Package 'etm'

April 12, 2025

Type Package

Title Empirical Transition Matrix

Version 1.1.2

Description The etm (empirical transition matrix) package permits to estimate the matrix of transition probabilities for any time-inhomogeneous multi-state model with finite state space using the Aalen-Johansen estimator. Functions for data preparation and for displaying are also included (Allignol et al., 2011 <doi:10.18637/jss.v038.i04>). Functionals of the Aalen-Johansen estimator, e.g., excess length-of-stay in an intermediate state, can also be computed (Allignol et al. 2011 <doi:10.1007/s00180-010-0200-x>).

License MIT + file LICENSE

Depends R (>= 3.0.0)

Imports survival, lattice, data.table, Rcpp (>= 0.11.4)

Suggests ggplot2, kmi, geepack

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Author Arthur Allignol [aut],
Mark Clements [cre, aut]

Maintainer Mark Clements <`mark.clements@ki.se`>

Repository CRAN

Date/Publication 2025-04-12 08:40:02 UTC

Contents

`abortion` *Pregnancies exposed to coumarin derivatives*

Description

Outcomes of pregnancies exposed to coumarin derivatives. The aim is to investigate whether exposition to coumarin derivatives increases the probability of spontaneous abortions. Apart from spontaneous abortion, pregnancy may end in induced abortion or live birth, leading to a competing risks situation. Moreover, data are left-truncated as women usually enter the study several weeks after conception.

Usage

```
data(abortion)
```

Format

A data frame with 1186 observations on the following 5 variables.

`id` Identification number

`entry` Entry times into the cohort

`exit` Event times

`group` Group. 0: control, 1: exposed to coumarin derivatives

`cause` Cause of failure. 1: induced abortion, 2: life birth, 3: spontaneous abortion

Source

Meiester, R. and Schaefer, C (2008). Statistical methods for estimating the probability of spontaneous abortion in observational studies – Analyzing pregnancies exposed to coumarin derivatives. Reproductive Toxicology, 26, 31–35

Examples

```
data(abortion)
```

`clos` *Change in Length of Stay*

Description

The function estimates the expected change in length of stay (LOS) associated with an intermediate event.

Usage

```
clos(x, aw, ratio, ...)
## S3 method for class 'etm'
clos(x, aw = FALSE, ratio = FALSE, ...)
## S3 method for class 'msfit'
clos(x, aw = FALSE, ratio = FALSE, cox_model, ...)
```

Arguments

`x`	An object of class `etm`. Argument `delta.na` in `etm` must be set to `TRUE` in order to use this function.
`aw`	Logical. Whether to compute the expected change of LOS using alternative weighting. Default is `FALSE`.
`ratio`	Logical. Compute the ratio of the expected length-of-stay given instermediate event status instead of a difference. Default value is `FALSE`
`cox_model`	TODO
`...`	Further arguments

Details

The approach for evaluating the impact of an intermediate event on the expected change in length of stay is based on Schulgen and Schumacher (1996). They suggested to consider the difference of the expected subsequent stay given infectious status at time s.

Extensions to the methods of Schulgen and Schumacher and the earlier implementation in the **changeLOS** include the possibility to compute the extra length of stay both for competing endpoints and the more simple case of one absorbing state, as well as the possibility to compute this quantity for left-truncated data.

Value

An object of class `clos.etm` with the following components:

`e.phi`	Change in length of stay
`phi.case`	Estimates of $E(\text{LOS} \mid X_s = \text{intermediate event})$ for all observed transition times s, where $X_s$denotes the state by time s
`phi.control`	Estimates of $E(\text{LOS} \mid X_s = \text{initial state})$ for all observed transition times s.
`e.phi2`	Weighted average of the difference between `phi2.case` and `phi2.control`.
`phi2.case`	Estimates of $E(\text{LOS}\mathbf{1}(X_{\text{LOS}} = \text{discharge}) \mid X_s = \text{intermediate event})$, where $\mathbf{1}$ denotes the indicator function.
`phi2.control`	$E(\text{LOS}\mathbf{1}(X_{\text{LOS}} = \text{discharge}) \mid X_s = \text{initial state})$.
`e.phi3`	Weighted average of the difference between `phi3.case` and `phi3.control`.
`phi3.case`	Estimates of $E(\text{LOS}\mathbf{1}(X_{\text{LOS}} = \text{death}) \mid X_s = \text{intermediate event})$.
`phi3.control`	$E(\text{LOS}\mathbf{1}(X_{\text{LOS}} = \text{death}) \mid X_s = \text{initial state})$.
`weights`	Weights used to compute the weighted averages.
`w.time`	Times at which the weights are computed.
`time`	All transition times.
`e.phi.weights.1`	Expected change in LOS using `weights.1`
`e.phi.weights.other`	Expected change in LOS using `weights.other`
`weights.1`	Weights corresponding to the conditional waiting time in the intial state given one experiences the intermediate event.
`weights.other`	Weights corresponding to the conditional waiting time given one does not experience the intermediate event.

Author(s)

Arthur Allignol <`arthur.allignol@gmail.com`>, Matthias Wangler, Jan Beyersmann

References

G Schulgen and M Schumacher (1996). Estimation of prolongation of hospital stay attributable to nosocomial infections. *Lifetime Data Analysis* 2, 219-240.

J Beyersmann, P Gastmeier, H Grundmann, S Baerwolf, C Geffers, M Behnke, H Rueden, and M Schumacher (2006). Use of Multistate Models to Assess Prolongation of Intensive Care Unit Stay Due to Nosocomial Infection. *Infection Control and Hospital Epidemiology* 27, 493-499.

Allignol A, Schumacher M, Beyersmann J: Estimating summary functionals in multistate models with an application to hospital infection data. *Computation Stat*, 2011; 26: 181-197.

M Wrangler, J Beyersmann and M Schumacher (2006). changeLOS: An R-package for change in length of hospital stay based on the Aalen-Johansen estimator. *R News* 6(2), 31–35.

