Package: ZTRecur (via r-universe)

November 12, 2026

Type Package

Title Analysis for Zero-Truncated Recurrent Event Data

Version 0.2.1

Date 2026-10-17

Description Provides nonparametric and Cox-type regression methods for analyzing zero-truncated recurrent event data with supplementary information. Key features include estimating rate and mean functions under nonparametric marginal models; estimating time-constant regression coefficients under standard and stratified Andersen-Gill (AG) models; and estimating time-varying covariate effects under stratified and nonstratified Cox-type models. Standard errors (SEs) are estimated using sandwich-based variance estimators for nonstratified models and resampling-based procedures for stratified models. For further details, see Chen, Hu, and Rosychuk (2026) <doi:10.1214/25-AOAS2130>, Chen et al. (2026) <doi:10.1002/cjs.70064>, and Chen, Hu, and Rosychuk (2026) <doi:10.48550/arXiv.2607.26210>.

License GPL (>= 2)

Encoding UTF-8

Imports Rcpp (>= 1.0.0), dplyr, graphics, grDevices, stats

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

ByteCompile yes

Depends R (>= 3.5)

LazyData true

Config/roxygen2/version 8.0.0

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Author Anqi A. Chen [aut, cre], X. Joan Hu [aut], Yi Xiong [aut], Rhonda J. Rosychuk [aut]

Maintainer Anqi A. Chen <anqi6@ualberta.ca>

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

Date/Publication 2026-11-12 12:57:50 UTC

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

RemoteRef HEAD

RemoteSha 9675fd5a76bebd6104d9b15f2585403406d521a1

Contents

ZTRecur-package 2
birth.interval 3
my.agge.info 4
my.base 4
my.visit 5
nonpara_esti 5
nonstrati.reg 8
plot_nonpar 12
plot_nonstrati_baseline 13
plot_nonstrati_coef 15
plot_stratified_baseline 17
plot_stratified_coef 20
strati.reg 23

Index **28**

`ZTRecur-package` *ZTRecur: Analysis for Zero-Truncated Recurrent Event Data*

Description

This package provides nonparametric and Cox-type regression methods for analyzing zero-truncated recurrent event data with supplementary information. Key features include estimating rate and mean functions under nonparametric marginal models; estimating time-constant regression coefficients under standard and stratified Andersen-Gill (AG) models; and estimating time-varying covariate effects under stratified and nonstratified Cox-type models. Standard errors (SEs) are estimated using sandwich-based variance estimators for nonstratified models and resampling-based procedures for stratified models. For further details, see Chen, Hu, and Rosychuk (2026) doi:10.1214/25AOAS2130, Chen et al. (2026) doi:10.1002/cjs.70064, and Chen, Hu, and Rosychuk (2026) https://arxiv.org/abs/2607.26210.

Author(s)

Maintainer: Anqi A. Chen <anqi6@ualberta.ca>

Authors:

- Anqi A. Chen <anqi6@ualberta.ca>
- X. Joan Hu
- Yi Xiong
- Rhonda J. Rosychuk

`birth.interval` *Generate Birthdate Intervals Based on Integer Ages at Event Times*

Description

Generate Birthdate Intervals Based on Integer Ages at Event Times

Usage

```
birth.interval(visit, MAXAGE = 18, WINDOWL, WINDOWR)
```

Arguments

`visit`	Matrix of event-level information. It contains subject ID (`patid`), Julian date of the event (`date`), and integer age at that event date (`age`).
`MAXAGE`	Maximum age considered. The default value is 18.
`WINDOWL`	The start date of data extraction window, such as as.Date("2010-04-01").
`WINDOWR`	The end date of data extraction window, such as as.Date("2025-03-31").

Details

The calculation of birthdate interval is described in Hu & Rosychuk (2016).

Value

A subject-level data frame with patient ID (`patid`), left (`birthL`) and right (`birthR`) endpoints of birthdate intervals in Julian date.

References

Hu, X. J., & Rosychuk, R. J. (2016). Marginal regression analysis of recurrent events with coarsened censoring times. *Biometrics*, 72(4), 1113-1122.

Examples

```
# Note: The \code{visit} must be a data frame.
birth.b <- birth.interval(visit=as.data.frame(my.visit),
WINDOWL=as.Date("2010-04-01"), WINDOWR=as.Date("2025-03-31"))
```

my.agge.info *Sample Population Aggregate Dataset*

Description

A sample data frame containing aggregated population census information.

Usage

```
my.agge.info
```

Format

A data frame with n rows and the following variables:

AGE Integer age.

MaleInd Indicator for male sex.

CityAInd Indicator for residence in region A.

CityBInd Indicator for residence in region B.

pop Number of subjects in the corresponding demographic group.

my.base *Sample Base Dataset*

Description

A sample matrix containing subject-level demographic information and the number of observed events for each subject.

Usage

```
my.base
```

Format

A matrix with n rows and the following variables:

patid Subject ID.

MaleInd Indicator for male sex.

CityAInd Indicator for residence in region A.

CityBInd Indicator for residence in region B.

nmvst Number of observed events.

p1stvt Row index of the subject's first observed event in the `my.visit` matrix.

birthl Lower endpoint of the birthdate interval, represented as a Julian date.

birthr Upper endpoint of the birthdate interval, represented as a Julian date.

birthJulian Exact birthdate, represented as a Julian date.

my.visit *Sample Visit Dataset*

Description

A sample matrix containing event-level information for each subject.

Usage

```
my.visit
```

Format

A matrix with n rows and the following variables:

patid Subject ID.

reptind Indicator for a subsequent event: 0 for the first observed event and 1 for subsequent events for the subject.

date Julian date of the event.

age Integer age at the date of the event.

nonpara_esti *Nonparametric Estimation of Marginal Mean and Rate Functions*

Description

Performs nonparametric estimation of marginal mean and rate functions across covariate groups using zero-truncated recurrent event data integrated with supplementary population information. The procedure accounts for coarsened individual follow-up information and provides variance estimates via bootstrap or multiplier resampling techniques.

