Package ‘ahw’

December 22, 2022

Type Package

Title Calculates Continuous Time Likelihood Ratio Weights Assuming Multiplicative Intensity Models and Additive Hazard Models

Version 0.1.0

Depends R (>= 3.5.0)

Imports methods, timereg, plyr, data.table (>= 1.10.4)

LazyData Yes

Description Estimates continuous time weights for performing causal survival analysis. For instance, weighted Nelson-Aalen or Kaplan-Meier estimates can be given a causal interpretation. See Ryalen, Stensrud, and Røysland (2019) <doi:10.1007/s10985-019-09468-y> and Ryalen (2019) <`https://www.duo.uio.no/handle/10852/70353`> for theory and examples.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

Suggests testthat (>= 3.0.0), survival

Config/testthat/edition 3

NeedsCompilation yes

Author Kjetil Røysland [ths],
Pål Christie Ryalen [aut, cre]
(<`https://orcid.org/0000-0002-3236-6782`>)

Maintainer Pål Christie Ryalen <`p.c.ryalen@medisin.uio.no`>

Repository CRAN

Date/Publication 2022-12-22 20:40:05 UTC

R topics documented:

addNoiseAtEventTimes 2
ahw 3

fFrame 3
fr1 3
makeContWeights 3
naReplace 6
plotContWeights 7
refineTable 7
Tmat 8
weightPredict 8

Index **10**

`addNoiseAtEventTimes` *Tie removal*

Description

Removes ties from by adding noise at tied times

Usage

```
addNoiseAtEventTimes(fr, id, from, to)
```

Arguments

`fr`	Data on long format
`id`	Name of column in `dataFr` that identifies individuals
`from`	Name of the variable that contains the name of start state for each at-risk interval
`to`	Name of column with stop time of the at risk period

Value

Longitudinally refined `data.table` of the input `data.table` `fr` with random noise added for tiebreaking.

Author(s)

Pål Christie Ryalen <p.c.ryalen@medisin.uio.no>

Examples

```
library(data.table)
data(fr1)
fr1 <- as.data.table(fr1)
fr1 <- addNoiseAtEventTimes(fr1)
head(fr1)
```

`ahw` *Re-weighting Point Processes Based on Additive Hazard Models*

Description

Estimates continuous time weights for performing causal survival analysis. For instance, weighted Nelson-Aalen or Kaplan-Meier estimates can be given a causal interpretation.

`fFrame` *fFrame*

Description

fFrame

`fr1` *fr1*

Description

fr1

`makeContWeights` *Continuous time weight estimation based on* `aalen.predict`

Description

Refines data longitudinally in order to do estimate parameters(e.g. Nelson-Aalen or Kaplan-Meier) using continuous time weights. The weights can be assessed using the plot option.

Usage

```
makeContWeights(
  faFit,
  cfaFit,
  dataFr,
  atRiskState,
  eventState,
  startTimeName,
  stopTimeName,
  startStatusName,
  endStatusName,
  idName,
  b,
  weightRange = c(0, 10),
  willPlotWeights = TRUE,
  withLeftLim = FALSE
)
```

Arguments

`faFit`	The `aalen` fit for the factual hazard
`cfaFit`	The `aalen` fit for the hypothetical hazard
`dataFr`	`data.frame` or `data.table` on long format
`atRiskState`	At risk state for the event of interest
`eventState`	State for the event of interest
`startTimeName`	Name of column with start time of the at risk period
`stopTimeName`	Name of column with stop time of the at risk period
`startStatusName`	Name of the column that contains the starting state for each interval
`endStatusName`	Name of the column that contains the end state for each interval
`idName`	Name of column in `dataFr` that identifies individuals
`b`	Smoothing bandwidth parameter
`weightRange`	Truncates weights outside this interval
`willPlotWeights`	Plot indicator
`withLeftLim`	Calculates left limit at jump if desired

Value

Longitudinally refined `data.table` of the initial `dataFr` with `weights` column added.

Author(s)

