Package: wnpmle (via r-universe)

June 18, 2026

Title Weighted NPMLE for Recurrent Events with a Competing Terminal Event

Version 0.1.2

Description Provides regression modeling and prediction for the marginal mean of recurrent events in the presence of a competing terminal event using the weighted nonparametric maximum likelihood estimator (wNPMLE) of Bellach and Kosorok (2026) <doi:10.48550/arXiv.2605.25934>. Two classes of transformation models are implemented: Box-Cox transformation models and logarithmic transformation models. These extend the proportional means model of Ghosh and Lin (2002) <doi:10.17615/pt0g-y207> and the transformation model framework of Zeng and Lin (2006) <doi:10.1093/biomet/93.3.627>. Parameter estimation is performed using automatic differentiation through the Template Model Builder (TMB) framework. Standard errors are computed using sandwich variance estimators that account for estimation of the inverse-probability censoring weights following Bellach, Kosorok, Rüschendorf and Fine (2019) <doi:10.1080/01621459.2017.1401540>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports TMB (>= 1.9.0), survival, methods, MASS, graphics, grDevices

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

URL https://github.com/abellach/wnpmle

BugReports https://github.com/abellach/wnpmle/issues

NeedsCompilation no

Author Anna Bellach [aut, cre]

Maintainer Anna Bellach <abellach.biostat@gmail.com>
Repository https://cran.r-universe.dev
Date/Publication 2026-06-18 17:10:02 UTC
RemoteUrl https://github.com/cran/wnpmle
RemoteRef HEAD
RemoteSha c493d08e09659e5a7857bfd9e646e1bb08df740a

Contents

`AIC.wnpmle` *AIC for wnpmle objects*

Description

AIC for wnpmle objects

Usage

```
## S3 method for class 'wnpmle'
AIC(object, ..., k = 2)
```

Arguments

`object`	A `wnpmle` object.
`...`	Ignored.
`k`	Penalty per parameter (default 2 for AIC).

Value

A numeric scalar giving the AIC value.

`baseline` *Extract the estimated baseline mean function*

Description

Returns a data frame with the estimated cumulative baseline mean function Lambda(t) and its increments lambda(t), together with standard errors and pointwise 95% confidence intervals.

Usage

```
baseline(object, conf_level = 0.95, ...)
```

Arguments

`object`	A `wnpmle` object.
`conf_level`	Confidence level for the pointwise intervals (default 0.95).
`...`	Ignored.

Value

A data frame with columns:

`time`	Recurrent event times.
`lambda`	Estimated baseline increments.
`Lambda`	Estimated cumulative baseline mean.
`se_Lambda`	Standard error of Lambda (if SE was estimated).
`lower`	Lower confidence band for Lambda.
`upper`	Upper confidence band for Lambda.

Examples

```
bdata <- bladder_prep()
bdata_clean <- bdata[, c("id", "time", "status", "treat", "num", "size")]
fit <- wnpmle_fit(Surv(time, status) ~ treat + num + size,
                  data = bdata_clean, id = "id", model = "log", rho = 1)
bl <- baseline(fit)
head(bl)
```

BIC.wnpmle *BIC for wnpmle objects*

Description

BIC for wnpmle objects

Usage

```
## S3 method for class 'wnpmle'
BIC(object, ...)
```

Arguments

object	A wnpmle object.
...	Ignored.

Value

A numeric scalar giving the BIC value.

bladder_prep *Prepare bladder cancer data for wnpmle analysis*

Description

Prepares the bladder1 dataset from the **survival** package for use with wnpmle_fit. Subjects randomised to pyridoxine are excluded (only placebo and thiotepa arms are retained). Status codes are recoded to 0 (censored), 1 (recurrent tumour), 2 (terminal/dropout), and event times beyond tau are truncated to tau.

Usage

```
bladder_prep(tau = 59)
```

Arguments

tau	Follow-up truncation time (default: 59 months). Event times beyond tau are set to tau.

Details

The bladder cancer trial (Veterans Administration Cooperative Urological Research Group) randomised patients to placebo, pyridoxine, or thiotepa. Only the placebo and thiotepa arms are used here (pyridoxine arm excluded). Subjects with event times equal to tau and status 0 have their times truncated to tau, consistent with administrative censoring.

Value

A data frame with columns:

`id`	Subject identifier (integer).
`status`	Event status: 0 = censored, 1 = recurrent event, 2 = terminal event.
`status0`	Indicator: 1 if censored.
`status1`	Indicator: 1 if recurrent event.
`status2`	Indicator: 1 if terminal event.
`time`	Event time (months).
`treat`	Treatment indicator: 1 = thiotepa, 0 = placebo.
`num`	Initial number of tumours.
`size`	Initial tumour size (cm).

