0.1 coxph: Cox Proportional Hazards Regression for Duration Dependent Variables

Choose the Cox proportional hazards regression model if the values in your dependent variable are duration observations. The advantage of the semi-parametric Cox proportional hazards model over fully parametric models such as the exponential or Weibull models is that it makes no assumptions about the shape of the baseline hazard. The model only requires the proportional hazards assumption that the baseline hazard does not vary across observations. The baseline hazard can be estimated from the model via post-hoc analysis.

Syntax

```
> z.out <- zelig(Surv(Y, C) ~ X1 + X2, model = "coxph", data = mydata)
> x.out <- setx(z.out)
> s.out <- sim(z.out, x = x.out)
```

Cox proportional hazards models require that the dependent variable be in the form `Surv(Y, C)`, where `Y` and `C` are vectors of length n. For each observation i in 1, ..., n, the value y_i is the duration (lifetime, for example), and the associated c_i is a binary variable such that $c_i = 1$ if the duration is not censored (*e.g.*, the subject dies during the study) or $c_i = 0$ if the duration is censored (*e.g.*, the subject is still alive at the end of the study). If c_i is omitted, all Y are assumed to be completed; that is, c_i defaults to 1 for all observations.

Additional Inputs

In addition to the standard inputs, `zelig()` takes the following additional options for Cox proportional hazards regression:

- `robust`: defaults to `FALSE`. If `TRUE`, `zelig()` computes robust standard errors based on sandwich estimators (see `?` and `?`) based on the options in `cluster`.
- `cluster`: if `robust = TRUE`, you may select a variable to define groups of correlated observations. Let `X3` be a variable that consists of either discrete numeric values, character strings, or factors that define the clusters. Then

  ```
  > z.out <- zelig(Surv(Y,C) ~ X1 + X2, robust = TRUE, cluster = "X3",
                   model = "coxph", data = mydata)
  ```

  means that the observations can be correlated within the clusters defined by the variable `X3`, and that robust standard errors should be calculated according to those clusters. If `robust = TRUE` but `cluster` is not specified, `zelig()` assumes that each observation falls into its own cluster.

- `method`: defaults to `"efron"`. Use this argument to specify how to handle ties within event times. The model assumes that no two event times should theoretically ever be the same, and any ties that occur are simply because the observation mechanism is not precise enough. In practice, ties often exist in the data so the model commonly uses one of three methods to deal with ties.
  - **Breslow method** (`method = "breslow"`): This method is the simplest computationally but also the least precise, especially as the number of tied events increases.
  - **Efron method** (`method = "efron"`): This is the default method and is more intensive computationally but also more precise than the Breslow method.
  - **Exact discrete method** (`method = "exact"`): This is the preferred method if the number of distinct events is rather small due to a large number of ties. Although it can be very computationally intensive, the exact discrete method, which computes the exact partial likelihood, is the most precise method when there are many ties.

Stratified Cox Model

In addition, `zelig()` also supports the stratified Cox model, where the baseline hazards are assumed to be different across different strata but the coefficients are restricted to be the same across strata. Let `id` be a variable that consists of either discrete numeric values, character strings, or factors that define the strata. Then the stratified Cox model can be estimated using `strata()` in the formula. The user can then find quantities of interest for a specific stratum by defining the stratum of choice in `setx()`. If no strata are defined, `setx` takes the mode. Strata on `setx` are defined as followed:

- If strata were defined by a variable (`strata(id)`), then strata should be defined as `strata = "id=5"`.
- If strata were defined by a mathematical expression (`strata(id>10)`), then strata should be defined as `strata = "id>10=TRUE"` or `strata = "id>10=FALSE"`.

```
> z.out <- zelig(Surv(Y,C) ~ X1 + X2 + strata(id), model = "coxph",
                 data = mydata)
> x.out <- setx(z.out, strata = "id=5")
> s.out <- sim(z.out, x = x.out)
```

Time-Varying Covariates

`zelig()` also supports the use of time-varying covariates for the Cox model, where some or all of the covariates change over time for each case. Let "case" refer to each unit in the data. Then each case can have one or more "observations", where each observation has a different

value for one or more covariates for a specific case.

