Grund, Birgit; Yang, Lijian

Working Paper

Hazard regression

SFB 373 Discussion Paper, No. 1999,83

Provided in Cooperation with:
Collaborative Research Center 373: Quantification and Simulation of Economic Processes, Humboldt University Berlin

Suggested Citation: Grund, Birgit; Yang, Lijian (1999) : Hazard regression, SFB 373 Discussion Paper, No. 1999,83, Humboldt University of Berlin, Interdisciplinary Research Project 373: Quantification and Simulation of Economic Processes, Berlin, https://nbn-resolving.de/urn:nbn:de:kobv:11-10046700

This Version is available at:
https://hdl.handle.net/10419/61724

Hazard Regression

Birgit Grund [1], *Lijian Yang* [2]
October 14, 1999

Hazard regression models are convenient tools to discover the dynamic structure of survival data. In medical research, the length of patients' survival are often observed and analyzed with hazard regression model, see, for instance, Cox and Oakes (1984). More recent areas of application of the hazard regression model include insurance industry and employment data, see, for example, Heckman and Singer (1985), Lancaster (1990).

The XploRe library `hazreg` contains a collection of quantlets for the analysis of right-censored survival data. They provide the pointwise Kaplan-Meier estimates and confidence intervals of the survival function, estimation of the regression coefficients and the baseline hazard and regression functions of Cox's proportional hazards model, and significance tests for the regression coefficients. In this chapter, we illustrate how these quantlets work with brief theory and some examples. Section 1 introduces quantlets that provide basic description of a survival data and arrange it in a form suitable for analysis. Section 2 discusses the Kaplan-Meier estimator and Greenwood confidence intervals. Section 3 covers the estimation and hypothesis testing of the semiparametric Cox's proportional model using the partial likelihood method.

1 Data Structure

```
{data,ties} = hazdat(t, delta {,z})
```
sorts the right-censored data, covariates, and labels

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

```
inrisk = hazrisk(data,i)
```
determines all observations at risk at time T_i

The data treated in library `hazreg` are right-censored: that is, the data consists of triples (T_i, δ_i, Z_i), $i = 1, \ldots, N$, where T_i denotes the observed survival time of the i-th individual, Z_i is a p covariate vector, respectively, and δ_i is the censoring indicator.

Let Y_i denote the uncensored survival time, which is observed when $\delta_i = 1$, and C_i the random censoring time, which is observed when $\delta_i = 0$. The observed survival time of the i-th individual is then given by $T_i = \min(Y_i, C_i)$.

For many computations, additional information is required and has to be passed on to the functions. For example, the method for estimating the Cox regression coefficients depends on whether or not the data contain ties. Obviously, we could locate the ties each time that a method requires information on ties. However, in a typical session the same dataset will be studied for various purposes. It is much

[1] School of Statistics, University of Minnesota, St. Paul, MN 55108, U.S.A. Supported in part by NSF Grant DMS-9501893, and in part by Sonderforschungsbereich 373 at Humboldt-Universität zu Berlin.

[2] Department of Statistics and Probability, Michigan State University, East Lansing, MI 48824, U.S.A. Supported in part by Sonderforschungsbereich 373 at Humboldt-Universität zu Berlin, and by NSF Grant DMS-9971186.

more efficient to gather the tie information once, and then include it as an argument to the various functions. In the following, we denote by d_i and r_i the number of events observed at time T_i and the number of observations that are still in the risk set at time T_i, respectively.

The quantlet `hazdat` sorts the right-censored data (T_i, δ_i, Z_i), $i = 1, \ldots, N$ in the order of T. Its syntax is the following:

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


`hazdattest.xpl`

where

`t`
: $N \times 1$ vector, consisting of survival times T_i,

`delta`
: $N \times 1$ vector, consisting of censoring indicators δ_i,

`z`
: $N \times p$ matrix, consisting of covariates Z_i, default option is empty matrix.

This quantlet returns *data* and *ties*:

`data`
: $N \times (p + 4)$ matrix, the data sorted according to T_i, the sorted δ_i, the sorted orinigal labels l, a column containing the number of ties d_i, and the sorted covariates Z_i,

`ties`
: scalar, indicator of ties, 0 means there are ties in T_i, 1 means no ties.

