

Grund, Birgit; Yang, Lijian

**Working Paper**

**Hazard regression**

SFB 373 Discussion Paper, No. 2000,56

**Provided in Cooperation with:**

Collaborative Research Center 373: Quantification and Simulation of Economic Processes,  
Humboldt University Berlin

*Suggested Citation:* Grund, Birgit; Yang, Lijian (2000) : Hazard regression, SFB 373 Discussion Paper, No. 2000,56, Humboldt University of Berlin, Interdisciplinary Research Project 373: Quantification and Simulation of Economic Processes, Berlin,  
<https://nbn-resolving.de/urn:nbn:de:kobv:11-10047823>

This Version is available at:

<https://hdl.handle.net/10419/62162>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*

# Hazard Regression

*Birgit Grund and Lijian Yang*

**Hazard regression models** are convenient tools to discover the structure and dependencies in time-to-event data with covariates. In medical research, the influence of certain covariates on the length of patients' survival is often evaluated with hazard regression models, see, for instance, Cox and Oakes (1984). In econometrics, hazard regression is being used, among others, to model insurance industry and employment data; see, for example, Heckman and Singer (1985), Lancaster (1990).

The XploRe quantlib `hazreg` provides a number of quantlets for the analysis of right-censored time-to-event data. These include Kaplan-Meier estimates of the survival function and pointwise confidence intervals for the Kaplan-Meier estimates. For the Cox proportional hazards model, we provide estimates for the regression coefficients and their covariance matrix, significance tests for the regression coefficients, and estimates for the baseline hazard and conditional survival functions. This chapter is a tutorial for the quantlets in the `hazreg` quantlib. We provide the syntax, shortly describe the underlying statistical theory, and illustrate their use with examples. In Section 1, we introduce right-censored time-to-event data and present quantlets that arrange the data into a form suitable for analysis in XploRe. Section 2 is dedicated to Kaplan-Meier estimates and corresponding confidence intervals for the survival function. Section 3 describes semiparametric estimation and hypothesis testing in the Cox proportional hazards model. We apply these methods to a data set on the length of stay in nursing homes.

## 1 Data Structure

```
{data, ties} = hazdat(t, delta{, z})
    sorts the times t in ascending order, cosorts the censoring indi-
    cator delta and the covariates in z, and provides tie information

nar = haznar(data)
    calculates the size of the risk set at each observed time point

atrisk = hazrisk(data, i)
    determines which observations are at risk at time ti
```

The quantlib `hazreg` provides methods for analyzing right-censored time-to-event data. The observed data are triples  $(t_i, \delta_i, z_i)$ ,  $i = 1, \dots, n$ , where  $t_i$  denotes the observed survival time of the  $i$ -th individual,  $z_i = (z_{i1}, \dots, z_{ip})^T$  denotes the  $p$ -dimensional covariate vector associated with the  $i$ -th individual, and  $\delta_i$  is the censoring indicator.

Let  $y_i$  denote the uncensored survival time, and  $c_i$  the random censoring time. The observed survival time of the  $i$ -th individual is then given by  $t_i = \min(y_i, c_i)$ . The censoring indicator takes the value  $\delta_i = 1$  when  $y_i \leq c_i$ ; in this case, the observed time,  $t_i = y_i$ , is called **event time**. Otherwise,  $\delta_i = 0$ , and the observed time is **censored**,  $t_i = c_i$ . We assume that censoring is uninformative; this means, given the covariate values, the conditional distributions of the survival time and of the censoring time are independent.

For many computations, information on the presence and location of ties is required. Obviously, we could locate the ties each time that a method requires this information. However, in a typical session the same dataset will be studied for various purposes. It is much more efficient to gather the tie information once, and link it to the data set. We address this problem by compiling most of the necessary data information into a matrix `data`, which is passed on as an argument to the various data analysis quantlets.

The quantlet `hazdat` sorts the right-censored data  $(t_i, \delta_i, z_i)$ ,  $i = 1, \dots, n$  in ascending order with respect to time  $t$ , cosorts the censoring indicator and covariate values, evaluates ties, and organizes the data and tie information in the matrix `data`.

It has the following syntax:

```
{data,ties} = hazdat(t, delta {,z})
```

Input:

**t**

$n \times 1$  vector of survival times  $t_i$ ,

**delta**

$n \times 1$  vector of censoring indicators  $\delta_i$ ,

**z**

$n \times p$  matrix of covariate values, with rows  $z_i^T$ ; default is an empty matrix.

Output:

**data**

$n \times (p + 4)$  matrix of cosorted time-to-event data, with  
column 1: observed times  $t_i$ , sorted in ascending order,  
column 2: censoring indicator  $\delta_i$ , cosorted,  
column 3: original observation labels  $(1, \dots, n)$ , cosorted,  
column 4: number of tied observations in time  $t_i$ , cosorted,  
columns 5 through  $(p+4)$ : covariate values  $z_i^T = (z_{i1}, \dots, z_{ip})$ , cosorted;

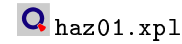
**ties**

scalar, indicator of ties, with **ties=1** when ties in the  $t_i$  are present, and **ties=0** when there are no ties.

**Example 1.** With this example, we illustrate the use of the quantlet `hazdat`. The censoring and the observed times are chosen to better demonstrate the handling of ties (column 4 in `data`, and tie indicator `ties=1`). There are no covariates. Note that at the start of each session, the quantlib `hazreg` has to be loaded manually, with the command `library("hazreg")`.

```
library("hazreg")
y = 2|1|3|2|4|7|1|3|2      ; uncensored event times
c = 3|1|5|6|1|6|2|4|5     ; censoring times
t = min(y~c,2)            ; observed (censored) times
delta = (y<=c)           ; censoring indicator
```

```
{data,ties} = hazdat(t,delta)
data
ties
```



The variables `data` and `ties` take the following values:

```
data =
  1      0      5      3
  1      1      7      3
  1      1      2      3
  2      1      4      3
  2      1      9      3
  2      1      1      3
  3      1      8      2
  3      1      3      2
  6      0      6      1

ties =
  1
```

The first column of `data` provides the observed times in ascending order. Column 3 gives the original order of the sample. The elements of Column 4 count how many observations (censored or uncensored) are tied at the corresponding times. In our data, three observations are tied at time points  $t = 1$  and  $t = 2$ , each.

**Remark 1.1** *Most of our hazard regression quantlets require an input variable `data`, which provides the time-to-event data and tie information in exactly the same format as the `hazdat` output variable `data` (first element in the output list). Therefore, we recommend to run the quantlet `hazdat` at the beginning of each session, or whenever a different set of covariates or a subset of time points is to be considered.*

In order to simplify notation, we assume from now on that the observed times are sorted,  $t_1 \leq t_2 \leq \dots \leq t_n$ .

For many calculations we need to know which observations are in the risk set for any given event time. The risk set at time  $t$  is defined as  $R(t) = \{j: t \leq t_j\}$ . It consists of all observations that did not have an event or were censored prior to

time  $t$ , and thus are still **at risk** for an event. The quantlet `hazrisk` determines the observations at risk at a given observed time point,  $t_i$ . The syntax is given below:

```
atrisk = hazrisk(data,i)
```

Input:

`data`  
 $n \times (p + 4)$  matrix, the sorted data matrix given by the output `data` of `hazdat`;


`i`  
 scalar, the position of  $t_i$  in the ordered list  $t_1 \leq t_2 \leq \dots \leq t_n$ .