Estimating a time-varying covariate model with `zelig()` involves setting up the data differently to reflect a counting process. In the typical non-time-varying covariate model, the cases include a duration time (Y), a censoring mechanism (C), and covariates (X). A typical dataset would look like this:

Case	Y	C	X1	X2
1	35	0	4	7
2	56	1	6	11

The user would then estimate the model and find quantities of interest using the following syntax:

```
> z.out <- zelig(Surv(Y,C) ~ X1 + X2, model = "coxph", data = mydata)
> x.out <- setx(z.out)
> s.out <- sim(z.out, x = x.out)
```

With time-varying covariates, each case is composed of multiple observations with start times, stop times, censoring (event) mechanisms, and covariates. The covariates are assumed to be constant within the intervals defined by the start and stop times. The covariates change only between intervals. Thus, the covariates are constant at each observation. The censoring mechanism equals 1 when an event occurs at the stop time and equals 0 if the observation is censored or if no event occurs at the stop time. A typical time-varying dataset would look like this:

Case	Start	Stop	C	X1	X2
1	0	26	0	4	7
1	26	35	0	4	10
2	0	39	0	6	11
2	39	56	1	9	5

The user would then estimate the model and find quantities of interest using the following syntax:

```
> z.out <- zelig(Surv(Start,Stop,C) ~ X1 + X2, model = "coxph",
                 data = mydata)
> x.out <- setx(z.out)
> s.out <- sim(z.out, x = x.out)
```

Examples

1. Example 1: Basic Example

   Attaching the sample dataset:

```
> data(coalition)
```

   Estimating parameter values for the coxph regression:

```
> z.out1 <- zelig(Surv(duration, ciep12) ~ invest + numst2 + crisis,
+     robust = TRUE, cluster = "polar", model = "coxph", data = coalition)
```

   Setting values for the explanatory variables:

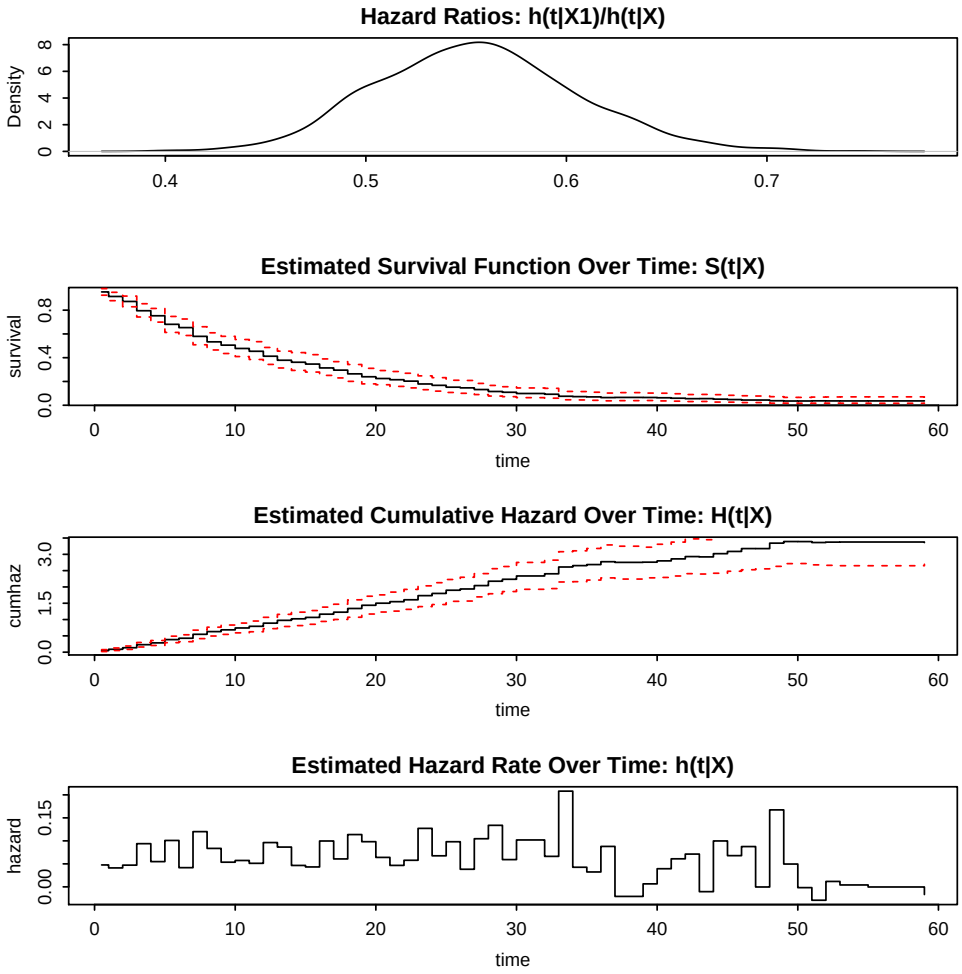
```
> x.low1 <- setx(z.out1, numst2 = 0)
> x.high1 <- setx(z.out1, numst2 = 1)
```