References

Byar, D.P. (1980). The Veterans Administration study of chemoprophylaxis for recurrent stage I bladder tumors. In *Bladder Tumors and Other Topics in Urological Oncology*, 363-370. Plenum, New York.

Examples

```
bdata <- bladder_prep(tau = 59)
head(bdata)
table(bdata$status)
```

`coef.wnpmle` *Extract coefficients from a wnpmle object*

Description

Extract coefficients from a wnpmle object

Usage

```
## S3 method for class 'wnpmle'
coef(object, ...)
```

Arguments

`object`	A `wnpmle` object.
`...`	Ignored.

Value

A named numeric vector of regression coefficients.

logLik.wnpmle *Log-likelihood for wnpmle objects*

Description

Log-likelihood for wnpmle objects

Usage

```
## S3 method for class 'wnpmle'
logLik(object, ...)
```

Arguments

object	A wnpmle object.
...	Ignored.

Value

An object of class "logLik" with the log-likelihood value, degrees of freedom (df) equal to the number of regression coefficients, and number of observations (nobs).

plot.wnpmle *Plot method for wnpmle objects*

Description

Plots the estimated cumulative baseline mean function Lambda(t) with optional pointwise confidence bands.

Usage

```
## S3 method for class 'wnpmle'
plot(x, conf_bands = TRUE, ...)
```

Arguments

x	A wnpmle object.
conf_bands	Logical; if TRUE (default), adds pointwise 95% confidence bands when available.
...	Additional graphical parameters passed to plot.

Value

No return value, called for side effects (produces a plot).

plot_loglik *Log-likelihood profile plot for the transformation parameter*

Description

Fits the model over a fine grid of transformation parameter values and plots the profile log-likelihood for both the Box-Cox and logarithmic transformation models on a single plot. The log transformation parameter r is shown on the left (negative axis) and the Box-Cox parameter rho on the right (positive axis), meeting at zero where both models coincide.

Usage

```
plot_loglik(
  formula,
  data,
  id = "id",
  rho_grid = seq(0.01, 1.2, by = 0.01),
  r_grid = seq(0.01, 1.2, by = 0.01),
  tau = NULL,
  mark_points = TRUE,
  file = NULL,
  verbose = TRUE,
  ...
)
```

Arguments

formula	A formula as passed to wnpmle_fit.
data	A data frame.
id	Name of the subject identifier column.
rho_grid	A numeric vector of rho values for the Box-Cox model (default: seq(0.01, 1.2, by = 0.01)).
r_grid	A numeric vector of r values for the log model (default: seq(0.01, 1.2, by = 0.01)).
tau	Optional truncation time. If NULL (default), uses the maximum observed time in the data.
mark_points	Logical; if TRUE (default), marks reference points at rho=1 (filled circle, Ghosh-Lin) and r=1 (open circle, proportional odds model).
file	Optional path to save the plot as a PDF (e.g. "loglik_profile.pdf"). If NULL (default), plots to the current device.
verbose	Logical; print progress (default TRUE).
...	Additional arguments passed to wnpmle_fit.

Value

A data frame with columns `model`, `param` and `loglik`, invisibly.

Examples

```
bdata <- bladder_prep()
bdata_clean <- bdata[, c("id", "time", "status", "treat", "num", "size")]
plot_loglik(Surv(time, status) ~ treat + num + size,
            data = bdata_clean, id = "id")
```

`predict.wnpmle` *Predict marginal mean for new covariate values*

Description

Predict marginal mean for new covariate values

Usage

```
## S3 method for class 'wnpmle'
predict(object, newdata = NULL, times = NULL, ...)
```

Arguments

`object`	A `wnpmle` object.
`newdata`	A data frame with the same covariates used in fitting. If `NULL`, returns the estimated Lambda(t) for the baseline (all covariates = 0).
`times`	Time points at which to evaluate the marginal mean. If `NULL`, uses the observed recurrent event times.
`...`	Ignored.

Value

A data frame with columns `time` and one column per row of `newdata` (or a single column `mu` for baseline).

print.wnpmle *Print method for wnpmle objects*

Description

Print method for wnpmle objects

Usage

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

Arguments

x	A wnpmle object.
...	Ignored.

Value

Invisibly returns x.

summary.wnpmle *Summary method for wnpmle objects*

Description

Summary method for wnpmle objects

Usage

```
## S3 method for class 'wnpmle'
summary(object, tau_grid = TRUE, ...)
```

Arguments

object	A wnpmle object.
tau_grid	Logical; if TRUE (default), also show Lambda at tau/4, tau/2, and tau.
...	Ignored.

Value

Invisibly returns object.

vcov.wnpmle — *Extract variance-covariance matrix from a wnpmle object*

Description

Returns the full variance-covariance matrix for (beta, Lambda). To get only the beta part, use `vcov(fit)[1:p, 1:p]`.