The following example, based on hypothetical data, illustrates the use of this quantlet

```
library("hazreg")
y = 2|1|3|2|4|7|1|3|2        ; hypothetical survival
c = 3|1|5|6|1|6|2|4|5        ; hypothetical censoring
t = min(y~c,2)               ; censored time
delta = (y<=c)               ; censoring indicator
{data,ties} = hazdat(t,delta)
```

The output is the following:

```
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 =
        0
```

The quantlet `haznar` calculates the size of the risk set at each point of data, obtained from `hazdat`. Its syntax is the following:

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

haznartest.xpl

where

`data`
: $N \times (p+4)$ matrix, the sorted data matrix as output of hazdat.

This quantlet returns *nar*:

`nar`
: $N \times 1$ vector, the number at risk r_i at each data point, a vector of length N.

The following example, based on hypothetical data, illustrates the use of this quantlet

```
library("hazreg")
y = 2|1|3|2|4|7|1|3|2        ; hypothetical survival
c = 3|1|5|6|1|6|2|4|5        ; hypothetical censoring
t = min(y~c,2)               ; censored time
delta = (y<=c)               ; censoring indicator
{data,ties} = hazdat(t,delta)
nar = haznar(data)
```

The output is the following:

```
nar = (9 9 9 6 6 6 3 3 1)'
```

The quantlet `hazrisk` determines all observations at risk at time T_i, a data point obtained from `hazdat`. Its syntax is the following:

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

hazrisktest.xpl

where

`data`
: $N \times (p+4)$ matrix, the sorted data matrix as output of hazdat,

`i`
: scalar, the position of the risk time point.

This quantlet returns *inrisk*:

`inrisk`
: $N \times 1$ vector, with elements 0 or 1 that indicate whether observations are in the risk set or not at the ith time point.

The following example, based on hypothetical data, illustrates the use of this quantlet

```
library("hazreg")
y = 2|1|3|2|4|7|1|3|2        ; hypothetical survival
c = 3|1|5|6|1|6|2|4|5        ; hypothetical censoring
t = min(y~c,2)               ; censored time
delta = (y<=c)               ; censoring indicator
{data,ties} = hazdat(t,delta)
inrisk = hazrisk(data,6)     ; the risk set at observation 6
```

The output is the following:

```
inrisk = (0 0 0 1 1 1 1 1 1)'
```

2 Kaplan-Meier Estimates

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

The Kaplan-Meier estimate for a survival function, also called "Product-Limit estimator", is given by

$$\hat{S}(t) = \begin{cases} 1 & \text{if} \quad t < T_1 \\ \prod_{T_i \le t} \left[1 - \frac{d_i}{r_i}\right], & \text{if} \quad T_1 \le t. \end{cases},$$

where, as defined in the previous section, d_i and r_i denote the number of ties and the size of the risk set at time T_i, respectively.

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

$$\hat{V}(t) = \hat{S}(t)^2 \sum_{T_i \le t} \frac{d_i}{r_i(r_i - d_i)}. \tag{1}$$

The following pointwise confidence intervals for the survival function is based on the asymptotically normal distribution of the Kaplan-Meier estimator $\hat{S}(t)$, whose variance is estimated by $\hat{V}(t)$, given in (1),

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

where $1 - \alpha$ is the coverage probability and z_p is the p^{th} percentile of the standard normal distribution. Note that Greenwood's estimator (1) tends to slightly underestimate the true variance, so that the true coverage probability of the confidence intervals might be somewhat smaller than stated.

For any data already sorted by `hazdat`, the quantlet `hazkpm` gives the Kaplan-Meier estimates and confidence bounds of the survival function using formula (2). Its syntax is the following:

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


`hazkpmtest.xpl`

where

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

`alpha`
scalar, the specified coverage probability, default option is 0.05.

This quantlet returns *cil*, *kme*, and *ciu*:

`cil`
$N \times 2$ matrix, the first column consists of the sorted T_i, the second column the Greenwood lower confidence bounds at T_i,

`kme`
$N \times 2$ matrix, the first column consists of the sorted T_i, the second column the Kaplan-Meier estimates at T_i,

`ciu`

$N \times 2$ matrix, the first column consists of the sorted T_i, the second column the Greenwood upper confidence bounds at T_i.

