Package: FrailtyCompRisk (via r-universe)

May 28, 2026

Title Competing Risks Models for Multi-Center Survival Data with Frailty

Version 0.1.1

Description Implements methods for analyzing competing risks data in multi-center survival studies using frailty models. The approach relies on a mixed proportional hazards model for the sub-distribution, allowing for cluster-specific random effects. The package provides tools for model estimation with or without frailty using Maximum Likelihood (ML) and Restricted Maximum Likelihood (REML). It supports flexible modeling of between-center heterogeneity and is particularly suited for multi-center clinical trials or registries. Core features include data simulation, likelihood computation, cluster-dependent censoring options, and testing of frailty effects. For methodological details, see Katsahian et al. (2006) <doi:10.1002/sim.2684>.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 3.5.0)

Imports Matrix, stats

Suggests dplyr, tibble, kableExtra, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

URL https://github.com/TeamHeKA/FrailtyCompRisk

BugReports https://github.com/TeamHeKA/FrailtyCompRisk/issues

Config/testthat/edition 3

Repository https://teamheka.r-universe.dev

Date/Publication 2026-04-23 11:38:45 UTC

RemoteUrl https://github.com/teamheka/frailtycomprisk

RemoteRef HEAD

RemoteSha 1ecb40de29151eba3d6364a24ccf39adb86821a1

Contents

`check_data_format` *Check the Format of a Data Frame for Multicentre Competing Risks with Frailty*

Description

Validates whether the submitted data frame is correctly structured for the analysis of multicentre competing risks with frailty.

Usage

```
check_data_format(df)
```

Arguments

`df`	A data frame to be checked.

Value

`TRUE` if `df` has the expected structure. Otherwise, the function stops with an error describing the formatting issue.

Examples

```
n_cov = 1
n_cluster = 5
n_per_cluster = 100
n = n_cluster * n_per_cluster
a1 = 0
b1 = 1
G <- rep(1:n_cluster, each = n_per_cluster)
Z <- matrix(runif(n * n_cov, a1, b1), ncol = n_cov)
df = simulate_data(
  G,
  Z,
  prop = 0.6,
  beta = c(0),
  theta = 0.01,
  cens = TRUE,
  pcens = 0.25,
  tau = 0
)
check_data_format(df)
```

---

compute_M_optimized *Compute the Weight Matrix M (Optimized)*

Description

This function computes the matrix of inverse probability of censoring weights (IPCW), used in multicenter competing risks models with frailty. It supports both Kaplan-Meier and Nelson-Aalen estimators for the censoring distribution.

Usage

```
compute_M_optimized(times, status, f, u_bis, clusters)
```

Arguments

`times`	A numeric vector of observed times (events or censoring).
`status`	A numeric vector of status indicators (0 = censored, 1 or 2 = cause of failure).
`f`	A string, either `"KaplanMeier_Censoring_vectorized"` or `"Nelson_Censoring_vectorized"`, specifying the method to estimate the censoring distribution.
`u_bis`	A numeric vector of cluster-level random effects (used only if `f == "Nelson_Censoring_vectorized"`).
`clusters`	A vector indicating cluster membership for each subject (same length as `times`).

Details

- If `f == "KaplanMeier_Censoring_vectorized"`, then `u_bis` is ignored.
- The weight matrix accounts for pairwise contributions, depending on event times and causes.

Value

A square matrix `M` of size N x N, where `M[i, j]` is the IPCW weight between subject i and subject j.

See Also

`KaplanMeier_Censoring_vectorized()`, `Nelson_Censoring_vectorized()`, `create_G_function()`

`create_G_function` *Create Step Function for Estimated Survival (G)*

Description

This function creates a right-continuous step function (vectorized) based on estimated values of the censoring survival function at given unique times.

