Package: survivalMPL (via r-universe)

October 11, 2026

Title Penalised Maximum Likelihood for Survival Analysis Models

Version 0.2-4

Date 2025-03-23

Description Estimate the regression coefficients and the baseline hazard of proportional hazard Cox models with left, right or interval censored survival data using maximum penalised likelihood. A 'non-parametric' smooth estimate of the baseline hazard function is provided.

Depends R (>= 3.5.0), survival, stats, MASS

NeedsCompilation no

LazyData true

License LGPL (>= 2)

Author Dominique-Laurent Couturier [aut, cre] (<https://orcid.org/0000-0001-5774-5036>), Jun Ma [aut], Stephane Heritier [aut], Maurizio Manuguerra [aut], Serigne Lo [aut]

Maintainer Dominique-Laurent Couturier <dominique.couturier@mrc-bsu.cam.ac.uk>

Repository https://ptuc-stats.r-universe.dev

Date/Publication 2025-03-24 12:00:02 UTC

RemoteUrl https://github.com/cran/survivalMPL

RemoteRef HEAD

RemoteSha 932d09b66348c8c8a6d9eb30b78ad17043f72dfb

Contents

survivalMPL-package 2
bcos2 3
coef.coxph_mpl 3
coxph_mpl 5
coxph_mpl.control 7

coxph_mpl.object 10
plot.coxph_mpl 11
predict.coxph_mpl 12
residuals.coxph_mpl 14
summary.coxph_mpl 15

Index **18**

`survivalMPL-package` *Penalised Maximum Likelihood for Survival Analysis Models*

Description

Simultaneously estimate the regression coefficients and provide a 'non-parametric' smooth estimate of the baseline hazard function for proportional hazard Cox models using maximum penalised likelihood (MPL).

Details

This package allows to perform simultaneous estimation of the regression coefficients and baseline hazard function in Cox proportional hazard models, with right, left and interval censored data and independent censoring, by maximising a penalised likelihood, in which a penalty function is used to smooth the baseline hazard estimate.

Optimisation is achieved using a new iterative algorithm, which combines Newton's method and the multiplicative iterative algorithm by Ma (2010), and respects the non-negativity constraints on the baseline hazard estimate (refer to Ma, Couturier, Heritier and Marschner (2021)).

Valid inferences for the regression coefficients and the baseline hazard, cumulative baseline hazard and survival functions as well as for their predictions are available.

This software is accepted by users "as is" and without warranties or guarantees of any kind.

Author(s)

Dominique-Laurent Couturier, Jun Ma, Stephane Heritier, Maurizio Manuguerra.

Maintainer: Dominique-Laurent Couturier <dominique-laurent.couturier@cruk.cam.ac.uk>.

References

Ma, J. and Couturier, D.-L., and Heritier, S. and Marschner, I.C. (2021), Penalized likelihood estimation of the proportional hazards model for survival data with interval censoring. *International Journal of Biostatistics*,doi:10.1515/ijb-2020-0104.

Ma, J. and Heritier, S. and Lo, S. (2014), On the Maximum Penalised Likelihood Approach for Proportional Hazard Models with Right Censored Survival Data. *Computational Statistics and Data Analysis* **74**, 142-156.

Ma, J. (2010), Positively constrained multiplicative iterative algorithm for maximum penalised likelihood tomographic reconstruction. *IEEE Transactions On Signal Processing* **57**, 181-192.

bcos2 *Breast Cosmesis Data*

Description

Interval censored data described and given in full in Finkelstein and Wolfe (1985), discussed by More (2016, example 12.2) and available in the R package 'interval' (refer to ?bcos). Compared to the interval package version, bcos2 simply recode the lower value of left-censored data (NA instead of 0) and upper value of right-censored data (NA instead of Inf) to allow an easy identification of left-censored data by means of the function `Surv` when `type=="interval2"`.

Usage

```
data(bcos2)
```

Format

A data frame with 94 observations on the following 3 variables.

