# Package 'TPmsm'

August 4, 2019

**Encoding** UTF-8

**Type** Package

**Depends** R (>= 2.12.0),graphics,grDevices,KernSmooth

**Enhances** p3state.msm,etm

**Title** Estimation of Transition Probabilities in Multistate Models

**Version** 1.2.2

**Date** 2019-08-04

**Author** Artur Araujo, Javier Roca-Pardinas <roca@uvigo.es>
and Luis Meira-Machado <lmachado@math.uminho.pt>

**Maintainer** Artur Araujo <artur.stat@gmail.com>

**Description** Estimation of transition probabilities for the
illness-death model and or the three-state progressive model.

**License** GPL (>= 2)

**LazyLoad** yes

**LazyData** yes

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2019-08-04 20:30:03 UTC

## R topics documented:

---

`TPmsm-package` *Estimation of Transition Probabilities in Multistate Models*

---

## Description

The **TPmsm** software contains functions that compute estimates for the transition probabilities in the illness-death model and or the three-state progressive model. This package implements seven different estimators. Being five of them non-parametric and two of them semi-parametric (PAJ and KMPW). The implemented estimators are the Aalen-Johansen estimator (AJ), Presmoothed Aalen-Johansen estimator (PAJ), Kaplan-Meier Weighted estimator (KMW), Presmoothed Kalpan-Meier Weighted estimator (KMPW), Inverse Probability of Censoring estimator (IPCW), Lin estimator (LIN) and Location-Scale estimator (LS). The Inverse Probability of Censoring (IPCW) and Lin (LIN) estimators also permit to compute transition probabilities conditioned on a single covariate. Bootstrap confidence bands can be computed for each of the mentioned estimators. Several graphical plots of the transition probabilities with or without confidence bands can be drawn. To aid in the study of the statistical properties of the implemented estimators, functions to generate pseudo-random data for some well-known multivariate distributions were implemented.

## Details

| | |
|---|---|
| Package: | TPmsm |
| Type: | Package |
| Version: | 1.2.1 |
| Date: | 2015-10-02 |
| License: | GPL (>= 2) |
| LazyLoad: | yes |
| LazyData: | yes |

## Author(s)

Artur Araújo, Javier Roca-Pardiñas <roca@uvigo.es>
and Luís Meira-Machado <lmachado@math.uminho.pt>
Maintainer: Artur Araújo <artur.stat@gmail.com>

## References

Aalen O. O., Johansen S. (1978) An Empirical Transition Matrix for Nonhomogeneous Markov Chains Based on Censored Observations. *Scandinavian Journal of Statistics* **5(3)**, 141–150.

Allignol A., Schumacher M., Beyersmann J. (2011) Empirical Transition Matrix of Multi-State Models: The etm Package. *Journal of Statistical Software* **38(4)**, 1–15.

Amorim A. P., de Uña-Álvarez J., Meira Machado L. F. (2011) Presmoothing the transition probabilities in the illness-death model. *Statistics and Probability Letters* **81**, 797–806.

Davison, A. C., Hinkley, D. V. (1997) *Bootstrap Methods and their Application* Chapter 5, Cambridge University Press.

Devroye L. (1986) *Non-Uniform Random Variate Generation* New-York: Springer-Verlag.

Johnson M. E. (1987) *Multivariate Statistical Simulation* John Wiley and Sons.

Johnson N., Kotz S. (1972) *Distributions in statistics: continuous multivariate distributions* John Wiley and Sons.

Karl A. T., Eubank R., Milovanovic J., Reiser M., Young D. (2014) Using RngStreams for parallel random number generation in C++ and R. *Computational Statistics* **29(5)**, 1301–1320.

L'Ecuyer, P. (1999) Good parameters and implementations for combined multiple recursive random number generators. *Operations Research* **47(1)**, 159—164.

L'Ecuyer P., Simard R., Chen E. J., Kelton W. D. (2002) An object-oriented random-number package with many long streams and substreams. *Operations Research* **50(6)**, 1073—1075.

Lu J., Bhattacharya G. (1990) Some new constructions of bivariate weibull models. *Annals of Institute of Statistical Mathematics* **42(3)**, 543–559.

Meira Machado L. F., de Uña-Álvarez J., Cadarso-Suárez C. (2006) Nonparametric estimation of transition probabilities in a non-Markov illness-death model. *Lifetime Data Anal* **12(3)**, 325–344.