See Also

`etm`

Examples

```
data(los.data)

## putting los.data in the long format
my.observ <- prepare.los.data(x=los.data)

tra <- matrix(FALSE, 4, 4)
tra[1, 2:4] <- TRUE
tra[2, 3:4] <- TRUE

tr.prob <- etm(my.observ, c("0","1","2","3"), tra, NULL, 0)

cLOS <- etm::clos(tr.prob)
plot(cLOS)


### Compute bootstrapped SE

## function that performs the bootstrap
## nboot: number of bootstrap samples. Other arguments are as in etm()
boot.clos <- function(data, state.names, tra, cens.name, s = 0, nboot) {
    res <- double(nboot)
    for (i in seq_len(nboot)) {
        index <- sample(unique(data$id), replace = TRUE)
        inds <- new.id <- NULL
        for (j in seq_along(index)){
            ind <- which(data$id == index[j])
            new.id <- c(new.id, rep(j, length(ind)))
            inds <- c(inds, ind)
        }
        dboot <- cbind(data[inds, ], new.id)
        dboot[, which(names(dboot) == "id")]
        dboot$id <- dboot$new.id
        tr.prob <- etm(dboot, state.names, tra, cens.name, s, cova = FALSE)
        res[i] <- etm::clos(tr.prob)$e.phi
    }
    res
}

## bootstrap
se <- sqrt(var(boot.clos(my.observ, c("0","1","2","3"), tra, NULL, 0,
                         nboot = 10)))
```

`closPseudo` *Pseudo Value Regression for the Extra Length-of-Stay*

Description

Pseudo Value Regression for the Extra Length-of-Stay

Usage

```
closPseudo(data, state.names, tra, cens.name, s = 0,
           formula, na.action,
           aw = FALSE, ratio = FALSE,
           ncores = 1,
           trick_ties = FALSE)
```

Arguments

`data`	data.frame of the form data.frame(id,from,to,time) or (id,from,to,entry,exit) **id:** patient id **from:** the state from where the transition occurs **to:** the state to which a transition occurs **time:** time when a transition occurs **entry:** entry time in a state **exit:** exit time from a state
`state.names`	A vector of characters giving the states names.
`tra`	A quadratic matrix of logical values describing the possible transitions within the multistate model.
`cens.name`	A character giving the code for censored observations in the column 'to' of `data`. If there is no censored observations in your data, put 'NULL'.
`s`	Starting value for computing the transition probabilities.
`formula`	A formula with the covariates at the right of a ~ operator. Leave the left part empty.
`na.action`	A function which indicates what should happen when the data contain 'NA's. The default is set by the 'na.action' setting of 'options', and is 'na.fail' if that is unset. The 'factory-fresh' default is 'na.omit'.
`aw`	Logical. Whether to compute the expected change of LOS using alternative weighting. Default is `FALSE`.
`ratio`	Logical. Compute the ratio of the expected length-of-stay given instermediate event status instead of a difference. Default value is `FALSE`
`ncores`	Number of cores used if doing parallel computation using the **parallel** package
`trick_ties`	If `TRUE`, pseudo values are computed only one per subject sharing the same entry, exit times / transition types.

Details

The function calculates the pseudo-observations for the extra length-of-stay for each individual. These pseudo-observations can then be used to fit a direct regression model using generalized estimating equation (e.g., package **geepack**).

Computation of the pseudo-observations can be parallelised using the `mclapply` function of the **parallel** package. See argument `ncores`.

Recent versions of R have changed the `data.frame` function, where the default for the `stringsAsFactors` argument from `TRUE` to `FALSE`. `etm` currently depends on the states being factors, so that the user should use `data.frame(..., stringsAsFactors=TRUE)`.

Value

An object of class `closPseudo` with the following components:

`pseudoData`	a data.frame containing `id`, computed pseudo values (see details) and the covariates as specified in the formula
`theta`	Estimates of excess LoS in the whole sample
`aw`	like in the function call
`call`	Function call

Author(s)

Arthur Allignol <`arthur.allignol@gmail.com`>

References

Andersen, P.K, Klein, J.P, Rosthoj, S. (2003). Generalised linear models for correlated pseudo-observations, with applications to multi-state models. *Biometrika*, 90(1):15–27.

See Also

`mclapply`, `clos`

Examples

```
if(require("kmi", quietly = TRUE)) {

    ## data in kmi package
    data(icu.pneu)
    my.icu.pneu <- icu.pneu

    my.icu.pneu <- my.icu.pneu[order(my.icu.pneu$id, my.icu.pneu$start), ]
    masque <- diff(my.icu.pneu$id)

    my.icu.pneu$from <- 0
    my.icu.pneu$from[c(1, masque) == 0] <- 1

    my.icu.pneu$to2 <- my.icu.pneu$event
    my.icu.pneu$to2[my.icu.pneu$status == 0] <- "cens"
    my.icu.pneu$to2[c(masque, 1) == 0] <- 1


    my.icu.pneu$to <- ifelse(my.icu.pneu$to2 %in% c(2, 3), 2,
                             my.icu.pneu$to2)

    my.icu.pneu <- my.icu.pneu[, c("id", "start", "stop", "from", "to",
                                   "to2", "age", "sex")]
    names(my.icu.pneu)[c(2, 3)] <- c("entry", "exit")

    ## computation of the pseudo-observations
    ## Not run:
```

```
        ps.icu.pneu <- closPseudo(my.icu.pneu, c("0", "1", "2"), tra_ill(), "cens",
                                  formula = ~ sex + age)

        ## regression model using geepack
        require(geepack)
        fit <- geeglm(ps.e.phi ~ sex + age, id = id, data = ps.icu.pneu$pseudoData,
                      family = gaussian)

        summary(fit)

## End(Not run)
} else {
    print("This example requires the kmi package")
}
```

`etm` *Computation of the empirical transition matrix*

Description

This function computes the empirical transition matrix, also called Aalen-Johansen estimator, of the transition probability matrix of any multistate model. The covariance matrix is also computed.