Output:

`atrisk`  
 $n \times 1$  vector, with elements 0 or 1 that indicate whether observations are in the risk set at time  $t_i$ .  
`atrisk[j] = 1` when  $t_i \leq t_j$ , and `atrisk[j] = 0`, otherwise.

**Example 2.** We illustrates the use of the quantlet `hazrisk` with the data set of Example 1. Note that the first 6 lines of the XploRe code are identical. In line 6, we call `hazdat` to organize the observations and the tie information into the matrix `data`, which is displayed as output of `hazdat` in Example 1. In line 7, `data` is passed as input argument to the quantlet `hazrisk`.

```
library("hazreg")
y = 2|1|3|2|4|7|1|3|2      ; uncensored event times
c = 3|1|5|6|1|6|2|4|5     ; censoring times
t = min(y~c,2)            ; observed (censored) times
delta = (y<=c)           ; censoring indicator
{data,ties} = hazdat(t,delta) ; organize data
atrisk = hazrisk(data,6)   ; risk set at observation 6
atrisk
```

 haz02.xpl

The variable `atrisk` takes the value `atrisk = (0,0,0,1,1,1,1,1,1)`<sup>T</sup>. In this

example, the times  $t_4 = t_5 = t_6$  are tied. Therefore, the risk set at time  $t_6$  includes all observations with index  $j \geq 4$ .

The quantlet `haznar` returns the size of the risk set at each observed time  $t_i$ ,  $i = 1, \dots, n$ . Its syntax follows below:

```
nar = haznar(data)
```

Input:

`data`

$n \times (p + 4)$  matrix, the sorted data matrix given by the output `data` of `hazdat`.


Output:

`nar`

$n \times 1$  vector, the number (of observations) at risk at each time point.

**Example 3.** The use of the quantlet `haznar` is illustrated with the same data set used in the previous two examples. Again, the first 6 lines of code are identical to Example 1, preparing the data. The input matrix `data` is obtained as part of the output of the `hazdat` call; `data` is displayed in Example 1.

```
library("hazreg")
y = 2|1|3|2|4|7|1|3|2           ; uncensored event times
c = 3|1|5|6|1|6|2|4|5           ; censoring times
t = min(y~c,2)                   ; observed (censored) times
delta = (y<=c)                   ; censoring indicator
{data,ties} = hazdat(t,delta)
nar = haznar(data)               ; calculate the number at risk
nar
```

 haz03.xpl

The output variable `nar` takes the value `nar = (9, 9, 9, 6, 6, 6, 3, 3, 1)T`. The first three observations are tied, and, therefore, have identical risk sets.

## 2 Kaplan-Meier Estimates

```
{cil, kme, ciu} = hazkpm(data{, alpha})
calculates Kaplan-Meier estimates and confidence bounds for the
survival function
```

Let  $t_{(1)} < t_{(2)} < \dots < t_{(m)}$  denote the distinct times in which an event was observed,  $d_i$  the number of events that occurred at time  $t_{(i)}$ , and  $r_i$  the size of the risk set at time  $t_{(i)}$ . The Kaplan-Meier estimate for a survival function, also called **product-limit estimate**, is given by

$$\hat{S}(t) = \begin{cases} 1, & \text{if } t < t_{(1)}, \\ \prod_{t_{(i)} \leq t} \left[1 - \frac{d_i}{r_i}\right], & \text{if } t_{(1)} \leq t. \end{cases} \quad (1)$$

The Kaplan-Meier estimate  $\hat{S}(t)$  is a right-continuous step function with jumps in the event times. Censoring times affect the estimate only by reducing the risk set for next event, and thereby increasing the height of the next jump.

In the presence of censoring, Greenwood (1926) suggested the following estimate for the variance of the Kaplan-Meier estimate:

$$\hat{V}(t) = \hat{S}(t)^2 \sum_{t_{(i)} \leq t} \frac{d_i}{r_i(r_i - d_i)}. \quad (2)$$

The Kaplan-Meier estimate  $\hat{S}(t)$  is asymptotically normally distributed. This leads to the following pointwise confidence intervals for the survival function,  $\hat{S}(t)$ ,

$$\left[ \hat{S}(t) - z_{1-\alpha/2} \hat{V}(t)^{1/2}, \hat{S}(t) + z_{1-\alpha/2} \hat{V}(t)^{1/2} \right], \quad (3)$$

where  $(1 - \alpha)$  is the coverage probability,  $z_p$  denotes the  $p \times 100$ -th percentile of the standard normal distribution, and  $\hat{V}(t)$  is Greenwood's estimate of the variance of  $\hat{S}(t)$ , given in formula (2). Note that Greenwood's estimate tends to slightly underestimate the true variance, so that the true coverage probability of the confidence intervals might be somewhat smaller than stated.

The quantlet `hazkpm` computes the Kaplan-Meier estimates and confidence bounds of the survival function using formulae (1) and (3). It requires that the data are organized in the specific form as provided by `hazdat`. The syntax is given below:



```
{cil,kme,ciu} = hazkpm(data {,alpha})
```

Input:

**data**

$n \times (p + 4)$  matrix, the sorted data matrix given by the output data of `hazdat`;

**alpha**

scalar, the specified error rate of the confidence interval, default option is 0.05 (coverage probability of 0.95).

Output:

**cil**

$n \times 2$  matrix, the first column consists of the sorted  $t_i$ , the second column contains the Greenwood lower confidence bounds at  $t_i$ , defined in (3);

**kme**

$n \times 2$  matrix, the first column consists of the sorted  $t_i$ , the second column contains the Kaplan-Meier estimates at  $t_i$ ;

**ciu**

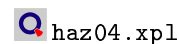
$n \times 2$  matrix, the first column consists of the sorted  $t_i$ , the second column contains the Greenwood upper confidence bounds at  $t_i$ , defined in (3).

By definition, the Kaplan-Meier estimate  $\hat{S}(t)$  is a right-continuous step function. The quantlet `hazkpm` supplies the coordinates  $(t_i, \hat{S}(t_i))$  of the upper left corners of each step, as well as coordinates of pointwise confidence limits for the  $S(t_i)$ ,  $(t_i, \text{cil}(t_i))$  and  $(t_i, \text{ciu}(t_i))$ . Note that the output of `hazkpm` provides one row for each **observed** time  $t_i$ , censored or uncensored. In the case of ties, the rows are repeated.

The quantlet `steps4plot` provides support for plotting step functions. Given the coordinates of the upper left corners and the leftmost starting point, quantlet `steps4plot` adds the coordinates of the lower right corner points in the correct order. Optionally, a right endpoint may be specified. The output is a  $(2n + 2) \times 2$  matrix of point coordinates. The step function may then be drawn into a graph by connecting consecutive output points with line segments.