Usage

```
create_G_function(unique_times, G_vals)
```

Arguments

`unique_times`	A numeric vector of increasing unique times (e.g., censoring times).
`G_vals`	A numeric vector of the same length as `unique_times`, corresponding to estimated values of the survival function at each time.

Details

The resulting function can be used to evaluate $G(t)$ for any vector of time points.

Value

A function `G(t)` that can be evaluated on a numeric vector of time points.

KaplanMeier_Censoring_vectorized *Kaplan-Meier Estimator for Censoring Distribution (Vectorized)*

Description

Computes the Kaplan-Meier estimator of the censoring distribution at given time points.

Usage

```
KaplanMeier_Censoring_vectorized(times, status, unique_times)
```

Arguments

`times`	A numeric vector of observed times.
`status`	A numeric vector of status indicators (0 = censored, >0 = event).
`unique_times`	A numeric vector of time points at which to evaluate the estimator.

Value

A numeric vector of estimated survival probabilities at each unique time.

logLikelihood_1 *Log-Likelihood Contribution for Cause 1*

Description

Computes the contribution to the log-likelihood from individuals who experienced event type 1, using a weighting matrix `W` and an IPCW matrix `M`.

Usage

```
logLikelihood_1(status, M, W)
```

Arguments

`status`	A numeric vector indicating the event type: `0` for censored, `1` for cause 1, `2` for other causes.
`M`	A numeric matrix (usually computed with `compute_M_optimized`), representing IPCW weights between individuals.
`W`	A sparse matrix (e.g., of class `dgCMatrix` from the **Matrix** package) representing estimated weights or hazards.

Details

For each individual `i` such that `status[i] == 1`, the function computes:

$$\log W_{ii} - \log \left(\sum_j M_{ij} W_{ij} \right)$$

and sums this across all such individuals.

Value

A numeric scalar: the log-likelihood contribution from all individuals who failed from cause 1.

See Also

`compute_M_optimized()`, Matrix::Matrix

`logLikelihood_2` *Compute Log-Likelihood Penalty for Random Effects*

Description

This function computes the second component of the log-likelihood (penalty term) associated with the random effects (frailty terms) in a mixed-effects model.

Usage

```
logLikelihood_2(u, theta, K)
```

Arguments

`u`	A numeric vector of length `K` containing the random effects (frailties).
`theta`	A positive numeric scalar, representing the variance component of the random effects.
`K`	An integer representing the number of clusters (i.e., the length of `u`).

Details

The penalty is derived from the Gaussian assumption on the random effects and involves both the dimension of the random effects vector and the variance parameter `theta`.

Value

A numeric scalar representing the log-likelihood penalty contribution from the random effects.

`Ml_CompRisk` *Maximum Likelihood Estimation for Cause 1 in Competing Risks*

Description

Performs maximum likelihood estimation (MLE) for the effect of covariates on the cause-specific hazard function of cause 1 in a competing risks framework.

Usage

```
Ml_CompRisk(data, max_iter = 100, tol = 1e-06)
```

Arguments

`data`	A data frame containing:
	times Observed survival or censoring times.
	status Event indicator: 0 = censored, 1 = cause 1 (event of interest), 2+ = other causes.
	Covariates All columns from the 4th onward are considered as covariates.
`max_iter`	Maximum number of Newton-Raphson iterations (default = 100).
`tol`	Convergence tolerance for the change in parameter estimates (default = 1e-6).

Details

The function fits a Cox-type model for the subdistribution hazard of cause 1, treating all other causes (including censoring) as censored (i.e., status != 1 becomes 0). It uses a Newton-Raphson procedure to maximize the partial likelihood.

The design matrix is reordered internally by event time to facilitate risk set computation.

Value

A named list with one element:

beta A numeric vector of estimated regression coefficients for the subdistribution hazard of cause 1. Each coefficient represents the log effect of the corresponding covariate on the subdistribution hazard.