The following example, based on hypothetical data, illustrates the use of this quantlet

```
dat=read("haz.dat")
y = dat[,1]                          ; survival time
c = dat[,2]                          ; censoring variable
z = dat[,3:4]                        ; covariates
t = min(y~c,2)                       ; censored time
delta = (y<=c)                       ; censoring indicator
{data,ties} = hazdat(t,delta, z)     ; preparing data
setsize(600,400)
t1=createdisplay(1,1)
{cil,kme,ciu} = hazkpm(data)
n = rows(data)
t = cil[2:n,1]
c = cil[1:n-1,2]
cil = ((cil[1:n-1,])|(t~c))|((t~c)|(cil[2:n,]))
pm = (#(1,n)'+ (0:n-2))|(#(2*n-2,3*n-3)'+ (0:n-2))
cn = matrix(2*n-2)                    ;  color_num, controls colors
ar = matrix(2*n-2)                    ;  art, controls line types
th = 2*matrix(2*n-2)                  ;  thick, controls line thickness
setmaskl(cil ,pm ,cn , ar, th)
setmaskp(cil, 4, 0, 8)
c = ciu[1:n-1,2]
ciu = ((ciu[1:n-1,])|(t~c))|((t~c)|(ciu[2:n,]))
setmaskl(ciu ,pm ,cn , ar, th)
setmaskp(ciu, 4, 0, 8)
k = kme[1:n-1,2]
kme = ((kme[1:n-1,])|(t~k))|((t~k)|(kme[2:n,]))
setmaskl(kme ,pm ,cn , ar, 2*th)
setmaskp(kme, 4, 0, 8)
show(t1,1,1,cil,kme,ciu)
setgopt(t1,1,1, "title","Kaplan-Meier Estimates",
"xlabel","Time","ylabel","Survival Function",  "ymajor", 0.2)
print (t1,"hazkpmtest.ps")
```

Figure 1 depicts the three estimated functions.

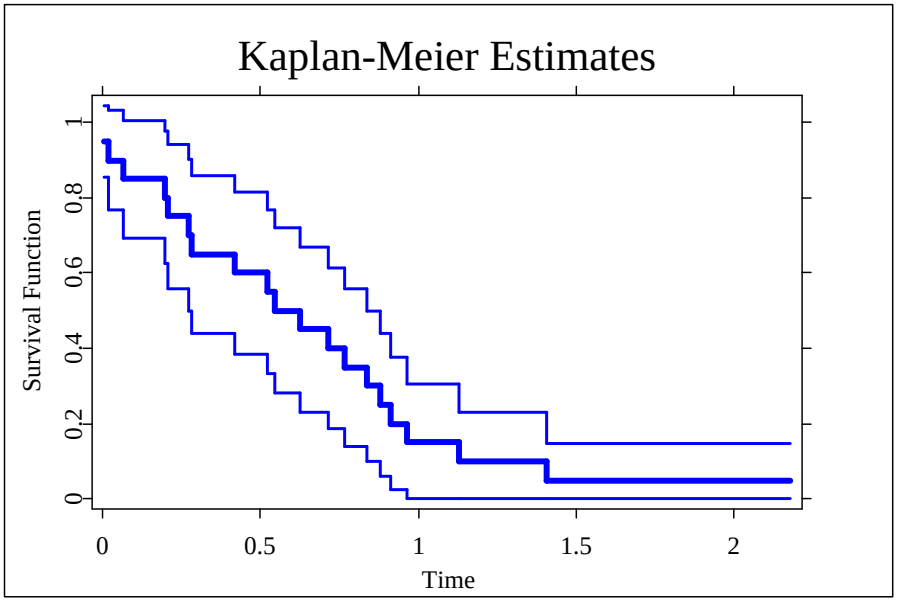

Figure 1: Output display for for simulated data

3 Cox's Proportional Hazards Model

`{ll,ll1,ll2} = hazregll(data,beta)`
calculates derivatives of the log-likelihood function up to order 2

`{betahat, betak, ck} = hazbeta(data {,maxit})`
estimates the parameter for Cox's proportional model

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

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

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

The semiparametric Cox's proportional hazards model is the most commonly used model in hazard regression. In this model, the conditional hazard function given that the covariates take the values Z, is assumed to be of the form

$$\lambda(t|Z) = \lambda_0(t) \exp\{\beta^T Z\}, \tag{3}$$

where β is the vector of regression coefficients, and $\lambda_0(t)$ denote the baseline hazard. The baseline hazard has to be estimated nonparametrically, since no particular shape is assumed. The contributions of the covariate values to the hazard are multiplicative. An accessible introduction to Cox's proportional hazards model is given, for example, in Klein and Moeschberger (1997).

The library `hazreg` provides estimates for the regression coefficients, β, standard deviations for the estimates, and testing procedures for the hypothesis that one or more of the $\beta_j, j = 1, \ldots, p$ are zero. All is carried out through maximizing the partial likelihood function. Additionally, estimates are provided for the baseline hazard function, and the conditional survival and hazard function for given covariate values.

