Package 'calibmsm'

November 30, 2023

Title Calibration Plots for the Transition Probabilities from Multistate Models

Version 1.0.0

Description Assess the calibration of an existing (i.e. previously developed) multistate model through calibration plots.
Calibration is assessed using one of three methods. 1) Calibration methods for binary logistic regression models applied at a fixed time point in conjunction with inverse probability of censoring weights. 2) Calibration methods for multinomial logistic regression models applied at a fixed time point in conjunction with inverse probability of censoring weights. 3) Pseudo-values estimated using the Aalen-Johansen estimator of observed risk. All methods are applied in conjunction with landmarking when required. These calibration plots evaluate the calibration (in a validation cohort of interest) of the transition probabilities estimated from an existing multistate model. While package development has focused on multistate models, calibration plots can be produced for any model which utilises information post baseline to update predictions (e.g. dynamic models); competing risks models; or standard single outcome survival models, where predictions can be made at any landmark time. The underpinning methodology is currently undergoing peer review; see Pate et al. (2023) <arXiv:2308.13394>
and Pate et al. (2023) <https://alexpate30.github.io/calibmsm/articles/Overview.html>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Imports boot, dplyr, ggplot2, ggpubr, Hmisc, magrittr, mstate, rms, stats, survival, tidyr, VGAM

Depends R (>= 2.10)

LazyData true

Suggests covr, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

URL https://alexpate30.github.io/calibmsm/

Config/testthat/edition 3

NeedsCompilation no

Author Alexander Pate [aut, cre, cph]
(<https://orcid.org/0000-0002-0849-3458>),
Glen P Martin [fnd, rev] (<https://orcid.org/0000-0002-3410-9472>)

Maintainer Alexander Pate <alexander.pate@manchester.ac.uk>

Repository CRAN

Date/Publication 2023-11-30 19:50:02 UTC

R topics documented:

calc_aj 2
calc_pv_aj 3
calc_weights 4
calib_blr 5
calib_mlr 9
calib_pv 12
ebmtcal 15
extract_ids_states 16
msebmtcal 17
msebmtcal.cmprsk 18
plot.calib_blr 19
plot.calib_mlr 20
plot.calib_pv 21
tp.cmprsk.j0 23
tps0 24
tps100 25

Index **26**

`calc_aj` *Estimate Aalen-Johansen estimator for a cohort of individuals*

Description

Estimates Aalen-Johansen estimator for the transition probabilities in cohort data.mstate. Estimates transition probabilities at time t if in state j at time 0 The Aalen-Johansen estimator for the entire cohort (including individual person_id.eval) is inputted manually (obs.aj), to speed up computational time if calculating pseudo-values for multiple individuals from the same cohort.

Usage

```
calc_aj(data.mstate, tmat, t, j)
```

Arguments

data.mstate	Validation data in msdata format
tmat	Transition probability matrix
t	Follow up time at which calibration is to be assessed
j	Landmark state at which predictions were made

---

calc_pv_aj *Estimate pseudo-values for the transition probabilities based on the Aalen-Johansen estimator*

Description

Estimates the pseudo-values for an individual (person_id.eval) from cohort data.mstate. Calculates psuedo-values for transition probabilities at time t if in state j at time 0 The Aalen-Johansen estimator for the entire cohort (including individual person_id.eval) is inputted manually (obs.aj), to speed up computaitonal time if calculating pseudo-values for multiple individuals from the same cohort.

Usage

```
calc_pv_aj(person_id.eval, data.mstate, obs.aj, tmat, n.cohort, t, j)
```

Arguments

person_id.eval	id of individual to calculate the pseudo-value for
data.mstate	Validation data in msdata format
obs.aj	Aalen-Johansen estimator of the transition probabilities in the entire cohort (not excluding person_id.eval)
tmat	Transition probability matrix
n.cohort	Size of cohort (number of unique entries in data.mstate)
t	Follow up time at which calibration is to be assessed
j	Landmark state at which predictions were made

`calc_weights` *Calculate inverse probability of censoring weights at time* `t`.

Description

Estimates the inverse probability of censoring weights by fitting a cox-propotinal hazards model in a landmark cohort of individuals. Primarily used internally, this function has been exported to allow users to reproduce results in the vignette when estimating confidence intervals using bootstrapping manually.

Usage

```
calc_weights(
  data.mstate,
  data.raw,
  covs = NULL,
  t,
  s,
  landmark.type = "state",
  j = NULL,
  max.weight = 10,
  stabilised = FALSE,
  max.follow = NULL
)
```

Arguments

`data.mstate`	Validation data in msdata format
`data.raw`	Validation data in data.frame (one row per individual)
`covs`	Character vector of variable names to adjust for when calculating inverse probability of censoring weights
`t`	Follow up time at which to calculate weights
`s`	Landmark time at which predictions were made
`landmark.type`	Whether weights are estimated in all individuals uncensored at time s ('all') or only in individuals uncensored and in state j at time s ('state')
`j`	Landmark state at which predictions were made (only required in landmark.type = 'state')
`max.weight`	Maximum bound for weights
`stabilised`	Indicates whether weights should be stabilised or not
`max.follow`	Maximum follow up for model calculating inverse probability of censoring weights. Reducing this to `t + 1` may aid in the proportional hazards assumption being met in this model.

Details

Estimates inverse probability of censoring weights (Hernan M, Robins J, 2020). Fits a cox proportional hazards model to individuals in a landmark cohort, predicting the probability of being censored at time `t`. This landmark cohort may either be all individuals uncensored at time `s`, or those uncensored and in state `j` at time `s`. All predictors in `w.covs` are assumed to have a linear effect on the hazard. Weights are estimated for all individuals in `data.raw`, even if they will not be used in the analysis as they do not meet the landmarking requirements. If an individual enters an absorbing state prior to `t`, we estimate the probability of being censored before the time of entry into the absorbing state, rather than at `t`. Details on all the above this are provided in vignette *overview*.

