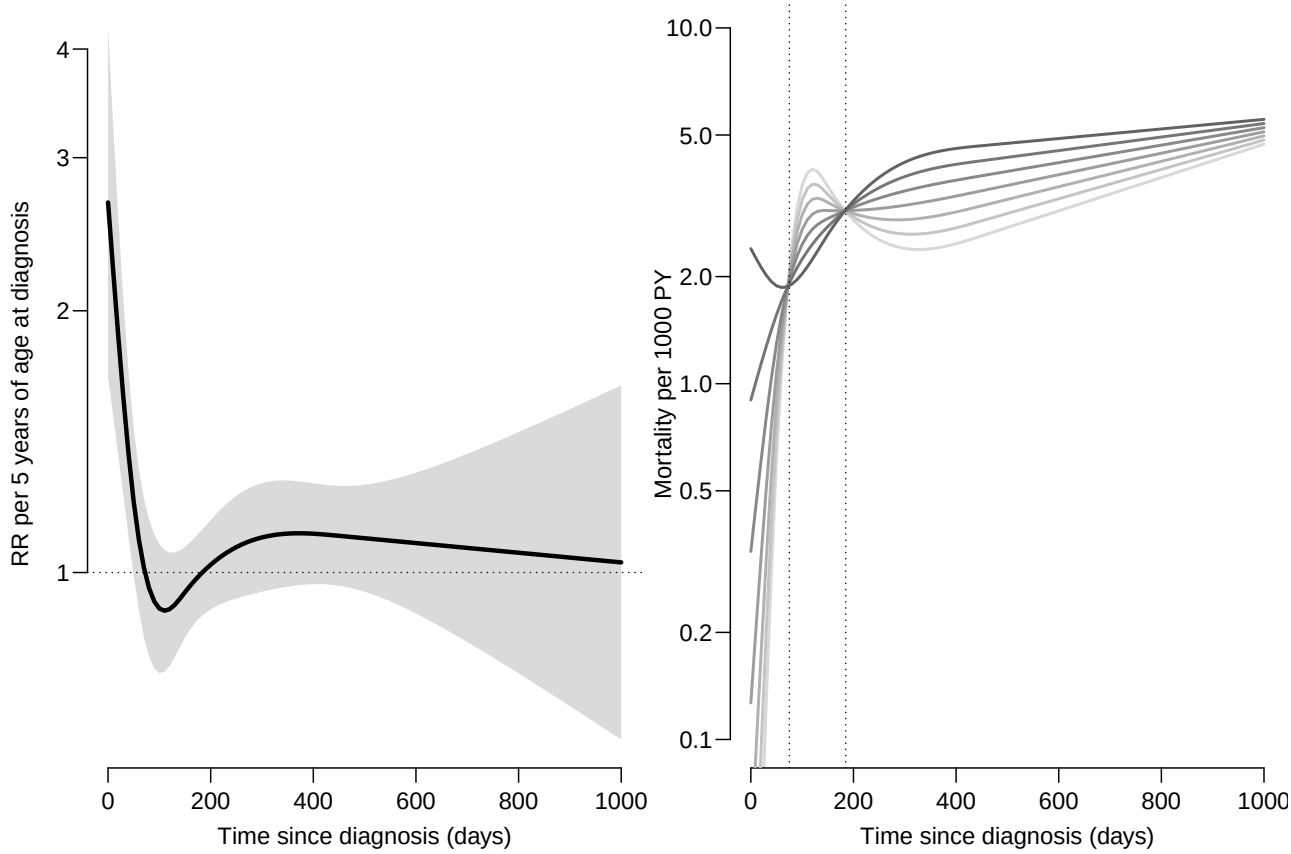

Figure 2.6: *RR of death for the lung cancer patients, per 5 years of age at diagnosis. Results from a "varying-coefficients" model — interaction between two continuous variables, where the effect of age is constrained to be linear at any time since diagnosis. The right panel shows the estimated mortality rates from the varying coefficients model for ages at diagnosis 40,45,...,70 (light to dark).* `./time-var-Aint`

```
> par( mfrow=c(1,2), mar=c(3,3,0.1,0.1), mgp=c(3,1,0)/1.6, bty="n", las=1 )
> for( aa in c(0,0.15) )
+ matshade( nx$tfe, pra*1000, col=gray((7:1+4)/15), plot=TRUE,
+           lwd=3, lty=1, log="y", ylim=c(0.1,10), alpha=aa,
+           xlab="Time since diagnosis (days)",
+           ylab="Mortality per 1000 PY" )
```

It is clear from this richer model that the age-effect is largest in the beginning, and that beyond 300 days, there is very little effect of age at diagnosis. This is not very different from the varying coefficients model, but the funny restrictions that mortality is linearly related to age at any time is relieved.

