

# Package: calibmsm (via r-universe)

November 8, 2024

**Title** Calibration Plots for the Transition Probabilities from  
Multistate Models

**Version** 1.1.1

**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. Please see Pate et al. (2024) <[doi:10.1002/sim.10094](https://doi.org/10.1002/sim.10094)> and Pate et al. (2024) <<https://alexpate30.github.io/calibmsm/articles/Overview.html>>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.1

**Imports** boot, dplyr, ggplot2, ggpubr, ggExtra, gridExtra, Hmisc,  
mstate, rms, stats, survival, tidyr, VGAM

**Depends** R (>= 2.10)

**LazyData** true

**Suggests** covr, knitr, rmarkdown, R.rsp, testthat (>= 3.0.0),  
survminer, flexsurv

**VignetteBuilder** knitr, R.rsp  
**URL** <https://alexpate30.github.io/calibmsm/>  
**Config/testthat/edition** 3  
**Config/pak/sysreqs** cmake make libicu-dev zlib1g-dev  
**Repository** <https://alexpate30.r-universe.dev>  
**RemoteUrl** <https://github.com/alexpate30/calibmsm>  
**RemoteRef** HEAD  
**RemoteSha** 1a9266c28b6689498e6aae469b4a0a21d172530b

Contents

calc_weights . . . . .	2
calib_msm . . . . .	4
ebmtcal . . . . .	9
ebmtcal_cmprsk . . . . .	10
metadata . . . . .	12
msebmcal . . . . .	12
msebmcal_cmprsk . . . . .	13
plot.calib_mlr . . . . .	14
plot.calib_msm . . . . .	16
tps0 . . . . .	18
tps100 . . . . .	19
tp_cmprsk_j0 . . . . .	20
<b>Index</b>	<b>22</b>

---

calc_weights	<i>Calculate inverse probability of censoring weights at time t.</i>
--------------	--

---

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_ms,  
  data_raw,  
  covs = NULL,  
  t,  
  s,  
  landmark_type = "state",
```

```

    j = NULL,
    max_weight = 10,
    stabilised = FALSE,
    max_follow = NULL
  )

```

## Arguments

data_ms	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 data frame with two 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 uncensored). If stabilised = TRUE was specified, a third 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_ms = msebmtcal,
  data_raw = ebmtcal,
  covs = c("year", "agec1", "proph", "match"),
  t = 1826,
  s = 0,
  landmark_type = "state",
  j = 1)

str(weights_manual)
```

---

calib\_msm

---

*Assess the calibration of a multistate model*


---

## Description

Calculates the underlying data for calibration plots of the predicted transition probabilities from a multistate model using three methods.

1. BLR-IPCW: Binary logistic regression with inverse probability of censoring weights.
2. MLR-IPCW: Multinomial logistic regression with inverse probability of censoring weights, based on the nominal calibration framework of van Hoorde et al. (2014, 2015)
3. Pseudo-values: Pseudo-values estimated using the Aalen-Johansen estimator (Aalen OO, Johansen S, 1978).

## Usage

```
calib_msm(
  data_ms,
  data_raw,
  j,
  s,
  t,
  tp_pred,
  tp_pred_plot = NULL,
  calib_type = "blr",
  curve_type = "rcs",
  rcs_nk = 3,
  loess_span = 0.75,
  loess_degree = 2,
  loess_surface = c("interpolate", "direct"),
  loess_statistics = c("approximate", "exact", "none"),
```

```

    loess_trace_hat = c("exact", "approximate"),
    loess_cell = 0.2,
    loess_iterations = 4,
    loess_iterTrace = FALSE,
    mlr_smoother_type = c("sm.ps", "sm.os", "s"),
    mlr_ps_int = 4,
    mlr_degree = 3,
    mlr_s_df = 4,
    mlr_niknots = 4,
    weights = NULL,
    w_function = NULL,
    w_covs = NULL,
    w_landmark_type = "state",
    w_max = 10,
    w_stabilised = FALSE,
    w_max_follow = NULL,
    pv_group_vars = NULL,
    pv_n_pctls = NULL,
    pv_precalc = NULL,
    pv_ids = NULL,
    CI = FALSE,
    CI_type = "bootstrap",
    CI_R_boot = NULL,
    CI_seed = NULL,
    transitions_out = NULL,
    assess_moderate = TRUE,
    assess_mean = TRUE,
    ...
)