Usage

```
nonpara_esti(
  base,
  visit,
  mycovars,
  AGEUNIT,
  ONUM,
  WINDOWL,
  WINDOWR,
  agge.info,
  knownBirth = FALSE,
  reptw = 100,
  B = 200,
```

```
  seeds = 1234,
  resampling = FALSE,
  r.distn = "std Normal",
  save.res = FALSE,
  resultpath = NULL
)
```

Arguments

base	Matrix of subject-level information. Each row corresponds to one subject and contains demographic information (covariates and either the exact birthdate (birthJulian) or birthdate interval (birthl and birthr)), the number of observed events (nmvst), and the row index of the subject's first observed event in the visit matrix (p1stvt). The variables must be provided in the following order: covariates, nmvst, p1stvt, and either birthJulian or birthl and birthr.
visit	Matrix of event-level information. Each row corresponds to one observed event and contains the subject ID (patid), an indicator of whether the event is a subsequent visit (reptind; 0 for the first observed visit and 1 for subsequent visits for the same subject), the Julian date of the event (date), and the integer age at the time of the event (age). Subject IDs must be integers from 1 through the number of rows in the base matrix. If birthdate information is unavailable, the integer age (age) must be provided for each event.
mycovars	Character vector specifying the variable names of covariates to define subgroups.
AGEUNIT	Integer; the number of time units per year (e.g., 6 for bimonthly data).
ONUM	Integer; the maximum number of time units under observation (e.g., 18 * AGEUNIT for <18 years).
WINDOWL	Numeric; Julian date representing the left boundary of the data extraction window.
WINDOWR	Numeric; Julian date representing the right boundary of the data extraction window.
agge.info	Data frame containing aggregated population census data, including integer age (AGE),mycovars, and number of subjects in that subgroup (pop).
knownBirth	Logical; indicates whether exact birthdates are known. Defaults to FALSE. If FALSE, exact birthdates are unknown, and birth.interval can be used to calculate the corresponding birthdate intervals in Julian dates.
reptw	Integer; number of birthdate sets generated for the missing birthdate procedure A (Hu & Rosychuk, 2016). Default is 100.
B	Integer; number of bootstrap samples or resampling replicates for variance estimation. Default is 200.
seeds	Numeric; seed value for reproducible sampling. Default is 1234.
resampling	Logical; if TRUE, applies the multiplier resampling method described in Chen, Hu, & Rosychuk (2026). If FALSE (default), standard bootstrapping is used.
r.distn	Character string specifying the multiplier distribution used for resampling when resampling = TRUE. Options are the standard normal distribution ("std Normal"; default) or a Poisson distribution with variance 1 ("Poisson").

save.res	Logical; if TRUE, writes output files to disk. Default is FALSE.
resultpath	Character string indicating the file path destination for saved results, such as "C:\project\". Required when save.res = TRUE.

Details

The detailed estimation procedure is described in Chen, Hu & Rosychuk (2026a). The approach accounts for missing birthdates using Procedure A of Hu & Rosychuk (2016) and calculates variance estimates using the multiplier resampling procedure described in Chen, Hu & Rosychuk (2026b).

Value

A list containing the estimated marginal mean functions and rate functions, their corresponding variance estimates, and numerical results for each bootstrap or resampling sample.

References

Chen, A. A., Hu, X. J., & Rosychuk, R. J. (2026a). Statistical Learning of Pediatric Mental Health-Related Emergency Department Visits Across COVID-19 Pandemic Periods. *arXiv preprint arXiv:2607.26210*.

Chen, A. A., Hu, X. J., & Rosychuk, R. J. (2026b). Stratified regression analysis of zero-truncated recurrent event data. *The Annals of Applied Statistics*, 20(1), 434-451.

Hu, X. J., & Rosychuk, R. J. (2016). Imputation procedures for missing birthdates in administrative health data. *Statistics in Medicine*, 35(14), 2341-2355.

Examples

```
# When birthdates are unknown, bootstrap-based variance estimates are used.
nonpara.res.uk <- nonpara_esti(base = my.base[,c(-1,-9)],
visit = my.visit, mycovars = c("MaleInd","CityAInd","CityBInd"),
  AGEUNIT = 6, ONUM = (18*6), WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")), agge.info = my.agge.info,
  seeds = 1234)

# When birthdates are known, bootstrap-based variance estimates are used.
nonpara.res.k <- nonpara_esti(base = my.base[,c(-1,-7,-8)],
visit = my.visit, mycovars = c("MaleInd","CityAInd","CityBInd"),
  AGEUNIT = 6, ONUM = (18*6), WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")), agge.info = my.agge.info,
  knownBirth=TRUE,seeds = 1234)

# When birthdates are known, multiplier resampling-based variance estimates
# are obtained using standard normal multipliers.
nonpara.res.k.r.std <- nonpara_esti(base = my.base[,c(-1,-7,-8)],
visit = my.visit, mycovars = c("MaleInd","CityAInd","CityBInd"),
  AGEUNIT = 6, ONUM = (18*6), WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")), agge.info = my.agge.info,
  knownBirth=TRUE,seeds = 1234,resampling=TRUE)

# When birthdates are known, multiplier resampling-based variance estimates
```

```
# are obtained using Poisson multipliers.
nonpara.res.k.r.poi <- nonpara_esti(base = my.base[,c(-1,-7,-8)],
visit = my.visit, mycovars = c("MaleInd","CityAInd","CityBInd"),
  AGEUNIT = 6, ONUM = (18*6), WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")), agge.info = my.agge.info,
  knownBirth=TRUE,seeds = 1234,resampling=TRUE,r.distn="Poisson")
```

`nonstrati.reg` *Non-Stratified Regression Analysis for Zero-Truncated Recurrent Event Data*

Description

Fits a non-stratified Cox-type model with time-constant or time-varying regression coefficients using zero-truncated recurrent event data integrated with population census information. The method accounts for coarsened follow-up information and yields sandwich-form variance estimates along with empirical bootstrap or multiplier resampling alternatives.

Usage

```
nonstrati.reg(
  base,
  visit,
  AGEUNIT,
  ONUM,
  WINDOWL,
  WINDOWR,
  etapara,
  reptmax,
  windowKH,
  tauL,
  tauR,
  method,
  N_star,
  n,
  knownBirth = FALSE,
  reptw = 100,
  B = NA,
  seeds = 1234,
  resampling = FALSE,
  r.distn = "std Normal",
  mycovars,
  save.res = FALSE,
  resultpath = NULL,
  output.lambda = FALSE
)
```