   Simulating quantities of interest:

```
> s.out1 <- sim(z.out1, x = x.low1, x1 = x.high1)
```

```
> summary(s.out1)
```

```
> plot(s.out1)
```



2. Example 2: Example with Stratified Cox Model

   Estimating parameter values for the stratified coxph regression:

```
> z.out2 <- zelig(Surv(duration, ciep12) ~ invest + strata(polar) +
+     numst2 + crisis, model = "coxph", data = coalition)
```

   Setting values for the explanatory variables:

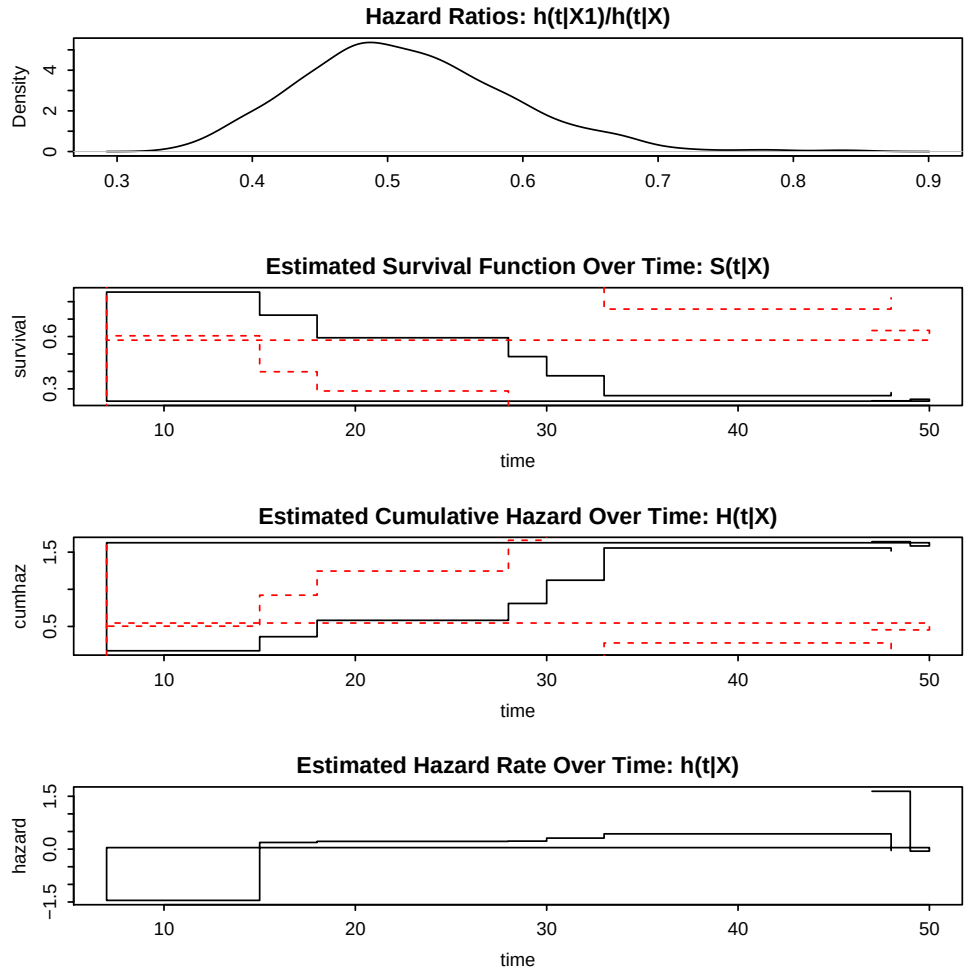
```
> x.low2 <- setx(z.out2, numst2 = 0, strata = "polar=3")
> x.high2 <- setx(z.out2, numst2 = 1, strata = "polar=3")
```