Usage

```
## S3 method for class 'wnpmle'
vcov(object, ...)
```

Arguments

`object`	A `wnpmle` object.
`...`	Ignored.

Value

A numeric matrix containing the variance-covariance matrix for the regression coefficients and cumulative baseline mean function. Returns an error if `se = "none"` was used in `wnpmle_fit`.

wnpmle_fit — *Fit Weighted NPMLE for Survival Data with Recurrent or Competing Events*

Description

Estimates the weighted nonparametric maximum likelihood estimator (wNPMLE) for the marginal mean function of recurrent events in the presence of a competing terminal event. Two transformation models are supported: the Box-Cox model and the logarithmic model.

Usage

```
wnpmle_fit(
  formula,
  data,
  id = "id",
  type = c("recurrent", "cmprsk", "cmprsk_ltrc"),
  model = c("boxcox", "log"),
  rho = 1,
  tau = NULL,
  se = c("sandwich_adj", "sandwich", "fisher", "none"),
  init_beta = NULL,
  control = list(),
  silent = TRUE
)
```

Arguments

formula	A formula of the form `Surv(time, status) ~ covariates`, where `status` takes value `1` for a recurrent event, `2` for the terminal event (e.g. death), and `0` for censoring.
data	A data frame containing the variables in `formula` and an `id` column identifying subjects.
id	Name of the subject identifier column in `data` (default: `"id"`).
type	Type of analysis. Currently only `"recurrent"` is supported (recurrent events with a competing terminal event). Future versions will add `"cmprsk"` (competing risks) and `"cmprsk_ltrc"` (competing risks with left truncation and right censoring).
model	Transformation model: `"boxcox"` (default) or `"log"`.
rho	Transformation parameter. For `model = "boxcox"`, this is the Box-Cox parameter rho (default 1, i.e. linear/proportional means model). For `model = "log"`, this is the parameter r in `G(x) = log(1 + r*x) / r` (default 1).
tau	Follow-up truncation time. Kaplan-Meier censoring weights are truncated at `tau`. If `NULL` (default), uses the maximum observed time.
se	Variance estimation method: `"sandwich_adj"` (default, sandwich with censoring correction), `"sandwich"` (plain sandwich), `"fisher"` (inverse Hessian), or `"none"`.
init_beta	Initial values for regression coefficients (default: all zeros).
control	A list of control parameters passed to `nlminb`.
silent	Suppress TMB output (default: `TRUE`).

Details

Analysis types: Currently only `type = "recurrent"` is implemented, for the recurrent events with competing terminal event setting. Support for `"cmprsk"` (Bellach, Kosorok, Ruschendorf and Fine, 2019) and `"cmprsk_ltrc"` (left truncation and right censoring) will be added in future versions.

Transformation models: The two models differ in the link function G:

- **Box-Cox**: $G(x) = ((1+x)^\rho - 1)/\rho$, with $G(x) \to \log(1+x)$ as $\rho \to 0$.
- **Logarithmic**: $G(x) = \log(1 + r \cdot x)/r$, with $G(x) \to x$ as $r \to 0$.

Both are estimated via automatic differentiation using TMB. The sandwich variance estimator accounts for the estimation of the censoring weights. The censoring-corrected sandwich (`sandwich_adj`) adds an influence function correction for the Kaplan-Meier censoring weight estimation.

Value

An object of class `"wnpmle"` with components:

coefficients	Estimated regression coefficients (beta).
se	Standard errors for the regression coefficients.
Lambda	Estimated cumulative baseline mean function, evaluated at the recurrent event times.

lambda	Estimated baseline increments.
event_times	Recurrent event times at which Lambda is estimated.
loglik	Log-likelihood at the optimum.
vcov	Full variance-covariance matrix for (beta, Lambda).
model	Transformation model used.
rho	Transformation parameter used.
tau	Truncation time used.
n	Number of subjects.
n_events	Named vector: recurrent events, terminal events, censored.
convergence	Convergence message from nlminb.
call	The matched call.

References

Bellach, A. and Kosorok, M.R. (2026). Weighted NPMLE for the marginal mean of recurrent events with a competing terminal event. *arXiv preprint* arXiv:2605.25934.

Bellach, A., Kosorok, M.R., Ruschendorf, L. and Fine, J.P. (2019). Weighted NPMLE for the subdistribution of a competing risk. *Journal of the American Statistical Association*, 114(525), 259-270.

Examples

```
library(survival)
data("bladder2", package = "survival")
bladder2_prepped <- bladder_prep()

fit <- wnpmle_fit(Surv(time, status) ~ treat + num + size,
                  data = bladder2_prepped, id = "id",
                  model = "log", rho = 1)
summary(fit)
```

Index