Arguments

base	Matrix of subject-level information. Each row corresponds to one subject and contains demographic information (covariates and either the exact birthdate (birthJulian) or birthdate interval (birthl and birthr)), the number of observed events (nmvst), and the row index of the subject's first observed event in the visit matrix (p1stvt). The variables must be provided in the following order: covariates, nmvst, p1stvt, and either birthJulian or birthl and birthr.
visit	Matrix of event-level information. Each row corresponds to one observed event and contains the subject ID (patid), an indicator of whether the event is a subsequent visit (reptind; 0 for the first observed visit and 1 for subsequent visits for the same subject), the Julian date of the event (date), and the integer age at the time of the event (age). Subject IDs must be integers from 1 through the number of rows in the base matrix. If birthdate information is unavailable, the integer age (age) must be provided for each event.
AGEUNIT	Integer; the number of time units per year (e.g., 6 for bimonthly data).
ONUM	Integer; the maximum number of time units under observation (e.g., 18 * AGEUNIT for <18 years).
WINDOWL	Numeric; Julian date representing the left boundary of the data extraction window.
WINDOWR	Numeric; Julian date representing the right boundary of the data extraction window.
etapara	Numeric; tolerance parameter controlling the convergence criterion when solving the estimating equations.
reptmax	Integer; the maximum number of iterations allowed when solving estimating equations.
windowKH	Numeric; bandwidth size of the smoothing kernel, specified in time units.
tauL	Numeric (in time units); left boundary used for local constant estimation to avoid boundary issues.
tauR	Numeric (in time units); right boundary used for local constant estimation to avoid boundary issues.
method	Character string specifying the estimation method. Options are estimation under the Andersen-Gill model ("L"), local constant estimation under a Cox-type model with time-varying coefficients ("LC"), or both estimation procedures ("all").
N_star	Data frame containing aggregated population census data, including integer age (AGE),mycovars, and number of subjects in that subgroup (pop).
n	Integer; the approximate size of O, the random sample drawn from the target population.
knownBirth	Logical; indicates whether exact birthdates are known. Defaults to FALSE. If FALSE, exact birthdates are unknown, and birth.interval can be used to calculate the corresponding birthdate intervals in Julian dates.
reptw	Integer; number of birthdate sets generated for the missing birthdate procedure A (Hu & Rosychuk, 2016). Default is 100.

B	Integer or NA; number of bootstrap or resampling replicates for variance estimation. Set to NA (default) to compute only the robust sandwich-form variance estimator.
seeds	Numeric; seed value for reproducible sampling. Default is 1234.
resampling	Logical; if TRUE, applies the multiplier resampling method described in Chen, Hu, & Rosychuk (2026). If FALSE (default), standard bootstrapping is used.
r.distn	Character string specifying the multiplier distribution used for resampling when resampling = TRUE. Options are the standard normal distribution ("std Normal"; default) or a Poisson distribution with variance 1 ("Poisson").
mycovars	Character vector specifying the variable names of covariates to define subgroups.
save.res	Logical; if TRUE, writes output files to disk. Default is FALSE.
resultpath	Character string indicating the file path destination for saved results, such as "C:\project\". Required when save.res = TRUE.
output.lambda	Logical; if TRUE, includes baseline function estimates in the returned output. Default is FALSE.

Details

The estimation procedure is similar to that described in Xiong, Hu & Rosychuk (2026). The approach accounts for missing birthdates using Procedure A of Hu & Rosychuk (2016) and obtains variance estimates using the multiplier resampling procedure described in Chen, Hu & Rosychuk (2026).

Value

A list containing estimates of the regression coefficients, standard error estimates (sandwich and/or bootstrap- or resampling-based), estimated cumulative baseline functions, and estimated baseline functions when output.lambda = TRUE.

References

Chen, A. A., Hu, X. J., & Rosychuk, R. J. (2026). Stratified regression analysis of zero-truncated recurrent event data. *The Annals of Applied Statistics*, 20(1), 434-451.

Hu, X. J., & Rosychuk, R. J. (2016). Imputation procedures for missing birthdates in administrative health data. *Statistics in Medicine*, 35(14), 2341-2355.

Xiong, Y., Hu, X. J., & Rosychuk, R. J. (2026). Learning differences between two decades of mental health related emergency department visits by youth via recurrent events data analyses. *Journal of the Royal Statistical Society Series C: Applied Statistics*, qlag001.

Examples

```
# Example non-stratified regression model with unknown birthdates.
# Sandwich-form variance estimates.
nonstrati.fit.uk <- nonstrati.reg(base = my.base[,c(-1,-9)], visit = my.visit,
AGEUNIT = 6, ONUM = 108,
  WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")),
```

```
  etapara = 1e-4, reptmax = 50, windowKH = rep(9,108),
  tauL = 9, tauR = 105, method = "all", N_star = my.agge.info, n=50000,
  knownBirth = FALSE, reptw = 100, B = NA, seeds = 12345, resampling = TRUE,
  r.distn = "std Normal",mycovars = c("MaleInd","CityAInd","CityBInd"),
  save.res = FALSE, output.lambda = TRUE)


# Example non-stratified regression model with known birthdates.
# Sandwich-form variance estimates.
nonstrati.fit.k <- nonstrati.reg(base = my.base[,c(-1,-7,-8)],
visit = my.visit,
AGEUNIT = 6, ONUM = 108,
  WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")),
  etapara = 1e-4, reptmax = 50, windowKH = rep(9,108),
  tauL = 9, tauR = 105, method = "all", N_star = my.agge.info, n=50000,
  knownBirth = TRUE, reptw = 100, B = NA, seeds = 12345, resampling = TRUE,
  r.distn = "std Normal",mycovars = c("MaleInd","CityAInd","CityBInd"),
  save.res = FALSE, output.lambda = TRUE)

# Example with known birthdates and bootstrap-based variance estimates.
nonstrati.fit.k.B <- nonstrati.reg(base = my.base[,c(-1,-7,-8)],
visit = my.visit,
AGEUNIT = 6, ONUM = 108,
  WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")),
  etapara = 1e-4, reptmax = 50, windowKH = rep(9,108),
  tauL = 9, tauR = 105, method = "all", N_star = my.agge.info, n=50000,
  knownBirth = TRUE, reptw = 100, B = 200, seeds = 12345, resampling = FALSE,
  r.distn = "std Normal",mycovars = c("MaleInd","CityAInd","CityBInd"),
  save.res = FALSE, output.lambda = TRUE)

# Example with known birthdates and multiplier resampling-based variance
# estimates using standard normal multipliers.
nonstrati.fit.k.R <- nonstrati.reg(base = my.base[,c(-1,-7,-8)],
visit = my.visit,
AGEUNIT = 6, ONUM = 108,
  WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")),
  etapara = 1e-4, reptmax = 50, windowKH = rep(9,108),
  tauL = 9, tauR = 105, method = "all", N_star = my.agge.info, n=50000,
  knownBirth = TRUE, reptw = 100, B = 200, seeds = 12345, resampling = TRUE,
  r.distn = "std Normal",mycovars = c("MaleInd","CityAInd","CityBInd"),
  save.res = FALSE, output.lambda = TRUE)

# Example with known birthdates and multiplier resampling-based variance
# estimates using Poisson multipliers with variance 1.
nonstrati.fit.k.R.p <- nonstrati.reg(base = my.base[,c(-1,-7,-8)],
visit = my.visit,
AGEUNIT = 6, ONUM = 108,
  WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")),
  etapara = 1e-4, reptmax = 50, windowKH = rep(9,108),
```

```
tauL = 9, tauR = 105, method = "all", N_star = my.agge.info, n=50000,
knownBirth = TRUE, reptw = 100, B = 200, seeds = 12345, resampling = TRUE,
r.distn = "Poisson",mycovars = c("MaleInd","CityAInd","CityBInd"),
save.res = FALSE, output.lambda = TRUE)
```

`plot_nonpar` *Plot Nonparametric Marginal Mean or Rate Estimates*

Description

Displays estimated marginal mean or rate functions along with pointwise confidence intervals for different covariate groups based on output from `nonpara_esti`.