See Also

`Reml_Cox_frailty`, `Ml_Cox`, `logLikelihood_1`

Examples

```
data <- data.frame(
 times = c(0.099, 0.006, 0.260, 0.151, 0.262,
           0.019, 0.026, 0.241, 0.017, 0.195,
           0.007, 0.057, 0.241, 0.132, 0.044,
           0.242, 0.416, 0.096, 0.172, 0.015),
 status = c(1, 0, 1, 1, 0,
            0, 0, 0, 0, 0,
            0, 0, 0, 0, 0,
            0, 0, 0, 0, 0),
 clusters = c(1,1,1,1,1,
              1,1,1,1,1,
             2,2,2,2,2,
              2,2,2,2,2),
 V1 = c(0.720, 0.289, 0.382, 0.995, 0.045,
        0.048, 0.042, 0.810, 0.964, 0.781,
        0.443, 0.303, 0.902, 0.596, 0.462,
        0.521, 0.656, 0.164, 0.261, 0.635)
)
  beta_hat <- Ml_CompRisk(data)
```

`Ml_Cox` *Maximum Likelihood Estimation for the Cox Model (Optimized)*

Description

This function computes the maximum likelihood estimates of the regression coefficients in the Cox proportional hazards model using Newton-Raphson iterations.

Usage

```
Ml_Cox(data, max_iter = 100, tol = 1e-06)
```

Arguments

`data`	A data frame with columns: `times`, `status`, and covariates starting from the 4th column.
`max_iter`	Maximum number of iterations (default = 100).
`tol`	Convergence tolerance for the Euclidean norm (default = 1e-6).

Value

A named list with one element:

beta A numeric vector of estimated regression coefficients in the Cox proportional hazards model. Each coefficient represents the log hazard ratio associated with the corresponding covariate.

Nelson_Censoring_vectorized *Nelson-Aalen Estimator of Censoring Distribution (Vectorized)*

Description

Computes the estimated censoring survival probabilities using a Nelson-Aalen-type estimator, accounting for shared frailty through a cluster-specific random effect.

Usage

```
Nelson_Censoring_vectorized(times, unique_times, status, clusters, u_bis)
```

Arguments

`times`	A numeric vector of observed times.
`unique_times`	A numeric vector of distinct, sorted time points.
`status`	A numeric vector where 0 indicates censoring.
`clusters`	A vector indicating the cluster/group for each observation.
`u_bis`	A numeric vector of cluster-specific random effects (e.g., frailty values).

Value

A numeric vector of estimated censoring survival probabilities for each individual.

Parameters_estimation *Unified Interface for Parameter Estimation in (Competing Risks) Frailty Models*

Description

Provides a unified wrapper to estimate model parameters under four different modeling strategies:

- Competing risks with shared frailty for cause 1 (`"CompRisk_frailty"`)
- Competing risks without frailty for cause 1 (`"CompRisk"`)
- Standard Cox model with shared frailty (`"Cox_frailty"`)
- Standard Cox model without frailty (`"Cox"`)

Usage

```
Parameters_estimation(
  data,
  method = "CompRisk_frailty",
  cluster_censoring = FALSE,
  max_iter = 300,
  tol = 1e-06,
  threshold = 1e-06
)
```

Arguments

`data`	A data frame containing at least the following columns: **times** Event or censoring times. **status** Event indicator: 0 = censored, 1 = event of interest, >1 = other types (for competing risks). **clusters** Cluster/group variable (e.g., center ID). **covariates** (Optional) Additional covariates from the 4th column onward.
`method`	Character string indicating the estimation method. One of: • `"CompRisk_frailty"` (default): Competing risks model for cause 1 with shared frailty. • `"CompRisk"`: Competing risks model for cause 1 without frailty. • `"Cox_frailty"`: Cox proportional hazards model with shared frailty. • `"Cox"`: Standard Cox proportional hazards model.
`cluster_censoring`	Logical. If `TRUE`, adjusts for cluster-specific censoring. Only applicable when `method = "CompRisk_frailty"` (default = `FALSE`).
`max_iter`	Maximum number of iterations for the selected model estimation (default = 300).
`tol`	Convergence tolerance (default = 1e-6).
`threshold`	Lower bound for the frailty variance parameter θ. If the estimated value falls below this threshold, frailty is considered negligible (default = 1e-5).