`left` a numeric vector

`right` a numeric vector

`treatment` a factor with levels `Rad` and `RadChem`

Source

Finkelstein, D.M., and Wolfe, R.A. (1985). A semiparametric model for regression analysis of interval-censored failure time data. Biometrics 41: 731-740.

Moore, D. K. (2016), Applied Survival Analysis Using R, *Springer*.

Examples

```
data(bcos2)
```

coef.coxph_mpl *Extract coefficients of a coxph_mpl Object or of its* `summary`

Description

Extract the coefficients of the model part of interest of a `coxph_mpl` object, and the matrix of coefficients of the model part of interest and corresponding standard errors, z-statistics and p-values of a `summary.coxph_mpl` object.

Usage

```
## S3 method for class 'coxph_mpl'
coef(object, parameters="Beta",...)

## S3 method for class 'summary.coxph_mpl'
coef(object, parameters="Beta",...)
```

Arguments

object	an object inheriting from class coxph_mpl or summary.coxph_mpl.
parameters	the set of parameters of interest. Indicate parameters="Beta" for the regression parameters and parameters="Theta" for the baseline hazard parameters. Default is parameters="Beta".
...	other arguments.

Details

When the input is of class summary.coxph_mpl and parameters=="Theta", only the parameter estimates larger than min.Theta (see coxph_mpl.control) are reported.

Value

a vector of coefficients or a matrix of coefficients with standard errors, z-statistics and corresponding p-values.

Author(s)

Dominique-Laurent Couturier, Maurizio Manuguerra

See Also

coxph_mpl and summary.coxph_mpl.

Examples

```
## Not run:
data(lung)

fit_mpl <- coxph_mpl(Surv(time, status == 2) ~ age + sex + ph.karno + wt.loss, data = lung)

coef(fit_mpl)
coef(summary(fit_mpl))

## End(Not run)
```

coxph_mpl *Fit Cox Proportional Hazards Regression Model Via MPL*

Description

Simultaneously estimate the regression coefficients and the baseline hazard function of proportional hazard Cox models using maximum penalised likelihood (MPL).

Usage

```
coxph_mpl(formula, data, subset, na.action, control, ...)

## S3 method for class 'coxph_mpl'
print(x,...)
```

Arguments

formula	a formula object, with the response on the left of a ~ operator, and the terms on the right. The response must be a survival object as returned by the Surv function (in the Surv, use type = "interval2" with interval censored data).
data	a data.frame in which to interpret the variables named in the formula, or in the subset argument. If no dataset is indicated, variables will be taken from the global environment.
subset	expression indicating which subset of the rows of data should be used in the fit. All observations are included by default.
na.action	a missing-data filter function. This is applied to the model.frame after any subset argument has been used. Default is options()\$na.action.
x	an object inheriting from class coxph_mpl, representing a fitted Cox proportional hazard model.
control	Object of class coxph_mpl.control specifying control options like basis choice, smoothing parameter value and maximum number of itereations, for example. Refer to coxph_mpl.control to see the defaults.
...	Other arguments. In coxph_mpl, these elements, will be passed to coxph_mpl.control. In print.coxph_mpl, these elements will be passed to the print function.

Details

coxph_mpl allows to simultaneously estimate the regression coefficients and baseline hazard function of Cox proportional hazard models, with right censored data and independent censoring, by maximising a penalised likelihood, in which a penalty function is used to smooth the baseline hazard estimate.

Optimisation is achieved using a new iterative algorithm, which combines Newton's method and the multiplicative iterative algorithm proposed by Ma (2010), and respects the non-negativity constraints on the baseline hazard estimate (refer to Ma, Couturier, Heritier and Marschner (2021)).

The centered **X** matrix is used in the optimisation process to get a better shaped (penalised) log-likelihood. Baseline hazard parameter estimates and covariance matrix are then respectively corrected using a correction factor and the delta method.

When the chosen basis is not uniform, estimates of zero are possible for baseline hazard parameters and will correspond to active constraints as defined by Moore and Sadler (2008). Inference, as described by Ma, Heritier and Lo (2014), is then corrected accordingly (refer to Moore and Sadler (2008)) by adequately 'cutting' the corresponding covariance matrix.