3.1 Estimating the Regression Coefficients

Let us assume that there are no ties between the event times, and let $R(t) = \{j : T_j \geq t\}$ denote the risk set at time point t. In this case, the partial likelihood is given by

$$L(\beta) = \prod_{i=1}^{n} \frac{\exp\left[\beta^T Z_{(i,\cdot)}\right]}{\sum_{k \in R(T_i)} \exp\left[\beta^T Z_{(k,\cdot)}\right]}, \tag{4}$$

and the (partial) maximum likelihood estimate of β is obtained by maximizing the log-likelihood function, $l(\beta) = \ln L(\beta)$. From (4) it follows immediately that

$$l(\beta) = \sum_{i=1}^{n} \beta^T Z_{(i,\cdot)} - \sum_{i=1}^{n} \ln \left\{ \sum_{k \in R(T_i)} \exp\left[\beta^T Z_{(k,\cdot)}\right] \right\}. \tag{5}$$

The first derivatives of $l(\beta)$ with respect to β are called "efficient scores", given by

$$U(\beta) = \frac{dl}{d\beta} = \mathbf{1}_n^T Z - \sum_{i=1}^{n} \frac{\sum_{k \in R(T_i)} Z_{(k,\cdot)} \exp\left[\beta^T Z_{(k,\cdot)}\right]}{\sum_{k \in R(T_i)} \exp\left[\beta^T Z_{(k,\cdot)}\right]}, \tag{6}$$

where $\mathbf{1}_n$ is the $n-$dimensional vector with all elements equal 1. Let $\hat{\beta}$ denote the (partial) maximum likelihood estimate for β.

For the case that there are ties, we still use formula (4), taking into account that the risk set $R(T_i)$ for a time T_i with tied events includes all observations tied at t_i. 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 $I(\beta)$ is given by the negative of the second derivative of $l(\beta)$. Let $\mathbf{I}_{R(i)} \in \Re^n$ denote the indicator vector of the risk set $R(T_i)$, this means the j^{th} element of $\mathbf{I}_{R(i)}$ is 1 when $T_j \geq T_i$, and 0, otherwise. Then, the information matrix takes the form

$$\begin{aligned} I(\beta) &= -\frac{d^2 l}{d\beta^2} \\ &= \sum_{i=1}^{n} \left[\mathbf{1}_n^T \exp[\bar{Z}(i)\beta]\right]^{-2} \bar{Z}(i)^T \\ &\quad \times \left\{ \mathbf{1}_n^T \exp[\bar{Z}(i)\beta] Diag\left\{\exp\left[Z\beta\right]\right\} - \exp\left[Z\beta\right] \exp\left[Z\beta\right]^T \right\} \bar{Z}(i) \end{aligned}$$

where $\bar{Z}(i) = Diag\left\{I_{R(i)}\right\} Z$, and $Diag\{\nu\}$ denotes the diagonal matrix with the main diagonal given by vector ν. This means that $\bar{Z}(i)$ is a modification of the design matrix Z, where the j^{th} row of $\bar{Z}(i)$ is set zero when $T_j < T_i$. Note that the index i lists all n observations. When ties are present, each of the tied observations appears with the same risk set, and contributes the same term to the sum.

Using a Newton-Raphson algorithm, $\hat{\beta}$ is calculated by solving the nonlinear equation system $\frac{dl}{d\beta} = 0$. The following stopping criterion is used

$$C(\beta_k) = \frac{|\hat{\beta}_k - \hat{\beta}_{k-1}|}{|l(\hat{\beta}_{k-1})|}.$$

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

$$I(\beta)^{1/2}\left\{\hat{\beta}-\beta\right\}\longrightarrow_{n\to\infty}N(0,I_p). \tag{7}$$

The covariance matrix of $\hat{\beta}$ will be consistently estimated by $I^{-1}(\hat{\beta})$.

The first and second derivatives of the log-likelihood function (5) are used for the iteration algorithm for $\hat{\beta}$, as well as to calculate the standard deviation of the asymptotic distribution of $\hat{\beta}$, and to compute test statistics for local tests on β used for model building. The partial log-likelihood function and its derivatives are computed by the quantlet `hazregll`. Its syntax is the following:

```
{ll,ll1,ll2} = hazregll(data,beta)
```


`hazreglltest.xpl`

where

`data`
: $N\times(p+4)$ matrix, the sorted data matrix as output of hazdat,

`beta`
: $p\times 1$ vector, the regression coefficients.