Value

A dataframe with three columns. `id` corresponds to the patient ids from `data.raw`. `ipcw` contains the inverse probability of censoring weights (specifically the inverse of the probability of being uncesored). `pcw = 1/ipcw`. If `stabilised = TRUE` was specified, a fourth variable `ipcw.stab` will be returned, which is the stabilised inverse probability of censoring weights.

References

Hernan M, Robins J (2020). "12.2 Estimating IP weights via modeling." In *Causal Inference: What If*, chapter 12.2. Chapman Hall/CRC, Boca Raton.

Examples

```
# Estimate inverse probability of censoring weights for individual in cohort ebmtcal.
# Specifically the probability of being uncensored at t = 1826 days.
# Weights are estimated using a model fitted in all individuals uncensored at time s = 0.
weights.manual <-
calc_weights(data.mstate = msebmtcal,
  data.raw = ebmtcal,
  covs = c("year", "agecl", "proph", "match"),
  t = 1826,
  s = 0,
  landmark.type = "state",
  j = 1)

 str(weights.manual)
```

`calib_blr` *Estimate calibration curves for a multistate model using binary logistic regression calibration techniques and inverse probability of censoring weights.'*

Description

Creates the underlying data for the calibration curves. `calib_blr` estimates the observed event probabilities for a given set of predicted transition probabilities in a cohort of interest. This is done using techniques for assessing calibration of binary logistic regression models, in combination with inverse probability of censoring weights and landmarking.

Usage

```
calib_blr(
  data.mstate,
  data.raw,
  j,
  s,
  t,
  tp.pred,
  curve.type = "rcs",
  rcs.nk = 3,
  loess.span = 0.75,
  loess.degree = 2,
  weights = NULL,
  w.function = NULL,
  w.covs = NULL,
  w.landmark.type = "state",
  w.max = 10,
  w.stabilised = FALSE,
  w.max.follow = NULL,
  CI = FALSE,
  CI.type = "bootstrap",
  CI.R.boot = NULL,
  data.pred.plot = NULL,
  transitions.out = NULL,
  ...
)
```

Arguments

`data.mstate`	Validation data in `msdata` format
`data.raw`	Validation data in `data.frame` (one row per individual)
`j`	Landmark state at which predictions were made
`s`	Landmark time at which predictions were made
`t`	Follow up time at which calibration is to be assessed
`tp.pred`	Matrix of predicted transition probabilities at time t, if in state j at time s. There must be a seperate column for the predicted transition probabilities into every state, even if these predicted transition probabilities are 0.
`curve.type`	Whether calibration curves are estimated using restricted cubic splines ('rcs') or loess smoothers ('loess')

rcs.nk	Number of knots when curves are estimated using restricted cubic splines
loess.span	Span when curves are estimated using loess smoothers
loess.degree	Degree when curves are estimated. using loess smoothers
weights	Vector of inverse probability of censoring weights
w.function	Custom function for estimating the inverse probability of censoring weights
w.covs	Character vector of variable names to adjust for when calculating inverse probability of censoring weights
w.landmark.type	Whether weights are estimated in all individuals uncensored at time s ('all') or only in individuals uncensored and in state j at time s ('state')
w.max	Maximum bound for inverse probability of censoring weights
w.stabilised	Indicates whether inverse probability of censoring weights should be stabilised or not
w.max.follow	Maximum follow up for model calculating inverse probability of censoring weights. Reducing this to t + 1 may aid in the proportional hazards assumption being met in this model.
CI	Size of confidence intervals as a %
CI.type	Method for estimating confidence interval (currently restricted to bootstrap)
CI.R.boot	Number of bootstrap replicates when estimating the confidence interval for the calibration curve
data.pred.plot	Data frame or matrix of predicted risks for each possible transition over which to plot the calibration curves. Must have one column for every possible transition.
transitions.out	Transitions for which to calculate calibration curves. Will do all possible transitions if left as NULL.
...	Extra arguments to be passed to w.function (custom function for estimating weights)

Details

Observed event probabilities at time t are estimated for predicted transition probabilities tp.pred out of state j at time s. calib_blr estimates calibration curves using techniques for assessing the calibration of binary logistic regression models (Van Calster et al., 2016). A choice between restricted cubic splines and loess smoothers for estimating the calibration curve can be made using curve.type. Landmarking is applied to only assess calibration in individuals who are uncensored and in state j at time s. Censoring is dealt with using inverse probability of censoring weights (Hernan M, Robins J, 2020).

Two datasets for the same cohort of inidividuals must be provided. Firstly data.mstate must be a dataset of class msdata, generated using the [mstate] package. This dataset is used to apply the landmarking. Secondly, data.raw must be a data.frame with one row per individual, containing the desired variables for estimating the weights, and variables for the time until censoring (dtcens), and an indicator for censoring dtcens.s, where (dtcens.s = 1) if an individual is censored at time dtcens, and dtcens.s = 0 otherwise. When an individual enters an absorbing state, this prevents censoring from happening (i.e. dtcens.s = 0). Unless the user specifies the weights using weights,

the weights are estimated using a cox-proportional hazard model, assuming a linear functional form of the variables defined in `w.covs`. We urge users to specify their own model for estimating the weights. The `weights` argument must be a vector with length equal to the number of rows of `data.raw`. Confidence intervals for the calibration curves can be estimated using bootstrapping. This procedure uses the internal method for estimating weights, we therefore encourage users to specify their own bootstrapping procedure, which incorporates their own model for estimating the weights. Details on how to do this are provided in the vignette *BLR-IPCW-manual-bootstrap*.