There are currently 3 ways to perform inference on model parameters:

Let H denote the Hessian matrix of the unpenalised likelihood, Q denote the product of the first order derivative of the penalised likelihood by its transpose, and M_2 denote the second order derivative of the penalised likelihood. Then,

- `'H'` refers to H^{-1}, the inverse of the Hessian matrix,
- `'M2QM2'`, refers to the sandwich formula $M_2^{-1}QM_2^{-1}$,
- `'M2HM2'`, refers to the sandwich formula $M_2^{-1}HM_2^{-1}$.

Simulations analysing the coverage levels of confidence intervals for the regression parameters seem to indicate that $M_2^{-1}HM_2^{-1}$ performs better when using the uniform basis, and that $M_2^{-1}QM_2^{-1}$ performs when using other bases.

Value

an object of class `coxph_mpl` representing the fit. See `coxph_mpl.object` for details.

Author(s)

Dominique-Laurent Couturier, Jun Ma, Stephane Heritier, Maurizio Manuguerra. Design inspired by the function `coxph` of the `survival` package.

References

Ma, J. and Couturier, D.-L., and Heritier, S. and Marschner, I.C. (2021), Penalized likelihood estimation of the proportional hazards model for survival data with interval censoring. *International Journal of Biostatistics*,doi:10.1515/ijb-2020-0104.

Ma, J. and Heritier, S. and Lo, S. (2014), On the Maximum Penalised Likelihood Approach for Proportional Hazard Models with Right Censored Survival Data. *Computational Statistics and Data Analysis* **74**, 142-156.

Ma, J. (2010), Positively constrained multiplicative iterative algorithm for maximum penalised likelihood tomographic reconstruction. *IEEE Transactions On Signal Processing* **57**, 181-192.

Moore, T. J. and Sadler, B. M. and Kozick R. J. (2008), Maximum-Likelihood Estimation, the Cramer-Rao Bound, and the Method of Scoring With Parameter Constraints, *IEEE Transactions On Signal Processing* **56, 3**, 895-907.

Moore, D. K. (2016), Applied Survival Analysis Using R, *Springer* .

See Also

`coxph_mpl.object`, `coxph_mpl.control`, `summary.coxph_mpl` and `plot.coxph_mpl`.

Examples

```
## Not run:
## right censoring example based on the dataset 'lung'
## of the survival package (refer to ?lung)
## with hazard approximated by means of a step function (default).
data(lung)

fit_mpl <- coxph_mpl(Surv(time, status == 2) ~ age + sex + ph.karno + wt.loss, data = lung)

summary(fit_mpl)

## interval censoring example
## (refer to ?bcos2 and to Moore (2016, example 12.2))
## with hazard approximated by means of m-splines

data(bcos2)

fit_mpl <- coxph_mpl(Surv(left, right, type="interval2") ~ treatment,
                     data = bcos2, basis="m")
summary(fit_mpl)


## End(Not run)
```

`coxph_mpl.control` *Ancillary arguments for controling coxph_mpl fits*

Description

This is used to set various numeric parameters controling a Cox model fit using `coxph_mpl`. Typically it would only be used in a call to `coxph_mpl`. Some basic checks are performed on inputs, such that impossible argument values (like a negative number of events per base, for example) are avoided.

Usage

```
coxph_mpl.control(n.obs=NULL, basis = "uniform",
        smooth = NULL, max.iter=c(150,7.5e+04,1e+06),
        tol=1e-7, n.knots = NULL, n.events_basis = NULL,
        range.quant = c(0.075,.9), cover.sigma.quant = .25,
        cover.sigma.fixed=.25, min.theta = 1e-10, penalty = 2L,
        order = 3L, kappa = 1/.6, epsilon = c(1e-16, 1e-10),
        ties = "epsilon", seed = NULL)
```

Arguments

`n.obs`	the number of fully observed (i.e., non censored) outcomes. This argument is only required when `basis=="uniform"` to define an acceptable range of values for `n.events_basis`.