```

### Arguments

<code>data_ms</code>	Validation data in msdata format
<code>data_raw</code>	Validation data in data.frame (one row per individual)
<code>j</code>	Landmark state at which predictions were made
<code>s</code>	Landmark time at which predictions were made
<code>t</code>	Follow up time at which calibration is to be assessed
<code>tp_pred</code>	Data frame or matrix of predicted transition probabilities at time t, if in state j at time s. There must be a separate column for the predicted transition probabilities into every state, even if these predicted transition probabilities are 0.
<code>tp_pred_plot</code>	Data frame or matrix of predicted risks for each possible transition over which to plot the calibration curves. Argument provided to enable user to apply bootstrapping manually.
<code>calib_type</code>	Whether calibration plots are estimated using BLR-IPCW ('blr'), MLR-IPCW ('mlr') or pseudo-values ('pv')
<code>curve_type</code>	Whether calibration curves are estimated using restricted cubic splines ('rcs') or loess smoothers ('loess')

r <sub>cs</sub> _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
loess_surface	see <a href="#">loess.control</a>
loess_statistics	see <a href="#">loess.control</a>
loess_trace_hat	see <a href="#">loess.control</a>
loess_cell	see <a href="#">loess.control</a>
loess_iterations	see <a href="#">loess.control</a>
loess_iterTrace	see <a href="#">loess.control</a>
mlr_smoother_type	Type of smoothing applied. Takes values s (see <a href="#">s</a> ), sm.ps (see <a href="#">sm.ps</a> ) or sm.os (see <a href="#">sm.os</a> ).
mlr_ps_int	the number of equally-spaced B spline intervals in the vector spline smoother (see <a href="#">sm.ps</a> )
mlr_degree	the degree of B-spline basis in the vector spline smoother (see <a href="#">sm.ps</a> )
mlr_s_df	degrees of freedom of vector spline (see <a href="#">s</a> )
mlr_niknots	number of interior knots (see <a href="#">sm.os</a> )
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.
pv_group_vars	Variables to group by before calculating pseudo-values
pv_n_pctls	Number of percentiles of predicted risk to group by before calculating pseudo-values
pv_precalc	Pre-calculated pseudo-values
pv_ids	Id's of individuals to calculate pseudo-values for
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
CI_seed	Seed for bootstrapping procedure
transitions_out	Transitions for which to calculate calibration curves. Will do all possible transitions if left as NULL.
assess_moderate	TRUE/FALSE whether to estimate data for calibration plots
assess_mean	TRUE/FALSE whether to estimate mean calibration
...	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_type = 'blr'` estimates calibration curves using techniques for assessing the calibration of a binary logistic regression model (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 (van Houwelingen HC, 2007) is applied to only assess calibration in individuals who are uncensored and in state  $j$  at time  $s$ . Calibration can only be assessed in individuals who are also uncensored at time  $t$ , which is accounted for using inverse probability of censoring weights (Hernan M, Robins J, 2020). See method BLR-IPCW from Pate et al., (2024) for a full explanation of the approach.

`calib_type = 'mlr'` estimates calibration scatter plots using 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  $s$ . Calibration can only be assessed in individuals who are also uncensored at time  $t$ , which is accounted for using inverse probability of censoring weights (Hernan M, Robins J, 2020). See method BLR-IPCW from Pate et al., (2024) for a full explanation of the approach.

`calib_type = 'pv'` estimates calibration curves using 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. A choice between restricted cubic splines and loess smoothers for estimating the calibration curve can be made using `curve_type`. Landmarking (van Houwelingen HC, 2007) is applied to only assess calibration in individuals who are uncensored and in state  $j$  at time  $s$ . The nature of pseudo-values means calibration can be assessed in all landmarked individuals, regardless of their censoring time. See method Pseudo-value approach from Pate et al., (2024) for a full explanation of the approach.

Two datasets for the same cohort of individuals must be provided. Firstly, `data_raw` must be a `data.frame` with one row per individual containing the 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`). `data_raw` must also contain the desired variables for estimating the weights. Secondly, `data_ms` must be a dataset of class `msdata`, generated using the `[mstate]` package. This dataset is used to apply the landmarking and identify which

state individuals are in at time  $t$ . While `data_ms` can be derived from `data_raw`, it would be inefficient to do this within `calibmsm::calib_msm` due to the bootstrapping procedure, and therefore they must be inputted separately.

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 produced for the calibration scatter plots (`calib_type = 'mlr'`). For calibration curves estimated using `calib_type = 'blr'`, confidence intervals can only be estimated using bootstrapping (`CI_type = 'bootstrap'`). 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*. For calibration curves estimated using `calib_type = 'pv'`, confidence intervals can be estimated using bootstrapping (`CI_type = 'bootstrap'`) or parametric formulae (`CI_type = 'parametric'`). For computational reasons we recommend using the parametric approach.

The calibration plots can be plotted using `plot.calib_msm` and `plot.calib_mlr`.