This quantlet returns *ll*, *ll1*, and *ll2*:

`ll`
: scalar, the log-likelihood function at parameter value beta,

`ll1`
: $p\times 1$ vector, the first derivatives at parameter value beta of the log-likelihood function,

`ll2`
: $p\times p$ matrix, the negative Hessian matrix at parameter value beta of the log-likelihood function.

The following example, based on hypothetical data, illustrates the use of this quantlet

```
library("hazreg")
dat=read("haz.dat")
y = dat[,1]                          ; survival time
c = dat[,2]                          ; censoring variable
z = dat[,3:4]                        ; covariates
t = min(y~c,2)                       ; censored time
delta = (y<=c)                       ; censoring indicator
{data,ties} = hazdat(t,delta, z)     ; preparing data
beta = 1|2
{ll,ll1,ll2} = hazregll(data,beta)
```

The calculation yields values of *ll*, *ll1*, and *ll2* as -43.306, $(2.4277,-2.6719)^T$, and $\begin{pmatrix} 1.5556 & -0.14401 \\ -0.14401 & 2.093 \end{pmatrix}$ respectively.

The Newton-Raphson routine is contained in the quantlet `hazbeta`. Its syntax is the following:

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

where

`data`
: $N \times (p+4)$ matrix, the sorted data matrix as output of hazdat,

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

This quantlet returns *betahat*, *betak*, and *ck*:

`betahat`
: $p \times 1$ vector, estimate of the regression parameter beta,

`betak`
: maxit $\times$ p matrix, iterated parameter values through the Newton-Raphson procedure,

`ck`
: maxit $\times$ 1 vector, convergence criteria values through the Newton-Raphson procedure.

The following example, based on hypothetical data, illustrates the use of this quantlet

```
library("hazreg")
dat=read("haz.dat")
y = dat[,1]                          ; survival time
c = dat[,2]                          ; censoring variable
z = dat[,3:4]                        ; covariates
t = min(y~c,2)                       ; censored time
delta = (y<=c)                       ; censoring indicator
{data,ties} = hazdat(t,delta, z)     ; preparing data
{betahat,betak,ck} = hazbeta(data)
```

The calculation yields value of *betahat* as $(1.9214, 0.83433)^T$, *ck* up to the last 4 iterations until it stops as $(2.4646e-05, 1.5725e-05, 1.0032e-05, 6.3996e-06)^T$, *betak* is too large to present here.

3.2 Estimating the Hazard and Survival Functions

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

$$\hat{\Lambda}_0(t) = \sum_{i:T_i \le t} \frac{d_i}{\exp\left[Z\hat{\beta}\right]^T I_{R(i)}}. \tag{8}$$

The estimator $\hat{\Lambda}_0(t)$ can be justified through a profile likelihood approach, see Klein and Moeschberger (1996), p.260 and p.237, and Johansen (1983).

As an estimator for the baseline survival function, $S_0(t) = \exp\left[-\Lambda_0(t)\right]$, we use

$$\hat{S}_0(t) = \exp\left[-\hat{\Lambda}_0(t)\right]. \tag{9}$$

In the Cox proportional hazards model, the survival function $S(t|Z)$ of an individual with covariate values Z is connected to the baseline survival function through a multiplicative factor,

$$S(t|Z) = S_0(t)^{\exp\left[Z^T\beta\right]}. \tag{10}$$

Consequently, we estimate the conditional survival function by

$$\hat{S}(t|Z) = \exp\left[-\hat{\Lambda}_0(t)\right]^{\exp\left[Z^T\hat{\beta}\right]}. \tag{11}$$

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

The quantlet `hazbase` gives the estimates $\hat{\Lambda}(t)$, $\hat{S}_0(t)$. Its syntax is the following:

```
{bhaz, bsurv} = hazbase(data)
```


`hazbasetest.xpl`

where

`data`
: $N \times (p+4)$ matrix, the sorted data matrix as output of hazdat.

This quantlet returns *bhaz* and *bsurv*:

`bhaz`
: $N \times 2$ matrix, the first column is the sorted T_i, followed by the estimated baseline hazard function at T_i,

`bsurv`
: $N \times 2$ matrix, the first column is the sorted T_i, followed by the estimated baseline survival function at T_i.