`basis` the name of the basis to use to approximate the baseline hazard function. Available options are `"uniform"`, for a step function approximation); `"gaussian"`, using truncated Gaussian densities; `"msplines"`, as defined by Ramsay (1988); and `"epanechikov"`. Default is `basis="uniform"`.

`smooth` the smoothing parameter value. When specified, it should be larger or equal to zero. By default, the smoothing value is set to `NULL` and its optimal value is estimated via REML. Maximum likelihood estimates are obtained by specifying `smooth=0`. The effect of the smooting parameter over the estimates (currently) depends on the response range, such that its value is (currently) difficult to interpret.

`max.iter` a vector of 3 integers defining the maximum number of iterations for the smooth parameter (first value) and for the Beta and Theta (second value) parameters to attempt for convergence. The third value is the total number of iterations allawed. Default is `max.iter=c(150,7.5e+04,1e+06)`.

`tol` the convergence tolerence value. Convergence is achieved when the maximum absolute difference between the parameter estimates at iteration k and iteration k-1 is smaller than `tol`. Default is `tol=1e-7`.

`n.knots` a vector of 2 integers defining how the internal knot sequence (the minimum and maximum observations define the external knots) of non-uniform bases should be set. The first value specify the number of quantile knots to be set between the `range.quant` quantiles of the fully observed (i.e., non censored) outcomes. The second value specify the number of equally spaced knots to be set outside the range of the quantile knots. The first and last equally spaced knots equal the minimum and maximum response values. When the number of quantile knots is larger than 0, other equally spaced knots are set between the largest quantile knot and the maximum outcome value. The minimal total number of knots is 3. Default is `n.knots=c(8,2)` when `basis=="msplines"` and `n.knots=c(0,20)` otherwise.

`n.events_basis` an integer specifing the number of fully observed (i.e., non censored) outcome per uniform base. The value has to be larger or equal to one and smaller than `n.obs` divided by 2. Default is `round(3.5*log(n.obs)-7.5)` if it belongs to the accepted range of values.

`range.quant` a vector of length 2 defining the range of the quantile knots when a non uniform basis is chosen. By default, `range.quant = c(0.075,.9)` such that `n.knots[1]` quantile knots are set between the quantiles `0.075` and `0.9` of the fully observed (i.e., non censored) outcomes.

`cover.sigma.quant`
the proportion of fully observed (i.e., non censored) outcomes that should belong to the interval defined by the quantiles 0.025 and 0.975 of each truncated Gaussian base corresponding to a quantile knot (see `n.kots`). This value allows to define the standard deviation of these bases. Default is `cover.sigma.quant=.25`.

`cover.sigma.fixed`
the proportion of the outcome range that should belong to the interval defined by the quantiles 0.025 and 0.975 of each *un*truncated Gaussian base corresponding to each fixed knot (see `n.kots`). Default is `cover.sigma.fixed=.25`.

`min.theta` a value indicating the minimal baseline hazard parameter value in the output (i.e., after the fit). Baseline hazard parameter estimates lower than `min.theta`

will be considered as zero. Consequenlty, in the inference, these zero estimates will correspond to active constraints as defined by Moore and Sadler (2008). Default is `1e-10`.

`penalty` an integer specifying the order of the penalty matrix (see Ma, Heritier and Lo (2008)). Currently, the first and second order penalty matrices are available for the `"uniform"` and `"gaussian"` bases, the second order penalty matrix is available for the `"epanechikov"` basis, and the penalty matrix of the `"msplines"` basis is set to `order-1` (see `order` below). Default is `penalty=2`.

`order` an integer specifying the order of the `"msplines"` (as defined by Ramsay (1988)) and `"epanechikov"` basis. Default is `order=3`. M-splines of order 1 correspond to a uniform base (with density equal to one) and M-splines of order 2 correspond to a triangular base.

`kappa` a value larger than 1 used in the fitting algorithm to decrease the step size when the penalised likelihood doesn't increase during the iterative process. Default is `kappa=1/.6`.