Meira-Machado L., de Uña-Álvarez J., Datta S. Conditional Transition Probabilities in a non-Markov Illness-death Model. Discussion Papers in Statistics and Operations Research n 11/03, 2011. Department of Statistics and Operations Research, University of Vigo (ISSN: 1888-5756, Deposito Legal VG 1402 - 2007). This file can be downloaded from: http://webs.uvigo.es/depc05/reports/12_05.pdf

Meira-Machado L., Roca-Pardiñas J. (2011) p3state.msm: Analyzing Survival Data from an Illness-Death Model. *Journal of Statistical Software* **38(3)**, 1–18.

Meira-Machado L., Roca-Pardiñas J., Van Keilegom I., Cadarso-Suárez C. (2013) Bandwidth Selection for the Estimation of Transition Probabilities in the Location-Scale Progressive Three-State Model. *Computational Statistics* **28(5)**, 2185–2210.

Meira-Machado L., Roca-Pardiñas J., Van Keilegom I. Cadarso-Suárez C. Estimation of transition probabilities in a non-Markov model with successive survival times. Discussion paper 2010. This

file can be downloaded from: http://sites.uclouvain.be/IAP-Stat-Phase-V-VI/ISBApub/dp2010/DP1053.pdf

Moreira A., de Uña-Álvarez J., Meira-Machado L. Presmoothing the Aalen-Johansen estimator of transition probabilities. Discussion Papers in Statistics and Operation Research n 11/03, 2011. Department of Statistics and Operations Research, University of Vigo (ISSN: 1888-5756, Deposito Legal VG 1402 - 2007). This file can be downloaded from: http://webs.uvigo.es/depc05/reports/11_03.pdf

Van Keilegom I., de Uña-Álvarez J., Meira-Machado L. (2011) Nonparametric location-scale models for successive survival times under dependent censoring. *Journal of Statistical Planning and Inference* **141(3)**, 1118–1131.

---

`as.data.frame.survTP` *as.data.frame method for a survTP object*

---

## Description

as.data.frame method for an object of class 'survTP'.

## Usage

```
## S3 method for class 'survTP'
as.data.frame(x, ..., package="TPmsm")
```

## Arguments

| | |
|---|---|
| `x` | An object of class 'survTP'. |
| `...` | Additional arguments to be passed to or from method. |
| `package` | The format of the data.frame. Possible options are "TPmsm", "p3state.msm" and "etm". Defaults to "TPmsm". |

## Value

A data.frame in the format specified by argument `package`.

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## References

Allignol A., Schumacher M., Beyersmann J. (2011) Empirical Transition Matrix of Multi-State Models: The etm Package. *Journal of Statistical Software* **38(4)**, 1–15.

Meira-Machado L., Roca-Pardiñas J. (2011) p3state.msm: Analyzing Survival Data from an Illness-Death Model. *Journal of Statistical Software* **38(3)**, 1–18.

## See Also

as.data.frame, survTP.

## Examples

```
# Example for the "TPmsm" format
weiTP <- dgpTP(n=100, corr=1, dist="weibull", dist.par=c(2, 7, 2, 7),
model.cens="exponential", cens.par = 6, state2.prob=0.6)
weidata <- as.data.frame(weiTP)
head(weidata)

# Example for the "etm" format
expTP <- dgpTP(n=100, corr=1, dist="exponential", dist.par=c(1, 1),
model.cens="uniform", cens.par=3, state2.prob=0.5)
expdata <- as.data.frame(expTP, package="etm")
head(expdata)
```

---

# bladderTP *Bladder cancer recurrences*

---

## Description

This contains the bladder cancer recurrences data in a different format. In this study, patients had superficial bladder tumors that were removed by transurethral resection. Many patients had multiple recurrences (up to a maximum of 9) of tumors during the study, and new tumors were removed at each visit. Only the first two recurrence times (in months) are considered.

## Usage

```
data(bladderTP)
```

## Format

A data frame with 85 observations on the following 4 variables.

`time1` Time until first recurrence/censoring time.

`event1` First recurrence indicator.

`Stime` Time until second recurrence/censoring time, i.e. the total time of the process.

`event` First or second recurrence indicator, i.e. the censoring indicator of the total time.

## References

Wei L. J., Lin D. Y., Weissfeld L. (1989) Regression analysis of multivariate incomplete failure time data by modeling marginal distributions. *Journal of the American Statistical Association* **84(408)**, 1065–1073.

colonTP *Chemotherapy for Stage B/C colon cancer*

## Description

These are data from one of the first successful trials of adjuvant chemotherapy for colon cancer. Levamisole is a low-toxicity compound previously used to treat worm infestations in animals; 5-FU is a moderately toxic (as these things go) chemotherapy agent.