   Simulating quantities of interest:

```
> s.out2 <- sim(z.out2, x = x.low2, x1 = x.high2)
```

```
> summary(s.out2)
```

```
> plot(s.out2)
```



3. Example 3: Example with Time-Varying Covariates

   Create sample toy dataset (from `survival` package):

```
> toy <- as.data.frame(list(start = c(1, 2, 5, 2, 1, 7, 3, 4, 8,
+     8), stop = c(2, 3, 6, 7, 8, 9, 9, 9, 14, 17), event = c(1,
+     1, 1, 1, 1, 1, 0, 0, 0), x = c(1, 0, 0, 1, 0, 1, 1, 1,
+     0, 0), x1 = c(5, 5, 7, 4, 5, 6, 3, 2, 7, 4)))
```

   Estimating parameter values for the coxph regression:

```
> z.out3 <- zelig(Surv(start, stop, event) ~ x + x1, model = "coxph",
+     data = toy)
```

   Setting values for the explanatory variables:

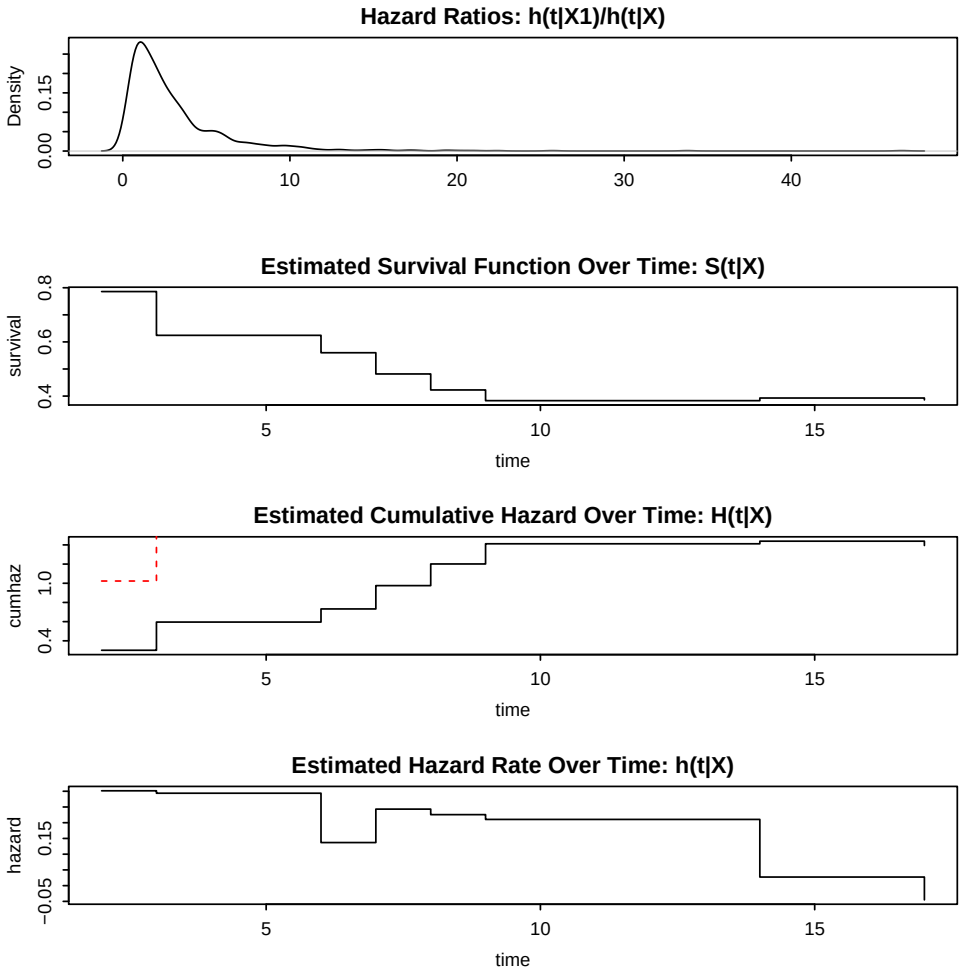
```
> x.low3 <- setx(z.out3, x = 0)
> x.high3 <- setx(z.out3, x = 1)
```