`epsilon` a vector of 2 values indicating the minimum distance from 1 and from 0 for - respectively - the survival function and the baseline parameter estimates in order to avoid problems with logarithms **in** the fitting algorithm . Default is `epsilon=c(1e-16, 1e-10)`

`ties` a character string indicating a method to handle duplicated outcomes when defining the knots sequence (see `n.events_basis` and `n.knots`). Current available options are `"epsilon"` which add a random noise smaller than 1e-10 to each duplicate fully observed (i.e., non censored) outcomes, and `"unique"` which delete duplicated fully observed (i.e., non censored) outcomes when defing the knot sequence. Default is `ties="epsilon"`.

`seed` `NULL` or an integer vector compatible with `.Random.seed`: the seed to be used when adding a random noise to duplicate events when `ties="epsilon"`. The current value of `.Random.seed` will be preserved if `seed` is set, i.e. non-`NULL`; otherwise, as by default, `.Random.seed` will be used and modified as usual from calls to `runif()` etc.

Value

a list containing the values of each of the above arguments (except `n.obs`).

Author(s)

Dominique-Laurent Couturier, Maurizio Manuguerra

References

Ma, J. and Heritier, S. and Lo, S. (2014), On the Maximum Penalised Likelihood Approach for Proportional Hazard Models with Right Censored Survival Data. *Computational Statistics and Data Analysis* **74**, 142-156.

Moore, T. J. and Sadler, B. M. and Kozick R. J. (2008), Maximum-Likelihood Estimation, the Cramer-Rao Bound, and the Method of Scoring With Parameter Constraints, *IEEE Transactions On Signal Processing* **56, 3**, 895-907.

Ramsay, J. O. (1988), Monotone Regression Splines in Action, *Statistical Science* **3, 4**, 425-441.

See Also

coxph_mpl

---

coxph_mpl.object *MPL Proportional Hazards Regression Object*

Description

This class of objects is returned by the coxph_mpl class of functions to represent a proportional hazards model fitted by maximum penalised likelihood. Objects of this class have methods for the functions print, summary, plot, residuals and predict.

All components described under **Arguments** must be included in a legitimate coxph_mpl object.

Arguments

coef	a list of length 2 containg the parameter estimates of each model part. The first list, named 'Beta', contains the vector of regression parameter estimates of length p. The second list, named 'Theta', contains the vector of the baseline hazard parameter estimates of length m.
se	a list of length 2 containg the parameter standard errors of each model part. The first list, named 'Beta', is a (p x 5) matrix indicating the standard errors of each regression parameter according to 5 inference methods. See **Details** in coxph_mpl. The second list, named 'Theta', is a (m x 5) matrix indicating the standard errors of each baseline hazard parameter according to 5 inference methods. See **Details** in coxph_mpl.
covar	a list of length 5 containg the ($p + m$ x $p + m$) covariance matrix according to 5 inference methods. See **Details** in coxph_mpl.
ploglik	a vector of length 2. The first element is the penalised log-likelihood with the final values of the coefficients. (The second element is a correction factor for the baseline hazard parameters due to the use of a centered **X** matrix in the estimation process.)
iter	a vector of length 3 indicating the number of iterations used to estimate the smoothing parameter (first value, equal to 1 when the user specified a chosen value), the Beta and Theta parameters during the entire process (second value), and Beta and Theta parameters during the last smoothing parameter iteration (third value).
knots	list of length 3 to 4 containg parameters of the chosen basis: 'm', the number of used bases; 'Alpha', the knot sequence of length $m + 1$ for the uniform basis, and of length m otherwise; 'Delta', the value of the integral of each base over the data support (which equals 1 when basis != 'uniform'); 'Sigma', only available for the Gaussian basis, corresponds to the standard deviation of each truncated Gaussian base.
control	Object of class coxph_mpl.control specifying the basis, smoothing parameter value and other options. See coxph_mpl.control.

dim	a list of length 5 with following elements: 'n', the sample size; 'n.events', the number of events; 'n.ties', the number of duplicated observations; 'p', the number of regression parameters; and 'm', the number of baseline hazard parameters.
call	the matched call.
data	a list of length 3 with following elements: 'time', the outcome vector with an added noise applied to duplicated observation if ties == "epsilon" in coxph_mpl.control; 'observed', a logical vector indicating if outcomes are fully observed or censored; 'X', the **X** matrix corresponding to the model formula indicated in coxph_mpl.