Usage

```
## S3 method for class 'data.frame'
etm(data, state.names, tra, cens.name, s, t = "last",
    covariance = TRUE, delta.na = TRUE, modif = FALSE,
    c = 1, alpha = NULL, strata, ...)
```

Arguments

`data`	data.frame of the form data.frame(id,from,to,time) or (id,from,to,entry,exit) **id:** patient id **from:** the state from where the transition occurs **to:** the state to which a transition occurs **time:** time when a transition occurs **entry:** entry time in a state **exit:** exit time from a state This data.frame is transition-oriented, *i.e.* it contains one row per transition, and possibly several rows per patient. Specifying an entry and exit time permits to take into account left-truncation.
`state.names`	A vector of characters giving the states names.
`tra`	A quadratic matrix of logical values describing the possible transitions within the multistate model.

`cens.name`	A character giving the code for censored observations in the column 'to' of `data`. If there is no censored observations in your data, put 'NULL'.
`s`	Starting value for computing the transition probabilities.
`t`	Ending value. Default is "last", meaning that the transition probabilities are computed over $(s, t]$, t being the last time in the data set.
`covariance`	Logical. Decide whether or not computing the covariance matrix. May be useful for, say, simulations, as the variance computation is a bit long. Default is TRUE.
`delta.na`	Logical. Whether to export the array containing the increments of the Nelson-Aalen estimator. Default is `TRUE`.
`modif`	Logical. Whether to apply the modification of Lai and Ying for small risk sets
`c`	Constant for the Lai and Ying modification. Either `c` contains only one value that will be used for all the states, otherwise `c` should be the same length as `state.names`.
`alpha`	Constant for the Lai and Ying modification. If NULL (the default) then only `c` is used and the Lai and Ying modification discards the event times for which $Y(t) \geq t$. Otherwise cn^{α} is used. It is recommanded to let `alpha` equal NULL for multistate models.
`strata`	Character vector giving variables on which to stratify the analysis.
`...`	Not used

Details

Data are considered to arise from a time-inhomogeneous Markovian multistate model with finite state space, and possibly subject to independent right-censoring and left-truncation.

The matrix of the transition probabilities is estimated by the Aalen-Johansen estimator / empirical transition matrix (Andersen et al., 1993), which is the product integral over the time period $(s, t]$ of I + the matrix of the increments of the Nelson-Aalen estimates of the cumulative transition hazards. The $(i, j) - th$ entry of the empirical transition matrix estimates the transition probability of being in state j at time t given that one has been in state j at time s.

The covariance matrix is computed using the recursion formula (4.4.19) in Anderson et al. (1993, p. 295). This estimator of the covariance matrix is an estimator of the Greenwood type.

If the multistate model is not Markov, but censorship is entirely random, the Aalen-Johansen estimator still consistently estimates the state occupation probabilities of being in state i at time t (Datta & Satten, 2001; Glidden, 2002)

Recent versions of R have changed the `data.frame` function, where the default for the `stringsAsFactors` argument from `TRUE` to `FALSE`. `etm` currently depends on the states being factors, so that the user should use `data.frame(..., stringsAsFactors=TRUE)`.

Value

`est`	Transition probability estimates. This is a 3 dimension array with the first dimension being the state from where transitions occur, the second the state to which transitions occur, and the last one being the event times.

`cov`	Estimated covariance matrix. Each cell of the matrix gives the covariance between the transition probabilities given by the rownames and the colnames, respectively.
`time`	Event times at which the transition probabilities are computed. That is all the observed times between $(s, t]$.
`s`	Start of the time interval.
`t`	End of the time interval.
`trans`	A `data.frame` giving the possible transitions.
`state.names`	A vector of character giving the state names.
`cens.name`	How the censored observation are coded in the data set.
`n.risk`	Matrix indicating the number of individuals at risk just before an event
`n.event`	Array containing the number of transitions at each times
`delta.na`	A 3d array containing the increments of the Nelson-Aalen estimator.
`ind.n.risk`	When `modif` is true, risk set size for which the indicator function is 1

If the analysis is stratified, a list of `etm` objects is returned.

Note

Transitions into a same state, mathematically superfluous, are not allowed. If transitions into the same state are detected in the data, the function will stop. Equally, `diag(tra)` must be set to FALSE, see the example below.

Author(s)

Arthur Allignol, `<arthur.allignol@gmail.com>`

References

Beyersmann J, Allignol A, Schumacher M: Competing Risks and Multistate Models with R (Use R!), Springer Verlag, 2012 (Use R!)

Allignol, A., Schumacher, M. and Beyersmann, J. (2011). Empirical Transition Matrix of Multi-State Models: The etm Package. *Journal of Statistical Software*, 38.

Andersen, P.K., Borgan, O., Gill, R.D. and Keiding, N. (1993). *Statistical models based on counting processes*. Springer Series in Statistics. New York, NY: Springer.

Aalen, O. and Johansen, S. (1978). An empirical transition matrix for non-homogeneous Markov chains based on censored observations. *Scandinavian Journal of Statistics*, 5: 141-150.

Gill, R.D. and Johansen, S. (1990). A survey of product-integration with a view towards application in survival analysis. *Annals of statistics*, 18(4): 1501-1555.

Datta, S. and Satten G.A. (2001). Validity of the Aalen-Johansen estimators of stage occupation probabilities and Nelson-Aalen estimators of integrated transition hazards for non-Markov models. *Statistics and Probability Letters*, 55(4): 403-411.

Glidden, D. (2002). Robust inference for event probabilities with non-Markov data. *Biometrics*, 58: 361-368.

