Package 'recurrentpseudo'

May 9, 2026

Title Creates Pseudo-Observations and Analysis for Recurrent Event Data

Version 1.0.0

Description Computation of one-, two- and three-dimensional pseudo-observations based on recurrent events and terminal events. Generalised linear models are fitted using generalised estimating equations. Technical details on the bivariate procedure can be found in ``Bivariate pseudo-observations for recurrent event analysis with terminal events" (Furberg et al., 2021) <doi:10.1007/s10985-021-09533-5>.

License GPL (>= 2)

URL https://github.com/JulieKFurberg/recurrentpseudo

Encoding UTF-8

RoxygenNote 7.2.1

Imports survival, geepack, stringr, prodlim

Depends magrittr, dplyr

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Julie Kjærulff Furberg [aut, cre] (ORCID: <https://orcid.org/0000-0001-8785-1462>)

Maintainer Julie Kjærulff Furberg <julie_furberg@hotmail.com>

Repository CRAN

Date/Publication 2022-09-19 14:16:16 UTC

Contents

`pseudo.geefit` *Function that makes GEE model fit for recurrent pseudo-observations*

Description

This function fits a GEE model based on pseudo-observations of the marginal mean function, and the survival probability or cumulative incidences of two death causes as returned by `pseudo.onedim()` (marginal mean function), or `pseudo.twodim()` (marginal mean function and survival probability), or `pseudo.threedim()` (marginal mean function and cumulative incidences of death causes 1 and 2)

Usage

```
pseudo.geefit(pseudodata, covar_names)
```

Arguments

`pseudodata`	Data set containing pseudo-observations. Expecting output from pseudo.twodim()
`covar_names`	Vector with covariate names to be found in "pseudodata". E.g. covar_names = c("Z", "Z1")

Value

An object of class `pseudo.geefit`.

- `xi` contains the estimated model parameters
- `sigma` contains the estimated variance matrix corresponding to `xi`

References

Furberg, J.K., Andersen, P.K., Korn, S. et al. Bivariate pseudo-observations for recurrent event analysis with terminal events. Lifetime Data Anal (2021). https://doi.org/10.1007/s10985-021-09533-5

Examples

```
# Bladder cancer data from survival package
require(survival)

# Make a three level status variable
bladder1$status3 <- ifelse(bladder1$status %in% c(2, 3), 2, bladder1$status)

# Add one extra day for the two patients with start=stop=0
# subset(bladder1, stop <= start)
bladder1[bladder1$id == 1, "stop"] <- 1
bladder1[bladder1$id == 49, "stop"] <- 1
```

```
# Restrict the data to placebo and thiotepa
bladdersub <- subset(bladder1, treatment %in% c("placebo", "thiotepa"))

# Make treatment variable two-level factor
bladdersub$Z <- as.factor(ifelse(bladdersub$treatment == "placebo", 0, 1))
levels(bladdersub$Z) <- c("placebo", "thiotepa")

head(bladdersub)

# Two-dimensional (bivariate pseudo-obs) model fit

# Computation of pseudo-observations
pseudo_bladder_2d <- pseudo.twodim(tstart = bladdersub$start,
                                   tstop = bladdersub$stop,
                                   status = bladdersub$status3,
                                   id = bladdersub$id,
                                   covar_names = "Z",
                                   tk = c(30),
                                   data = bladdersub)

# Data in wide format
head(pseudo_bladder_2d$outdata)

# Data in long format
head(pseudo_bladder_2d$outdata_long)

# GEE fit
fit_bladder_2d <- pseudo.geefit(pseudodata = pseudo_bladder_2d,
                                covar_names = c("Z"))
fit_bladder_2d
```

`pseudo.onedim` *Function that computes univariate pseudo-observations*

Description

This function computes univariate pseudo-observations of the marginal mean function (in the presence of terminal events)

Usage

```
pseudo.onedim(tstart, tstop, status, covar_names, id, tk, data)
```

Arguments

`tstart`	Start time - expecting counting process notation
`tstop`	Stop time - expecting counting process notation
`status`	Status variable (0 = censoring, 1 = recurrent event, 2 = death)
`covar_names`	Vector containing names of covariates intended for further analysis

id	ID variable for subject
tk	Vector of time points to calculate pseudo-observations at
data	Data set which contains variables of interest

Value

An object of class pseudo.onedim.