## Usage

```
data(colonTP)
```

## Format

A data frame with 929 observations on the following 15 variables.

`time1` Time to recurrence/censoring/death, whichever occurs first.

`event1` Recurrence/censoring/death indicator (recurrence/dead=1, alive=0).

`Stime` Time to death/censoring.

`event` Censoring indicator (dead=1, alive=0).

`rx` Treatment - Obs(ervation), Lev(amisole), Lev(amisole)+5-FU.

`sex` 1=male.

`age` Age in years.

`obstruct` Obstruction of colon by tumour.

`perfor` Perforation of colon.

`adhere` Adherence to nearby organs.

`nodes` Number of lymph nodes with detectable cancer.

`differ` Differentiation of tumour (1=well, 2=moderate, 3=poor).

`extent` Extent of local spread (1=submucosa, 2=muscle, 3=serosa, 4=contiguous structures).

`surg` Time from surgery to registration (0=short, 1=long).

`node4` More than 4 positive lymph nodes.

## Note

The study is originally described in Laurie (1989). The main report is found in Moertel (1990). This data set is closest to that of the final report in Moertel (1991). A version of the data with less follow-up time was used in the paper by Lin (1994).

## References

Laurie J. A., Moertel C. G., Fleming T. R., Wieand H. S., Leigh J. E., Rubin J., McCormack G. W., Gerstner J. B., Krook J. E., Malliard J. (1989) Surgical adjuvant therapy of large-bowel carcinoma: An evaluation of levamisole and the combination of levamisole and fluorouracil: The North Central Cancer Treatment Group and the Mayo Clinic. *Journal of Clinical Oncology* **7(10)**, 1447–1456.

Lin D. Y. (1994) Cox regression analysis of multivariate failure time data: the marginal approach. *Statistics in Medicine* **13(21)**, 2233–2247.

Moertel C. G., Fleming T. R., MacDonald J. S., Haller D. G., Laurie J. A., Goodman P.J., Ungerleider J.S., Emerson W.A., Tormey D.C., Glick J.H., Veeder M.H., Maillard J.A. (1990) Levamisole and fluorouracil for adjuvant therapy of resected colon carcinoma. *New England Journal of Medicine* **322(6)**, 352–358.

Moertel C. G., Fleming T. R., MacDonald J. S., Haller D. G., Laurie J. A., Tangen C. M., Ungerleider J. S., Emerson W. A., Tormey D. C., Glick J. H., Veeder M. H., Maillard J. A. (1991) Fluorouracil plus Levamisole as an effective adjuvant therapy after resection of stage II colon carcinoma: a final report. *Annals of Internal Medicine* **122(5)**, 321–326.

---

`contour.TPCmsm` *contour method for a TPCmsm object*

---

## Description

contour method for an object of class 'TPCmsm'. Creates a contour plot of the transition probabilities.

## Usage

```
## S3 method for class 'TPCmsm'
contour(x, contour.type="tc", tr.choice, nlevels=20, levels=pretty(zlim, nlevels),
xlim, ylim, zlim=c(0, 1), col=grey(0.4), xlab, ylab, main="", sub="",
add=FALSE, las=1, conf.int=FALSE, legend=TRUE, curvlab, ...)
```

## Arguments

| | |
|---|---|
| `x` | An object of class 'TPCmsm'. |
| `contour.type` | A character string specifying the type of contour. If "tc" the contour with time in the x axis, covariate in the y axis and transition probability in the z axis is drawn. If "ct" the contour with covariate in the x axis, time in the y axis and transition probability in the z axis is drawn. Defaults to "tc". |
| `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. |
| `nlevels` | The number of levels to divide the range of z. Defaults to 20 levels. |
| `levels` | Numeric vector of levels at which to draw contour lines. Defaults to `pretty(zlim,nlevels)`. |
| `xlim` | Limits of x-axis for the plot. |
| `ylim` | Limits of y-axis for the plot. |

| | |
|---|---|
| zlim | Limits of z-axis for the plot. Defaults to c(0,1). |
| col | Color for the lines drawn. Defaults to grey(0.4). |
| xlab | x-axis label. If contour.type="tc" defaults to "Time". If contour.type="ct" defaults to "Covariate". |
| ylab | y-axis label. If contour.type="tc" defaults to "Covariate". If contour.type="ct" defaults to "Time". |
| main | The main title for the plot. By default no main title is added. |
| sub | A sub title for the plot. By default no sub title is added. |
| add | logical. If TRUE, add to a current plot. |
| las | The style of labeling to be used. The default is to use horizontal labeling. |
| conf.int | Logical. Whether to display contour plots of confidence regions. Default is FALSE. |
| legend | A logical specifying if a legend should be added. |
| curvlab | A character or expression vector to appear in the legend. Default is the name of the transitions. |
| ... | Further arguments for contour. |

## Value

No value is returned.