Usage

```
plot_nonpar(
  res.nonpar,
  rate.p = TRUE,
  c = 1.96,
  max.L.m = NA,
  ONUM,
  AGEUNIT,
  cex = 1,
  two.plot = FALSE,
  xlab = "Age (in years)"
)
```

Arguments

`res.nonpar`	An object returned by `nonpara_esti` containing nonparametric estimates.
`rate.p`	Logical; if `TRUE` (default), plots marginal rate estimates. If `FALSE`, plots marginal mean estimates.
`c`	Numeric; critical value used for constructing pointwise confidence intervals. Default is `1.96` (corresponding to a 95% confidence interval).
`max.L.m`	Numeric; upper limit for the y-axis. If `NA` (default), the maximum value is automatically calculated based on the data.
`ONUM`	Integer; maximum number of time units (i.e., `AGEUNIT`* number of years).
`AGEUNIT`	Integer; number of time units per year (e.g., 6 for bimonthly data).
`cex`	Numeric; magnification factor for legend text size. Default is `1`.
`two.plot`	Logical; if `TRUE`, splits display into two separate panels, useful when visualizing a large number of covariate combinations. Default is `FALSE`.
`xlab`	Character string specifying the x-axis label. Defaults to `"Age (in years)"`.

Value

No return value; generates a plot of marginal mean or rate functions.

References

Chen, A. A., Hu, X. J., & Rosychuk, R. J. (2026). Statistical Learning of Pediatric Mental Health-Related Emergency Department Visits Across COVID-19 Pandemic Periods. *arXiv preprint arXiv:2607.26210*.

Examples

```
# The object 'nonpara.res.uk' is the output from nonpara_esti().
nonpara.res.uk <- nonpara_esti(base = my.base[,c(-1,-9)],
visit = my.visit, mycovars = c("MaleInd","CityAInd","CityBInd"),
  AGEUNIT = 6, ONUM = (18*6), WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")), agge.info = my.agge.info,
  seeds = 1234)

# Plot estimated marginal means across covariate subgroups.
plot_nonpar(res.nonpar = nonpara.res.uk, rate.p = FALSE, c = 1.96, ONUM = 108,
  AGEUNIT = 6, two.plot = FALSE)

# Plot estimated marginal rates across covariate subgroups.
plot_nonpar(res.nonpar = nonpara.res.uk, rate.p = TRUE, c = 1.96, ONUM = 108,
  AGEUNIT = 6, two.plot = FALSE,max.L.m=0.20)
```

---

plot_nonstrati_baseline

Plot (Cumulative) Baseline Functions from Non-Stratified Cox-Type Models

---

Description

Displays estimated (cumulative) baseline functions along with pointwise confidence intervals for non-stratified models with both time-varying and time-constant regression coefficients.

Usage

```
plot_nonstrati_baseline(
  fit.non.s,
  ONUM,
  AGEUNIT,
  subtitle = "non-stratified",
  leg.loc = "topright",
  c = 1.96,
  leg.cex = 1.2,
  cumulative = TRUE,
```

```
  ylim = NA,
  cex.lab = 1.3,
  cex.axis = 1.3,
  model.v = "Age-varying coef",
  model.c = "Age-constant coef",
  xlab = "Age (in years)"
)
```

Arguments

fit.non.s	Object containing the output from a non-stratified model fit (e.g., from nonstrati.reg).
ONUM	Integer; maximum number of time units (i.e., AGEUNIT* number of years).
AGEUNIT	Integer; number of time units per year (e.g., 6 for bimonthly data).
subtitle	Character string specifying plot subtitle. Default is "non-stratified".
leg.loc	Character string indicating legend placement (e.g., "topright").
c	Numeric; critical value used for constructing pointwise confidence intervals. Default is 1.96 (corresponding to a 95% confidence interval).
leg.cex	Numeric magnification factor for legend text size. Default is 1.2.
cumulative	Logical; if TRUE (default), plots estimated cumulative baseline functions; if FALSE, plots estimated baseline functions.
ylim	Numeric vector of length 2 specifying y-axis limits c(ymin, ymax). If NA (default), limits are computed automatically.
cex.lab	Numeric magnification factor for axis labels relative to current setting. Default is 1.3.
cex.axis	Numeric magnification factor for axis tick labels relative to current setting. Default is 1.3.
model.v	Character string; legend label for the model with time-varying coefficients. Default is "Age-varying coef".
model.c	Character string; legend label for the model with time-constant coefficients. Default is "Age-constant coef".
xlab	Character string specifying the x-axis label. Defaults to "Age (in years)".

Details

The function assumes that method = "all" and B!=NA was used in nonstrati.reg.

Value

No return value; generates a (cumulative) baseline function plot.

Examples

```
# The 'nonstrati.fit.k.B' is the output from nonstrati.reg()

# Note: 95% pointwise confidence intervals for the baseline functions can
# be plotted only when bootstrap- or multiplier-resampling-based variance
```

```
# estimates are available from nonstrati.reg().
nonstrati.fit.k.B <- nonstrati.reg(base = my.base[,c(-1,-7,-8)],
visit = my.visit,
AGEUNIT = 6, ONUM = 108,
  WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")),
  etapara = 1e-4, reptmax = 50, windowKH = rep(9,108),
  tauL = 9, tauR = 105, method = "all", N_star = my.agge.info, n=50000,
  knownBirth = TRUE, reptw = 100, B = 200, seeds = 12345, resampling = FALSE,
  r.distn = "std Normal",mycovars = c("MaleInd","CityAInd","CityBInd"),
  save.res = FALSE, output.lambda = TRUE)

# Plot the estimated cumulative baseline functions:
plot_nonstrati_baseline(fit.non.s = nonstrati.fit.k.B, ONUM = 108, AGEUNIT = 6,
  subtitle = "non-stratified analysis", cumulative = TRUE, c = 1.96,
  leg.loc="topleft")

# Plot the estimated baseline functions:
plot_nonstrati_baseline(fit.non.s = nonstrati.fit.k.B, ONUM = 108, AGEUNIT = 6,
  subtitle = "non-stratified analysis", cumulative = FALSE, c = 1.96,
  leg.loc="topleft",ylim=c(0.05,0.17))
```

`plot_nonstrati_coef` *Plot Estimated Regression Coefficients from Non-Stratified Cox-Type Models*

Description

Displays estimates and pointwise confidence intervals for time-varying and time-constant regression coefficients from fitted non-stratified models.