- outdata contains the semi-wide version of the computed pseudo-observations (one row per time, tk, per id).
- outdata_long contains the long version of the computed pseudo-observations (one row per observation, several per id).
- indata contains the input data which the pseudo-observations are based on.
- ts vector with time points used for computation of pseudo-observations.
- k number of time points used for computation of pseudo-observations (length(ts)).

References

Furberg, J.K., Andersen, P.K., Korn, S. et al. Bivariate pseudo-observations for recurrent event analysis with terminal events. Lifetime Data Anal (2021). https://doi.org/10.1007/s10985-021-09533-5

Examples

```
# Example: Bladder cancer data from survival package
require(survival)

# Make a three level status variable
bladder1$status3 <- ifelse(bladder1$status %in% c(2, 3), 2, bladder1$status)

# Add one extra day for the two patients with start=stop=0
# subset(bladder1, stop <= start)
bladder1[bladder1$id == 1, "stop"] <- 1
bladder1[bladder1$id == 49, "stop"] <- 1

# Restrict the data to placebo and thiotepa
bladdersub <- subset(bladder1, treatment %in% c("placebo", "thiotepa"))

# Make treatment variable two-level factor
bladdersub$Z <- as.factor(ifelse(bladdersub$treatment == "placebo", 0, 1))
levels(bladdersub$Z) <- c("placebo", "thiotepa")
head(bladdersub)

# Pseudo observations
pseudo_bladder_1d <- pseudo.onedim(tstart = bladdersub$start,
                                   tstop = bladdersub$stop,
                                   status = bladdersub$status3,
                                   id = bladdersub$id,
                                   covar_names = "Z",
                                   tk = c(30),
```

```
                                    data = bladdersub)
head(pseudo_bladder_1d$outdata)

# GEE fit
fit_bladder_1d <- pseudo.geefit(pseudodata = pseudo_bladder_1d,
                                covar_names = c("Z"))
fit_bladder_1d
```

`pseudo.threedim` *Function that computes 3-dim pseudo-observations*

Description

This function computes 3-dimensional pseudo-observations of the marginal mean function (in the presence of terminal events) and cumulative incidences of death causes 1 and 2

Usage

```
pseudo.threedim(tstart, tstop, status, covar_names, id, tk, data, deathtype)
```

Arguments

`tstart`	Start time - expecting counting process notation
`tstop`	Stop time - expecting counting process notation
`status`	Status variable (0 = censoring, 1 = recurrent event, 2 = death)
`covar_names`	Vector containing names of covariates intended for further analysis
`id`	ID variable for subject
`tk`	Vector of time points to calculate pseudo-observations at
`data`	Data set which contains variables of interest
`deathtype`	Type of death (cause 1 or cause 2)

Value

An object of class `pseudo.threedim`.

- `outdata` contains the semi-wide version of the computed pseudo-observations (one row per time, tk, per id).
- `outdata_long` contains the long version of the computed pseudo-observations (one row per observation, several per id).
- `indata` contains the input data which the pseudo-observations are based on.
- `ts` vector with time points used for computation of pseudo-observations.
- k number of time points used for computation of pseudo-observations (length(ts)).