Author(s)

Dominique-Laurent Couturier, Maurizio Manuguerra

See Also

coxph_mpl, summary.coxph_mpl, coef.coxph_mpl, plot.coxph_mpl, residuals.coxph_mpl and predict.coxph_mpl.

---

plot.coxph_mpl *Plot a coxph_mpl Object*

Description

Plot the bases used to estimate the baseline hazard parameters, as well as the estimate and confidence interval of the baseline hazard, cumulative baseline hazard and baseline survival functions (plots are selectable by which).

Usage

```
## S3 method for class 'coxph_mpl'
plot(x, se="M2QM2", ask=TRUE, which=1:4, upper.quantile=.95,...)
```

Arguments

x	an object inheriting from class coxph_mpl, representing a fitted Cox proportional hazard model.
se	an inference method (to build confidence intevals for the baseline hazard, cumulative baseline hazard and baseline survival functions). Possibilites are "H", "M2QM2" and "M2HM2". Refer to the **Details** Section of coxph_mpl. Default is se="M2QM2".
ask	logical. If TRUE, the user is asked to hit the enter keyboard before each plot. See par(ask=.). Default is ask=TRUE.
which	integer vector indicating the list of wished plots. If a subset of the plots is required, specify a subset of the numbers 1:4. By default, all plots are provided.

upper.quantile	quantile of the model response defining the upper limit of the x-axis of the plots of the baseline hazard, cumulative baseline hazard and baseline survival functions. Default is upper.quantile=.95.
...	other parameters to be passed through to plotting functions.

Details

In the first plot, the bases corresponding to zero (or close to zero) estimates appear in dashed line. An estimate is considered as a zero if it is smaller than min.Theta (See coxph_mpl.control).

Confidence intervals for the baseline hazard, cumulative baseline hazard and baseline survival functions are obtained using the delta method.

Author(s)

Dominique-Laurent Couturier, Maurizio Manuguerra

See Also

coxph_mpl, coxph_mpl.control, coxph_mpl.object and summary.coxph_mpl.

Examples

```
## Not run:
data(lung)
fit_mpl <- coxph_mpl(Surv(time, status == 2) ~ age + sex + ph.karno + wt.loss, data = lung)

par(mfrow=c(2,2))
plot(fit_mpl, ask=FALSE, cex.main=.75)

## End(Not run)
```

---

predict.coxph_mpl *Predictions for a Cox model*

Description

Compute (and plot) predictions of instantaneous risk and of probability of survival for a model fitted by coxph_mpl.

Usage

```
## S3 method for class 'coxph_mpl'
predict(object,se="M2QM2",type="risk",i=NULL,time=NULL,upper.quantile=.95,...)

## S3 method for class 'predict.coxph_mpl'
plot(x,...)
```

Arguments

object	an object inheriting from class coxph_mpl, representing a fitted Cox proportional hazard model.
se	a character string indicating a method to build confidence intevals for the predictions. Possibilites are "H", "M2QM2" and "M2HM2". Refer to the **Details** Section of coxph_mpl. Default is se="M2QM2".
type	character string indicating the type of wished predictions. Possibilies are "risk", for predicted instantaneous risk at time t (see argument "time"); and "survival", for predicted probability of survival above time t. Default is type="risk".
i	an integer indicating the covariate vector of interest (i.e., line of the **X** matrix). If i==NULL, the mean of each covariate will be used. Default is i=NULL.
time	a double-precision vector indicating at which time the preditions should be computed. If time==NULL, predictions are computed for 1000 equally spaced times in the range of the outcomes. Default is time=NULL.
upper.quantile	quantile of the model response defining the upper limit of the x-axis of the plot of the predictions. This argument is passed through to plot.predict.coxph_mpl. Default is upper.quantile=.95.
x	an object inheriting from class predict.coxph_mpl, representing the prediction of a fitted Cox proportional hazard model.
...	other parameters to be passed through to printing or plotting functions.