Syntax of `steps4plot`:

```
{xyline}=steps4plot(xy {,xymin} {,xmax})
```



Input:

`xy`

$n \times 2$  matrix, coordinates  $(x_i, y_i)$  of the jump points of a right-continuous step function which jumps in  $x_i$  to value  $y_i$ . The  $x_i$  (first column) are required to be sorted in ascending order.

`xymin`

$1 \times 2$  matrix, coordinates of the leftmost starting point of the plotted step function. Default is the first row in `xymin`. If `xymin[1,1] > xy[1,1]`, then the leftmost starting point is set to the first row of `xy`.

`xmax`

scalar,  $x$ -coordinate of the rightmost endpoint.

Default: `xmax = xy[n,1] + 0.01*(xy[n,1] - xy[1,1])`, adding 1 % of the  $x$  range to the last jump point. If `xmax < xy[n,1]`, then `xmax` is set to `xy[n,1]`, the last jump point.

Output:

`xyline`

$(2n + 2) \times 2$  matrix, rows are coordinates of the starting point, the lower right and the upper left corner points, and the end point of a step function with jumps in  $x_i$  to value  $y_i$  (given in input `xy`). Connecting consecutive points with lines draws a plot of the step function.

**Example 4.** We illustrate the use of `hazkpm` and `steps4plot` by plotting a Kaplan-Meier estimate and Greenwood's confidence limits for simulated data. The data are provided in the file `haz01`. They were obtained by generating  $n = 20$  independent, uniformly distributed covariate values  $z_i = (z_{1i}, z_{2i})^T$ , with  $z_{ki} \sim U[-0.5, 0.5]$ ,  $k = 1, 2$ ,  $i = 1, \dots, n$ ; uniformly distributed censoring times,  $c_i \sim U[0, 4]$ ; and exponentially distributed survival times  $y_i | z_i \sim \text{Exp}(\lambda(z_i))$ , with  $\lambda(z) = \exp(z_1 + 2z_2)$ . The first column in `haz01` contains the observed times,  $t_i = \min(c_i, y_i)$ , the second column is

the censoring indicator, and the third and fourth columns contain the covariate values. In this particular sample, three of the observations are censored, including the largest time,  $t_{20}$ .

In this example, we display the confidence limits as step functions, although `hazkpm` provides only pointwise confidence intervals at the event points  $t_i$ . Alternatively, readers may choose to draw vertical lines connecting the confidence limits  $(t_i, cil(t_i))$  and  $(t_i, ciu(t_i))$  to emphasize the pointwise nature of the confidence intervals.

```

library("hazreg")
dat=read("haz01.dat")
t = dat[,1] ; observed times
delta = dat[,2] ; censoring indicator
z = dat[,3:4] ; covariates
{data,ties} = hazdat(t,delta, z) ; preparing data
{cil,kme,ciu} = hazkpm(data)
; compute kme and confidence limits

setsize(600,400) ; initiating graph
plot1=createdisplay(1,1) ; initiating graph
n = rows(data) ; sample size
pm = ( #(1,n+2)' + (0:n) ) | ( #(2*n+2,3*n+3)' + (0:n) )
; points to be connected
cn = matrix(2*n+2) ; color_num, controls colors
ar = matrix(2*n+2) ; art, controls line types
th = matrix(2*n+2) ; thick, controls line thickness

cilline = steps4plot(cil) ; points for step function plot
setmaskl(cilline, pm, cn, ar, th) ; lines control
setmaskp(cilline, 4, 0, 8) ; points control

ciuline = steps4plot(ciu) ; points for step function plot
setmaskl(ciuline, pm, cn, ar, th) ; lines control
setmaskp(ciuline, 4, 0, 8) ; points control

kmeline = steps4plot(kme, 0~1)
; points for step function plot
setmaskl(kmeline, pm, cn, ar, 2*th) ; lines control
setmaskp(kmeline, 4, 0, 8) ; points control

```

```

show(plot1, 1, 1, cilline, kmeline, ciuline)
setgopt(plot1, 1, 1, "title","Kaplan-Meier Estimates")
setgopt(plot1, 1, 1, "xlabel","Time")
setgopt(plot1, 1, 1, "ylabel","Survival Function")
setgopt(plot1, 1, 1, "ymajor",0.2)
print (plot1,"hazkptest.ps")

```


 haz04.xpl

Figure 1 displays the three estimated functions. The pointwise confidence limits are truncated to 0 or 1 when the asymptotic confidence intervals exceed these values. Each step in the Kaplan-Meier estimate corresponds to one event time. In our sample, the event times  $t_2$  and  $t_3$  are very close, and the two jumps merge into one on the plot.

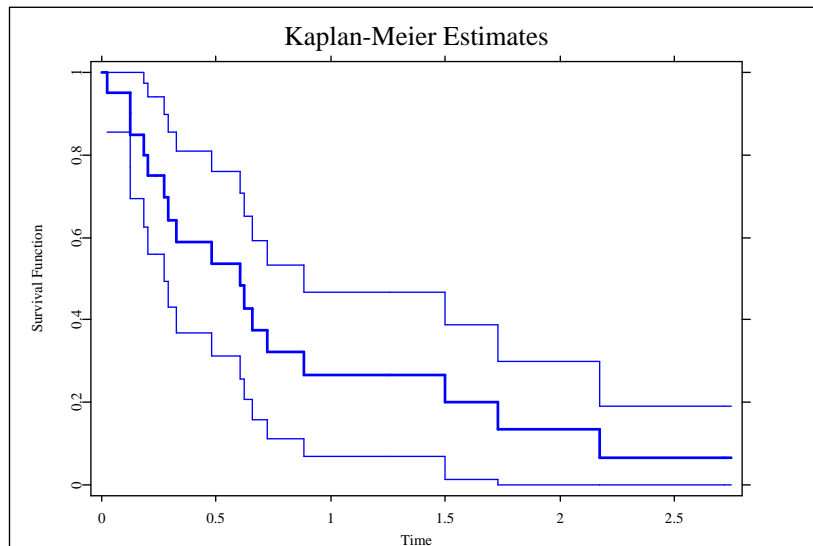


Figure 1: Kaplan-Meier estimate (bold line) and pointwise confidence limits for the survival function. Estimates are based on the simulated data in haz01.

The Kaplan-Meier step function is plotted starting at the point  $(0, 1)$ , while the

step functions for the confidence limits start at the first event point,  $t_1 > 0$ . This is achieved through the argument `xymin` in the `steps4plot` calls. In defining `kmeline` for the Kaplan-Meier step function, `xymin` is set to  $(0, 1)$ , while this argument is omitted when defining `cilline` and `ciuline` for the confidence limits.