Usage

```
plot_nonstrati_coef(
  fit.non.s,
  cov_name,
  i = 1,
  age,
  tauL,
  tauR,
  yRange = NA,
  resampling.se = TRUE,
  model.v = "Age-varying coef",
  model.c = "Age-constant coef",
  location = "topright",
  cex.axis = 1.2,
  cex.lab = 1.3,
  cex.legend = 1.2,
```

```
  c = 1.96,
  xlab = "Age (in years)"
)
```

Arguments

fit.non.s	Object containing the output from a non-stratified model fit (e.g., from nonstrati.reg).
cov_name	Character vector of covariate names matching the model specification.
i	Integer; index of the covariate in cov_name to be plotted. Default is 1.
age	Numeric vector representing the age or time points corresponding to the estimates.
tauL	Numeric (in time units); left boundary used for local constant estimation to avoid boundary issues.
tauR	Numeric (in time units); right boundary used for local constant estimation to avoid boundary issues.
yRange	Numeric vector of length 2 specifying the y-axis limits c(ymin, ymax). If NA (default), limits are calculated automatically.
resampling.se	Logical; if TRUE (default), uses standard errors derived from bootstrap/resampling. If FALSE, uses sandwich standard errors.
model.v	Character string; legend label for the model with time-varying coefficients. Default is "Age-varying coef".
model.c	Character string; legend label for the model with time-constant coefficients. Default is "Age-constant coef".
location	Character string specifying the position of the legend (e.g., "topright").
cex.axis	Numeric magnification factor for axis annotation relative to current setting. Default is 1.2.
cex.lab	Numeric magnification factor for axis labels relative to current setting. Default is 1.3.
cex.legend	Numeric magnification factor for legend text. Default is 1.2.
c	Numeric; critical value used for constructing pointwise confidence intervals. Default is 1.96 (corresponding to a 95% confidence interval).
xlab	Character string specifying the x-axis label. Defaults to "Age (in years)".

Details

The function assumes that method = "all" was used in nonstrati.reg.

Value

No return value; generates a coefficient plot for the specified covariate.

Examples

```
# The 'nonstrati.fit.uk' and 'nonstrati.fit.k' are outputs from nonstrati.reg()
nonstrati.fit.uk <- nonstrati.reg(base = my.base[,c(-1,-9)], visit = my.visit,
AGEUNIT = 6, ONUM = 108,
  WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")),
  etapara = 1e-4, reptmax = 50, windowKH = rep(9,108),
  tauL = 9, tauR = 105, method = "all", N_star = my.agge.info, n=50000,
  knownBirth = FALSE, reptw = 100, B = NA, seeds = 12345, resampling = TRUE,
  r.distn = "std Normal",mycovars = c("MaleInd","CityAInd","CityBInd"),
  save.res = FALSE, output.lambda = TRUE)

# 95% pointwise confidence intervals using sandwich-form standard errors.
plot_nonstrati_coef(fit.non.s = nonstrati.fit.uk, cov_name = c("MaleInd","CityAInd","CityBInd"),
i = 1, age = c(1:108)/6, tauL = 9, tauR = 105,resampling.se = FALSE, c = 1.96)

plot_nonstrati_coef(fit.non.s = nonstrati.fit.uk, cov_name = c("MaleInd","CityAInd","CityBInd"),
i = 2, age = c(1:108)/6, tauL = 9, tauR = 105,resampling.se = FALSE, c = 1.96)

plot_nonstrati_coef(fit.non.s = nonstrati.fit.uk, cov_name = c("MaleInd","CityAInd","CityBInd"),
i = 3, age = c(1:108)/6, tauL = 9, tauR = 105,resampling.se = FALSE, c = 1.96)

# 95% pointwise confidence intervals using bootstrap- or
# multiplier-resampling-based standard errors.
nonstrati.fit.k <- nonstrati.reg(base = my.base[,c(-1,-7,-8)],
visit = my.visit,
AGEUNIT = 6, ONUM = 108,
  WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")),
  etapara = 1e-4, reptmax = 50, windowKH = rep(9,108),
  tauL = 9, tauR = 105, method = "all", N_star = my.agge.info, n=50000,
  knownBirth = TRUE, reptw = 100, B = 100, seeds = 12345, resampling = TRUE,
  r.distn = "std Normal",mycovars = c("MaleInd","CityAInd","CityBInd"),
  save.res = FALSE, output.lambda = TRUE)

plot_nonstrati_coef(fit.non.s = nonstrati.fit.k, cov_name = c("MaleInd","CityAInd","CityBInd"),
i = 1, age = c(1:108)/6, tauL = 9, tauR = 105,resampling.se = TRUE, c = 1.96)
```

---

plot_stratified_baseline

Plot (Cumulative) Baseline Functions from Stratified Cox-Type Models

---

Description

Displays estimated (cumulative) baseline functions along with pointwise confidence intervals for stratified models with both time-varying and time-constant regression coefficients.

Usage

```
plot_stratified_baseline(
  fit,
  ONUM,
  AGEUNIT,
  subtitle = "stratified",
  leg.loc = "topright",
  c = 1.96,
  leg.cex = 1.2,
  cumulative = TRUE,
  ylim = NA,
  comLamb = FALSE,
  cex.lab = 1.3,
  cex.axis = 1.3,
  model.v = "Age-varying coef",
  model.c = "Age-constant coef",
  xlab = "Age (in years)"
)
```

Arguments

fit	Object containing output from a stratified Cox-type model fit (e.g., from strati.reg).
ONUM	Integer; maximum number of time units (i.e., AGEUNIT* number of years).
AGEUNIT	Integer; number of time units per year (e.g., 6 for bimonthly data).
subtitle	Character string specifying plot subtitle. Default is "stratified".
leg.loc	Character string indicating legend placement (e.g., "topright").
c	Numeric; critical value used for constructing pointwise confidence intervals. Default is 1.96 (corresponding to a 95% confidence interval).
leg.cex	Numeric magnification factor for legend text size. Default is 1.2.
cumulative	Logical; if TRUE (default), plots estimated cumulative baseline functions; if FALSE, plots estimated baseline functions.
ylim	Numeric vector of length 2 specifying y-axis limits c(ymin, ymax). If NA (default), limits are computed automatically.
comLamb	Logical; indicates whether common baseline functions across strata were assumed in the model fit. Default is FALSE.
cex.lab	Numeric magnification factor for axis labels relative to current setting. Default is 1.3.
cex.axis	Numeric magnification factor for axis tick labels relative to current setting. Default is 1.3.
model.v	Character string; legend label for the model with time-varying coefficients. Default is "Age-varying coef".
model.c	Character string; legend label for the model with time-constant coefficients. Default is "Age-constant coef".
xlab	Character string specifying the x-axis label. Defaults to "Age (in years)".

Details

The function assumes that `method = "all"` and `B!=NA` was used in `strati.reg`.