See Also

`print.etm`, `summary.etm`, `sir.cont`, `xyplot.etm`

Examples

```
data(sir.cont)

# Modification for patients entering and leaving a state
# at the same date
# Change on ventilation status is considered
# to happen before end of hospital stay
sir.cont <- sir.cont[order(sir.cont$id, sir.cont$time), ]
for (i in 2:nrow(sir.cont)) {
  if (sir.cont$id[i]==sir.cont$id[i-1]) {
    if (sir.cont$time[i]==sir.cont$time[i-1]) {
      sir.cont$time[i-1] <- sir.cont$time[i-1] - 0.5
    }
  }
}

### Computation of the transition probabilities
# Possible transitions.
tra <- matrix(ncol=3,nrow=3,FALSE)
tra[1, 2:3] <- TRUE
tra[2, c(1, 3)] <- TRUE

# etm
tr.prob <- etm(sir.cont, c("0", "1", "2"), tra, "cens", 1)

tr.prob
summary(tr.prob)

# plotting
if (require("lattice")) {
xyplot(tr.prob, tr.choice=c("0 0", "1 1", "0 1", "0 2", "1 0", "1 2"),
       layout=c(2, 3), strip=strip.custom(bg="white",
         factor.levels=
     c("0 to 0", "1 to 1", "0 to 1", "0 to 2", "1 to 0", "1 to 2")))
}

### example with left-truncation

data(abortion)

# Data set modification in order to be used by etm
names(abortion) <- c("id", "entry", "exit", "from", "to")
abortion$to <- abortion$to + 1

## computation of the matrix giving the possible transitions
tra <- matrix(FALSE, nrow = 5, ncol = 5)
tra[1:2, 3:5] <- TRUE
```

```
## etm
fit <- etm(abortion, as.character(0:4), tra, NULL, s = 0)

## plot
xyplot(fit, tr.choice = c("0 0", "1 1", "0 4", "1 4"),
       ci.fun = c("log-log", "log-log", "cloglog", "cloglog"),
       strip = strip.custom(factor.levels = c("P(T > t) -- control",
                                              "P(T > t) -- exposed",
                                 "CIF spontaneous abortion -- control",
                                 "CIF spontaneous abortion --
exposed")))
```

`etmCIF` *Cumulative incidence functions of competing risks*

Description

`etmCIF` is a wrapper around the `etm` function for facilitating the computation of the cumulative incidence functions in the competing risks framework.

Usage

```
etmCIF(formula, data, etype, subset, na.action, failcode = 1)
```

Arguments

`formula`	A `formula` object, that must have a `Surv` object on the left of ~ operator, and a discrete covariate (or 1) on the right. The status indicator should be 1 (or TRUE) for an event (whatever the type of this event, 0 (or FALSE) for censored observations.)
`data`	A data.frame in which to interpret the terms of the formula
`etype`	Competing risks event indicator. When the status indicator is 1 (or TRUE) in the formula, `etype` describes the type of event, otherwise, for censored observation, the value of `etype` is ignored
`subset`	Expression saying that only a subset of the data should be used.
`na.action`	Missing-data filter function. Default is `options()$na.action`.
`failcode`	Indicates the failure type of interest. Default is one. This option is only relevant for some options of the `plot` function.

Details

This function computes the cumulative incidence functions in a competing risks setting using the `etm` machinery, without having to specify the matrix of possible transitions and using the more usual formula specification with `Surv`

Value

Returns a list of `etm` objects (1 per covariate level) plus additional informations:

`failcode`	As in function call
`call`	Function call
`X`	A matrix giving the name of the covariate (if present) and the levels of this covariate.

Author(s)

Arthur Allignol <`arthur.alignol@gmail.com`>

See Also

`etm`, `print.etmCIF`, `summary.etmCIF`, `plot.etmCIF`

Examples

```
data(abortion)

cif.ab <- etmCIF(survival::Surv(entry, exit, cause != 0) ~ group, abortion,
                 etype = cause, failcode = 3)

cif.ab

plot(cif.ab, ci.type = "bars", pos.ci = 24,
     col = c(1, 2), lty = 1, curvlab = c("Control", "Exposed"))
```

`etmprep` *Data transformation function for using etm*

Description

The function transforms a data set in the wide format (i.e., one raw per subject) into the long format (i.e., one raw per transition, and possibly several raws per subjects) in a suitable way for using the `etm` function

Usage

```
etmprep(time, status, data, tra, state.names, cens.name = NULL,
start = NULL, id = NULL, keep)
```

Arguments

`time`	A character vector giving the name of the columns containing the transition times or last follow-up times. The length of `time` have to be equal to the number of states, some elements may be NA. See Details.
`status`	A character vector giving the name of the columns indicating whether a state has been visited (0 if not, 1 otherwise).
`data`	A data frame in which to look for the columns specified in `time` and `status`.
`tra`	A quadratic matrix of logical values describing the possible transitions within the multistate model. The (i, j)th element of `tra` is TRUE if a transition from state i to state j is possible, FALSE otherwise. The diagonal must be set to FALSE.
`state.names`	A vector of characters giving the states names. If missing, state names are set to be 0:(number of states).
`cens.name`	A character string specifying how censored observations will be indicated in the new data set. Default is NULL, i.e., no censored observation.
`start`	A list containing two elements, `state` and `time`, giving the starting states and times for all individuals. Default is NULL, in which case all individuals are considered to start in the initial state at time 0.
`id`	A character string specifying in which column of `data` the user ids are. Default is NULL, and the ids will be `1:n`.
`keep`	A character vector indicating the column names of the covariate one might want to keep in the new data.frame.

Details

This function only works for irreversible acyclic Markov processes. Therefore, the multistate model will have initial states, into which no transition are possible. For these, NAs are allowed in `time` and `status`.