## Value

`calib_msm` returns a list containing two elements: `plotdata` and `metadata`. The `plotdata` element contains the data for the calibration plots. 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 `tp_pred_plot` is specified, 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
- 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.
- Pate, A., Sperrin, M., Riley, R. D., Peek, N., Van Staa, T., Sergeant, J. C., Mamas, M. A., Lip, G. Y. H., Flaherty, M. O., Barrowman, M., Buchan, I., & Martin, G. P. Calibration plots for multistate risk predictions models. *Statistics in Medicine.* 2024;April:1–23. doi: 10.1002/sim.10094.
- 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>



- 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

```
# 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.
dat_calib <-
  calib_msm(data_ms = msebmtcal,
    data_raw = ebmtcal,
    j=1,
    s=0,
    t = 1826,
    tp_pred = tp_pred,
    w_covs = c("year", "agec1", "proph", "match"))

# Summarise the output
summary(dat_calib)
```

---

ebmtcal

*European Group for Blood and Marrow Transplantation data (one row per individual)*

---

## 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 identifier  
**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 occurred, 0: absorbing state entered before censoring occurred

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).

---

ebmtcal_cmprsk	<i>European Group for Blood and Marrow Transplantation data (one row per individual)</i>
----------------	--

---

Description

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

Usage

ebmtcal\_cmprsk

## Format

**'ebmtcal\_cmprsk':**

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

**id** Patient identifier

**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 occurred, 0: absorbing state entered before censoring occurred

## Details

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](#). Specifically, the time until censoring is calculated in the setting of a competing risks model out of the first state, where no further transitions can be made. This means entry into any state (as they are all absorbing states) will have the effect of preventing censoring from occurring, and `dtcens` and `dtcens_s` will be different than the values found in [ebmtcal](#). This dataset has been designed to be used alongside dataset [msebmtcal\\_cmprsk](#), when assessing the calibration of a competing risks model. 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).

---

metadata	Create S3 generic for printing metadata
----------	---

---

**Description**

Create S3 generic for printing metadata

**Usage**

```
metadata(x, ...)
```

**Arguments**

- x                    Object generated from [calib\\_msm](#).
- ...                  Extra arguments

---

msebmtool	European Group for Blood and Marrow Transplantation data in msdata format.
-----------	--

---

**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**

```
msebmtool
```

**Format**

- 'msebmtool':**  
A data frame in msdata format (see [msprep](#)) with 15,512 rows and 8 columns:
- id** Patient identifier
  - 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	<i>European Group for Blood and Marrow Transplantation data in competing risks format, for transitions out of the initial state only</i>
------------------	--

---

## Description

Used in vignette/article: [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 identifier

**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_mlr	<i>Plots calibration scatter plots for objects of class calib_mlr estimated using using <a href="#">calib_msm</a>.</i>
----------------	--

---

## Description

Plots calibration scatter plots for the transition probabilities of a multistate model estimated using the MLR-IPCW approach.

## Usage

```
## S3 method for class 'calib_mlr'
plot(
  x,
  ...,
  combine = TRUE,
  ncol = NULL,
  nrow = NULL,
  size_point = 0.5,
  size_text = 12,
  transparency_plot = 0.25,
  marg_density = FALSE,
  marg_density_size = 5,
  marg_density_type = "density",
  marg_rug = FALSE,
  marg_rug_transparency = 0.1,
  titles_include = TRUE,
  titles = NULL,
  axis_titles_x = NULL,
  axis_titles_text_x = "Predicted risk",
```

```

    axis_titles_y = NULL,
    axis_titles_text_y = "Predicted-observed risk"
  )

```

## Arguments

<code>x</code>	Object of class <code>calib_mlr</code> generated from <code>calib_msm</code>
<code>...</code>	Other
<code>combine</code>	Whether to combine into one plot using <code>ggarrange</code> , or return as a list of individual plots
<code>ncol</code>	Number of columns for combined calibration plot
<code>nrow</code>	Number of rows for combined calibration plot
<code>size_point</code>	Size of points in scatter plot
<code>size_text</code>	Size of text in plot
<code>transparency_plot</code>	Degree of transparency for points in the calibration scatter plot
<code>marg_density</code>	Whether to produce marginal density plots TRUE/FALSE
<code>marg_density_size</code>	Size of the main plot relative to the density plots (see <code>ggMarginal</code> )
<code>marg_density_type</code>	What type of marginal plot to show (see <code>ggMarginal</code> )
<code>marg_rug</code>	Whether to produce marginal rug plots TRUE/FALSE
<code>marg_rug_transparency</code>	Degree of transparency for the density rug plot along each axis
<code>titles_include</code>	Whether to include titles for each individual calibration plots
<code>titles</code>	Vector of titles for the calibration plots_ Defaults to "State k" for each plot_
<code>axis_titles_x</code>	Position of plots for which to include title on x-axis
<code>axis_titles_text_x</code>	x-axis title
<code>axis_titles_y</code>	Position of plots for which to include title on y-axis
<code>axis_titles_text_y</code>	y-axis title