The calibration curves can be plotted using `plot.calib_blr`.

Value

`calib_blr` returns a list containing two elements: `plotdata` and `metadata`. The `plotdata` element contains the data for the calibration curves. This will itself be a list with each element containing calibration plot data for the transition probabilities into each of the possible states. Each list element contains patient ids (`id`) from `data.raw`, the predicted transition probabilities (`pred`) and the estimated observed event probabilities (`obs`). If a confidence interval is requested, upper (`obs.upper`) and lower (`obs.lower`) bounds for the observed event probabilities are also returned. If data.pred.plot is defined manually, column (`id`) is not returned. The `metadata` element contains metadata including: a vector of the possible transitions, a vector of which transitions calibration curves have been estimated for, the size of the confidence interval, the method for estimating the calibration curve and other user specified information.

References

Hernan M, Robins J (2020). "12.2 Estimating IP weights via modeling." In *Causal Inference: What If*, chapter 12.2. Chapman Hall/CRC, Boca Raton.

Van Calster B, Nieboer D, Vergouwe Y, De Cock B, Pencina MJ, Steyerberg EW (2016). "A calibration hierarchy for risk models was defined: From utopia to empirical data." *Journal of Clinical Epidemiology*, 74, 167–176. ISSN 18785921. doi:10.1016/j.jclinepi.2015. 12.005. URL http://dx.doi.org/10.1016/j.jclinepi.2015.12.005

Examples

```
# Estimate BLR-IPCW calibration curves for the predicted transition
# probabilities at time t = 1826, when predictions were made at time
# s = 0 in state j = 1. These predicted transition probabilities are stored in tps0.

# Extract the predicted transition probabilities out of state j = 1
tp.pred <- dplyr::select(dplyr::filter(tps0, j == 1), any_of(paste("pstate", 1:6, sep = "")))

# Now estimate the observed event probabilities for each possible transition.
# 95% confidence intervals are estimated using bootstrapping with 200
# bootstrap repicates. In practice we recommend using a higher number.
dat.calib.blr <-
calib_blr(data.mstate = msebmtcal,
 data.raw = ebmtcal,
 j=1,
 s=0,
 t = 1826,
 tp.pred = tp.pred,
```

```
 w.covs = c("year", "agecl", "proph", "match"))

# The data for each calibration curve are stored in the "plotdata" list
# element.
str(dat.calib.blr)
```

---

calib_mlr *Create data for calibration curves using a multinomial logistic regression framework with inverse probability of censoring weights*

Description

Creates the underlying data for the calibration plots. Observed event probabilities at time `t` are estimated for inputted predicted transition probabilities `tp.pred` out of state `j` at time `s`. `calib_mlr` estimates calibration scatter plots using a multinomial logistic framework in combination with landmarking and inverse probability of censoring weights.

Two datasets for the same cohort of inidividuals must be provided. A `msdata` format dataset generated using the `mstate` package. A `data.frame` with one row per individual, relevant variables for estimating the weights, and a time until censoring varaible (`dtcens`) and indicator (`dtcens.s`). Weights are estimated using a cox-proportional hazard model and assuming linear functional form of the variables defined in `w.covs`. We urge users to specify their own modwl for estimating the weights. Confidence intervals for the calibration scatter plots cannot be produced as it is currently unclear how to present such data.

Usage

```
calib_mlr(
  data.mstate,
  data.raw,
  j,
  s,
  t,
  tp.pred,
  smoother.type = "sm.ps",
  ps.int = 4,
  degree = 3,
  s.df = 4,
  niknots = 4,
  weights = NULL,
  w.function = NULL,
  w.covs = NULL,
  w.landmark.type = "state",
  w.max = 10,
  w.stabilised = FALSE,
  w.max.follow = NULL,
  ...
)
```

Arguments

data.mstate	Validation data in msdata format
data.raw	Validation data in data.frame (one row per individual)
j	Landmark state at which predictions were made
s	Landmark time at which predictions were made
t	Follow up time at which calibration is to be assessed
tp.pred	Vector of predicted transition probabilities at time t
smoother.type	Type of smoothing applied. Takes values s (see s), sm.ps (see sm.ps) or sm.os (see sm.os).
ps.int	the number of equally-spaced B spline intervals in the vector spline smoother (see sm.ps)
degree	the degree of B-spline basis in the vector spline smoother (see sm.ps)
s.df	degrees of freedom of vector spline (see s)
niknots	number of interior knots (see sm.os)
weights	Vector of inverse probability of censoring weights
w.function	Custom function for estimating the inverse probability of censoring weights
w.covs	Character vector of variable names to adjust for when calculating inverse probability of censoring weights
w.landmark.type	Whether weights are estimated in all individuals uncensored at time s ('all') or only in individuals uncensored and in state j at time s ('state')
w.max	Maximum bound for inverse probability of censoring weights
w.stabilised	Indicates whether inverse probability of censoring weights should be stabilised or not
w.max.follow	Maximum follow up for model calculating inverse probability of censoring weights. Reducing this to t + 1 may aid in the proportional hazards assumption being met in this model.
...	Extra arguments to be passed to w.function (custom function for estimating weights)

Details

Observed event probabilities at time t are estimated for predicted transition probabilities tp.pred out of state j at time s. calib_mlr estimates calibration scatter plots uses a technique for assessing the calibration of multinomial logistic regression models, namely the nominal calibration framework of van Hoorde et al. (2014, 2015). Landmarking (van Houwelingen HC, 2007) is applied to only assess calibration in individuals who are uncensored and in state j at time s. Censoring is dealt with using inverse probability of censoring weights (Hernan M, Robins J, 2020).