Value

The function returns a data.frame suitable for using the `etm` function. The data frame contains the following components:

`id`	Individual id number
`entry`	Entry time into a state
`exit`	Exit time from a state
`from`	State from which a transition occurs
`to`	State into which a transition occurs
`...`	Further columns specified in `keep`

Author(s)

Arthur Allignol, `<arthur.allignol@gmail.com>`

See Also

`etm`

Examples

```
### creation of fake data in the wild format, following an illness-death model
## transition times
tdisease <- c(3, 4, 3, 6, 8, 9)
tdeath <- c(6, 9, 8, 6, 8, 9)

## transition status
stat.disease <- c(1, 1, 1, 0, 0, 0)
stat.death <- c(1, 1, 1, 1, 1, 0)

## a covariate that we want to keep in the new data
cova <- rbinom(6, 1, 0.5)

dat <- data.frame(tdisease, tdeath,
                  stat.disease, stat.death,
                  cova, stringsAsFactors = TRUE)

## Possible transitions
tra <- matrix(FALSE, 3, 3)
tra[1, 2:3] <- TRUE
tra[2, 3] <- TRUE

## data preparation
newdat <- etmprep(c(NA, "tdisease", "tdeath"),
                  c(NA, "stat.disease", "stat.death"),
                  data = dat, tra = tra, cens.name = "cens")
```

`fourD` *Placebo data from the 4D study*

Description

Data from the placebo group of the 4D study. This study aimed at comparing atorvastatin to placebo for patients with type 2 diabetes and receiving hemodialysis in terms of cariovascular events. The primary endpoint was a composite of death from cardiac causes, stroke and non-fatal myocardial infarction. Competing event was death from other causes.

Usage

```
data(fourD)
```

Format

A data frame with 636 observations on the following 7 variables.

`id` Patients' id number

`sex` Patients' gender

`age` Patients' age

`medication` Character vector indicating treatment affiliation. Here only equal to `"Placebo"`

`status` Status at the end of the follow-up. 1 for the event of interest, 2 for death from other causes and 0 for censored observations

`time` Survival time

`treated` Numeric vector indicated whether patients are treated or not. Here always equal to zero

Source

Wanner, C., Krane, V., Maerz, W., Olschewski, M., Mann, J., Ruf, G., Ritz, E (2005). Atorvastatin in patients with type 2 diabetes mellitus undergoing hemodialysis. New England Journal of Medicine, 353(3), 238–248.

References

Allignol, A., Schumacher, M., Wanner, C., Dreschler, C. and Beyersmann, J. (2010). Understanding competing risks: a simulation point of view. Research report.

Examples

```
data(fourD)
```

`lines.etm` *Lines method for 'etm' objects*

Description

Lines method for `etm` objects

Usage

```
## S3 method for class 'etm'
lines(x, tr.choice, col = 1, lty,
      conf.int = FALSE, level = 0.95, ci.fun = "linear",
      ci.col = col, ci.lty = 3, ...)
```

Arguments

`x`	An object of class `etm`.
`tr.choice`	character vector of the form `c("from to","from to")` specifying which transitions should be plotted. By default, all the direct transition probabilities are plotted
`col`	Vector of colours. Default is black.
`lty`	Vector of line type. Default is 1:number of transitions
`conf.int`	Logical specifying whether to plot confidence intervals. Default is FALSE.
`level`	Level of the confidence interval. Default is 0.95.
`ci.fun`	Transformation applied to the confidence intervals. It could be different for all transition probabilities, though if `length(ci.fun) != number of transitions`, only `ci.fun[1]` will be used. Possible choices are "linear", "log", "log-log" and "cloglog". Default is "linear".
`ci.col`	Colours of the confidence intervals. Default value is the same as `col`.
`ci.lty`	Line types for the confidence intervals. Default is 3.
`...`	Further arguments for `lines`.

Value

No value returned.

Author(s)

Arthur Allignol, `<arthur.allignol@gmail.com>`

See Also

`etm`, `plot.etm`, `xyplot.etm`

`los.data` *Length of hospital stay*

Description

The `los.data` data frame has 756 rows, one row for each patient, and 7 columns.

Usage

```
data(los.data)
```

Format

A data frame with the following columns:

adm.id admision id of the patient

j.01 observed time for jump from 0 (initial state) to 1 (intermediate state)

j.02 observed time for jump from 0 to 2 (discharge)

j.03 observed time for jump from 0 to 3 (death)

j.12 observed time for jump from 1 to 2

j.13 observed time for jump from 1 to 3

cens censoring time (either in initial or intermediate state)

Examples

```
data(los.data)
my.data <- prepare.los.data(los.data)
```

`plot.clos.etm` *Plot method for 'clos.etm' objects*

Description

Plot method for objects of class `clos.etm`.

Usage

```
## S3 method for class 'clos.etm'
plot(x, xlab = "Time", ylab.e = "Expected LOS",
ylab.w = "Weights", xlim, ylim.e, ylim.w, col.e = c(1, 2), col.w = 1,
lty.e = c(1, 1), lty.w = 1, legend = TRUE, legend.pos, curvlab,
legend.bty = "n", ...)
```

Arguments

`x`	An object of class `clos.etm`
`xlab`	Label for the x-axis
`ylab.e`	Label for the y-axis in the plot of the expected LOS
`ylab.w`	Label for the y-axis in the plot of the weights
`xlim`	Limits of x-axis for the plots
`ylim.e`	Limits of the y-axis for the expected LOS plot
`ylim.w`	Limits of the y-axis for the weights plot
`col.e`	Vector of colours for the plot of expected LOS
`col.w`	Vector of colours for the plot of the weights
`lty.e`	Vector of line type for the plot of expected LOS

`lty.w`	Vector of line type for the plot of the weights
`legend`	Logical. Whether to draw a legend for the plot of expected LOS
`legend.pos`	A vector giving the legend's position. See `legend` for details
`curvlab`	Character or expression vector to appear in the legend. Default is `c("Intermediate event by time t", "No intermediate event by time t")`
`legend.bty`	Box type for the legend
`...`	Further arguments for plot

Details

Two graphs are drawn. The lower graph displays the expected LOS for patients who have experienced the intermediate event and for those who have not. The upper graph displays the weights used to compute the weighted average.