## Note

The device is divided by the number of transitions specified by argument tr.choice. Being the number of columns equal to the number of transitions. If argument conf.int=TRUE the device is further divided to make room for the confidence regions. In this case two rows are added, one for each side of the confidence region. So if conf.int=TRUE the center row provides the contour of the estimates. The upper row provides the upper side of the confidence region. And the lower row provides the lower side of the confidence region.

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## References

Meira-Machado L., de Uña-Álvarez J., Datta S. Conditional Transition Probabilities in a non-Markov Illness-death Model. Discussion Papers in Statistics and Operations Research n 11/03, 2011. Department of Statistics and Operations Research, University of Vigo (ISSN: 1888-5756, Deposito Legal VG 1402 - 2007). This file can be downloaded from: http://webs.uvigo.es/depc05/reports/12_05.pdf

## See Also

contour.

## Examples

```
# Set the number of threads
nth <- setThreadsTP(2)

# Create survTP object
data(colonTP)
colonTP_obj <- with( colonTP, survTP(time1, event1, Stime, event, age=age) )

# Compute IPCW conditional transition probabilities without confidence band
TPCmsm_obj <- transIPCW(colonTP_obj, s=57, t=310, x=0)

# Plot contour with Time in the x-axis
contour(TPCmsm_obj, contour.type="tc", tr.choice=c("1 1", "1 2", "2 2"), ylab="Age")

# Plot contour with Time in the y-axis
contour(TPCmsm_obj, contour.type="ct", tr.choice=c("1 1", "1 2", "1 3"), xlab="Age")

# Restore the number of threads
setThreadsTP(nth)
```

---

# corrTP *Correlation between two gap times*

---

## Description

Provides the correlation between the bivariate times for some copula distributions.

## Usage

```
corrTP(dist, corr, dist.par)
```

## Arguments

| | |
|---|---|
| `dist` | The distribution. Possible bivariate distributions are "exponential" and "weibull". |
| `corr` | Correlation parameter. Possible values for the bivariate exponential distribution are between -1 and 1 (0 for independency). Any value between 0 (not included) and 1 (1 for independency) is accepted for the bivariate Weibull distribution. |
| `dist.par` | Vector of parameters for the allowed distributions. Two (scale) parameters for the bivariate exponential distribution and four (2 location parameters and 2 scale parameters) for the bivariate Weibull distribution. See details below. |

## Details

The bivariate exponential distribution, also known as Farlie-Gumbel-Morgenstern distribution is given by

$$F(x, y) = F_1(x)F_2(y)[1 + \alpha(1 - F_1(x))(1 - F_2(y))]$$

for $x \geq 0$ and $y \geq 0$. Where the marginal distribution functions $F_1$ and $F_2$ are exponential with scale parameters $\theta_1$ and $\theta_2$ and correlation parameter $\alpha$, $-1 \leq \alpha \leq 1$.

The bivariate Weibull distribution with two-parameter marginal distributions. It's survival function is given by

$$S(x,y) = P(X > x, Y > y) = e^{-[(\frac{x}{\theta_1})^{\frac{\beta_1}{\delta}} + (\frac{y}{\theta_2})^{\frac{\beta_2}{\delta}}]^{\delta}}$$

Where $0 < \delta \leq 1$ and each marginal distribution has shape parameter $\beta_i$ and a scale parameter $\theta_i$, $i = 1, 2$.

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## References

Johnson N., Kotz S. (1972) *Distributions in statistics: continuous multivariate distributions* John Wiley and Sons.

Lu J., Bhattacharya G. (1990) Some new constructions of bivariate weibull models. *Annals of Institute of Statistical Mathematics* **42(3)**, 543–559.

## See Also

dgpTP.

## Examples

```
# Example for the bivariate Weibull distribution
corrTP(dist = "weibull", corr = 0.5, dist.par = c(2, 7, 2, 7))

# Example for the bivariate Exponential distribution
corrTP(dist = "exponential", corr = 1, dist.par = c(1, 1))
```

---

# dgpTP *Generates bivariate survival data*

---

## Description

Generates bivariate censored gap times from some known copula functions.