The following example, based on hypothetical data, illustrates the use of this quantlet

```
library("hazreg")
dat=read("haz.dat")
y = dat[,1]                          ; survival time
c = dat[,2]                          ; censoring variable
z = dat[,3:4]                        ; covariates
t = min(y~c,2)                       ; censored time
delta = (y<=c)                       ; censoring indicator
{data,ties} = hazdat(t,delta, z)     ; preparing data
setsize(600,400)
t1=createdisplay(1,1)
t2=createdisplay(1,1)
{bhaz,bsurv} = hazbase(data)
n = rows(data)
t = bsurv[2:n,1]
b = bsurv[1:n-1,2]
bsurv = ((bsurv[1:n-1,])|(t~b))|((t~b)|(bsurv[2:n,]))
pm = (#(1,n)'+ (0:n-2))|(#(2*n-2,3*n-3)'+ (0:n-2))
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
setmaskl(bsurv ,pm ,cn , ar, th)
setmaskp(bsurv, 4, 0, 8)
b = bhaz[1:n-1,2]
bhaz = ((bhaz[1:n-1,])|(t~b))|((t~b)|(bhaz[2:n,]))
setmaskl(bhaz, pm ,cn , ar, th)
setmaskp(bhaz, 4, 0, 8)
```

```
show(t1, 1, 1, bhaz)                ; plot baseline hazard
setgopt(t1,1,1, "title","Baseline Hazard Function","xlabel","Time",
"ylabel","Hazard Function",  "ymajor", 0.2)
print (t1,"hazbhaztest.ps")
show(t2,1, 1, bsurv)                ; plot baseline survival
setgopt(t2,1,1, "title","Baseline Survival Function","xlabel","Time",
"ylabel","Survival  Function",  "ymajor", 0.2)
print (t2,"hazbsurvtest.ps")
```

Figures 2 and 3 depict the baseline hazard and survival functions estimated from the above example.

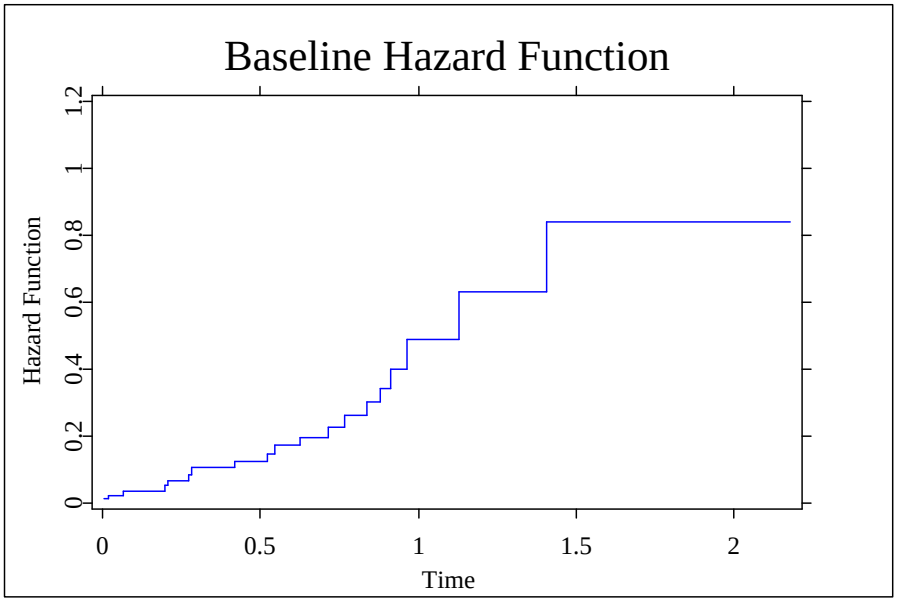

Figure 2: Output display for for simulated data

The quantlet `hazsurv` gives the estimate $\hat{S}(t|Z)$. Its syntax is the following:

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

`hazsurvtest.xpl`

where

`data`
: $N \times (p+4)$ matrix, the sorted data matrix as output of hazdat,

`z`
: $p \times 1$ vector, value of the covariates.

This quantlet returns *surv*:

`surv`
: $N \times 2$ matrix, the first column is the sorted T_i, followed by the estimated survival function at T_i, conditional on Z.