Details

This wrapper allows seamless switching between various modeling frameworks. It automatically calls:

- `Reml_CompRisk_frailty` for `"CompRisk_frailty"`
- `Ml_CompRisk` for `"CompRisk"`
- `Reml_Cox_frailty` for `"Cox_frailty"`
- `Ml_Cox` for `"Cox"`

Data format is validated via the helper function `check_data_format`. An error is raised if input format is invalid or if `cluster_censoring` is used with an incompatible method.

Value

A named list whose structure depends on `method`. For `"CompRisk"` and `"Cox"`, the returned list contains:

beta Numeric vector of estimated regression coefficients.

For `"CompRisk_frailty"` and `"Cox_frailty"`, the returned list may contain:

beta Numeric vector of estimated fixed-effect coefficients.

u Numeric vector of estimated cluster-specific frailties.

theta Estimated frailty variance parameter.

p_value P-value for testing the null hypothesis of no frailty effect.

Returns `NULL` if estimation fails.

See Also

`Reml_CompRisk_frailty`, `Ml_CompRisk`, `Reml_Cox_frailty`, `Ml_Cox`, `check_data_format`

Examples

```
data <- data.frame(
  times = c(4, 6, 10, 12, 3),
  status = c(1, 0, 2, 1, 0),
  clusters = c(1, 1, 2, 2, 3),
  x1 = rnorm(5),
  x2 = sample(0:1, 5, replace = TRUE)
)
result <- Parameters_estimation(data, method = "CompRisk_frailty")
result$beta
```

Reml_CompRisk_frailty *REML Estimation for Cause 1 in a Competing Risks Model with Shared Frailty*

Description

Fits a cause-specific hazard model for cause 1 in a competing risks framework using restricted maximum likelihood (REML) with a shared frailty term for clusters.

Usage

```
Reml_CompRisk_frailty(
  data,
  cluster_censoring = FALSE,
  max_iter = 300,
  tol = 1e-06,
  threshold = 1e-05
)
```

Arguments

`data`	A data frame with at least the following columns: **times** Observed times to event or censoring. **status** Event type: 0 = censored, 1 = cause of interest, >1 = other competing risks. **clusters** Cluster membership (e.g., centers or subjects). **covariates** (Optional) Columns from the 4th onward are used as covariates.
`cluster_censoring`	Logical. If `TRUE`, accounts for center-specific censoring using a frailty model on censoring times. Default is `FALSE`.
`max_iter`	Maximum number of iterations for the Newton-Raphson algorithm (default = 300).
`tol`	Convergence tolerance for the likelihood and frailty variance (default = 1e-6).
`threshold`	Lower bound for the frailty variance parameter θ. If the estimated value falls below this threshold, frailty is considered negligible (default = 1e-5).

Details

The function fits a proportional hazards model for cause 1, accounting for unobserved heterogeneity via a shared frailty term on clusters. When `cluster_censoring = TRUE`, a frailty-adjusted survival curve is used to correct for informative censoring using either a Kaplan-Meier or Nelson-Aalen-based estimator.

If the estimated frailty variance θ becomes smaller than the provided `threshold`, the model reverts to a standard Cox model for cause 1 using `Ml_CompRisk`.

Value

A list with:

beta Estimated regression coefficients for the cause 1 hazard.

u Estimated cluster-specific random effects (frailties).

theta Estimated frailty variance.

p_value P-value testing the null hypothesis $H_0 : \theta = 0$.