Simulating quantities of interest:

```
> s.out3 <- sim(z.out3, x = x.low3, x1 = x.high3)

> summary(s.out3)

> plot(s.out3)
```



The Model

Let Y_i^* be the survival time for observation i. This variable might be censored for some observations at a fixed time y_c such that the fully observed dependent variable, Y_i, is defined as

$$Y_i = \begin{cases} Y_i^* & \text{if } Y_i^* \leq y_c \\ y_c & \text{if } Y_i^* > y_c \end{cases}$$

- The *stochastic component* is described by the distribution of the partially observed variable Y^*:

$$Y_i^* \sim f(y_i^* | \mu_i, \alpha)$$

where f is an unspecified distribution with some mean μ_i and shape α. In the Cox proportional hazards model, the distributional form of the duration times is unknown and left unparameterized. Instead it uses the proportional hazards assumption to model the set of (ordered) event times on particular covariates.

An important component of all survival models is the hazard function $h(t)$, which measures the probability of an observation not surviving past time t given survival up to t. The hazard function is given by

$$h_i(t) = \lambda(t) \times \lambda_i$$

where $\lambda(t)$ is the baseline hazard (when all covariates equal 0), which varies over t but not over i, and λ_i is the parameterized part of the hazard function, which varies over i but not over t (the proportional hazards assumption).

The model estimates the parameters without a distributional assumption on the duration times by focusing on the occurrence of events and ignoring the time between events. The data are reconceptualized from duration times to K discrete event times such that each y_i corresponds to exactly one event time t_i. The model assumes that no two y_i have the same event times.

For each event time, denote $R(t_i)$ as the set of all observations j that are at risk at t_i. Given that an event occurred at t_i, we are interested in the conditional probability that the event occurred in observation i. The conditional probability is given by

$$\begin{aligned} \Pr(y_i = t_i \mid \text{an event at } t_i) &= \frac{h_i(t_i)}{\sum_{j \in R(t_i)} h_j(t_i)} \\ &= \frac{\lambda(t_i)\,\lambda_i}{\sum_{j \in R(t_i)} \lambda(t_i)\,\lambda_j} \\ &= \frac{\lambda_i}{\sum_{j \in R(t_i)} \lambda_j} \end{aligned}$$

where the numerator denotes the probability of observation i experiencing the event at t_i and the denominator denotes the probability that an event occurred at t_i.

- The *systematic component* λ_i is modeled as

$$\lambda_i = \exp(x_i \beta)$$

where x_i is the vector of explanatory variables, and β is the vector of coefficients.

- Each risk set (and thus each event time) contributes one conditional probability to the partial likelihood function, given by

$$L(\beta|y) = \prod_{i=1}^{K} \left[\frac{\exp(x_i\beta)}{\sum_{j \in R(t_i)} \exp(x_j\beta)} \right]^{c_i}$$

  where c_i is the binary censoring variable. Note that event times corresponding to censored observations are not counted since their corresponding terms for the partial likelihood are exponentiated to 0. However, all censored observations are considered part of the risk sets $R(t_i)$ for all event times prior to their censoring, but otherwise do not contribute to the partial likelihood. For an example, see **?**, 53.