References

Furberg, J.K., Andersen, P.K., Korn, S. et al. Bivariate pseudo-observations for recurrent event analysis with terminal events. Lifetime Data Anal (2021). https://doi.org/10.1007/s10985-021-09533-5

Examples

```
# Example: Bladder cancer data from survival package
require(survival)

# Make a three level status variable
bladder1$status3 <- ifelse(bladder1$status %in% c(2, 3), 2, bladder1$status)

# Add one extra day for the two patients with start=stop=0
# subset(bladder1, stop <= start)
bladder1[bladder1$id == 1, "stop"] <- 1
bladder1[bladder1$id == 49, "stop"] <- 1

# Restrict the data to placebo and thiotepa
bladdersub <- subset(bladder1, treatment %in% c("placebo", "thiotepa"))

# Make treatment variable two-level factor
bladdersub$Z <- as.factor(ifelse(bladdersub$treatment == "placebo", 0, 1))
levels(bladdersub$Z) <- c("placebo", "thiotepa")
head(bladdersub)

# Add deathtype variable to bladder data
# Deathtype = 1 (bladder disease death), deathtype = 2 (other death reason)
bladdersub$deathtype <- with(bladdersub, ifelse(status == 2, 1, ifelse(status == 3, 2, 0)))
table(bladdersub$deathtype, bladdersub$status)

# Pseudo-observations
pseudo_bladder_3d <- pseudo.threedim(tstart = bladdersub$start,
                                     tstop = bladdersub$stop,
                                     status = bladdersub$status3,
                                     id = bladdersub$id,
                                     deathtype = bladdersub$deathtype,
                                     covar_names = "Z",
                                     tk = c(30),
                                     data = bladdersub)
pseudo_bladder_3d

# GEE fit
fit_bladder_3d <- pseudo.geefit(pseudodata = pseudo_bladder_3d,
                                covar_names = c("Z"))
fit_bladder_3d
```

`pseudo.twodim` *Function that computes bivariate pseudo-observations*

Description

This function computes bivariate pseudo-observations of the marginal mean function (in the presence of terminal events) and the survival probability

Usage

```
pseudo.twodim(tstart, tstop, status, covar_names, id, tk, data)
```

Arguments

tstart	Start time - expecting counting process notation
tstop	Stop time - expecting counting process notation
status	Status variable (0 = censoring, 1 = recurrent event, 2 = death)
covar_names	Vector containing names of covariates intended for further analysis
id	ID variable for subject
tk	Vector of time points to calculate pseudo-observations at
data	Data set which contains variables of interest

Value

An object of class `pseudo.twodim`.

- `outdata` contains the semi-wide version of the computed pseudo-observations (one row per time, tk, per id).
- `outdata_long` contains the long version of the computed pseudo-observations (one row per observation, several per id).
- `indata` contains the input data which the pseudo-observations are based on.
- `ts` vector with time points used for computation of pseudo-observations.
- k number of time points used for computation of pseudo-observations (length(ts)).

References

Furberg, J.K., Andersen, P.K., Korn, S. et al. Bivariate pseudo-observations for recurrent event analysis with terminal events. Lifetime Data Anal (2021). https://doi.org/10.1007/s10985-021-09533-5

Examples

```
# Example: Bladder cancer data from survival package
require(survival)

# Make a three level status variable
bladder1$status3 <- ifelse(bladder1$status %in% c(2, 3), 2, bladder1$status)

# Add one extra day for the two patients with start=stop=0
# subset(bladder1, stop <= start)
bladder1[bladder1$id == 1, "stop"] <- 1
```

```
bladder1[bladder1$id == 49, "stop"] <- 1

# Restrict the data to placebo and thiotepa
bladdersub <- subset(bladder1, treatment %in% c("placebo", "thiotepa"))

# Make treatment variable two-level factor
bladdersub$Z <- as.factor(ifelse(bladdersub$treatment == "placebo", 0, 1))
levels(bladdersub$Z) <- c("placebo", "thiotepa")
head(bladdersub)

# Pseudo observations
pseudo_bladder_2d <- pseudo.twodim(tstart = bladdersub$start,
                                   tstop = bladdersub$stop,
                                   status = bladdersub$status3,
                                   id = bladdersub$id,
                                   covar_names = "Z",
                                   tk = c(30),
                                   data = bladdersub)
head(pseudo_bladder_2d$outdata)

# GEE fit
fit_bladder_2d <- pseudo.geefit(pseudodata = pseudo_bladder_2d,
                                covar_names = c("Z"))
fit_bladder_2d
```

Index