See Also

`Ml_CompRisk`, `Reml_Cox_frailty`, `logLikelihood_1`, `compute_M_optimized`

Examples

```
data <- data.frame(
  times = c(4, 6, 10, 12, 3),
  status = c(1, 0, 2, 1, 0),
  clusters = c(1, 1, 2, 2, 3),
  x1 = rnorm(5),
  x2 = sample(0:1, 5, replace = TRUE)
)
```

```
result <- Reml_CompRisk_frailty(data)
result$beta
result$p_value
```

`Reml_Cox_frailty` *Cox Model with Random Frailty (REML Estimation)*

Description

Estimates the regression coefficients, random effects, and frailty variance θ in a Cox proportional hazards model with shared frailty using Restricted Maximum Likelihood (REML).

Usage

```
Reml_Cox_frailty(data, max_iter = 300, tol = 1e-06, threshold = 1e-05)
```

Arguments

`data`	A data frame containing: • `times`: observed survival or censoring times, • `status`: event indicator (1 = event, 0 = censored), • `clusters`: cluster identifiers (e.g., center, group), • covariates from column 4 onward.
`max_iter`	Maximum number of Newton-Raphson iterations (default = 300).
`tol`	Tolerance for convergence (default = 1e-6).
`threshold`	Lower bound for the frailty variance parameter θ. If the estimated value falls below this threshold, frailty is considered negligible (default = 1e-5).

Details

The function uses a REML-based Newton-Raphson approach to estimate the parameters. A penalized likelihood including a log-likelihood contribution for the frailty variance θ is maximized.

The hypothesis test of $H_0 : \theta = 0$ is based on a Wald-type statistic. A high p-value (e.g. > 0.05) may indicate that the random effects (frailties) are negligible.

Value

A list with the following elements:

beta Estimated fixed effects (regression coefficients).

u Estimated random effects (frailties) for each cluster.

theta Estimated variance of the random effects.

p_value P-value for testing the null hypothesis $H_0 : \theta = 0$ (no frailty).

See Also

Ml_Cox, logLikelihood_1, logLikelihood_2

---

simulate_data *Simulate clustered competing risks data with shared frailty*

Description

This function simulates clustered time-to-event data under a competing risks framework with shared frailty and the possibility of random censoring. Cause 1 is modeled through an inversion method of the subdistribution function, and cause 2 is introduced when inversion fails.

Usage

```
simulate_data(
  G,
  Z = NULL,
  prop = 0.6,
  beta = NULL,
  theta = 0.5,
  cens = FALSE,
  pcens = 0.25,
  tau = 0.5
)
```

Arguments

G	A vector of group or cluster identifiers (length N). Each value indicates which cluster the individual belongs to.
Z	A matrix of covariates (dimensions N x p). Can be set to NULL if no covariates are used.
prop	Proportion of individuals susceptible to cause 1 (default: 0.6).
beta	A numeric vector of regression coefficients, one per covariate (length p).
theta	Variance of the shared frailty term for event times (cause 1).
cens	Logical, indicating whether censoring should be simulated (default: FALSE).
pcens	Target censoring proportion (default: 0.25).
tau	Variance of the shared frailty term for censoring times (default: 0).

Value

A data.frame with one row per individual and at least the columns:

times Observed follow-up time.

status Event indicator: 0 for censoring, 1 for cause 1, 2 for cause 2.

clusters Cluster identifier.

If covariates are simulated, additional numeric covariate columns are included.

Examples

```
n_cov <- 1
n_cluster <- 5
n_per_cluster <- 100
n <- n_cluster * n_per_cluster

G <- rep(1:n_cluster, each = n_per_cluster)
Z <- matrix(runif(n * n_cov), ncol = n_cov)

df <- simulate_data(
  G = G,
  Z = Z,
  prop = 0.6,
  beta = c(0.5),
  theta = 0.01,
  cens = TRUE,
  pcens = 0.25,
  tau = 0.01
)

head(df)
table(df$status)
```

Index