### 3 The Cox Proportional Hazards Model

```
{ll, ll1, ll2} = hazregll(data, beta)
    calculates the value of the partial log-likelihood function and of
    the first two derivatives

{betahat, betak, ck} = hazbeta(data {,maxit})
    estimates the regression coefficients for the Cox proportional haz-
    ards model

{bhaz, bsurv} = hazbase(data)
    estimates the baseline hazard and survival functions

surv = hazsurv(data, z)
    estimates the conditional survival function

{val, df, pval} = haztest(data, index)
    performs the likelihood ratio test, Wald's test and the score test
```

The semiparametric Cox proportional hazards model is the most commonly used model in hazard regression. In this model, the conditional hazard function, given the covariate value  $z$ , is assumed to be of the form

$$\lambda(t|z) = \lambda_0(t) \exp\{\beta^T z\},$$

where  $\beta = (\beta_1, \dots, \beta_p)^T$  is the vector of regression coefficients, and  $\lambda_0(t)$  denotes the baseline hazard function. No particular shape is assumed for the baseline hazard; it is estimated nonparametrically. The contributions of covariates to the hazard are multiplicative. An accessible introduction to the Cox model is given, for example, in Klein and Moeschberger (1997).

The quantlib `hazreg` provides quantlets for estimating the regression coefficients,  $\beta$ , standard deviations of these estimates, and estimates for the cumulative baseline hazard and the conditional survival function. Our calculations

are based on standard partial likelihood methods in the proportional hazards model. Additionally, we provide three commonly used tests for the hypothesis that one or more of the  $\beta$ 's are zero. These tests are useful for model choice procedures.

### 3.1 Estimating the Regression Coefficients

Let us assume that there are no ties between the event times. In this case, the partial likelihood function is given by

$$L(\beta) = \prod_{i=1}^n \left\{ \frac{\exp(\beta^T z_i)}{\sum_{j \in R(t_i)} \exp(\beta^T z_j)} \right\}^{\delta_i}, \quad (4)$$

where  $R(t_i) = \{j: t_j \geq t_i\}$  denotes the risk set at time  $t_i$ . Note that only **event** times contribute their own factor to the partial likelihood. However, both censored and uncensored observations appear in the denominator, where the sum over the risk set includes all individuals who are still at risk immediately prior to  $t_i$ .

Let  $\hat{\beta}$  denote the maximum (partial) likelihood estimate of  $\beta$ , obtained by maximizing the partial log-likelihood function,  $l(\beta) = \ln L(\beta)$ . From (4), it follows immediately that

$$l(\beta) = \sum_{i=1}^n \delta_i (\beta^T z_i) - \sum_{i=1}^n \delta_i \ln \left\{ \sum_{j \in R(t_i)} \exp(\beta^T z_j) \right\}. \quad (5)$$

The first derivative of  $l(\beta)$  with respect to  $\beta$  is called **vector of efficient scores**, given by

$$U(\beta) = \frac{dl}{d\beta} = \delta^T Z - \sum_{i=1}^n \delta_i \frac{\sum_{j \in R(t_i)} \exp(\beta^T z_j) Z_{(j, \cdot)}}{\sum_{j \in R(t_i)} \exp(\beta^T z_j)}, \quad (6)$$

where  $\delta = (\delta_1, \dots, \delta_n)^T$  denotes the vector of censoring indicators, and  $Z$  is the  $(n \times p)$ -matrix of covariate values, with the  $j$ -th row containing the covariate values of the  $j$ -th individual,  $Z_{(j, \cdot)} = z_j^T$ .

For the case of ties in the event times, there are several ways to define a partial likelihood function. Currently, we are using formula (4) both for data with and without ties. Each event time contributes one factor to the likelihood

function; for tied events, all events in the tie appear with the same denominator. This approach was suggested by Breslow (1974), and is implemented in most statistical packages. When there are few ties, this approximation to the partial likelihood works rather well, see Klein and Moeschberger (1997), p.238.

The information matrix  $\mathbf{I}(\beta)$  is given by the negative of the second derivative of  $l(\beta)$ . Let  $\mathbf{1}_{R(i)} \in \mathbb{R}^n$  denote the indicator vector of the risk set  $R(t_i)$ ; this means, the  $j$ -th element of  $\mathbf{1}_{R(i)}$  is 1 when  $t_j \geq t_i$ , and 0, otherwise. Then, the information matrix takes the form

$$\begin{aligned} \mathbf{I}(\beta) &= -\frac{d^2 l}{d\beta^2} \\ &= \sum_{i=1}^n \frac{\delta_i}{w_i(\beta)^2} \bar{Z}(i)^T [w_i(\beta) \text{Diag}\{\exp(Z\beta)\} - \exp(Z\beta) \exp(Z\beta)^T] \bar{Z}(i), \end{aligned} \quad (7)$$

where the  $w_i(\beta) = \mathbf{1}_{R(i)}^T \exp(Z\beta)$  are scalars; for any vector  $\nu$ ,  $\text{Diag}\{\nu\}$  denotes the diagonal matrix with the main diagonal  $\nu$ , and  $\exp(\nu)$  is defined elementwise; and  $\bar{Z}(i) = \text{Diag}\{\mathbf{1}_{R(i)}\} Z$ . The matrices  $\bar{Z}(i)$  are modifications of the design matrix  $Z$ , setting the rows of  $\bar{Z}(i)$  to zero when the corresponding observation is not in the risk set for time  $t_i$ . Note that the index  $i$  runs through all  $n$  observations. When ties are present, each of the tied event times appears once, with the same risk set, and contributes the same term to the information matrix.

For large samples, the maximum likelihood estimate  $\hat{\beta}$  is known to follow an asymptotic  $p$ -variate normal distribution,

$$\mathbf{I}(\beta)^{1/2} \left\{ \hat{\beta} - \beta \right\} \rightarrow_{n \rightarrow \infty} N(0, I_p).$$

The inverse of the information matrix,  $\mathbf{I}^{-1}(\hat{\beta})$ , is a consistent estimate of the covariance matrix of  $\hat{\beta}$ . It may be used to construct confidence intervals for the components of  $\beta$ .

The quantlet `hazregll` computes the partial log-likelihood function, its first derivative (efficient scores), and the negative of the second derivative (information matrix). The first and second derivatives of the log-likelihood function (5) are later used to obtain  $\hat{\beta}$ , as well as for computing test statistics for local tests on  $\beta$ . The syntax of `hazregll` is given below:

```
{ll,111,112} = hazregll(data,beta)
```

Input:

**data**

$n \times (p + 4)$  matrix, the sorted data matrix obtained as output `data` of the quantlet `hazdat`;

**beta**

$p \times 1$  vector, the regression coefficient vector  $\beta$ .

Output:

**ll**

scalar, the log-likelihood function at parameter value  $\beta$ ;

**ll1**


$p \times 1$  vector, the first derivatives of the log-likelihood function at parameter value  $\beta$ ;

**ll2**

$p \times p$  matrix, the negative Hessian matrix of the log-likelihood function at parameter value  $\beta$  (information matrix).