Value

No return value; generates a plot of stratum-specific baseline functions.

References

Chen, A. A., Hu, X. J., & Rosychuk, R. J. (2026a). Stratified regression analysis of zero-truncated recurrent event data. *The Annals of Applied Statistics*, 20(1), 434-451.

Chen, A. A., Hu, X. J., Rosychuk, R. J., & Zeng, L. (2026b). Analyzing zero-truncated recurrent event data by stratified regression with time-varying coefficients. *Canadian Journal of Statistics*, e70064.

Examples

```
# The 'strati.fit.k.R.p' is the output from strati.reg().
strati.fit.k.R.p <- strati.reg(base = my.base[,c(-1,-7,-8)], visit = my.visit,
AGEUNIT = 6,ONUM = 108, WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")), etapara = 1e-4, reptmax = 50,
  windowKH = rep(9,108), tauL = 9, tauR = 105, method = "all",
  N_star =  my.agge.info, n=50000,
  B = 200, seeds = 12345, com.coef = 0, com.lamb = 0, knownBirth = TRUE,
  reptw = 100, resampling = TRUE,r.distn = "Poisson",
  mycovars = c("MaleInd","CityAInd","CityBInd"), save.res = FALSE,
  output.lambda = TRUE)

# Plot stratified cumulative baseline function estimates:
  plot_stratified_baseline(fit = strati.fit.k.R.p, ONUM = 108, AGEUNIT = 6,
  subtitle = "stratified analysis", leg.loc="topleft",cumulative = TRUE,
  comLamb = FALSE,c = 1.96)

  # Plot stratified baseline function estimates:
  plot_stratified_baseline(fit = strati.fit.k.R.p, ONUM = 108, AGEUNIT = 6,
  subtitle = "stratified analysis", leg.loc="topleft",cumulative = FALSE,
  comLamb = FALSE,c = 1.96)

  # The object 'strati.fit.k.R.p.combB' is the output from strati.reg().
  strati.fit.k.R.p.combB <- strati.reg(base = my.base[,c(-1,-7,-8)], visit = my.visit,
AGEUNIT = 6,ONUM = 108, WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")), etapara = 1e-4, reptmax = 50,
  windowKH = rep(9,108), tauL = 9, tauR = 105, method = "all",
  N_star =  my.agge.info, n=50000,
  B = 200, seeds = 12345, com.coef = 0, com.lamb = 1, knownBirth = TRUE,
  reptw = 100, resampling = TRUE,r.distn = "Poisson",
  mycovars = c("MaleInd","CityAInd","CityBInd"), save.res = FALSE,
  output.lambda = TRUE)

  # Plot the common cumulative baseline function estimates.
  plot_stratified_baseline(fit = strati.fit.k.R.p.combB, ONUM = 108, AGEUNIT = 6,
```

```
  subtitle = "stratified analysis", leg.loc="topleft",cumulative = TRUE,
  comLamb = TRUE,c = 1.96)


  # Plot the common baseline function estimates.
  plot_stratified_baseline(fit = strati.fit.k.R.p.combB, ONUM = 108, AGEUNIT = 6,
  subtitle = "stratified analysis", leg.loc="topleft",cumulative = FALSE,
  comLamb = TRUE,c = 1.96)
```

`plot_stratified_coef` *Plot Estimated Regression Coefficients from Stratified Cox-Type Models*

Description

Displays estimates and pointwise confidence intervals for time-varying and time-constant regression coefficients from fitted stratified models.

Usage

```
plot_stratified_coef(
  fit,
  cov_name,
  i = 1,
  age,
  tauL,
  tauR,
  yRange = NA,
  adjust.LC = FALSE,
  tauR.1 = NA,
  tauL.2 = NA,
  comCoef = FALSE,
  model.v = "Age-varying coef",
  model.c = "Age-constant coef",
  location = "topright",
  cex.axis = 1.2,
  cex.lab = 1.3,
  cex.legend = 1.2,
  c = 1.96,
  xlab = "Age (in years)"
)
```

Arguments

`fit`	Object containing output from a stratified Cox-type model fit (e.g., from `strati.reg`).
`cov_name`	Character vector of covariate names matching the model specification.

i	Integer; index of the covariate in cov_name to be plotted. Default is 1.
age	Numeric vector representing the age or time points corresponding to the estimates.
tauL	Numeric (in time units); left boundary used for local constant estimation to avoid boundary issues.
tauR	Numeric (in time units); right boundary used for local constant estimation to avoid boundary issues.
yRange	Numeric vector of length 2 specifying the y-axis limits c(ymin, ymax). If NA (default), limits are calculated automatically.
adjust.LC	Logical; if TRUE, applies boundary adjustment using tauR.1 and tauL.2 for stratum-specific data sparsity. Default is FALSE.
tauR.1	Numeric (in time units); right boundary parameter for stratum 1 when adjust.LC = TRUE.
tauL.2	Numeric (in time units); left boundary parameter for stratum 2 when adjust.LC = TRUE.
comCoef	Logical; indicates whether common regression coefficients across strata were assumed in the model. Default is FALSE.
model.v	Character string; legend label for the model with time-varying coefficients. Default is "Age-varying coef".
model.c	Character string; legend label for the model with time-constant coefficients. Default is "Age-constant coef".
location	Character string specifying the position of the legend (e.g., "topright").
cex.axis	Numeric magnification factor for axis annotation relative to current setting. Default is 1.2.
cex.lab	Numeric magnification factor for axis labels relative to current setting. Default is 1.3.
cex.legend	Numeric magnification factor for legend text. Default is 1.2.
c	Numeric; critical value used for constructing pointwise confidence intervals. Default is 1.96 (corresponding to a 95% confidence interval).
xlab	Character string specifying the x-axis label. Defaults to "Age (in years)".

Details

The function assumes that method = "all" was used in strati.reg.

Value

No return value; generates a stratum-specific coefficient plot for the specified covariate.

References

Chen, A. A., Hu, X. J., & Rosychuk, R. J. (2026a). Stratified regression analysis of zero-truncated recurrent event data. *The Annals of Applied Statistics*, 20(1), 434-451.

Chen, A. A., Hu, X. J., Rosychuk, R. J., & Zeng, L. (2026b). Analyzing zero-truncated recurrent event data by stratified regression with time-varying coefficients. *Canadian Journal of Statistics*, e70064.