Details

The available predictions incorporate the baseline hazard (instantaneous risk) or cumulated baseline hazard estimate (survival function) and are thus absolute instead of relative (see predict.coxph).

Prediction standard errors and confidence intervals are obtained by use of the delta method.

In the plots, the confidence intervals are forced to belong to the parameter range, which is $[0, inf[$ for instantaneous risk, and $[0, 1]$ for survival probabilities.

Value

a data.frame of class predict.coxph_mpl with following columns: 'time', the prediction time (as defined in argument 'time'); 'risk' or 'survival', the wished predictions; 'se', the standard error of each prediction; 'lower' and 'upper', the lower and upper bound of the prediction confidence interval.

Author(s)

Dominique-Laurent Couturier, Maurizio Manuguerra

See Also

coxph_mpl, coxph_mpl.control, coxph_mpl.object, residuals.coxph_mpl and summary.coxph_mpl.

Examples

```
## Not run:
data(lung)
fit_mpl <- coxph_mpl(Surv(time, status == 2) ~ age + sex + ph.karno + wt.loss, data = lung)

plot(predict(fit_mpl))

## End(Not run)
```

`residuals.coxph_mpl` *Residuals for a Cox model*

Description

Compute martingale and Cox and Snell residuals for a model fitted by `coxph_mpl`. Return objects are of class `residuals.coxph_mpl` and have methods for `plot`.

Usage

```
## S3 method for class 'coxph_mpl'
residuals(object, ...)

## S3 method for class 'residuals.coxph_mpl'
plot(x, ask=TRUE, which=1:2, upper.quantile=.95, ...)
```

Arguments

`object`	an object inheriting from class `coxph_mpl`, representing a fitted Cox proportional hazard model.
`x`	an object inheriting from class `residuals.coxph_mpl`, representing the residuals of a Cox proportional hazard model fit with `coxph_mpl`.
`ask`	logical. If `TRUE`, the user is asked to hit the `enter keyboard` before each plot. See `par(ask=.)`. Default is `ask=TRUE`.
`which`	integer vector indicating the list of wished plots. If a subset of the plots is required, specify a subset of the numbers `1:2`. See **Details**. By default, all plots are provided.
`upper.quantile`	quantile of the Cox and Snell residuals used when `which==3`. Default is `upper.quantile=.95`.
`...`	other parameters to be passed through to plotting or printing functions.

Details

Refer to Collet (2003, Chapter 4) for a review of model check in the Cox regression model, and specifically to Farrington (2000) for an overview on residuals with interval-censored survival data.

For object of class `residuals.coxph_mpl`, the available residual plots are, respectively, the martingale residual plot (`which==1`) and the Cox and Snell residual plot (`which==2`).

Value

A data.frame of class `residuals.coxph_mpl` of n rows with following columns: `'time1'`, the model outcome (with a random noise added to event ties if `ties=='epsilon'` in `coxph_mpl.control`); `'time2'`, ending time of the interval for interval censored data only (unused otherwise); `'censoring'`, the status indicator as in the `Surv()` function, i.e. 0=right censored, 1=event at time, 2=left censored, 3=interval censored; `'coxsnell'`, the Cox and Snell residuals; `'martingale'`, the martingale residuals.

Author(s)

Dominique-Laurent Couturier, Maurizio Manuguerra

References

Farrington C.P. (2000), *Residuals for Proportional Hazard Models with Interval-Censored Data*, Biometrics 56, 473-482.

Collett, D. (2003), and Moeschberger, M. L. (2003), *Modelling Survival Data in Medical Research*, Chapman and All.

See Also

`coxph_mpl`, `coxph_mpl.control`, `coxph_mpl.object`, `predict.coxph_mpl` and `summary.coxph_mpl`.