Two datasets for the same cohort of inidividuals must be provided. Firstly data.mstate must be a dataset of class msdata, generated using the [mstate] package. This dataset is used to apply the landmarking. Secondly, data.raw must be a data.frame with one row per individual, containing the desired variables for estimating the weights, and variables for the time until censoring (dtcens),

and an indicator for censoring dtcens.s, where (dtcens.s = 1) if an individual is censored at time dtcens, and dtcens.s = 0 otherwise. When an individual enters an absorbing state, this prevents censoring from happening (i.e. dtcens.s = 0). Unless the user specifies the weights using weights, the weights are estimated using a cox-proportional hazard model, assuming a linear functional form of the variables defined in w.covs. We urge users to specify their own model for estimating the weights. The weights argument must be a vector with length equal to the number of rows of data.raw.

Confidence intervals cannot be generated for the calibration scatter plots. While confidence intervals could be estimated for each data point, it is not clear how these would be plotted cohesively.

Calibration plots cannot be produced for specific transitions (i.e. transitions.out in calib_blr) because the nominal calibration framework (van Hoorde et al., 2014, 2015) assesses the calibration of all states simultaneously.

The calibration scatter plots can be plotted using plot.calib_mlr.

Value

calib_mlr returns a list containing two elements: plotdata and metadata. The plotdata element contains the data for the calibration scatter plots This will itself be a list with each element containing the data for the transition probabilities into each of the possible states. Each list element contains patient ids (id), the predicted transition probabilities (pred) and the estimated observed event probabilities (obs). The metadata element contains metadata including a vector of the possible transitions and other user specified information.

References

Hernan M, Robins J (2020). "12.2 Estimating IP weights via modeling." In *Causal Inference: What If*, chapter 12.2. Chapman Hall/CRC, Boca Raton.

Van Hoorde K, Vergouwe Y, Timmerman D, Van Huffel S, Steyerberg W, Van Calster B (2014). "Assessing calibration of multinomial risk prediction models." *Statistics in Medicine*, 33(15), 2585–2596. doi:10.1002/sim.6114.

Van Hoorde K, Van Huffel S, Timmerman D, Bourne T, Van Calster B (2015). "A spline-based tool to assess and visualize the calibration of multiclass risk predictions." *Journal of Biomedical Informatics*, 54, 283–293. ISSN 15320464. doi:10.1016/j.jbi.2014.12.016. URL http://dx.doi.org/10.1016/j.jbi.2014.12.016.

van Houwelingen HC (2007). "Dynamic Prediction by Landmarking in Event History Analysis." *Scandinavian Journal of Statistics*, 34(1), 70–85.

Yee TW (2015). *Vector Generalized Linear and Additive Models*. 1 edition. Springer New, NY. ISBN 978-1-4939-4198-8. doi:10.1007/978-1-4939-2818-7. URL https://link.springer.com/book/10.1007/978-1-4939-2818-7.

Examples

```
# Using competing risks data out of initial state (see vignette: ... -in-competing-risk-setting).
# Estimate and plot MLR-IPCW calibration scatter plots for the predicted transition
# probabilities at time t = 1826, when predictions were made at time
# s = 0 in state j = 1. These predicted transition probabilities are stored in tp.cmprsk.j0.

# To minimise example time we reduce the datasets to 150 individuals.
```

```
# Extract the predicted transition probabilities out of state j = 1 for first 150 individuals
tp.pred <- tp.cmprsk.j0 |>
 dplyr::filter(id %in% 1:150) |>
 dplyr::select(any_of(paste("pstate", 1:6, sep = "")))
# Reduce ebmtcal to first 150 individuals
ebmtcal <- ebmtcal |> dplyr::filter(id %in% 1:150)
# Reduce msebmtcal.cmprsk to first 150 individuals
msebmtcal.cmprsk <- msebmtcal.cmprsk |> dplyr::filter(id %in% 1:150)

# Now estimate the observed event probabilities for each possible transition.
dat.calib.mlr <-
calib_mlr(data.mstate = msebmtcal.cmprsk,
 data.raw = ebmtcal,
 j=1,
 s=0,
 t = 1826,
 tp.pred = tp.pred,
 w.covs = c("year", "agecl", "proph", "match"),
 ps.int = 2,
 degree = 2)

# The data for each calibration scatter plots are stored in the "plotdata"
# list element.
str(dat.calib.mlr)
```

`calib_pv` *Estimate calibration curves for a multistate model using pseudo-values.*

Description

Creates the underlying data for the calibration curves. `calib_pv` estimates the observed event probabilities for a given set of predicted transition probabilities in a cohort of interest. This is done using techniques for assessing calibration of binary logistic regression models, in combination with inverse probability of censoring weights and landmarking.

Usage

```
calib_pv(
  data.mstate,
  data.raw,
  j,
  s,
  t,
  tp.pred,
  curve.type = "rcs",
  rcs.nk = 3,
  loess.span = 0.75,
```

```
  loess.degree = 2,
  group.vars = NULL,
  n.pctls = NULL,
  CI = FALSE,
  CI.type = "parametric",
  CI.R.boot = NULL,
  data.pred.plot = NULL,
  transitions.out = NULL
)
```

Arguments

data.mstate	Validation data in msdata format
data.raw	Validation data in data.frame (one row per individual)
j	Landmark state at which predictions were made
s	Landmark time at which predictions were made
t	Follow up time at which calibration is to be assessed
tp.pred	Matrix of predicted transition probabilities at time t, if in state j at time s. There must be a seperate column for the predicted transition probabilities into every state, even if these predicted transition probabilities are 0.
curve.type	Whether calibration curves are estimated using restricted cubic splines ('rcs') or loess smoothers ('loess')
rcs.nk	Number of knots when curves are estimated using restricted cubic splines
loess.span	Span when curves are estimated using loess smoothers
loess.degree	Degree when curves are estimated. using loess smoothers
group.vars	Baseline variables to define groups within which to estimate pseudo-values
n.pctls	Number of percentiles to group individuals by with respect to predicted transition probabilities when estimating pseudo-values
CI	Size of confidence intervals as a %
CI.type	Method for estimating confidence interval (bootstrap or parametric)
CI.R.boot	Number of bootstrap replicates when estimating the confidence interval for the calibration curve using bootstrapping
data.pred.plot	Data frame or matrix of predicted risks for each possible transition over which to plot the calibration curves. Must have one column for every possible transition.
transitions.out	Transitions for which to calculate calibration curves. Will do all possible transitions if left as NULL.

