Package ‘marcox’

July 22, 2025

Type Package

Title Marginal Hazard Ratio Estimation in Clustered Failure Time Data

Version 1.0.0

Description Estimation of marginal hazard ratios in clustered failure time data. It implements the weighted generalized estimating equation approach based on a semiparametric marginal proportional hazards model (See Niu, Y. Peng, Y.(2015). ``A new estimating equation approach for marginal hazard ratio estimation"), accounting for within-cluster correlations. 5 different correlation structures are supported. The package is designed for researchers in biostatistics and epidemiology who require accurate and efficient estimation methods for survival analysis in clustered data settings.

Depends R (>= 4.4.0), Matrix

Imports Rcpp, RcppEigen, survival, ggplot2, stats

LinkingTo Rcpp, RcppEigen

Encoding UTF-8

RoxygenNote 7.3.2

LazyData true

Maintainer Junyi Chen <`2655088079@qq.com`>

License GPL-3

NeedsCompilation yes

Author Junyi Chen [aut, cre],
Siqi Zhou [ctb],
Shida Li [ctb],
Yi Niu [aut]

Repository CRAN

Date/Publication 2025-04-10 14:20:02 UTC

Contents

`diabetes` *Diabetes Study Data*

Description

A dataset containing clinical information from a diabetes study.

Usage

`data(diabetes)`

Format

A data frame with 166 rows and 6 variables:

`risk` Numeric: Risk score of the patient.

`cens` Binary (0/1): Censoring indicator (1 = event occurred, 0 = censored).

`time` Numeric: Time to event or censoring (in months).

`id` Integer: Patient ID.

`trt` Binary (0/1): Treatment indicator (1 = treated, 0 = control).

`age` Binary (0/1): Age group indicator (1 = older, 0 = younger).

Source

Hypothetical clinical study data.

Examples

```
data(diabetes)
summary(diabetes)
```

`gendat` *Generate Simulated Datasets for Cox Proportional Hazards Model*

Description

This function generates multiple datasets for survival analysis based on a Cox proportional hazards model. The baseline hazard function follows either a Weibull or an exponential distribution, depending on the values of `lambda`. The function ensures that the maximum observed time in both the control and treatment groups is checked for censoring. If the maximum time is not censored, it is forced to be censored to maintain the desired censoring rate.

Usage

```
gendat(
  type = "bin",
  dimension = 10,
  K = 30,
  n = 2,
  lambda = c(1, 2),
  b1 = c(log(2), -0.1),
  theta = 8,
  censrate = 0.3
)
```

Arguments

`type`	Character. If `type = 'bin'`, the covariates are generated as binary variables; if `type = 'cont'` continuous covariates are generated.
`dimension`	Integer. The number of datasets to be generated.
`K`	Integer. The number of clusters (groups) within each dataset.
`n`	Integer. The number of samples within each cluster.
`lambda`	Numeric vector. A two-element vector specifying the parameters for the baseline distribution: • If `lambda = c(a, b)`, where `a > 1`, the baseline follows a Weibull distribution. • If `lambda = c(1, b)`, the baseline follows an exponential distribution.
`b1`	Vector. The regression coefficient for the covariates, affecting the hazard function. We suggest that the maximum of `b1` should be lower than 2.
`theta`	Numeric. A parameter controlling the dependency structure between survival times within clusters. Higher values indicate stronger within-cluster correlation.
`censrate`	Numeric. The target censoring rate for the dataset.

Value

A list containing:

- `data` - A list of data frames, each containing a generated dataset.
- `censoringrates` - A numeric vector representing the censoring rate for each dataset.
- `mean(censoringrates)` - The mean censoring rate across all datasets.

Examples

```
# Generate binary covariate datasets with 1 datasets, 10 clusters, and 6 samples per cluster
print(gendat(type = 'bin', dimension = 1, K = 6, n = 10, lambda = c(1, 2),
      b1 = c(log(2),-log(2)), theta = 8, censrate = 0.5))
```

`kidney_data` *Kidney Disease Study Data*

Description

A dataset containing survival analysis information related to kidney disease patients.

Usage

```
data(kidney_data)
```

Format

A data frame with 76 rows and 5 variables:

`time` Numeric: Time to event or censoring (in days).

`cens` Binary (0/1): Censoring indicator (1 = event occurred, 0 = censored).

`age` Numeric: Age of the patient in years.

`sex` Binary (0/1): Sex of the patient (1 = male, 0 = female).

`type` Categorical (0,1,2,3): Kidney disease type classification.

Source

Hypothetical survival study data.

Examples

```
data(kidney_data)
summary(kidney_data)
```

`marcox` *Analysis for Cox Proportional Hazards Models*

Description

This function performs marcox analysis for Cox proportional hazards models, incorporating clustered data and handling time-dependent covariates. It estimates coefficients, standard errors, and p-values based on the specified formula and dataset.

Usage

```
marcox(
  formula,
  data,
  method = "exchangeable",
  sep = NULL,
  col_id = "id",
  div = NULL,
  k_value = 1,
  plot_x = NULL,
  x_axis = "Time",
  y_axis = "Survival Rates",
  size = 0.5
)
```

Arguments

`formula`	A model formula that uses the `Surv()` function to define the survival outcome. It should include both continuous and categorical covariates, where categorical variables must be specified using the `factormar()` function.
`data`	The file path or the dataset(matrix) to be analyzed. If a file path is provided, the file will be loaded into a matrix. The file should be in a tabular format (e.g., .csv, .txt).
`method`	The method employed to solve the correlation coefficient: • Exchangeable correlation structure: `method = 'exchangeable'` • Autoregressive(AR-1): `method = 'ar1'` • K-dependent: `method = 'kdependent'` • Toeplitz: `method = 'toeplitz'` • Independent: `method = 'independent'`
`sep`	Character. The `sep` parameter specifies the character that separates the fields in each line of the file. For instance, for a comma-separated file, set `sep = ","`, and for a tab-separated file, set `sep = "\t"`.
`col_id`	Character. The name of column that identifies the clusters.
`div`	Integer. The number of observation points per sample. If provided, the data will be divided accordingly. If the data has complex observational situations, please preprocess the data before using this function.
`k_value`	The k value only for k-dependent structure. The default value is 1.
`plot_x`	A character string specifying the column name of the covariate for which survival curves are generated; if not provided, no survival curves will be produced.
`x_axis`	A character string specifying the title for the x-axis.
`y_axis`	A character string specifying the title for the y-axis.
`size`	The size of the generated survival curve.

Details

The `marcox()` function is specifically designed for survival data analysis using Cox proportional hazards models. It handles both clustered and time-dependent covariates effectively. The survival outcome must be defined using the `Surv()` function in the model formula, and covariates can be included directly or by converting categorical variables with the `factormar()` function.

Value

A list containing the following components:

- `coef` - The estimated regression coefficients.
- `exp(coef)` - The exponentiated coefficients (hazard ratios).
- `se(coef)` - The standard errors of the estimated coefficients.
- `z` - The z-statistics for testing the significance of the coefficients.
- `p` - The p-values associated with the coefficients.
- (hidden).correlation - Correlation coefficients of the data.

Examples

```
 formula <- Surv(time, cens) ~ sex + factormar('type', d_v=c(1,2,3))
r <- marcox(formula, data = kidney_data, div = 2, method = 'exchangeable', plot_x = 'sex')
 print(r)
 print(r$plot)
```

Index