**Example 5.** The simulated data in the file `haz01` were generated from a proportional hazards model with  $p = 2$  covariates, the conditional hazard function  $\lambda(t|z) = \exp(\beta^T z)$ , and  $\beta = (1, 2)^T$ . The baseline hazard is constant,  $\lambda_0(t) = 1$ . We use the quantlet `hazregll` to calculate the partial log-likelihood function, the efficient scores and the information matrix at the true parameter value,  $\beta = (1, 2)^T$ .

```
library("hazreg")
dat=read("haz01.dat")
t = dat[,1]                ; observed times
delta = dat[,2]            ; censoring indicator
z = dat[,3:4]              ; covariates
{data,ties} = hazdat(t,delta, z) ; preparing data
beta = 1|2
{ll,ll1,ll2} = hazregll(data,beta)
```

 `haz05.xpl`

The calculations yield `ll = -34.679` for the value of the log-likelihood function,



$111 = (0.014323, 0.88238)^T$  for the first derivatives, and

$$112 = \begin{pmatrix} 1.3696 & -0.43704 \\ -0.43704 & 0.8285 \end{pmatrix}$$

for the information matrix.

The quantlet `hazbeta` calculates the maximum likelihood estimate  $\hat{\beta}$  by solving the nonlinear equation system  $U(\beta) = 0$ , defined in (6). We use a Newton-Raphson algorithm with the stopping criterion

$$C(\hat{\beta}_s) = \frac{|\hat{\beta}_s - \hat{\beta}_{s-1}|}{|U(\hat{\beta}_{s-1})|}.$$

The syntax of `hazbeta` is given below:

```
{betahat, betak, ck} = hazbeta(data {,maxit})
```

Input:

`data`

$n \times (p + 4)$  matrix, the sorted data matrix obtained as output `data` of `hazdat`;

`maxit`

scalar, maximum number of iteration for the Newton-Raphson procedure, default is 40.

Output:

`betahat`

$p \times 1$  vector, estimate of the regression parameter  $\beta$ ;

`betak`


$\text{maxit} \times p$  matrix, iterated parameter values through the Newton-Raphson procedure;

`ck`

$\text{maxit} \times 1$  vector, values of the convergence criterion at each iteration of the Newton-Raphson procedure.

**Example 6.** In this example, we compute  $\hat{\beta}$  for the data in `haz01`, and estimate the covariance matrix of  $\hat{\beta}$  by  $\mathbf{I}^{-1}(\hat{\beta})$ . We use the quantlets `hazbeta` and `hazregll`. The data was generated from a proportional hazards model with  $\beta = (1, 2)^T$ . Details are given in Examples 4 and 5.

```
library("hazreg")
dat=read("haz01.dat")
t = dat[,1] ; observed times
delta = dat[,2] ; censoring indicator
z = dat[,3:4] ; covariates
{data,ties} = hazdat(t,delta, z) ; preparing data
{betahat,betak,ck} = hazbeta(data)
{ll, ll1, ll2} = hazregll(data, betahat)
sigma = inv(ll2) ; covariance matrix estimate
```

 `haz06.xpl`

The calculation results in `betahat = (1.4599, 3.3415)^T`, with the estimated covariance matrix

$$\text{sigma} = \begin{pmatrix} 1.019 & 0.55392 \\ 0.55392 & 1.5847 \end{pmatrix}.$$

Both components  $\beta_1 = 1$  and  $\beta_2 = 2$  are within their respective asymptotic 95% confidence intervals that may be constructed with `betahat` and the square root of the diagonal elements of `sigma`.

### 3.2 Estimating the Hazard and Survival Functions

We estimate the cumulative baseline hazard function,  $\Lambda_0(t) = \int_0^t \lambda_0(s)ds$ , by

$$\hat{\Lambda}_0(t) = \sum_{i: t_i \leq t} \frac{\delta_i}{\sum_{j \in R(t_i)} \exp(\hat{\beta}^T z_j)}.$$

The estimate  $\hat{\Lambda}_0$  is a right-continuous step function, with jumps in the event times. The index  $i$  cycles through all observations,  $i = 1, \dots, n$ . In the case of tied events, each of the events in the tie contributes its own term to the sum; this term is the same for all events in a particular tie. The estimate  $\hat{\Lambda}_0$  can be derived through a profile likelihood approach, see Klein and Moeschberger (1997), pages 260 and 237, and Johansen (1983).

We estimate the baseline survival function,  $S_0(t) = \exp\{-\Lambda_0(t)\}$ , by

$$\hat{S}_0(t) = \exp\{-\hat{\Lambda}_0(t)\}.$$

In the Cox proportional hazards model, the survival function  $S(t|z)$  of an individual with covariate values  $z$  is given by

$$S(t|z) = S_0(t)\exp(\beta^T z). \quad (8)$$

Consequently, we estimate the conditional survival function by substituting estimates for  $S_0(t)$  and  $\beta$ ,

$$\hat{S}(t|z) = \exp\{-\hat{\Lambda}_0(t)\}^{\exp(\hat{\beta}^T z)}. \quad (9)$$

Note that the estimates  $\hat{\Lambda}_0(t)$ ,  $\hat{S}_0(t)$  and  $\hat{S}(t|z)$  are all step functions, with jumps at the event times. All three estimates are non-negative,  $\hat{\Lambda}_0(t)$  is monotonously increasing, and the survival function estimates are monotonously decreasing.

The quantlet `hazcoxb` provides the estimates  $\hat{\Lambda}_0(t)$  and  $\hat{S}_0(t)$ . It has the following syntax:

```
{bcumhaz, bsurv} = hazcoxb(data)
```

Input:

**data**

$n \times (p + 4)$  matrix, the sorted data matrix given by the output `data` of `hazdat`.

Output:

**bcumhaz**

$n \times 2$  matrix, with rows  $(t_i, \hat{\Lambda}_0(t_i))$ , sorted in the same order as the  $t_i$  in `data`;

**bsurv**

$n \times 2$  matrix, with rows  $(t_i, \hat{S}_0(t_i))$ , sorted in the same order as the  $t_i$  in `data`.

**Example 7.** In this example, we calculate and plot estimates of the cumulative baseline hazard  $\Lambda_0(t)$  and of the corresponding survival function  $S_0(t)$ , for the simulated data in `haz01`. The estimates are calculated using the quantlet `hazcoxb`. Plotting of the step functions is supported by `steps4plot`. The resulting plots are displayed in Figures 2 and 3. The data in `haz01` were generated from a proportional hazards model with  $\Lambda_0(t) = t$  and  $S_0(t) = \exp(-t)$ ; details are given in Examples 4 and 5.

```

library("hazreg")
dat=read("haz01.dat")
t = dat[,1] ; observed times
delta = dat[,2] ; censoring indicator
z = dat[,3:4] ; covariates
{data,ties} = hazdat(t,delta, z) ; preparing data
{bcumhaz,bsurv} = hazcoxb(data) ; compute estimates

setsize(600,400) ; initiating graph
plot1=createdisplay(1,1) ; initiating graph
plot2=createdisplay(1,1)
n = rows(data) ; sample size
pm = ( #(1,n+2)'+ (0:n)) | ( #(2*n+2,3*n+3)'+ (0:n))
; points to be connected
cn = matrix(2*n+2) ; color_num, controls colors
ar = matrix(2*n+2) ; art, controls line types
th = matrix(2*n+2) ; thick, controls line thickness

bsurvline = steps4plot(bsurv, 0~1)
; points for step function plot
setmaskl(bsurvline, pm, cn, ar, th) ; lines connected
setmaskp(bsurvline, 4, 0, 8) ; points controlled

bcumhazline = steps4plot(bcumhaz, 0~0)
; points for step function plot
setmaskl(bcumhazline, pm, cn, ar, th)
setmaskp(bcumhazline, 4, 0, 8)

show(plot1, 1, 1, bcumhazline) ; plot baseline hazard
setgopt(plot1, 1, 1, "title","Cumulative Baseline Hazard")
setgopt(plot1, 1, 1, "xlabel","Time")
setgopt(plot1, 1, 1, "ylabel","Cumulative Hazard")