Value

No value returned

Author(s)

Arthur Allignol <`arthur.allignol@gmail.com`>, Matthias Wangler

See Also

`clos`

`plot.etm` *Plot method for an etm object*

Description

Plot method for an object of class 'etm'. It draws the estimated transition probabilities in a basic scatterplot.

Usage

```
## S3 method for class 'etm'
plot(x, tr.choice, xlab = "Time",
     ylab = "Transition Probability", col = 1, lty, xlim, ylim,
     conf.int = FALSE, level = 0.95, ci.fun = "linear",
     ci.col = col, ci.lty = 3,
     legend = TRUE, legend.pos, curvlab, legend.bty = "n", ...)
```

Arguments

`x`	An object of class 'etm'
`tr.choice`	character vector of the form 'c("from to","from to")' specifying which transitions should be plotted. Default, all the transition probabilities are plotted
`xlab`	x-axis label. Default is "Time"
`ylab`	y-axis label. Default is "Transition Probability"
`col`	Vector of colour. Default is black
`lty`	Vector of line type. Default is 1:number of transitions
`xlim`	Limits of x-axis for the plot
`ylim`	Limits of y-axis for the plot
`conf.int`	Logical. Whether to display pointwise confidence intervals. Default is FALSE.
`level`	Level of the conficence intervals. Default is 0.95.
`ci.fun`	Transformation applied to the confidence intervals. It could be different for all transition probabilities, though if `length(ci.fun) != number of transitions`, only `ci.fun[1]` will be used. Possible choices are "linear", "log", "log-log" and "cloglog". Default is "linear".
`ci.col`	Colour of the confidence intervals. Default is `col`.
`ci.lty`	Line type of the confidence intervals. Default is 3.
`legend`	A logical specifying if a legend should be added
`legend.pos`	A vector giving the legend's position. See `legend` for further details
`curvlab`	A character or expression vector to appear in the legend. Default is the name of the transitions
`legend.bty`	Box type for the legend
`...`	Further arguments for plot

Details

By default, if the argument `strata` was used for creating the `etm` object, the first transition probability for all strata will be plotted. If there is no strata, all transition probabilities are plotted by default.

In any case, a legend with be created by the labels are likely to be ugly. Please use the `curvlab` argument to control the test or use `legend = FALSE` and build your own legend.

Value

No value returned

Author(s)

Arthur Allignol, `<arthur.allignol@gmail.com>`

See Also

`plot.default`, `legend`, `etm`

Examples

```
data(sir.cont)

# Modification for patients entering and leaving a state
# at the same date
sir.cont <- sir.cont[order(sir.cont$id, sir.cont$time), ]
for (i in 2:nrow(sir.cont)) {
  if (sir.cont$id[i]==sir.cont$id[i-1]) {
    if (sir.cont$time[i]==sir.cont$time[i-1]) {
      sir.cont$time[i-1] <- sir.cont$time[i-1] - 0.5
    }
  }
}

tra <- matrix(ncol=3,nrow=3,FALSE)
tra[1, 2:3] <- TRUE
tra[2, c(1, 3)] <- TRUE

my.etm <- etm(sir.cont,c("0","1","2"),tra,"cens", s = 0)

plot(my.etm, tr.choice = c("0 0"))
```

`plot.etmCIF` *Plot cumulative incidence functions*

Description

Plot function for `etmCIF` objects. The function plots cumulative incidence curves, possibly with pointwise confidence intervals.

Usage

```
## S3 method for class 'etmCIF'
plot(x, which.cif, xlim, ylim,
    ylab = "Cumulative Incidence", xlab = "Time", col = 1, lty, lwd = 1,
    ci.type = c("none", "bars", "pointwise"), ci.fun = "cloglog",
    ci.col = col, ci.lty = 3, legend = TRUE, legend.pos, curvlab,
    legend.bty = "n", pos.ci = 27, ci.lwd = 3, ...)
```

Arguments

`x`	A `etmCIF` object
`which.cif`	A numeric vector indicating which CIFs should be plotted. When missing, only the CIF of interest is plotted (determined through the `failcode` argument in `etmCIF`.)
`xlim`	x-axis limits for the plot. By default, `c(0, max(time))`
`ylim`	y-axis limits. Default is `c(0, 1)`

`ylab`	Label for y-axis. Default is `"Cumulative Incidence"`
`xlab`	Label for x-axis. Default is "Time"
`col`	Vector describing colours used for the CIF curves. Default is black
`lty`	Vector of line type
`lwd`	Thickness of the lines
`ci.type`	One of `c("none", "bars", "pointwise")`. `none` plots no confidence interval, `bars` plots the confidence intervals in the form of a segment for one time point, and `pointwise` draws pointwise confidence intervals for the whole follow-up period.
`ci.fun`	Transformation used for the confidence intervals. Default is "clolog", and is a better choice for cumulative incidences. Other choices are "log" and "log-log"
`ci.col`	Colour for the pointwise confidence interval curves. Default is same as the CIF curves
`ci.lty`	Line type for the confidence intervals. Default is 3
`legend`	Logical. Whether to draw a legend. Default is `TRUE`
`legend.pos`	A vector giving the legend's position. See `legend` for further details
`curvlab`	A character or expression vector to appear in the legend. Default is CIF + event label
`legend.bty`	Box type for the legend. Default is none ("n")
`pos.ci`	If `ci.type = "bars"`, vector of integers indicating at which time point to put the confidence interval bars. Default is 27
`ci.lwd`	Thickness of the confidence interval segment (for `ci.type = "bars"`)
`...`	Further graphical arguments

Details

The function relies on `plot.etm` and `lines.etm` with more or less the same options. Exception is the drawing of the confidence intervals, for which several displays are possible.