Examples

```
## Not run:
### lung data of the survival package (see ?lung)
data(lung)
fit_mpl <- coxph_mpl(Surv(time, status == 2) ~ age + sex + ph.karno + wt.loss, data = lung)

par(mfrow=c(1,2))
plot(residuals(fit_mpl), which=1:2, ask=FALSE)

## End(Not run)
```

`summary.coxph_mpl` *Summarise a coxph_mpl Object*

Description

Additional information about the Cox proportional hazard model fit represented by `object` is extracted and included in the returned object, which is suitable for printing with the generic `print` function. The generic `coef` function will extract the matrix of coefficients of interest with standard errors, z-statistics and p-values. See `coef.summary.coxph_mpl`.

Only the baseline hazard parameters larger than `min.Theta` (see `coxph_mpl.control`) are reported.

Usage

```
## S3 method for class 'coxph_mpl'
summary(object, se="M2QM2", full=FALSE, ...)

## S3 method for class 'summary.coxph_mpl'
print(x, se="M2QM2", ...)
```

Arguments

object	In an object inheriting from class coxph_mpl, representing a fitted Cox proportional hazard model.
se	an inference method. Possibilites are "H", "M2QM2" and "M2HM2". Refer to the **Details** Section of coxph_mpl. Default is se="M2QM2".
full	logical. If TRUE inference for the baseline hasard parameters is provided. Default is full=FALSE.
x	an object inheriting from class summary.coxph_mpl, representing the summary of a fitted Cox proportional hazard model.
...	Other arguments passed through to printing functions.

Value

an object of class summary.coxph_mpl representing the fit and additional information.

Beta	a matrix of p rows indicating the regression parameter estimates, standard errors, z-statistics values and according p-values.
Theta	If full == TRUE, a matrix of m rows indicating the baseline hazard parameter estimates, standard errors, z-statistics values and according p-values. If full == FALSE, the baseline hazard estimates.
inf	a list of elements extracted from the object of class coxph_mpl including the number of iterations and the penalised likelihood value, for example.

Author(s)

Dominique-Laurent Couturier, Maurizio Manuguerra

See Also

coxph_mpl, coxph_mpl.control, coxph_mpl.object and plot.coxph_mpl.

Examples

```
## Not run:
data(lung)

fit_mpl <- coxph_mpl(Surv(time, status == 2) ~ age + sex + ph.karno + wt.loss, data = lung)

summary(fit_mpl, full = TRUE)
summary(fit_mpl, se = "M2HM2")
```

```
## End(Not run)
```

Index

∗ **dataset**
  bcos2, 3
∗ **package**
  survivalMPL-package, 2
∗ **survival**
  coef.coxph_mpl, 3
  coxph_mpl, 5
  coxph_mpl.control, 7
  coxph_mpl.object, 10
  plot.coxph_mpl, 11
  predict.coxph_mpl, 12
  residuals.coxph_mpl, 14
  summary.coxph_mpl, 15
  survivalMPL-package, 2
.Random.seed, *9*

bcos2, 3

coef.coxph_mpl, 3, *11*
coef.summary.coxph_mpl, *15*
coef.summary.coxph_mpl
  *(*coef.coxph_mpl*)*, 3
coxph, *6*
coxph_mpl, *4*, 5, *7*, *10–16*
coxph_mpl.control, *4–6*, 7, *10–13*, *15, 16*
coxph_mpl.object, *6*, 10, *12, 13*, *15, 16*

par, *11*, *14*
plot.coxph_mpl, *6*, 11, *11*, *16*
plot.predict.coxph_mpl, *13*
plot.predict.coxph_mpl
  *(*predict.coxph_mpl*)*, 12
plot.residuals.coxph_mpl
  *(*residuals.coxph_mpl*)*, 14
predict.coxph, *13*
predict.coxph_mpl, *11*, 12, *15*
print.coxph_mpl *(*coxph_mpl*)*, 5
print.summary.coxph_mpl
  *(*summary.coxph_mpl*)*, 15

residuals.coxph_mpl, *11*, *13*, 14, *14*

runif, *9*

summary.coxph_mpl, *4*, *6*, *11–13*, 15, *15, 16*
Surv, *3*, *5*
survivalMPL *(*survivalMPL-package*)*, 2
survivalMPL-package, 2