```

```

setgopt(plot1, 1, 1, "ymajor", 0.5)
print (plot1,"hazbcumhaztest.ps")

show(plot2,1, 1, bsurvline)           ; plot baseline survival
setgopt(plot2, 1, 1, "title","Baseline Survival Function")
setgopt(plot2, 1, 1, "xlabel","Time")
setgopt(plot2, 1, 1, "ylabel","Survival Function")
setgopt(plot2, 1, 1, "ymajor", 0.2, "ylim", (0|1.01))
print (plot2,"hazbsurvtest.ps")

```

 haz07.xpl

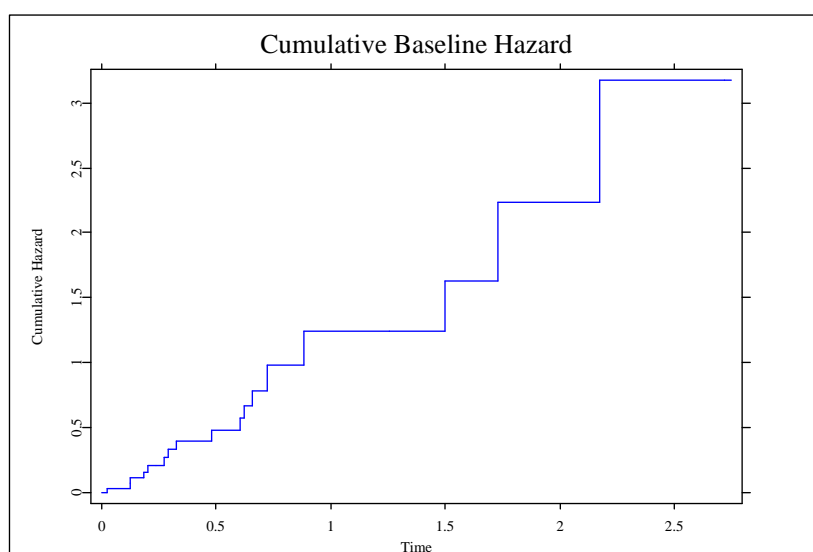


Figure 2: Estimate of the cumulative baseline hazard in the proportional hazards model. Data were generated in a model with  $\Lambda_0(t) = t$ .

The quantlet `hazsurv` provides an estimate of the conditional survival function,  $\hat{S}(t|z)$ , as defined in formula (9). It has the following syntax:

```
surv = hazsurv(data,z)
```

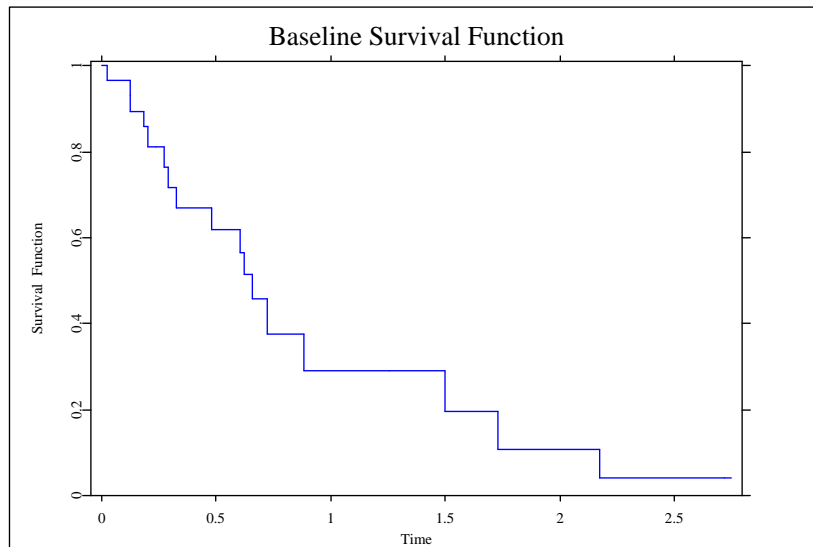


Figure 3: Estimate of the baseline survival function in the proportional hazards model. Data were generated in a model with  $S_0(t) = \exp(-t)$ .

Input:

**data**

$n \times (p + 4)$  matrix, the sorted data matrix given by the output `data` of `hazdat`;

**z**

$p \times 1$  vector, value of the covariates;

Output:

**surv**

$n \times 2$  matrix, the first column is the sorted  $t_i$ , followed by the estimated conditional survival function  $\hat{S}(t_i|\mathbf{z})$ .

**Example 8.** We calculate and plot the estimate  $\hat{S}(t|z)$  of the conditional survival function for  $z = (0.1, -0.3)$ , using the simulated data in haz01. The resulting graph is displayed in Figure 4.

```


library("hazreg")
dat=read("haz01.dat")
t = dat[,1] ; observed times
delta = dat[,2] ; censoring indicator
z = dat[,3:4] ; covariates
{data,ties} = hazdat(t,delta, z) ; preparing data
z1 = 0.1|-0.3 ; covariate values
surv = hazsurv(data, z1)
; estimate conditional survival function

setsize(600, 400) ; initiating graph
plot1=createdisplay(1,1) ; initiating graph
n = rows(data)
pm = ( #(1,n+2)' + (0:n) ) | ( #(2*n+2,3*n+3)' + (0:n) )
; points to be connected
cn = matrix(2*n+2) ; color_num, controls colors
ar = matrix(2*n+2) ; art, controls line types
th = matrix(2*n+2)
; thick, controls line thickness

survline = steps4plot(surv, 0~1)
; points for step function plot
setmaskl(survline, pm, cn , ar, th) ; lines connected
setmaskp(survline, 4, 0, 8) ; points controlled
setsize(600,400)

show(plot1, 1, 1, survline)
setgopt(plot1, 1, 1, "title","Conditional Survival Function")
setgopt(plot1, 1, 1, "xlabel","Time")
setgopt(plot1, 1, 1, "ylabel","Survival Function")
setgopt(plot1, 1, 1, "ylim", (0|1.01), "ymajor", 0.2)
print (plot1,"hazsurvttest.ps")

```

 haz08.xpl

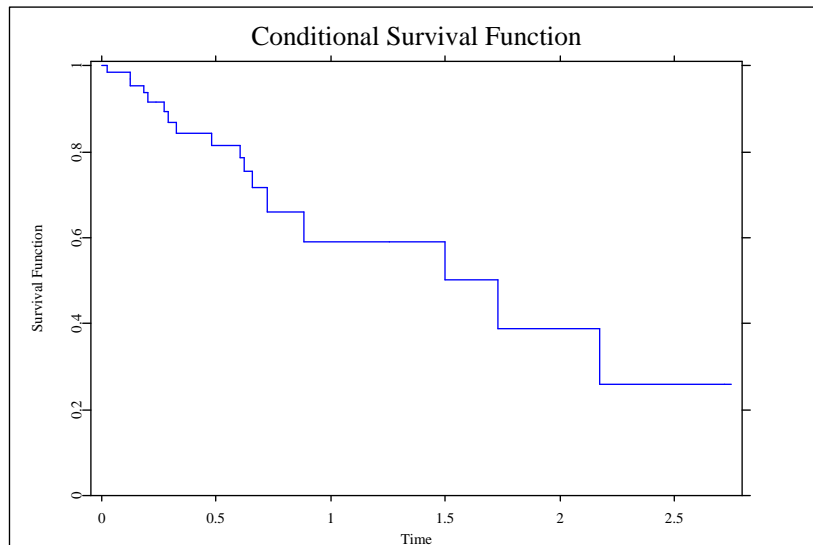


Figure 4: Estimated conditional survival function based on the data in `haz01`, for  $z = (0.1, -0.3)^T$ .