Details

Observed event probabilities at time t are estimated for predicted transition probabilities tp.pred out of state j at time s. calib_pv estimates the observed event probabilities using pseudo-values (Andersen PK, Pohar Perme M, 2010) calculated using the Aalen-Johansen estimator (Aalen OO, Johansen S, 1978) Calibration curves are generated by regressing the pseudo-values on the predicted

transition probabilities. Currently calibration curves can be produced using loess smoothers or restricted cubic splines. This will be updated to include restricted cubic splines. Landmarking (van Houwelingen HC, 2007) is applied to only assess calibration in individuals who are uncensored and in state `j` at time `s`.

Two datasets for the same cohort of inidividuals must be provided. Firstly `data.mstate` must be a dataset of class `msdata`, generated using the `[mstate]` package. This dataset is used to apply the landmarking. Secondly, `data.raw` must be a `data.frame` with one row per individual, containing the desired variables for calculating pseudo-values within (no baseline variables required if `group.vars = NULL`). Confidence intervals for the calibration curves can be estimated using bootstrapping.

The calibration curves can be plotted using `plot.calib_pv`.

Value

`calib_pv` returns a list containing two elements: `plotdata` and `metadata`. The `plotdata` element contains the data for the calibration curves. This will itself be a list with each element containing calibration plot data for the transition probabilities into each of the possible states. Each list element contains patient ids (`id`) from `data.raw`, the predicted transition probabilities (`pred`) and the estimated observed event probabilities (`obs`). If a confidence interval is requested, upper (`obs.upper`) and lower (`obs.lower`) bounds for the observed event probabilities are also returned. If data.pred.plot is defined manually, column (`id`) is not returned. The `metadata` element contains metadata including: a vector of the possible transitions, a vector of which transitions calibration curves have been estimated for, the size of the confidence interval, the method for estimating the calibration curve and other user specified information.

References

Aalen OO, Johansen S. An Empirical Transition Matrix for Non-Homogeneous Markov Chains Based on Censored Observations. *Scand J Stat*. 1978;5(3):141-150.

Andersen PK, Pohar Perme M. Pseudo-observations in survival analysis. *Stat Methods Med Res*. 2010;19(1):71-99. doi:10.1177/0962280209105020

van Houwelingen HC (2007). "Dynamic Prediction by Landmarking in Event History Analysis." *Scandinavian Journal of Statistics*, 34(1), 70–85.

Examples

```
# Using competing risks data out of initial state.
# See vignette: comparison-with-graphical-calibration-curves-in-competing-risk-setting.
# Estimate pseudo-value calibration curves for the predicted transition
# probabilities at time t = 1826, when predictions were made at time
# s = 0 in state j = 1. These predicted transition probabilities are stored in tp.cmprsk.j0.

# To minimise example time we reduce the datasets to 50 individuals.
# Extract the predicted transition probabilities out of state j = 1 for first 50 individuals
tp.pred <- tp.cmprsk.j0 |>
 dplyr::filter(id %in% 1:50) |>
 dplyr::select(any_of(paste("pstate", 1:6, sep = "")))
# Reduce ebmtcal to first 50 individuals
ebmtcal <- ebmtcal |> dplyr::filter(id %in% 1:50)
```

```
# Reduce msebmtcal.cmprsk to first 50 individuals
msebmtcal.cmprsk <- msebmtcal.cmprsk |> dplyr::filter(id %in% 1:50)

# Now estimate the observed event probabilities for each possible transition.
dat.calib.pv <- calib_pv(data.mstate = msebmtcal.cmprsk,
  data.raw = ebmtcal,
  j = 1,
  s = 0,
  t = 1826,
  tp.pred = tp.pred,
  curve.type = "loess",
  loess.span = 1,
  loess.degree = 1)

# The data for each calibration curve are stored in the "plotdata" list
# element.
str(dat.calib.pv)
```

`ebmtcal` *European Group for Blood and Marrow Transplantation data*

Description

A data frame of 2,279 individuals with blood cancer who have undergone a transplant. This data is identical to the `ebmt4` data, except two extra variables have been derived, time until censoring and a censoring indicator, which are required to assess calibration using some of the methods in `calibmsm`. Code for the derivation of this dataset is provided in the source code for the package.

Usage

```
ebmtcal
```

Format

'ebmtcal':

A data frame with 2,279 rows and 17 columns:

id Patient indentifier

rec, rec.s Time until and event indicator for recovery variable

ae, ae.s Time until and event indicator for adverse event variable

recae, recae.s Time until and event indicator for recovery + adverse event variable

rel, rel.s Time until and event indicator for relapse variable

srv, srv.s Time until and event indicator for death variable

year Year of transplant

agecl Age at transplant

proph Prophylaxis

match Donor-recipient match

dtcens Time of censoring

dtcens.s Event indicator, 1:censoring occured, 0: absorbing state entered before censoring occured

Source

This dataset was derived from data made available within the `mstate` package, see `ebmt4`. The data was originally provided by the European Group for Blood and Marrow Transplantation (https://www.ebmt.org/). We reiterate the source statement given by the developers of `mstate`: "We acknowledge the European Society for Blood and Marrow Transplantation (EBMT) for making available these data. Disclaimer: these data were simplified for the purpose of illustration of the analysis of competing risks and multi-state models and do not reflect any real life situation. No clinical conclusions should be drawn from these data."