- In the case of the Cox model with time-varying covariates, the partial likelihood function is similarly given by

$$L(\beta|y) = \prod_{i=1}^{K} \left[\frac{\exp(x_i(t_i)\ \beta)}{\sum_{j \in R(t_i)} \exp(x_j(t_i)\ \beta)} \right]^{c_i}$$

  where $x_i(t_i)$ is the value of the covariates at time t_i. Denote "cases" as the units in our data. Each case is composed of one or more observations corresponding to different values in one or more covariates. At each event time t_i, the partial likelihood evaluates the hazard of the case in which the event occurred in with its covariate values at t_i (the numerator) and the hazard of all the other cases at risk at t_i (risk set $R(t_i)$) with their covariate values at t_i (the denominator). See previous section for more information.

- Although the model assumes that there are no tied event times, in practice, data often have tied event times due to imprecise measurement. There are three commonly used methods to deal with tied event times.

  - **Breslow method**: The Breslow method simply treats the risk set as the same for all tied events in the risk set. Suppose observations 1 and 3 are tied in a risk set of observations 1, 2, 3, and 4. Theoretically, if the event occurred in 1 before in 3, then the risk set for observation 3 would have dropped observation 1. However, since we cannot tell which event occurred first, in the partial likelihood, the risk set for observation 1 and observation 3 are the same, consisting of both observations 1 and 3 as well as 2 and 4. For each risk set $R(t_i)$, let d_i equal the number of tied events in the ith risk set and let D_i denote the set of d_i tied events. For risk sets with no tied events, $d_i = 1$. The approximate partial likelihood for the Breslow method is given by

$$L(\beta|y) = \prod_{i=1}^{K} \left[\frac{\prod_{i \in D_i} \exp(x_i\beta)}{\left[\sum_{j \in R(t_i)} \exp(x_j\beta)\right]^{d_i}} \right]^{c_i}$$

– **Efron method**: The Efron method is more precise because it tries to account for how the risk set changes depending on the sequence of tied events. For an intuition behind the Efron approximation, suppose as in the previous example that observations 1 and 3 are tied in a risk set of observations 1, 2, 3, and 4. If the event occurred in 1 before 3, then the risk set for the second event would consist of observations $\{2, 3, 4\}$. On the other hand, if the event occurred in 3 before 1, then the risk set for the second event would consist of observations $\{1, 2, 4\}$. Since both cases are equally plausible with the tied event times, the Efron approximation suggests that the second risk set would consist of $\{2, 3, 4\}$ with 0.5 probability and $\{1, 2, 4\}$ with 0.5 probability. The Efron approximate partial likelihood is then given by

$$L(\beta|y) = \prod_{i=1}^{K} \left[\frac{\prod_{i \in D_i} \exp(x_i\beta)}{\prod_{r=1}^{d_i} \left[\sum_{j \in R(t_i)} \exp(x_j\beta) - \frac{r-1}{d_i} \sum_{j \in D_i} \exp(x_j\beta) \right]} \right]^{c_i}$$

where r indexes D_i, which is the set of d_i tied events for the ith risk set.

– **Exact discrete method**: Unlike the Breslow and Efron methods, which assume a continuous time process, the exact discrete method assumes a discrete time process where the tied events actually do occur at exactly the same time. The method begins by assuming that the data are grouped into risk sets $R(t_i)$. In each risk set and for each observation, denote a binary dependent variable which takes on the value of 1 for each observation that experiences the event and 0 for each observation that does not experience the event. Denote d_i as the number of 1s in $R(t_i)$ and D_i as the set of observations with 1s in $R(t_i)$. D_i represents a specific pattern of 0s and 1s (in our previous example, the specific pattern of 0s and 1s is that observations 1 and 3 experienced an event while 2 and 4 did not, so D_i is the set $\{1, 3\}$). Then for each $R(t_i)$, we are interested in the conditional probability of getting the specific pattern of 0s and 1s given the total number of 1s in $R(t_i)$. Thus, the conditional probability for each risk set is given as

$$\Pr(D_i|d_i) = \frac{\prod_{i \in D_i} \exp(x_i\beta)}{\sum_{m=1}^{M} \left[\prod_{j \in A_{im}} \exp(x_j\beta) \right]}$$

where A_{im} is a set of observations that represents one combination of d_i number of 1s in $R(t_i)$. There are M possible combinations for each risk set. The partial likelihood then takes the conditional probability over each i risk set. Note that the exact discrete approximation method is equivalent to a conditional logit model.