Examples

```
# The 'strati.fit.k.R.p' is the output from strati.reg().
strati.fit.k.R.p <- strati.reg(base = my.base[,c(-1,-7,-8)], visit = my.visit,
AGEUNIT = 6,ONUM = 108, WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")), etapara = 1e-4, reptmax = 50,
  windowKH = rep(9,108), tauL = 9, tauR = 105, method = "all",
  N_star =  my.agge.info, n=50000,
  B = 200, seeds = 12345, com.coef = 0, com.lamb = 0, knownBirth = TRUE,
  reptw = 100, resampling = TRUE,r.distn = "Poisson",
  mycovars = c("MaleInd","CityAInd","CityBInd"), save.res = FALSE,
  output.lambda = TRUE)

# Plot age-varying regression coefficient estimates for each stratum.
# The age range is adjusted for the local constant estimates in both strata.
plot_stratified_coef(fit = strati.fit.k.R.p,
cov_name = c("MaleInd","CityAInd","CityBInd"),
  i = 1, age = c(1:108)/6, tauL = 9, tauR = 105, yRange=c(-3.5,1),
  adjust.LC = TRUE, tauR.1=105,tauL.2=18,comCoef = FALSE)

plot_stratified_coef(fit = strati.fit.k.R.p,
cov_name = c("MaleInd","CityAInd","CityBInd"),
i = 2, age = c(1:108)/6, tauL = 9, tauR = 105, yRange=c(-2,1),
adjust.LC = TRUE, tauR.1=105,tauL.2=18,comCoef = FALSE, c = 1.96)

plot_stratified_coef(fit = strati.fit.k.R.p,
cov_name = c("MaleInd","CityAInd","CityBInd"),
i = 3, age = c(1:108)/6, tauL = 9, tauR = 105, yRange=c(-2,1),
adjust.LC = TRUE, tauR.1=105,tauL.2=18,comCoef = FALSE,c = 1.96)

# Plot common regression coefficient estimates across strata.
strati.fit.k.R.p.combreg <- strati.reg(base = my.base[,c(-1,-7,-8)], visit = my.visit,
AGEUNIT = 6,ONUM = 108, WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")), etapara = 1e-4, reptmax = 50,
  windowKH = rep(9,108), tauL = 9, tauR = 105, method = "all",
  N_star =  my.agge.info, n=50000,
  B = 200, seeds = 12345, com.coef = 1, com.lamb = 0, knownBirth = TRUE,
  reptw = 100, resampling = TRUE,r.distn = "Poisson",
  mycovars = c("MaleInd","CityAInd","CityBInd"), save.res = FALSE,
  output.lambda = TRUE)

plot_stratified_coef(fit = strati.fit.k.R.p.combreg,
cov_name = c("MaleInd","CityAInd","CityBInd"),
i = 1, age = c(1:108)/6, tauL = 9, tauR = 105, yRange=c(-2.5,-1),
adjust.LC = TRUE, tauR.1=105,tauL.2=18,comCoef = TRUE,c = 1.96)

plot_stratified_coef(fit = strati.fit.k.R.p.combreg,
cov_name = c("MaleInd","CityAInd","CityBInd"),
i = 2, age = c(1:108)/6, tauL = 9, tauR = 105, yRange=c(-1.4,-0.5),
adjust.LC = TRUE, tauR.1=105,tauL.2=18,comCoef = TRUE,c = 1.96)

plot_stratified_coef(fit = strati.fit.k.R.p.combreg,
cov_name = c("MaleInd","CityAInd","CityBInd"),
```

```
i = 3, age = c(1:108)/6, tauL = 9, tauR = 105, yRange=c(-2,-1),
adjust.LC = TRUE, tauR.1=105,tauL.2=18,comCoef = TRUE,c = 1.96)
```

---

strati.reg *Stratified Regression Analysis for Zero-Truncated Recurrent Event Data*

Description

Fits a stratified Cox-type model with time-constant or time-varying regression coefficients using zero-truncated recurrent event data integrated with population census information. The method accounts for coarsened follow-up information and providing bootstrap or multiplier resampling variance estimates.

Usage

```
strati.reg(
  base,
  visit,
  AGEUNIT,
  ONUM,
  WINDOWL,
  WINDOWR,
  etapara,
  reptmax,
  windowKH,
  tauL,
  tauR,
  method,
  adjust.LC = FALSE,
  tauR.1,
  tauL.2,
  N_star,
  n,
  B = NA,
  seeds,
  com.coef = 0,
  com.lamb = 0,
  knownBirth = FALSE,
  reptw = 100,
  resampling = FALSE,
  r.distn = "std Normal",
  mycovars,
  save.res = FALSE,
  resultpath = NULL,
  output.lambda = FALSE
)
```

Arguments

base	Matrix of subject-level information. Each row corresponds to one subject and contains demographic information (covariates and either the exact birthdate (birthJulian) or birthdate interval (birthl and birthr)), the number of observed events (nmvst), and the row index of the subject's first observed event in the visit matrix (p1stvt). The variables must be provided in the following order: covariates, nmvst, p1stvt, and either birthJulian or birthl and birthr.
visit	Matrix of event-level information. Each row corresponds to one observed event and contains the subject ID (patid), an indicator of whether the event is a subsequent visit (reptind; 0 for the first observed visit and 1 for subsequent visits for the same subject), the Julian date of the event (date), and the integer age at the time of the event (age). Subject IDs must be integers from 1 through the number of rows in the base matrix. If birthdate information is unavailable, the integer age (age) must be provided for each event.
AGEUNIT	Integer; the number of time units per year (e.g., 6 for bimonthly data).
ONUM	Integer; the maximum number of time units under observation (e.g., 18 * AGEUNIT for <18 years).
WINDOWL	Numeric; Julian date representing the left boundary of the data extraction window.
WINDOWR	Numeric; Julian date representing the right boundary of the data extraction window.
etapara	Numeric; tolerance parameter controlling the convergence criterion when solving the estimating equations.
reptmax	Integer; the maximum number of iterations allowed when solving estimating equations.
windowKH	Numeric; bandwidth size of the smoothing kernel, specified in time units.
tauL	Numeric (in time units); left boundary used for local constant estimation to avoid boundary issues.
tauR	Numeric (in time units); right boundary used for local constant estimation to avoid boundary issues.
method	Character string specifying the estimation method. Options are estimation under the Andersen-Gill model ("L"), local constant estimation under a Cox-type model with time-varying coefficients ("LC"), or both estimation procedures ("all").
adjust.LC	Logical; if TRUE, applies boundary adjustment using tauR.1 and tauL.2 for stratum-specific data sparsity. Default is FALSE.
tauR.1	Numeric (in time units); right boundary parameter for stratum 1 when adjust.LC = TRUE.
tauL.2	Numeric (in time units); left boundary parameter for stratum 2 when adjust.LC = TRUE.
N_star	Data frame containing aggregated population census data, including integer age (AGE),mycovars, and number of subjects in that subgroup (pop).
n	Integer; the approximate size of O, the random sample drawn from the target population.