We can make an illustrative film of this:

```
> for( aa in 40:75 )
+    {
+       pra <- NULL
+    for( al in c(5,10,25)/100 )
+       pra <- cbind( pra, ci.pred( mw, transform( nx, age=aa ), alpha=al ) )
+ par( mar=c(3,3,0.1,0.1), mgp=c(3,1,0)/1.6, bty="n", las=1 )
+ matshade( nx$tfe, pra*1000, plot=TRUE,
+           lwd=3, col='black', log="y", ylim=c(0.1,10),
```

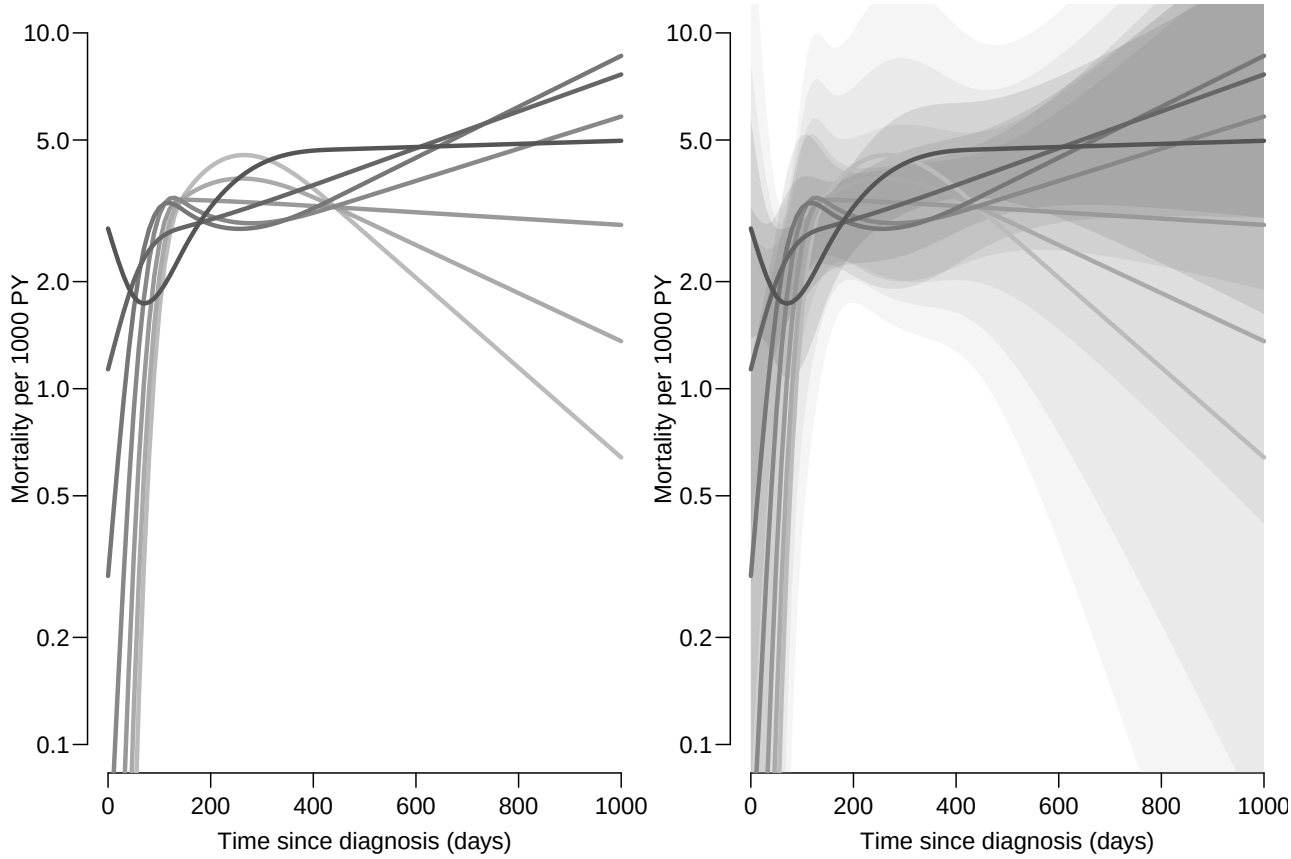

Figure 2.7: *The estimated mortality rates from the traditional interaction model for ages at diagnosis 40,45,...,70 (light to dark). The right panel is with shaded 95% confidence intervals.*
`./time-var-Xint`

```
+           xlab="Time since diagnosis (days)",
+           ylab="Mortality per 1000 PY" )
+ text( 100, 0.1, paste(aa,"years at dx: 95, 90 and 75% CI."), adj=0 )
+    }
```

The resulting film in the form of a multi-page `.pdf`-file is available at
http://bendixcarstensen.com/time-var-film.pdf.