Value

No value returned

Author(s)

Arthur Allignol <`arthur.allignol@gmail.com`>

See Also

`etmCIF`, `plot.etm`, `lines.etm`

Examples

```
data(abortion)

cif.ab <- etmCIF(survival::Surv(entry, exit, cause != 0) ~ group, abortion,
                 etype = cause, failcode = 3)

cif.ab

plot(cif.ab, ci.type = "bars", pos.ci = 24,
     col = c(1, 2), lty = 1, curvlab = c("Control", "Exposed"))

plot(cif.ab, which = c(1, 2))
```

`prepare.los.data` *Prepare the data for clos*

Description

Prepare data to be passed to clos() in package etm.

Usage

```
prepare.los.data(x)
```

Arguments

`x`	data.frame of the form data.frame(id, j.01, j.02, j.03, j.12, j.13, cens): **id:** id (patient id, admision id) **j.01:** observed time for jump from 0 to 1 **j.02:** observed time for jump from 0 to 2 **j.03:** observed time for jump from 0 to 3 **j.12:** observed time for jump from 1 to 2 **j.13:** observed time for jump from 1 to 3 **cens:** censoring time (either in initial or intermediate state)

Value

a data.frame of the form data.frame(id, from, to, time, oid):

`id:`	id (patient id, admision id)
`from:`	the state from where a transition occurs
`to:`	the state to which a transition occurs
`time:`	time of the transition
`oid:`	the observation id

Author(s)

Matthias Wangler

See Also

`clos`

Examples

```
data(los.data)
my.observ <- prepare.los.data(x=los.data)
```

`print.clos.etm` *Print function for 'clos.etm' objects*

Description

Print method for object of class `clos.etm`

Usage

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

Arguments

`x`	An object of class `clos.etm`
`...`	Further arguments

Value

No value returned

Author(s)

Arthur Allignol, `<arthur.allignol@gmail.com>`

See Also

`clos`

`print.etm` *Print method for object of class 'etm'*

Description

Print method for objects of class `etm`.

Usage

```
## S3 method for class 'etm'
print(x, covariance = FALSE, whole = TRUE, ...)
```

Arguments

`x`	An object of class `etm`.
`covariance`	Whether print the covariance matrix. Default is TRUE
`whole`	Whether to plot the entire covariance matrix. If set to FALSE, rows and columns containing only 0 will be removed for printing.
`...`	Further arguments for print or summary.

Details

The function prints a matrix giving the possible transitions, along with the estimates of $P(s,t)$ and $cov(P(s,t))$.

Value

No value returned

Author(s)

Arthur Allignol, `<arthur.allignol@gmail.com>`

See Also

`etm`

`print.etmCIF` *Print function for* `cifETM` *objects*

Description

Print method for `cifETM` objects

Usage

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

Arguments

`x`	An object of class `etmCIF`
`...`	Further arguments

Value

No value returned

Author(s)

Arthur Allignol `<arthur.allignol@gmail.com>`

See Also

`etmCIF`

`sir.cont` *Ventilation status in intensive care unit patients*

Description

Time-dependent ventilation status for intensive care unit (ICU) patients, a random sample from the SIR-3 study.

Usage

```
data(sir.cont)
```

Format

A data frame with 1141 rows and 6 columns:

id: Randomly generated patient id

from: State from which a transition occurs

to: State to which a transition occurs

time: Time when a transition occurs

age: Age at inclusion

sex: Sex. `F` for female and `M` for male

The possible states are:

0: No ventilation

1: Ventilation

2: End of stay

And `cens` stands for censored observations.

Details

This data frame consists in a random sample of the SIR-3 cohort data. It focuses on the effect of ventilation on the length of stay (combined endpoint discharge/death). Ventilation status is considered as a transcient state in an illness-death model.

The data frame is directly formated to be used with the `etm` function, i.e. it is transition-oriented with one row per transition.

References

Beyersmann, J., Gastmeier, P., Grundmann, H., Baerwolff, S., Geffers, C., Behnke, M., Rueden, H., and Schumacher, M. Use of multistate models to assess prolongation of intensive care unit stay due to nosocomial infection. *Infection Control and Hospital Epidemiology*, 27:493-499, 2006.

Examples

```
data(sir.cont)
```

`summary.etm` *Summary methods for an 'etm' object*

Description

Summary method for objects of class `etm`

Usage

```
## S3 method for class 'etm'
summary(object, tr.choice,
        ci.fun = "linear", level = 0.95, times, ...)
## S3 method for class 'summary.etm'
print(x, ...)
```

Arguments

`object`	An object of class `etm`.
`tr.choice`	Character vector of the form 'c("from to","from to")' specifying which transitions should be summarized. Default to all the transition probabilities
`ci.fun`	A character vector specifying the transformation to be applied to the pointwise confidence intervals. It could be different for each transition probability, though if `length(ci.fun) != number of transitions`, only `ci.fun[1]` will be used. The function displays the transition probabilities in the following order: first the direct transitions in alphabetical order, e.g., 0 to 1, 0 to 2, 1 to 2, ..., then the state occupation probabilities in alphabetical order, e.g., 0 to 0, 1 to 1, ... The possible transformations are "linear", "log", "log-log" and "cloglog". Default is "linear".
`level`	Level of the two-sided confidence intervals. Default is 0.95.
`x`	A `summary.cpf` object
`times`	Time points for which estimates should be returned. Default to all transition times.
`...`	Further arguments

Value

A list of data.frames giving the transition probability and stage occupation probability estimates. List items are named after the possible transition.