References

EBMT (2023). "Data from the European Society for Blood and Marrow Transplantation." URL https://search.r-project.org/CRAN/refmans/mstate/html/EBMT-data.html.

de Wreede LC, Fiocco M, Putter H (2011). "mstate: An R Package for the Analysis of Competing Risks and Multi-State Models." *Journal of Statistical Software*, 38(7).

`extract_ids_states` *Identify patids for individuals in state j at time t*

Description

Extract patids for individuals in state j at time t from a dataset in 'msdata' format. Used internally in calib_blr, calib_mlr and calib_pv.

Usage

```
extract_ids_states(data.mstate, tmat, j, t)
```

Arguments

`data.mstate`	Validation data in `msdata` format
`tmat`	Transition probability matrix
`j`	State j
`t`	Follow up time

`msebmtcal` *European Group for Blood and Marrow Transplantation data*

Description

The `ebmt4` data converted into `msdata` format (see `msprep`), using the processes implemented in the `mstate` package. Code for the derivation of this dataset is provided in the source code for the package.

Usage

```
msebmtcal
```

Format

'msebmtcal':

A data frame in `msdata` format (see `msprep`) with 15,512 rows and 8 columns:

id Patient indentifier

from transition from state

to transition to state

trans transition number

Tstart time entered state 'from'

Tstop time leaving state 'from'

time time in state 'from'

status event indicator, 1:transitioned to state 'to'

Source

This dataset was derived from data made available within the `mstate` package, see `ebmt4`. The data was originally provided by the European Group for Blood and Marrow Transplantation (https://www.ebmt.org/). We reiterate the source statement given by the developers of `mstate`: "We acknowledge the European Society for Blood and Marrow Transplantation (EBMT) for making available these data. Disclaimer: these data were simplified for the purpose of illustration of the analysis of competing risks and multi-state models and do not reflect any real life situation. No clinical conclusions should be drawn from these data."

References

EBMT (2023). "Data from the European Society for Blood and Marrow Transplantation." URL https://search.r-project.org/CRAN/refmans/mstate/html/EBMT-data.html.

de Wreede LC, Fiocco M, Putter H (2011). "mstate: An R Package for the Analysis of Competing Risks and Multi-State Models." *Journal of Statistical Software*, 38(7).

`msebmtcal.cmprsk` *European Group for Blood and Marrow Transplantation data in competing risks format, for transitions out of the initial state only*

Description

Used in vignette: Comparison-with-graphical-calibration-curves-in-competing-risks-setting.

Usage

```
msebmtcal.cmprsk
```

Format

'msebmtcal.cmprsk':

A data frame with 9,116 rows and 8 columns:

id Patient indentifier

from transition from state

to transition to state

trans transition number

Tstart time entered state 'from'

Tstop time leaving state 'from'

time time in state 'from'

status event indicator, 1:transitioned to state 'to'

Details

The `ebmt4` data converted into `msdata` format (see `msprep`), where all subsequent states are considered absorbing states. i.e. only transitions out of the initial state are considered, meaning this data constitutes a competing risks model out of the initial state. Code for the derivation of this dataset is provided in the source code for the package.

Source

This dataset was derived from data made available within the `mstate` package, see `ebmt4`. The data was originally provided by the European Group for Blood and Marrow Transplantation (https://www.ebmt.org/). We reiterate the source statement given by the developers of `mstate`: "We acknowledge the European Society for Blood and Marrow Transplantation (EBMT) for making available these data. Disclaimer: these data were simplified for the purpose of illustration of the analysis of competing risks and multi-state models and do not reflect any real life situation. No clinical conclusions should be drawn from these data."

References

EBMT (2023). "Data from the European Society for Blood and Marrow Transplantation." URL https://search.r-project.org/CRAN/refmans/mstate/html/EBMT-data.html.

de Wreede LC, Fiocco M, Putter H (2011). "mstate: An R Package for the Analysis of Competing Risks and Multi-State Models." *Journal of Statistical Software*, 38(7).

`plot.calib_blr` *Plots calibration curves estimated using* `calib_blr`.

Description

Plots calibration curves for the transition probabilities of a multistate model estimated using `calib_blr`.

Usage

```
## S3 method for class 'calib_blr'
plot(x, ..., combine = TRUE, ncol = NULL, nrow = NULL, transparency.rug = 0.1)
```

Arguments

`x`	Object of class 'calib_blr' generated from `calib_blr`.
`...`	Other
`combine`	Whether to combine into one plot using ggarrange, or return as a list of individual plots
`ncol`	Number of columns for combined calibration plot
`nrow`	Number of rows for combined calibration plot
`transparency.rug`	Degree of transparency for the density rug plot along each axis

Value

If `combine = TRUE`, returns an object of classes `gg`, `ggplot`, and `ggarrange`, as all ggplots have been combined into one object. If `combine = FALSE`, returns an object of class `list`, each element containing an object of class `gg` and `ggplot`.