## 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 <-
calib_msm(data_ms = msebmtcal_cmprsk,
  data_raw = ebmtcal,
  j=1,
  s=0,
  t = 1826,
  tp_pred = tp_pred,
  calib_type = "mlr",
  w_covs = c("year", "agecl", "proph", "match"),
  mlr_ps_int = 2,
  mlr_degree = 2)

# These are then plotted
plot(dat_calib, combine = TRUE, nrow = 2, ncol = 3)
```

---

plot.calib\_msm

*Plots calibration curves estimated using [calib\\_msm](#).*


---

## Description

Plots calibration curves for the transition probabilities of a multistate model estimated using BLR-IPCW and pseudo-value approaches.

## Usage

```
## S3 method for class 'calib_msm'
plot(
  x,
  ...,
  combine = TRUE,
  ncol = NULL,
```



```

nrow = NULL,
size_line = 0.5,
size_text = 12,
marg_density = TRUE,
marg_density_size = 5,
marg_density_type = "density",
marg_rug = FALSE,
marg_rug_transparency = 0.1,
titles_include = TRUE,
titles = NULL,
axis_titles_x = NULL,
axis_titles_text_x = "Predicted risk",
axis_titles_y = NULL,
axis_titles_text_y = "Predicted-observed risk",
legend_include = TRUE,
legend_separate = FALSE,
legend_title = NULL,
legend_position = "bottom"
)

```

### Arguments

x	Object of class 'calib_msm' generated from <a href="#">calib_msm</a> .
...	Other
combine	Whether to combine into one plot using <a href="#">ggarrange</a> , or return as a list of individual plots
ncol	Number of columns for combined calibration plot
nrow	Number of rows for combined calibration plot
size_line	Size of line plots
size_text	Size of text in plot
marg_density	Whether to produce marginal density plots TRUE/FALSE
marg_density_size	Size of the main plot relative to the density plots (see <a href="#">ggMarginal</a> )
marg_density_type	What type of marginal plot to show (see <a href="#">ggMarginal</a> )
marg_rug	Whether to produce marginal rug plots TRUE/FALSE
marg_rug_transparency	Degree of transparency for the density rug plot along each axis
titles_include	Whether to include titles for each individual calibration plots
titles	Vector of titles for the calibration plots. Defaults to "State k" for each plot.
axis_titles_x	Position of plots for which to include title on x-axis
axis_titles_text_x	x-axis title
axis_titles_y	Position of plots for which to include title on y-axis

```

axis_titles_text_y
      y-axis title
legend_include  Whether to produce a legend
legend_separate
      = Whether to include legend in plot (FALSE) or as a separate object (TRUE)
legend_title    Title of legend
legend_position
      Position of legend

```

### 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 <-
  calib_msm(data_ms = msebmtcal,
    data_raw = ebmtcal,
    j=1,
    s=0,
    t = 1826,
    tp_pred = tp_pred,
    w_covs = c("year", "agec1", "proph", "match"))

# These are then plotted
plot(dat_calib, combine = TRUE, nrow = 2, ncol = 3)

```

---

tps0	<i>Predicted transition probabilities out of transplant state made at time s = 0</i>
------	--

---

### 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 identifier**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	<i>Predicted transition probabilities out of every state made at time <math>s = 100</math></i>
--------	--

---

**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 identifier  
**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).

---

tp_cmprsk_j0	<i>Predicted risks for a competing risks model out of state j = 0</i>
--------------	---

---

Description

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

Usage

tp\_cmprsk\_j0

Format

**'tp\_cmprsk\_j0':**  
A data frame with 2,279 rows and 13 columns:  
**id** Patient identifier  
**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

## Details

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.

## 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, [9](#)
- ebmtcal\_cmprsk, [10](#)
- msebmcal, [12](#)
- msebmcal\_cmprsk, [13](#)
- tp\_cmprsk\_j0, [20](#)
- tps0, [18](#)
- tps100, [19](#)

calc\_weights, [2](#)  
calib\_msm, [4](#), [8](#), [12](#), [14–17](#)

ebmt4, [9–14](#), [18–21](#)  
ebmtcal, [9](#), [11](#)  
ebmtcal\_cmprsk, [10](#)

ggMarginal, [15](#), [17](#)

loess.control, [6](#)

metadata, [12](#)  
msebmcal, [12](#)  
msebmcal\_cmprsk, [11](#), [13](#), [21](#)  
msprep, [12](#), [13](#)

plot.calib\_mlr, [8](#), [14](#)  
plot.calib\_msm, [8](#), [16](#)

s, [6](#)  
sm.os, [6](#)  
sm.ps, [6](#)

tp\_cmprsk\_j0, [20](#)  
tps0, [18](#)  
tps100, [19](#)