2.4 Simplifying code

2.4.1 Parametrizations

Note that when we use prediction data frames to tease out the effects, the particular parametrization does not matter, so we could have used a simple expression for the r.h.s. of the model formula:

```
> ~ Ns(tfe,knots=kn) * age + sex
```

and we would have obtained the same results.

2.4.2 Using the `Lexis` structure

Since we are using a `Lexis` object as data base for the analysis, we already have specified the time-structure in data, so we can shorten the code even further using the `glm.Lexis` function for fitting multistate models; basically it is just a wrapper using the `poisreg` family. It will by default analyze all transitions *to* any absorbing state, which in this case is "Dead", and only one state precedes "Dead", namely "Alive". So the analysis could be done by:

```
> mL <- glm.Lexis( sL, formula= ~ Ns(tfe,knots=kn)*age + sex )
stats::glm Poisson analysis of Lexis object sL with log link:
Rates for the transition: Alive->Dead
> c( deviance(mv), deviance(mL) )
[1] 1334.232 1334.232
```

For this, you want to look up the help page:

```
> ?glm.Lexis
```

A similar function `coxph.Lexis` exists for Cox-modeling based on a `Lexis` object.

2.5 Testing the proportionality assumption

In the sections on stratified models and time-varying coefficient we addressed the question of interaction between the time scale (`tfe`) and a categorical variable (`sex`), respectively a quantitative variable (`age`). In both cases we found interactions, and graphed the shape of them.

Interactions with the time scale is usually termed "non-proportionality" and the test for interactions is called "checking the proportionality assumption". I most practical applications people just use the `cox.zph` function to get a series of tests for proportionality. Here we refit the model using the `coxph.Lexis` facility:

```
> m0 <- coxph.Lexis( sL, formula = tfe ~ age + sex )
model survival::coxph analysis of Lexis object sL:
Rates for the transition Alive->Dead
> summary( m0 )
Call:
coxph(formula = as.formula(paste("Sobj", as.character(formula[3]),
    sep = "~")), data = Lx)

  n= 3606, number of events= 165

          coef exp(coef)  se(coef)      z Pr(>|z|)
age   0.017053  1.017199  0.009218  1.850  0.06432
sexF -0.520328  0.594326  0.167512 -3.106  0.00189

     exp(coef) exp(-coef) lower .95 upper .95
age     1.0172     0.9831     0.999    1.0357
sexF    0.5943     1.6826     0.428    0.8253

Concordance= 0.603  (se = 0.025 )
Likelihood ratio test= 14.42  on 2 df,   p=7e-04
Wald test            = 13.75  on 2 df,   p=0.001
Score (logrank) test = 14.01  on 2 df,   p=9e-04
```

```
> cox.zph( m0 )
```

```
           rho chisq     p
age    -0.0249 0.105 0.746
sexF    0.1247 2.487 0.115
GLOBAL      NA 2.657 0.265
```

So in this case we can safely write in our paper that "we checked the proportionality assumption and found that it was not violated". Which is what most people do, instead of inspecting the interactions they are testing, and risking to find out that they were actually there...

2.6 The short version

This is a condensed piece of code showing how to derive baseline rate and survival function using a parametric spline approach. Few explanations and no bells and whistles on the plots.

```
> library(Epi)
> library(popEpi)
> library(mgcv)
> library(survival)
```

2.6.1 Data preparation

Get the data, and convert sex to a factor (factors make life easier and safer):

```
> data(lung)
> lung$sex <- factor(lung$sex,labels=c("M","F"))
```

Set up a Lexis object (outcome as a factor), and split time in small intervals:

```
> Lx <- Lexis( exit=list(tfe=time),
+              exit.status=factor(status,labels=c("Alive","Dead")),
+              data=lung )
NOTE: entry.status has been set to "Alive" for all.
NOTE: entry is assumed to be 0 on the tfe timescale.
> sL <- splitMulti( Lx, tfe=seq(0,1200,10) )
```