Examples

```
# Estimate and plot BLR-IPCW calibration curves for the predicted transition
# probabilities at time t = 1826, when predictions were made at time
# s = 0 in state j = 1. These predicted transition probabilities are stored in tps0.

# Extract the predicted transition probabilities out of state j = 1
tp.pred <- dplyr::select(dplyr::filter(tps0, j == 1), any_of(paste("pstate", 1:6, sep = "")))

# Now estimate the observed event probabilities for each possible transition.
dat.calib.blr <-
```

```
calib_blr(data.mstate = msebmtcal,
 data.raw = ebmtcal,
 j=1,
 s=0,
 t = 1826,
 tp.pred = tp.pred,
 w.covs = c("year", "agecl", "proph", "match"))

 # These are then plotted
 plot(dat.calib.blr, combine = TRUE, nrow = 2, ncol = 3)
```

`plot.calib_mlr` *Plots calibration scatter plots estimated using* `calib_mlr`.

Description

Plots calibration scatter plots for the transition probabilities of a multistate model estimated using `calib_mlr`.

Usage

```
## S3 method for class 'calib_mlr'
plot(
  x,
  ...,
  combine = TRUE,
  ncol = NULL,
  nrow = NULL,
  transparency.plot = 0.25,
  transparency.rug = 0.1
)
```

Arguments

`x`	Object of class 'calib_mlr' generated from `calib_mlr`
`...`	Other
`combine`	Whether to combine into one plot using ggarrange, or return as a list of individual plots
`ncol`	Number of columns for combined calibration plot
`nrow`	Number of rows for combined calibration plot
`transparency.plot`	Degree of transparency for the calibration scatter plot
`transparency.rug`	Degree of transparency for the density rug plot along each axis

Value

If combine = TRUE, returns an object of classes gg, ggplot, and ggarrange, as all ggplots have been combined into one object. If combine = FALSE, returns an object of class list, each element containing an object of class gg and ggplot.

Examples

```
# Using competing risks data out of initial state (see vignette: ... -in-competing-risk-setting).
# Estimate and plot MLR-IPCW calibration scatter plots for the predicted transition
# probabilities at time t = 1826, when predictions were made at time
# s = 0 in state j = 1. These predicted transition probabilities are stored in tp.cmprsk.j0.

# To minimise example time we reduce the datasets to 150 individuals.
# Extract the predicted transition probabilities out of state j = 1 for first 150 individuals
tp.pred <- tp.cmprsk.j0 |>
 dplyr::filter(id %in% 1:150) |>
 dplyr::select(any_of(paste("pstate", 1:6, sep = "")))
# Reduce ebmtcal to first 150 individuals
ebmtcal <- ebmtcal |> dplyr::filter(id %in% 1:150)
# Reduce msebmtcal.cmprsk to first 150 individuals
msebmtcal.cmprsk <- msebmtcal.cmprsk |> dplyr::filter(id %in% 1:150)

# Now estimate the observed event probabilities for each possible transition.
dat.calib.mlr <-
calib_mlr(data.mstate = msebmtcal.cmprsk,
 data.raw = ebmtcal,
 j=1,
 s=0,
 t = 1826,
 tp.pred = tp.pred,
 w.covs = c("year", "agecl", "proph", "match"),
 ps.int = 2,
 degree = 2)

 # These are then plotted
 plot(dat.calib.mlr, combine = TRUE, nrow = 2, ncol = 3)
```

---

plot.calib_pv *Plots calibration curves estimated using* calib_pv.

Description

Plots calibration curves for the transition probabilities of a multistate model estimated using calib_pv.

Usage

```
## S3 method for class 'calib_pv'
plot(x, ..., combine = TRUE, ncol = NULL, nrow = NULL, transparency.rug = 0.1)
```

Arguments

x	Object of class 'calib_pseudo' generated from calib_pv.
...	Other
combine	Whether to combine into one plot using ggarrange, or return as a list of individual plots
ncol	Number of columns for combined calibration plot
nrow	Number of rows for combined calibration plot
transparency.rug	Degree of transparency for the density rug plot along each axis

Value

If combine = TRUE, returns an object of classes gg, ggplot, and ggarrange, as all ggplots have been combined into one object. If combine = FALSE, returns an object of class list, each element containing an object of class gg and ggplot.

Examples

```
# Using competing risks data out of initial state (see vignette: ... -in-competing-risk-setting).
# Estimate and plot pseudo-value calibration curves for the predicted transition
# probabilities at time t = 1826, when predictions were made at time
# s = 0 in state j = 1. These predicted transition probabilities are stored in tp.cmprsk.j0.

# To minimise example time we reduce the datasets to 50 individuals.
# Extract the predicted transition probabilities out of state j = 1 for first 50 individuals
tp.pred <- tp.cmprsk.j0 |>
 dplyr::filter(id %in% 1:50) |>
 dplyr::select(any_of(paste("pstate", 1:6, sep = "")))
# Reduce ebmtcal to first 50 individuals
ebmtcal <- ebmtcal |> dplyr::filter(id %in% 1:50)
# Reduce msebmtcal.cmprsk to first 50 individuals
msebmtcal.cmprsk <- msebmtcal.cmprsk |> dplyr::filter(id %in% 1:50)

# Now estimate the observed event probabilities for each possible transition.
dat.calib.pv <- calib_pv(data.mstate = msebmtcal.cmprsk,
  data.raw = ebmtcal,
  j = 1,
  s = 0,
  t = 1826,
  tp.pred = tp.pred,
  curve.type = "loess",
  loess.span = 1,
  loess.degree = 1)

 # These are then plotted
 plot(dat.calib.pv, combine = TRUE, nrow = 2, ncol = 3)
```

`tp.cmprsk.j0` *Predicted risks for a competing risks model out of state j = 0*

Description

Used in vignette: Comparison-with-graphical-calibration-curves-in-competing-risks-setting. Data frame containing the predicted transition probabilities out of state j = 1 made at time s = 0, for a competing risks model out of the initial state (see `msebmtcal.cmprsk`). The predicted transition probabilities were estimated by fitting a competing risks model to the `msebmtcal.cmprsk` data using a leave-one-out approach. Code for deriving this dataset is provided in the source code for `calibmsm`. Code for the derivation of this dataset is provided in the source code for the package.