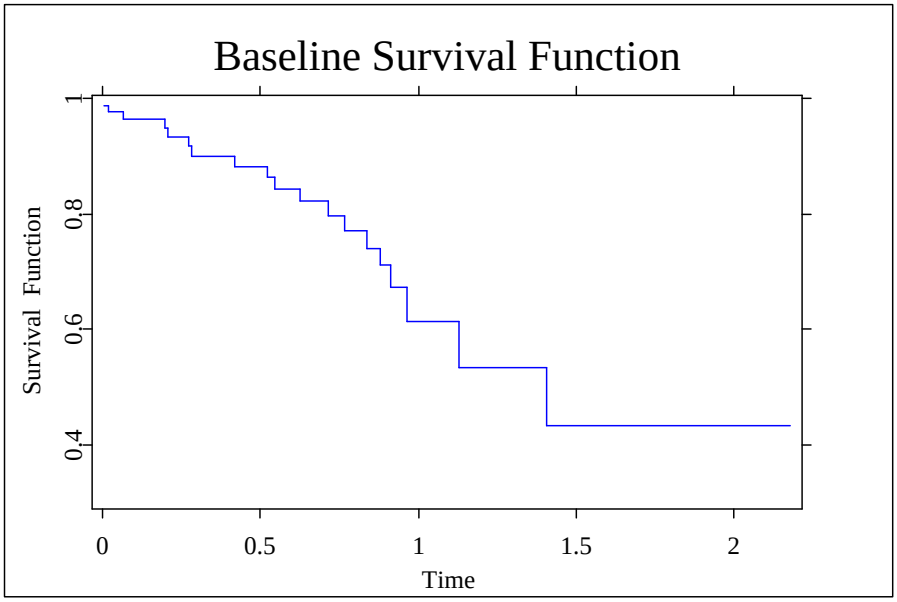

Figure 3: Output display for for simulated data

The following example, based on hypothetical data, illustrates the use of this quantlet

```
library("hazreg")
n = 20
p = 2
beta = 1|2                          ; regression parameter
z = 1 + uniform(n,p)                ; covariates
y = -log(1-uniform(n))              ; exponential survival
y = y./exp(z*beta)                  ; covariate effects
c = 4*uniform(n)                    ; uniform censoring
t = min(y~c,2)                      ; censored time
delta = (y<=c)                      ; censoring indicator
{data,ties} = hazdat(t,delta, z)    ; preparing data
z1 = 1.1|1.23
surv = hazsurv(data, z1)
t = surv[2:n,1]
s = surv[1:n-1,2]
surv = ((surv[1:n-1,])|(t~s))|((t~s)|(surv[2:n,]))
pm = (#(1,n)'+ (0:n-2))|(#(2*n-2,3*n-3)'+ (0:n-2))
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
setmaskl(surv ,pm ,cn , ar, th)
setmaskp(surv, 4, 0, 8)
setsize(600,400)
t1=createdisplay(1,1)
show(t1, 1, 1, surv)
setgopt(t1,1,1, "title","Conditional Survival Function","xlabel","Time",
```

```
"ylabel","Survival Function",  "ymajor", 0.2)
print (t1,"hazsurvtest.ps")
```

Figure 4 depicts the conditional survival function at $Z = (1.1, 1.23)^T$ estimated from the above example.