2.6.2 Baseline hazard and survival

Fit smooth parametric model for baseline:

```
> m0 <- gam.Lexis( sL, formula= ~ s(tfe) + sex + age )
mgcv::gam Poisson analysis of Lexis object sL with log link:
Rates for the transition: Alive->Dead
```

Prediction data frame for rates and survival — at what times do you want the rates and the survival shown:

```
> nd <- data.frame( tfe=seq(0,900,20)+10, sex="M", age=65 )
> rate <- ci.pred( m0, nd )*365.25 # per year, not per day
> surv <- ci.surv( m0, nd, int=20 )
```

Plot the rates

```
> matshade( nd$tfe, rate, log="y", plot=TRUE )
```

Plot the survival function — we supplied the midpoints, now use the (left) endpoints as x-variable

```
> matshade( nd$tfe-10, surv, ylim=c(0,1), plot=TRUE )
```

2.6.3 Stratified model

Stratified parametric model for baseline:

```
> ms <- gam.Lexis( sL, formula= ~ s(tfe,by=sex) + sex + age )
mgcv::gam Poisson analysis of Lexis object sL with log link:
Rates for the transition: Alive->Dead
```

Prediction data frame for men and women, separate baselines:

```
> nM <- data.frame( tfe=seq(0,900,20)+10, sex="M", age=65 )
> nF <- data.frame( tfe=seq(0,900,20)+10, sex="F", age=65 )
> rateM <- ci.pred( ms, nM )*365.25
> rateF <- ci.pred( ms, nF )*365.25
```

Plot the baseline rates for men and women

```
> matshade( nd$tfe, cbind(rateM,rateF), col=c(4,2), log="y", plot=TRUE )
```

Compute and plot the M/F RR

```
> MFrr <- ci.exp( ms, list(nM,nF) )
> matshade( nd$tfe, MFrr, log="y", plot=TRUE ); abline(h=1)
```

If you want the two survival functions, use `ci.surv` with the prediction frames:

```
> matshade( nM$tfe-10, cbind( ci.surv(ms,nM,int=20),
+                             ci.surv(ms,nF,int=20) ),
+           col=c("blue","red"), ylim=c(0,1), plot=TRUE )
```

Chapter 3

So who *does* need the Cox-model?

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

The Cox-model is computationally vastly more efficient, and it is easier to produce a survival curve by standard software, which is relevant in most clinical (survival) studies. One drawback is the overly detailed modeling of survival curves that may lead to over-interpretation of little humps and notches on the curve, another is that you only have access to the baseline hazard via the transformation to the survival scale.

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 modeled and in particular displayed. Moreover it 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 you are not really interested in interactions but feel obliged to "test the proportionality assumption".
- Studies analyzed on computing equipment pre-1985.

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

- it clarifies the distinction between (risk) time as response variable and time(scales) as covariates — reflected in the `poisreg` family in the `Epi` package.
- it enables smoothing of the effect of timescales using standard regression tools. In particular it allows more credible estimates of survival functions in the simple case with only time since entry as timescale.
- it enables modeling effects of multiple timescales
- it enables sensible modeling and display of interactions between timescales and other variables (and between timescales).

Moreover, as the necessary computing power and software is available, the computational problems encountered previously are now largely non-existent. Extraction of the relevant functions has been facilitated by the introduction of the possibility of supplying pairs of prediction data frames to extract rate-ratios (see the help pages for `ci.lin` and `ci.cum` in the `Epi` package).

However, the user-interface to the Poisson modeling is slightly more complex than that offered by standard packages for the Cox-model. This is partly because the Poisson approach requires an explicit specification of models for the timescales, but the upside is that you can actually quite easily show the shape of baseline hazards and potential interactions with the time scales.

References

[1] Bendix Carstensen and Martyn Plummer. Using Lexis objects for multi-state models in R. *Journal of Statistical Software*, 38(6):1–18, 1 2011.

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

[3] T R Holford. Life table with concomitant information. *Biometrics*, 32:587–597, 1976.

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

[5] Martyn Plummer and Bendix Carstensen. Lexis: An R class for epidemiological studies with long-term follow-up. *Journal of Statistical Software*, 38(5):1–12, 1 2011.

[6] K. Rostgaard. Methods for stratification of person-time and events - a prerequisite for Poisson regression and SIR estimation. *Epidemiol Perspect Innov*, 5:7, Nov 2008.