Usage

```
tp.cmprsk.j0
```

Format

'tp.cmprsk.j0':

A data frame with 2,279 rows and 13 columns:

id Patient indentifier

pstate1, pstate2, pstate3, pstate4, pstate5, pstate6 Predicted transition probabilities of transitions into states 1 to 6

se1, se2, se3, se4, se5, se6 Standard error of the predicted transition probabilities of transitions into states 1 to 6

Source

This dataset was derived from data made available within the `mstate` package, see `ebmt4`. The data was originally provided by the European Group for Blood and Marrow Transplantation (https://www.ebmt.org/). We reiterate the source statement given by the developers of `mstate`: "We acknowledge the European Society for Blood and Marrow Transplantation (EBMT) for making available these data. Disclaimer: these data were simplified for the purpose of illustration of the analysis of competing risks and multi-state models and do not reflect any real life situation. No clinical conclusions should be drawn from these data."

References

EBMT (2023). "Data from the European Society for Blood and Marrow Transplantation." URL https://search.r-project.org/CRAN/refmans/mstate/html/EBMT-data.html.

de Wreede LC, Fiocco M, Putter H (2011). "mstate: An R Package for the Analysis of Competing Risks and Multi-State Models." *Journal of Statistical Software*, 38(7).

`tps0` *Predicted transition probabilities out of transplant state made at time s = 0*

Description

Data frame containing the predicted transition probabilities out of state j = 1 made at time s = 0. The predicted transition probabilities were estimated by fitting a multistate model to the `ebmt4` data using a leave-one-out approach. Code for deriving this dataset is provided in the source code for `calibmsm`. Code for the derivation of this dataset is provided in the source code for the package.

Usage

```
tps0
```

Format

'tps0':

A data frame with 13,674 (CHANGE) rows and 14 columns:

id Patient indentifier

pstate1, pstate2, pstate3, pstate4, pstate5, pstate6 Predicted transition probabilities of transitions into states 1 to 6

se1, se2, se3, se4, se5, se6 Standard error of the predicted transition probabilities of transitions into states 1 to 6

j State from which the predicted transition probabilities are estimated from

Source

This dataset was derived from data made available within the `mstate` package, see `ebmt4`. The data was originally provided by the European Group for Blood and Marrow Transplantation (https://www.ebmt.org/). We reiterate the source statement given by the developers of `mstate`: "We acknowledge the European Society for Blood and Marrow Transplantation (EBMT) for making available these data. Disclaimer: these data were simplified for the purpose of illustration of the analysis of competing risks and multi-state models and do not reflect any real life situation. No clinical conclusions should be drawn from these data."

References

EBMT (2023). "Data from the European Society for Blood and Marrow Transplantation." URL https://search.r-project.org/CRAN/refmans/mstate/html/EBMT-data.html.

de Wreede LC, Fiocco M, Putter H (2011). "mstate: An R Package for the Analysis of Competing Risks and Multi-State Models." *Journal of Statistical Software*, 38(7).

`tps100` *Predicted transition probabilities out of every state made at time s = 100*

Description

Data frame containing the predicted transition probabilities out of states 1 (transplant), 2 (adverse event), 3 (recovery) and 4 (adverse event + recovery), made at time s = 100. The predicted transition probabilities were estimated by fitting a multistate model to the `ebmt4` data using a leave-one-out approach. Code for deriving this dataset is provided in the source code for `calibmsm`. Code for derivation of this dataset is provided in the source code for the package.

Usage

```
tps100
```

Format

'tps100':

A data frame with 13,674 (CHANGE) rows and 14 columns:

id Patient indentifier

pstate1, pstate2, pstate3, pstate4, pstate5, pstate6 Predicted transition probabilities of transitions into states 1 to 6

se1, se2, se3, se4, se5, se6 Standard error of the predicted transition probabilities of transitions into states 1 to 6

j State from which the predicted transition probabilities are estimated from

Source

This dataset was derived from data made available within the `mstate` package, see `ebmt4`. The data was originally provided by the European Group for Blood and Marrow Transplantation (https://www.ebmt.org/). We reiterate the source statement given by the developers of `mstate`: "We acknowledge the European Society for Blood and Marrow Transplantation (EBMT) for making available these data. Disclaimer: these data were simplified for the purpose of illustration of the analysis of competing risks and multi-state models and do not reflect any real life situation. No clinical conclusions should be drawn from these data."

References

EBMT (2023). "Data from the European Society for Blood and Marrow Transplantation." URL https://search.r-project.org/CRAN/refmans/mstate/html/EBMT-data.html.

de Wreede LC, Fiocco M, Putter H (2011). "mstate: An R Package for the Analysis of Competing Risks and Multi-State Models." *Journal of Statistical Software*, 38(7).

Index

∗ **datasets**
ebmtcal, 15
msebmtcal, 17
msebmtcal.cmprsk, 18
tp.cmprsk.j0, 23
tps0, 24
tps100, 25

calc_aj, 2
calc_pv_aj, 3
calc_weights, 4
calib_blr, 5, *8*, *11*, *19*
calib_mlr, 9, *11*, *20*
calib_pv, 12, *14*, *21, 22*

ebmt4, *15–18*, *23–25*
ebmtcal, 15
extract_ids_states, 16

msebmtcal, 17
msebmtcal.cmprsk, 18, *23*
msprep, *17, 18*

plot.calib_blr, *8*, 19
plot.calib_mlr, *11*, 20
plot.calib_pv, *14*, 21

s, *10*
sm.os, *10*
sm.ps, *10*

tp.cmprsk.j0, 23
tps0, 24
tps100, 25