Quantities of Interest

- The hazard ratio (`qi$hr`) is defined as

$$\text{HR} = \frac{h(t \mid x_1)}{h(t \mid x)} = \frac{\lambda(t)\exp(x_1\beta)}{\lambda(t)\exp(x\beta)} = \frac{\exp(x_1\beta)}{\exp(x\beta)}$$

given draws of β from its sampling distribution, where x and x_1 are values of the independent variables chosen by the user. Typically, x and x_1 should only differ over one independent variable to interpret the effect of that variable on the hazard rate. In a stratified Cox model, the strata should be the same in both x and x_1.

- The survival function (`qi$survival`) is defined as the fraction of observations surviving past time t. It is derived from the cumulative hazard function (`exp(-cumhaz)`). The confidence interval of the survival function is drawn on the `log(survival)` scale.
- The cumulative hazard function (`qi$cumhaz`) is defined as `-log(survival)`. Although there is no direct interpretation, the cumulative hazard function is estimated from the data and then other quantities of interest are derived from the cumulative hazard function.
- The hazard function (`qi$hazard`) is defined as the probability of an observation not surviving past time t given survival up to t. It is derived directly from the cumulative hazard function.
- For MI data, if survival times are multiply imputed, we suggest having a larger number of imputed datasets. Because the quantities of interest are derived semi-parametrically, there may be instances in which survival times appear only in one or a small fraction of the multiply imputed datasets, which may bias the results.

Output Values

The output of each Zelig command contains useful information which you may view. For example, if you run `z.out <- zelig(Surv(y,c) ~ x, model = "coxph", data)`, then you may examine the available information in `z.out` by using `names(z.out)`, see the `coefficients` by using `z.out$coefficients`, and a default summary of information through `summary(z.out)`. Other elements available through the `$` operator are listed below.

- From the `zelig()` output object `z.out`, you may extract:
  - `coefficients`: parameter estimates for the explanatory variables.
  - `var`: the variance-covariance matrix.
  - `residuals`: the working residuals of the fit.
  - `loglik`: the log-likelihood for the baseline and full models
  - `linear.predictors`: a mean-adjusted linear predictor $x_i\beta$, where $x_i = x_i - \text{mean}(x)$.
- From `summary(z.out)`, you may extract:
  - `coef`: the parameter estimates with their associated standard errors, p-values, and z-statistics.

  - `conf.int`: $\exp(\beta)$ and their associated confidence intervals.

- From the `sim()` output object `s.out`, you may extract quantities of interest arranged as matrices indexed by simulation $\times$ `x`-observation (for more than one `x`-observation). Available quantities are:

  - `qi$hr`: the simulated hazard ratios for the specified values of `x` and `x1`.
  - `qi$survival`: the estimated survival function for the values specified in `x`.
  - `qi$cumhaz`: the estimated cumulative hazard function for the values specified in `x`.
  - `qi$hazard`: the estimated hazard function for the values specified in `x`.

How To Cite

To cite Zelig as a whole, please reference these two sources:

Kosuke Imai, Gary King, and Olivia Lau. 2007. "Zelig: Everyone's Statistical Software," `http://GKing.harvard.edu/zelig`.

Imai, Kosuke, Gary King, and Olivia Lau. (2008). "Toward A Common Framework for Statistical Analysis and Development." Journal of Computational and Graphical Statistics, Vol. 17, No. 4 (December), pp. 892-913.

See also

The Cox proportional hazards model is part of the survival library by Terry Therneau (**?**), ported to R by Thomas Lumley. Advanced users may wish to refer to `help(coxph)` and `help(survfit)` in the survival library. Sample data are from **?**