`P`	Transition probability estimates
`var`	Variance estimates
`lower`	Lower confidence limit
`upper`	Upper confidence limit
`time`	Transition times
`n.risk`	Number of individuals at risk of experiencing a transition just before time t
`n.event`	Number of events at time t

Author(s)

Arthur Allignol <`arthur.allignol@gmail.com`>

See Also

`etm`

`summary.etmCIF` *Summary function for cifETM*

Description

Summary function for objects of class `cifETM`

Usage

```
## S3 method for class 'etmCIF'
summary(object, ci.fun = "cloglog",
        level = 0.95, ...)
## S3 method for class 'summary.etmCIF'
print(x, ...)
```

Arguments

`object`	An object of class `etmCIF`
`ci.fun`	Transformation applied to the pointwise confidence intervals. On of `"linear"`, `"log"`, `"log-log"`, `"cloglog"`. Default is `"cloglog"`.
`level`	Level of the confidence intervals. Default is 0.95.
`x`	An object of class `cifETM`.
`...`	Further arguments

Value

A data.frame per covariate level and competing event

`P`	Transition probability estimates
`var`	Variance estimates
`lower`	Lower confidence limit
`upper`	Upper confidence limit
`time`	Transition times
`n.risk`	Number of individuals at risk of experiencing a transition just before time t
`n.event`	Number of events at time t

Author(s)

Arthur Allignol <`arthur.allignol@gmail.com`>

See Also

`etmCIF`

---

tra *Matrix of possible transitions*

Description

Miscellaneous functions that compute the matrix of possible transitions used as argument in the `etm` function.

Usage

```
tra_ill(state.names = c("0", "1", "2"))
tra_ill_comp(nComp = 2,
     state.names = as.character(seq(0, nComp + 1, 1)))
tra_comp(nComp = 2,
         state.names = as.character(seq(0, nComp)))
tra_surv(state.names = c("0", "1"))
```

Arguments

`state.names`	A vector of characters giving the states names
`nComp`	For the competing risks models, the number of competing events

Details

These functions compute the matrix of possible transitions that is used as argument in, e.g., the `etm` function. `tra_surv` is for the usual survival model, `tra_comp` for the competing risks model, `tra_ill` for the illness-death model and `tra_ill_comp` for the illness-death model with competing terminal events. By default, state names are from 0 to …

Value

A quadratic matrix with `TRUE` if a transition is possible, `FALSE` otherwise.

Author(s)

Arthur Allignol <`arthur.allignol@gmail.com`>

See Also

`etm`

Examples

```
tra_ill()

## competing risks model with 4 competing events non-default state names
tra_comp(4, state.names = c("healthy", "Cardiac problems", "Cancer",
                            "Rhenal failure", "Other"))
```

`trprob.etm` *Function to extract transition probabilities and (co)variance*

Description

The `trprob` method is used to extract transition probabilities, while `trcov` is used to obtain the (co)variance.

Usage

```
## S3 method for class 'etm'
trprob(x, tr.choice, timepoints, ...)
## S3 method for class 'etm'
trcov(x, tr.choice, timepoints, ...)
```

Arguments

`x`	An object of class `etm`.
`tr.choice`	A character vector of the form "from to" describing for which transition one wishes to obtain the transition probabilities or covariance estimates. For `trprob`, `tr.choice` must be of length 1, while it can be of length 2 for `trcov`.
`timepoints`	Time points at which one want the estimates. When missing, estimates are obtained for all event times.
`...`	Further arguments.

Value

A vector containing the transition probabilities or covariance estimates either at the time specified in `timepoints` or at all transition times.

Author(s)

Arthur Allignol, <`arthur.allignol@gmail.com`>

See Also

`etm`

Examples

```
data(sir.cont)

# Modification for patients entering and leaving a state
# at the same date
# Change on ventilation status is considered
# to happen before end of hospital stay
sir.cont <- sir.cont[order(sir.cont$id, sir.cont$time), ]
```

```
for (i in 2:nrow(sir.cont)) {
  if (sir.cont$id[i]==sir.cont$id[i-1]) {
    if (sir.cont$time[i]==sir.cont$time[i-1]) {
      sir.cont$time[i-1] <- sir.cont$time[i-1] - 0.5
    }
  }
}

### Computation of the transition probabilities
# Possible transitions.
tra <- matrix(ncol=3,nrow=3,FALSE)
tra[1, 2:3] <- TRUE
tra[2, c(1, 3)] <- TRUE

# etm
fit.etm <- etm(sir.cont, c("0", "1", "2"), tra, "cens", 0)

## extract P_01(0, t) and variance
p01 <- trprob(fit.etm, "0 1")
var.p01 <- trcov(fit.etm, "0 1")

## covariance between P_00 and P_01
cov.00.01 <- trcov(fit.etm, c("0 0", "0 1"))

## P_01 at some time points
trprob(fit.etm, "0 1", c(0, 15, 50, 100))
```

`xyplot.etm` *xyplot method for object of class 'etm'*

Description

xyplot function for objects of class `etm`. Estimates of the transition probabilities are plotted as a function of time for all the transitions specified by the user.

Usage

```
## S3 method for class 'etm'
xyplot(x, data = NULL, tr.choice, col = c(1, 1, 1),
       lty = c(1, 3, 3),  xlab = "Time",
       ylab = "Transition probability",
       conf.int = TRUE, ci.fun = "linear", level = 0.95, ...)
```

Arguments

`x`	An object of class `etm`.
`data`	*Useless.*

`tr.choice`	A character vector of the form c("from to", "from to", ...) specifying the transition probabilities to be plotted. By default, all the direct transition probabilities are displayed.
`col`	Vector of colours for the curves.
`lty`	Vector of line types.
`xlab`	x-axis label. Default is "Time".
`ylab`	y-axis label. Default is "Estimated transition probability".
`conf.int`	Logical. Whether to draw pointwise confidence intervals. Default is TRUE.
`ci.fun`	A character vector specifying the transformation to be applied to the pointwise confidence intervals. It could be different for each transition probability, though if `length(ci.fun) != length(tr.choice)`, only `ci.fun[1]` will be used. The possible transformations are "linear", "log", "log-log" and "cloglog". Default is "linear".
`level`	Level of the two-sided confidence intervals. Default is 0.95.
`...`	Further arguments for `xyplot`.

Value

An object of class `trellis`.

Author(s)

Arthur Allignol, `<arthur.allignol@gmail.com>`

See Also

`etm`, `xyplot`

Index