Pål Christie Ryalen <p.c.ryalen@medisin.uio.no>

References

https://arxiv.org/abs/1802.01946

Examples

```
library(data.table)
library(timereg)

# fr1 is a longitudinal data set with subjects that are diagnosed at time 0, and may
# be treated as time evolves. Subjects can die before receiving treatment
# The method assumes there are no tied event times in the observed data. Although there are no
# tied event times in fr1, we use the function addNoiseAtEventTimes() for illustration here
fr1 <- as.data.table(fr1)
fr1 <- addNoiseAtEventTimes(fr1)

# Time to treatment and death are confounded by the baseline variable L. We want to
# mimic a scenario where time to treatment is randomized (and does not depend on L):
fr1_diag <- fr1[fr1$from.state == "diag", ]
fFit <- aalen(
  Surv(from, to, to.state == "treat") ~ 1 + L, data = fr1_diag, n.sim = 50L,
  robust = 0
)
cfFit <- aalen(
  Surv(from, to, to.state == "treat") ~ 1, data = fr1_diag, n.sim = 50L,
  robust = 0
)

# We calculate and plot the weights
frame1 <- makeContWeights(fFit, cfFit, fr1, "diag", "treat", "from", "to",
  "from.state", "to.state", "id",
  b = 0.4,
  weightRange = c(0, 5)
)

# We fit a weighted model for the outcome. A is a treatment indicator (A=1 means treated).
a1 <- aalen(
  Surv(from, to, to.state == "death") ~ 1 + A, data = frame1,
  weights = frame1$weights, n.sim = 50L, robust = 0
)

# We plot the A coefficient from the weighted regression,
# and compare with the true hypothetical coefficient
plot(a1$cum[, c(1, 3)],
  type = "s", ylim = c(-1.2, 0.5), xlim = c(0, 5),
  main = "Weighted additive hazard treatment coefficient"
)
lines(Tmat, col = 2)
legend("bottomleft", c("weighted estimate", "hypothetical treatment coef"),
  lty = 1, col = c(1, 2), bty = "n"
)

# Next we consider an example with dependent censoring.
# Subjects are censored depending on a baseline variable u. We wish to mimic the
# cumulative hazard for death we would have seen if the censoring were independent.

faFit <- aalen(
  Surv(from, to, to.state == "Censored") ~ 1 + u, data = fFrame, n.sim = 50L,
  robust = 0
)
cfaFit <- aalen(
  Surv(from, to, to.state == "Censored") ~ 1, data = fFrame, n.sim = 50L,
  robust = 0
)

frame <- makeContWeights(
  faFit, cfaFit, fFrame, "Alive", "Censored", "from", "to", "from.state",
  "to.state", "id", 100
)

fMod <- aalen(
  Surv(from, to, to.state == "Dead") ~ 1, data = fFrame, n.sim = 50L,
  robust = 0
)
wMod <- aalen(
  Surv(from, to, to.state == "Dead") ~ 1, data = frame, weights = frame$weights,
  n.sim = 50L, robust = 0
)

plot(fMod$cum, type = "s", main = "Nelson-Aalen for death", ylab = "")
lines(wMod$cum, type = "s", col = "red")
legend("topleft", c("factual", "weighted factual"), lty = 1, col = c(1, "red"), bty = "n")
```

`naReplace` *Replaces NA-values in* `vec` *with last non-NA value*

Description

Assumes first element is non-NA

Usage

```
naReplace(vec)
```

Arguments

`vec`	Vector of any type

Value

`numeric` `vector` with each `NA` entry replaced with the last previous non-`NA` entry.

Note

Can be replaced by `link[zoo]{na.locf0}`

Author(s)

Pål Christie Ryalen <p.c.ryalen@medisin.uio.no>

Examples

```
naReplace(c(1, 2, 3, NA, NA, 4))
naReplace(c("text", NA, NA))
```

`plotContWeights` *Plots mean and individual weight trajectories*

Description

Plotting all the individual weight trajectories in fr, along with the mean. Plots weights for assessment.

Usage

```
plotContWeights(
  fr,
  stopTimeName = "to",
  startStatusName = "from.state",
  endStatusName = "to.state",
  idName = "id"
)
```

Arguments

`fr`	Data with weight column
`stopTimeName`	Name of column with stop time of the at risk period
`startStatusName`	Name of the variable that contains the name of start state for each at-risk interval
`endStatusName`	Name of the variable that contains the name of end state for each at-risk interval
`idName`	Name of column in `dataFr` that identifies individuals

Value

No return value.

Author(s)

Pål Christie Ryalen <p.c.ryalen@medisin.uio.no>

`refineTable` *Expands data.table*

Description

Refines dataFr so that each individual at risk get a row for each of the provided event times

Usage

```
refineTable(dataFr, atRiskState, eventState)
```

Arguments

`dataFr`	data.frame or data.table on long format
`atRiskState`	At risk state(s)
`eventState`	Observed event times

Value

data.table

Author(s)

Pål Christie Ryalen <p.c.ryalen@medisin.uio.no>

`Tmat` *Tmat*

Description

Tmat

`weightPredict` *Continuous time weight estimation based on* `predict.aalen`

Description

Extracts cumulative hazard estimates for each individual. Each individual receives a weight process evaluated at pre-specified time points. The weight process is estimated as a cumulative product involving estimated cumulative hazard increments, and a hazard ratio estimated using a smoothing parameter b.

Usage

```
weightPredict(fPred, cfPred, wtFrame, ids, eventTimes, eventIds, b)
```

Arguments

`fPred`	`predict.aalen` object of the factual fit
`cfPred`	`predict.aalen` object of the counterfactual fit
`wtFrame`	`data.frame` or `data.table` for the at risk individuals
`ids`	All individuals in the data set
`eventTimes`	Observed event times
`eventIds`	Individuals that experience the event
`b`	Smoothing parameter

Value

data.table

Author(s)

Pål Christie Ryalen <p.c.ryalen@medisin.uio.no>

Index

aalen, *4*
addNoiseAtEventTimes, 2
ahw, 3

data.frame, *4*, *8*
data.table, *2*, *4*, *8*

fFrame, 3
fr1, 3

makeContWeights, 3

naReplace, 6

plotContWeights, 7
predict.aalen, *8*

refineTable, 7

Tmat, 8

weightPredict, 8