### 3.3 Hypothesis Testing

The quantlib `hazreg` offers three tests for hypotheses about regression parameters: the likelihood ratio test, Wald's test and the score test. Assume that  $\beta = (\beta_1^T, \beta_2^T)^T$ , where the  $q$ -dimensional subvector  $\beta_1$  consists of the regression coefficients of interest, and the  $(p - q)$ -dimensional subvector  $\beta_2$  contains the remaining parameters. We are testing the hypotheses  $H_0: \beta_1 = \mathbf{0}$  versus  $H_a: \beta_1 \neq \mathbf{0}$ , in the presence of the remaining unknown  $(p - q)$  regression coefficients;  $\mathbf{0}$  denotes the  $q$ -dimensional zero vector. This type of tests is often used in model choice procedures, testing whether a given model can be improved by including certain additional covariates or covariate combinations.



### 3.3.1 Likelihood Ratio Test

The test statistic for the likelihood ratio test is given by

$$T_{LR} = 2l(\hat{\beta}) - 2l(\hat{\beta}_0),$$

where  $\hat{\beta}_0 = (\mathbf{0}^T, \hat{\beta}_2^T)^T$ , and  $\mathbf{0}$  and  $\hat{\beta}_2$  are the  $q$ -dimensional zero vector and the conditional maximum likelihood estimate for  $\beta_2$ , given  $\beta_1 = \mathbf{0}$ , respectively. The estimate  $\hat{\beta}_2$  is obtained by substituting the fixed null hypothesis value,  $\beta_1 = \mathbf{0}$ , for the corresponding  $\beta$ 's in the partial log-likelihood function (5).

Under the null hypothesis, the asymptotic distribution of  $T_{LR}$  is  $\chi_q^2$ . We calculate  $p$ -values as tail probabilities of the  $\chi_q^2$  distribution,  $P(\chi_q^2 \geq T_{LR})$ .

### 3.3.2 Wald Test

Let  $\hat{\beta} = (\hat{\beta}_1^T, \hat{\beta}_2^T)^T$  denote the usual maximum partial likelihood estimate of the full parameter vector  $\beta = (\beta_1^T, \beta_2^T)^T$ . Now, let us partition the inverse of the information matrix  $\mathbf{I}(\beta)$  into

$$\mathbf{I}^{-1}(\beta) = \begin{pmatrix} \mathbf{I}^{11} & \mathbf{I}^{12} \\ \mathbf{I}^{21} & \mathbf{I}^{22} \end{pmatrix},$$

where  $\mathbf{I}^{11}$  denotes the  $q \times q$  submatrix corresponding to  $\beta_1$ . The information matrix is defined in (8). The test statistic for the Wald test is given by

$$T_W = \hat{\beta}_1^T \left[ \mathbf{I}^{11}(\hat{\beta}) \right]^{-1} \hat{\beta}_1.$$

Under the null hypothesis, the distribution of  $T_W$  converges to  $\chi_q^2$ .

### 3.3.3 Score Test

Let  $U_1(\beta)$  denote the subvector of the first  $q$  elements of the score function  $U(\beta)$ , defined in (6). The test statistic for the score test is

$$T_{SC} = U_1(\hat{\beta}_0)^T \mathbf{I}^{11}(\hat{\beta}_0) U_1(\hat{\beta}_0),$$

where  $\hat{\beta}_0 = (\mathbf{0}^T, \hat{\beta}_2^T)^T$  is the maximum likelihood estimate for  $\beta$  under the null hypothesis, introduced in Section 3.3.1. Again, the large sample distribution of the test statistic under the null hypothesis is  $\chi_q^2$ .

For more details on hypothesis testing in the Cox model, see Klein and Moeschberger (1997), Section 8.4.

### 3.3.4 Implementation

Values of the three test statistics  $T_{LR}$ ,  $T_W$  and  $T_{SC}$ , and the corresponding asymptotic  $p$ -values are provided by the quantlet `haztest`.  $p$ -values are computed as tail probabilities of the  $\chi_q^2$  distribution, which is the asymptotic distribution for each of the three tests. The syntax of `haztest` is given below:

```
{ttest, val, df, pval} = haztest(data,index)
```

Input:

`data`

$n \times (p + 4)$  matrix, the sorted data matrix given by the output `data` of `hazdat`;

`index`

$p \times 1$  vector, with `index[i]=0` when  $\beta_i = 0$  is part of the null hypothesis, and `index[i]=1`, otherwise;

Output:

`ttest`

printed output, table with values of the test statistics, degrees of freedom and  $p$ -values for the likelihood ratio test, Wald's test and the score test.

`val`

$3 \times 1$  vector, values of the test statistics, in the following order: likelihood ratio test, Wald's test, score test;

`df`

scalar, degrees of freedom of the  $\chi_q^2$  reference distribution;

`pval`


$3 \times 1$  vector,  $p$ -values of the tests.

**Example 9.** We are testing the null hypothesis  $H_0: \beta_2 = 0$  for the data in `haz01`. The quantlet `haztest` provides values for three test statistics and computes the corresponding  $p$ -values as tail probabilities of the  $\chi_1^2$  distribution.

```

library("hazreg")
dat=read("haz01.dat")
t = dat[,1]                ; observed times
delta = dat[,2]           ; censoring indicator
z = dat[,3:4]             ; covariates
{data,ties} = hazdat(t,delta, z) ; preparing data
index = 1|0                ; testing if the second
                           ; coefficient is zero
{testt, val,df,pval} = haztest(data, index)
testt                      ; print summary table

```

 `haz09.xpl`

The variable `testt` contains the summary table. The last line of the code prints the table into the XploRe output window:

```

"-----"
"Cox Proportional Hazards Model"
""
"Hypothesis: beta1=0 for a subvector of regression coefficients"
""
" - - - - - "
"           Test statistic      DF      P-value           "
" - - - - - "
"LR Test           7.56687         1      0.00595"
"Wald Test          7.04612         1      0.00794"
"Score Test         4.25763         1      0.03908"
"-----"
""

```

Additionally, the test statistic values and the  $p$ -values are stored in the variables `val` and `pval`, respectively. The data in `haz01` were generated from a proportional hazards model with  $\beta = (1, 2)^T$ . For this sample, all three tests result in small  $p$ -values, providing evidence against the null hypothesis  $H_0: \beta_2 = 0$ .