B	Integer or NA; number of bootstrap or multiplier-resampling replicates used for variance estimation. Defaults to NA, in which case no variance estimator is calculated.
seeds	Numeric; seed value for reproducible sampling. Default is 1234.
com.coef	Binary; 0 (default) allows stratum-specific regression coefficients, while 1 constrains coefficients to be common across strata.
com.lamb	Binary; 0 (default) allows stratum-specific baseline functions, while 1 constrains baselines to be common across strata.
knownBirth	Logical; indicates whether exact birthdates are known. Defaults to FALSE. If FALSE, exact birthdates are unknown, and birth.interval can be used to calculate the corresponding birthdate intervals in Julian dates.
reptw	Integer; number of birthdate sets generated for the missing birthdate procedure A (Hu & Rosychuk, 2016). Default is 100.
resampling	Logical; if TRUE, applies the multiplier resampling method described in Chen, Hu, & Rosychuk (2026). If FALSE (default), standard bootstrapping is used.
r.distn	Character string specifying the multiplier distribution used for resampling when resampling = TRUE. Options are the standard normal distribution ("std Normal"; default) or a Poisson distribution with variance 1 ("Poisson").
mycovars	Character vector specifying the variable names of covariates to define subgroups.
save.res	Logical; if TRUE, writes output files to disk. Default is FALSE.
resultpath	Character string indicating the file path destination for saved results, such as "C:\project\". Required when save.res = TRUE.
output.lambda	Logical; if TRUE, includes baseline function estimates in the returned output. Default is FALSE.

Details

The detailed estimation procedures are described in Chen, Hu & Rosychuk (2026a) for time-constant regression coefficients and in Chen et al. (2026b) for time-varying regression coefficients. The approach accounts for missing birthdates using Procedure A of Hu & Rosychuk (2016).

Value

A list containing estimated stratified regression coefficients, standard error estimates (bootstrap- or resampling-based), estimated stratified cumulative baseline functions, and estimated stratified baseline functions when output.lambda = TRUE.

References

Chen, A. A., Hu, X. J., & Rosychuk, R. J. (2026a). Stratified regression analysis of zero-truncated recurrent event data. *The Annals of Applied Statistics*, 20(1), 434-451.

Chen, A. A., Hu, X. J., Rosychuk, R. J., & Zeng, L. (2026b). Analyzing zero-truncated recurrent event data by stratified regression with time-varying coefficients. *Canadian Journal of Statistics*, e70064.

Hu, X. J., & Rosychuk, R. J. (2016). Imputation procedures for missing birthdates in administrative health data. *Statistics in Medicine*, 35(14), 2341-2355.

Examples

```
# When birthdates are unknown, obtain resampling-based variance estimates
# using standard normal multipliers..
strati.fit.uk.R <- strati.reg(base = my.base[,c(-1,-9)], visit = my.visit,
AGEUNIT = 6,ONUM = 108, WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")), etapara = 1e-4, reptmax = 50,
  windowKH = rep(9,108), tauL = 9, tauR = 105, method = "all",
  N_star =  my.agge.info, n=50000,
  B = 200, seeds = 12345, com.coef = 0, com.lamb = 0, knownBirth = FALSE,
  reptw = 100, resampling = TRUE,r.distn = "std Normal",
  mycovars = c("MaleInd","CityAInd","CityBInd"), save.res = FALSE,
  output.lambda = TRUE)

  # When birthdates are known, obtain resampling-based variance estimates
  # using Poisson multipliers with variance 1.
  strati.fit.k.R.p <- strati.reg(base = my.base[,c(-1,-7,-8)], visit = my.visit,
AGEUNIT = 6,ONUM = 108, WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")), etapara = 1e-4, reptmax = 50,
  windowKH = rep(9,108), tauL = 9, tauR = 105, method = "all",
  N_star =  my.agge.info, n=50000,
  B = 200, seeds = 12345, com.coef = 0, com.lamb = 0, knownBirth = TRUE,
  reptw = 100, resampling = TRUE,r.distn = "Poisson",
  mycovars = c("MaleInd","CityAInd","CityBInd"), save.res = FALSE,
  output.lambda = TRUE)

  # When birthdates are known, obtain bootstrap-based variance estimates.
  strati.fit.k.B <- strati.reg(base = my.base[,c(-1,-7,-8)], visit = my.visit,
AGEUNIT = 6,ONUM = 108, WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")), etapara = 1e-4, reptmax = 50,
  windowKH = rep(9,108), tauL = 9, tauR = 105, method = "all",
  N_star =  my.agge.info, n=50000,
  B = 200, seeds = 12345, com.coef = 0, com.lamb = 0, knownBirth = TRUE,
  reptw = 100, resampling = FALSE,r.distn = "Poisson",
  mycovars = c("MaleInd","CityAInd","CityBInd"), save.res = FALSE,
  output.lambda = TRUE)

  # Fit models with common regression coefficients across strata.
  strati.fit.k.R.p.combreg <- strati.reg(base = my.base[,c(-1,-7,-8)], visit = my.visit,
AGEUNIT = 6,ONUM = 108, WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")), etapara = 1e-4, reptmax = 50,
  windowKH = rep(9,108), tauL = 9, tauR = 105, method = "all",
  N_star =  my.agge.info, n=50000,
  B = 200, seeds = 12345, com.coef = 1, com.lamb = 0, knownBirth = TRUE,
  reptw = 100, resampling = TRUE,r.distn = "Poisson",
  mycovars = c("MaleInd","CityAInd","CityBInd"), save.res = FALSE,
  output.lambda = TRUE)

  # Fit models with a common baseline function across strata.
  strati.fit.k.R.p.combB <- strati.reg(base = my.base[,c(-1,-7,-8)], visit = my.visit,
AGEUNIT = 6,ONUM = 108, WINDOWL = as.numeric(as.Date("2010-04-01")),
  WINDOWR = as.numeric(as.Date("2025-03-31")), etapara = 1e-4, reptmax = 50,
  windowKH = rep(9,108), tauL = 9, tauR = 105, method = "all",
```

```
N_star =  my.agge.info, n=50000,
B = 200, seeds = 12345, com.coef = 0, com.lamb = 1, knownBirth = TRUE,
reptw = 100, resampling = TRUE,r.distn = "Poisson",
mycovars = c("MaleInd","CityAInd","CityBInd"), save.res = FALSE,
output.lambda = TRUE)
```

Index

∗ **datasets**
    my.agge.info, 4
    my.base, 4
    my.visit, 5

birth.interval, 3

my.agge.info, 4
my.base, 4
my.visit, 5

nonpara_esti, 5, *12*
nonstrati.reg, 8, *14*, *16*

plot_nonpar, 12
plot_nonstrati_baseline, 13
plot_nonstrati_coef, 15
plot_stratified_baseline, 17
plot_stratified_coef, 20

strati.reg, *18–21*, 23

ZTRecur *(*ZTRecur-package*)*, 2
ZTRecur-package, 2