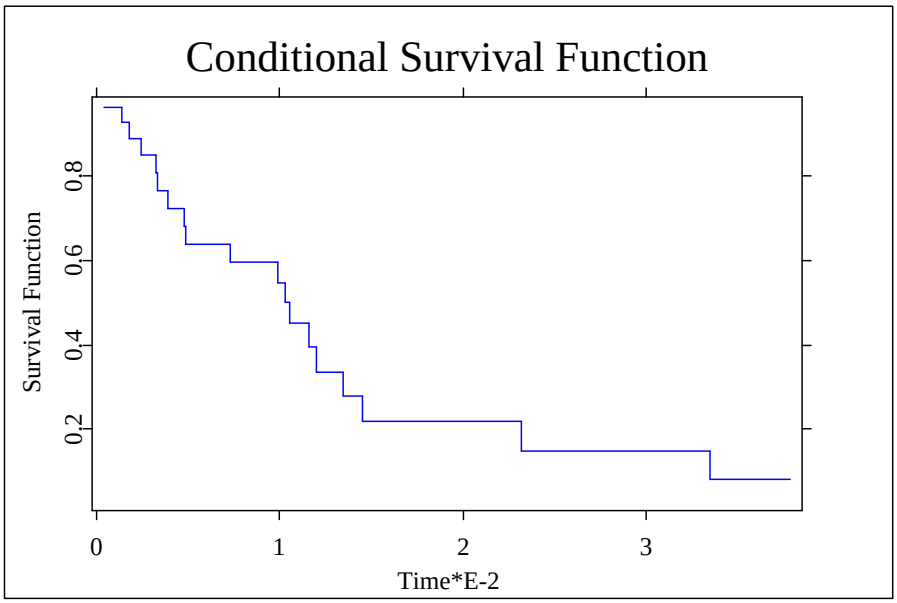

Figure 4: Output display for for simulated data

3.3 Hypothesis Testing

The library `hazreg` offers three tests for hypotheses about subsets of regression parameters, the likelihood ratio test, Wald's test and the scores test. Assume that $\beta = (\beta_1^T, \beta_2^T)^T$, where $\beta_1 \in \Re^q$ consists of the regression coefficients of interest, and $\beta_2 \in \Re^{p-q}$ contains the remaining parameters. We are testing the hypotheses $H_0 : \beta_1 = \beta_{10}$ vs. $H_1 : \beta_1 \neq \beta_{10}$, where $\beta_{10} \in \Re^p$ is given.

Likelihood-ratio Test. Let $\hat{\beta}_2|\beta_{10}$ denote the conditional maximum likelihood estimate for β_2, given β_{10}. It is obtained by substituting the fixed null hypothesis value β_{10} for the corresponding β's in the partial log-likelihood function (5).

The likelihood ratio test statistic is given by

$$T_{LR} = 2l(\hat{\beta}) - 2l(\hat{\beta}_0), \tag{12}$$

where $\hat{\beta}_0 = (\beta_{10}^T, \hat{\beta}_2^T|\beta_{10}^T)^T$. Under H_0, the large sample distribution of T_{LR} is χ_q^2.

Wald's Test. Let $\hat{\beta} = (\hat{\beta}_1^T, \hat{\beta}_2^T)$ denote the usual maximum partial likelihood estimate of the full parameter vector (β_1^T, β_2^T). Now, let us partition the information matrix $I(\beta)$, defined in (7), into

$$I(\beta) = \begin{pmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{pmatrix}, \tag{13}$$

where I_{11} denotes the $q \times q$ submatrix corresponding to β_1. The Wald test is defined through

$$T_W = (\hat{\beta}_1 - \beta_{10})^T I_{11}(\hat{\beta})^{-1}(\hat{\beta}_1 - \beta_{10}). \tag{14}$$

Under the null hypothesis, the distribution of T_W converges to χ^2_q.

Scores Test. Let $U_1(\beta)$ denote the sub-vector of the first q elements of the score function $U(\beta)$, defined in (6). The test statistic for the scores test is

$$T_{SC} = U_1(\hat{\beta}_0)^T I_{11}(\hat{\beta}_0) U_1(\hat{\beta}_0), \tag{15}$$

where $\hat{\beta}_0$ is defined as for the likelihood ratio test. Again, the large sample distribution of the test statistic under the null hypothesis is χ^2_q.

Implementation. The library `hazreg` contains routines for testing $H_0 : \beta_1 = 0$, where β_1 is an arbitrary $q-$dimensional sub-vector of β. In other words, the hypothesized vector β_{10} is taken to be zero.

Values of the three above test statistics and the corresponding asymptotic P-values, computed using the χ^2_q distribution are provided through the quantlet `haztest`. Its syntax is the following:

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


`haztesttest.xpl`

where

`data`
: $N \times (p+4)$ matrix, the sorted data matrix as output of hazdat,

`index`
: $p \times 1$ vector, with ith element $= 0$ when $\beta_i = 0$ is in the null hypothesis, and 1, otherwise.

This quantlet returns *val*, *df* and *pval*:

`val`
: 3×1 vector, values of the test statistics,

`df`
: scalar, degree of freedom,

`pval`
: 3×1 vector, P-values of the tests.

The following example, based on hypothetical data, illustrates the use of this quantlet

```
library("hazreg")
dat=read("haz.dat")
y = dat[,1]                          ; survival
c = dat[,2]                          ; censoring
z = dat[,3:4]                        ; covariates
t = min(y~c,2)                       ; censored time
delta = (y<=c)                       ; censoring indicator
{data,ties} = hazdat(t,delta, z)     ; preparing data
index = 1|0                          ; testing if the second
                                     ; coefficient is zero
{val,df,pval} = haztest(data, index)
```

The output is the following:

```
val~pval =
           0.76588  0.38148
           0.35145  0.55328
         4.426e-09  0.99995


df =
        1
```

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.

Marubini, E. and Valsecchi, M. G. (1994). *Analysing Survival Data from Clinical Trials and Observational Studies*, John Wiley & Sons, New York.

Miller, R. G. and Halpern, J.W. (1982). Regression with censored data, *Biometrika* **69**: 521–531.