### 3.4 Example: Length of Stay in Nursing Homes

Nursing homes provide both short-term and long-term care, and patients may stay from a few days to several years. In this section, we investigate how certain characteristics of nursing home patients influence their length of stay. We use a subset of the **nursing home data** presented by Morris, Norton, and Zhou (1994). The original study was sponsored by the National Center for Health Services in 1980–1982, and investigated the the impact of certain financial incentives on the nursing home care of Medicaid patients. Thirty six nursing homes were randomized into a treatment or control group; nursing homes in the treatment group received financial incentives for admitting more disabled Medicare patients, for improving their health status, and for discharging the patients to their homes within 90 days. The **nursing home data** consist of  $n = 1,601$  patients from this study. Patients were admitted to participating nursing homes between May 1, 1981, and April 30, 1982, and followed over a period of up to three years. The following characteristics were recorded: age, marital status, gender, health status. The data set is available on a floppy disk distributed with Lange et al. (1994), and at *StatLib*, <http://www.stat.cmu.edu/datasets/csb/>.

We restrict our analysis to nursing homes in the control group, which represents the standard care without additional financial incentives, and to patients that were at least 65 years old and of comparatively good health (health status = 2). Our subset consists of  $n = 214$  patients. The patients were followed from admission to either discharge, or death. For patients that were still in a nursing home on April 30, 1983, the length of stay is recorded as **censored** (25 % of the observations).

Our subset of the **nursing home data** is provided in `nursing`. The first column of the data file contains the **length of stay** in the nursing home (in days), the second column is the **censoring indicator**, and columns 3–5 contain the **age** (in years), the **gender** (1=male, 0=female), and the **marital status** (1=married, 0=unmarried), respectively. Twenty one percent of the patients are male, and 14 % are married.

In order to identify the impact of age, marital status and gender on the length of stay in a nursing home, we are fitting a Cox proportional hazards model with  $p = 3$  covariates: `agespline`, `gender` and `married`. The first variable measures the age of a patient as  $\text{agesp} = \min(\text{age}, 90) - 65$ ; this transformation was suggested in Morris, Norton, and Zhou (1994). The other two covariates are indicator variables for the **gender** and **marital status**, respectively. The

Covariate	Mean (SD)	$\hat{\beta}$ SE( $\hat{\beta}$ )	LR Test <i>p</i> -value	Wald Test <i>p</i> -value	Score Test <i>p</i> -value
<b>agespline</b>	15.6 (7.06)	-0.052 (0.011)	0.00001	0.00001	0.00001
<b>gender</b>	0.210	0.037 (0.213)	0.86	0.86	0.09
<b>married</b>	0.140	0.040 (0.246)	0.87	0.87	0.12

Table 1: Covariates in a Cox proportional hazards model fitted to the data in nursing. The **time-to-event** is the length of stay of patients in a nursing home (in days).

second column of Table 1 provides the sample means of the covariates and the standard deviation of **agespline**.

The following code reads in the data and calculates estimates of the regression coefficients,  $\hat{\beta}$ , and their covariance matrix:

```

library("hazreg")
dat=read("nursing.dat")           ; read data from file
t = dat[,1]                       ; time = length of stay
delta = dat[,2]                   ; censoring
age = dat[,3]                     ; covariate AGE
gender = dat[,4]                  ; covariate GENDER
married = dat[,5]                 ; covariate MARRIED
limit = matrix(rows(dat), 1)*90   ; transform AGE
agespline = min(age~limit,2) - 65
{data, ties} = hazdat(t, delta, agespline~gender~married)
                                ; prepare data
{betahat, betak, ck}=hazbeta(data) ; estimate beta
{ll, ll1, ll2} = hazregll(data,betahat)
sigma = inv(ll2)
                                ; covariance matrix of betahat

```

Table 1 presents the estimated regression coefficients in the fitted model ( $\hat{\beta}$ , the value of **betahat**) and their estimated standard deviation, SE( $\hat{\beta}$ ), obtained as square root of the diagonal elements of **sigma**.

Let us test, for each of the covariates, whether it contributes to the hazard in the presence of the other two variables:

```
{ttest1, val1, df1, pval1} = haztest(data, (0|1|1))
                                ;test for AGESPLINE
{ttest2, val2, df2, pval2} = haztest(data, (1|0|1))
                                ;test for GENDER
{ttest3, val3, df3, pval3} = haztest(data, (1|1|0))
                                ;test for MARRIED
                                haz10.xpl
```

The variables `ttest1`, `ttest2` and `ttest3` contain the summary tables for the covariates `agespline`, `gender`, and `married`, respectively. The  $p$ -values are provided in Table 1.

The only covariate with a significant contribution to the hazard is the age. In comparison, in Morris, Norton, and Zhou (1994), a Cox model is fitted to all  $n = 1,601$  observations, with additional variables that identify the health status at entry and the treatment group. Here, the age does not appear to be significant, while gender, marital status and poor health significantly contribute to the length of stay.

These results are an example that caution is advised in interpreting fitted models. In our case, gender and marital status are correlated with the health status: married patients tend to enter the nursing home with more advanced health problems, and men are more likely than women to be admitted in poorer health. In our restricted data set of patients with similar, good health at entry, neither gender nor marital status are helpful for modeling the expected length of stay in the framework of proportional hazards.

## References

- Breslow, N. E. (1974). Covariance analysis of censored survival data, *Biometrics* **30**: 579–594.
- Cox, D. R. and Oakes, D. (1984). *Analysis of Survival Data*, Chapman and Hall, London.
- Greenwood, M. (1926). *The Natural Duration of Cancer*, Reports on Public Health and Medical Subjects, His Majesty’s Stationary Office, London.

- Heckman, J. J. and Singer, B. (1985). Longitudinal Analysis of Labor Market Data, in *Econometric Society Monograph 10*, Cambridge University Press, Cambridge.
- Johansen, S. (1983). An extension of Cox's regression model, *International Statistical Review 51*: 258–262.
- Klein, J. P. and Moeschberger, M. L. (1997). *Survival Analysis: Techniques for Censored and Truncated Data*, Springer-Verlag, New York.
- Lancaster, T. (1990). The Econometric Analysis of Transition Data, in *Econometric Society Monograph 17*, Cambridge University Press, Cambridge.
- Lange, N., Ryan, L., Billard, L., Brillinger, D., Conquest, L., and Greenhouse, J., (1994). *Case Studies in Biometry*, John Wiley & Sons, New York.
- Morris, X., Norton, E., and Zhou, X. (1994). Parametric Duration Analysis of Nursing Home Usage, in *Case Studies in Biometry*, edited by Lange, N., Ryan, L., Billard, L., Brillinger, D., Conquest, L., and Greenhouse, J., John Wiley & Sons, New York.
- Miller, R. G., and Halpern, J. W. (1982). Regression with censored data, *Biometrika 69*: 521–531.