## Usage

```
dgpTP(n, corr, dist, dist.par, model.cens, cens.par, state2.prob)
```

## Arguments

| | |
|---|---|
| n | Sample size. |
| corr | Correlation parameter. Possible values for the bivariate exponential distribution are between -1 and 1 (0 for independency). Any value between 0 (not included) and 1 (1 for independency) is accepted for the bivariate Weibull distribution. |
| dist | Distribution. Possible bivariate distributions are "exponential" and "weibull". |
| dist.par | Vector of parameters for the allowed distributions. Two (scale) parameters for the bivariate exponential distribution and four (2 location parameters and 2 scale parameters) for the bivariate Weibull distribution. See details below. |
| model.cens | Model for censorship. Possible values are "uniform" and "exponential". |
| cens.par | Parameter for the censorship distribution. For censure model equal to "exponential" the argument cens.par must be greater than 0. For censure model equal to "uniform" the argument must be greater or equal than 0. |
| state2.prob | The proportion of individuals that enter state 2. |

## Details

The bivariate exponential distribution, also known as Farlie-Gumbel-Morgenstern distribution is given by

$$F(x,y) = F_1(x)F_2(y)[1 + \alpha(1 - F_1(x))(1 - F_2(y))]$$

for $x \geq 0$ and $y \geq 0$. Where the marginal distribution functions $F_1$ and $F_2$ are exponential with scale parameters $\theta_1$ and $\theta_2$ and correlation parameter $\alpha$, $-1 \leq \alpha \leq 1$.

The bivariate Weibull distribution with two-parameter marginal distributions. It's survival function is given by

$$S(x,y) = P(X > x, Y > y) = e^{-[(\frac{x}{\theta_1})^{\frac{\beta_1}{\delta}} + (\frac{y}{\theta_2})^{\frac{\beta_2}{\delta}}]^{\delta}}$$

Where $0 < \delta \leq 1$ and each marginal distribution has shape parameter $\beta_i$ and a scale parameter $\theta_i$, $i = 1, 2$.

## Value

An object of class 'survTP'.

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## References

Devroye L. (1986) *Non-Uniform Random Variate Generation* New-York: Springer-Verlag.

Johnson N., Kotz S. (1972) *Distributions in statistics: continuous multivariate distributions* John Wiley and Sons.

Lu J., Bhattacharya G. (1990) Some new constructions of bivariate weibull models. *Annals of Institute of Statistical Mathematics* **42(3)**, 543–559.

Johnson M. E. (1987) *Multivariate Statistical Simulation* John Wiley and Sons.

## See Also

corrTP.

## Examples

```
# Set the number of threads
nth <- setThreadsTP(2)

# Example for the bivariate Exponential distribution
dgpTP(n=100, corr=1, dist="exponential", dist.par=c(1, 1),
model.cens="uniform", cens.par=3, state2.prob=0.5)

# Example for the bivariate Weibull distribution
dgpTP(n=100, corr=1, dist="weibull", dist.par=c(2, 7, 2, 7),
model.cens="exponential", cens.par = 6, state2.prob=0.6)

# Restore the number of threads
setThreadsTP(nth)
```

---

# heartTP *More Stanford heart transplant data*

---

## Description

This contains the Stanford heart transplant data in a different format. The main data set is in (heart). Survival of patients on the waiting list for the Stanford heart transplant program.

## Usage

```
data(heartTP)
```

## Format

A data frame with 103 observations on the following 7 variables.

`time1` Time to transplant/censoring/death, whichever occurs first.

`event1` Transplant/censoring/death indicator (transplanted/dead=1, alive=0).

`Stime` Time to death/censoring.

`event` Censoring indicator (dead=1, alive=0).

`age` age-48 years.

`year` Year of acceptance; in years after 1 Nov 1967.

`surgery` Prior bypass surgery; 1=yes.

## References

Crowley J., Hu M. (1977) Covariance analysis of heart transplant survival data. *Journal of the American Statistical Association* **72(357)**, 27–36.

`image.TPCmsm` *image method for a TPCmsm object*

## Description

image method for an object of class 'TPCmsm'. Creates a grid of colored or gray-scale rectangles with colors corresponding to the values of the transition probabilities.

## Usage

```
## S3 method for class 'TPCmsm'
image(x, image.type="tc", tr.choice, xlim, ylim, zlim=c(0, 1), col, xlab, ylab,
main, sub, key.title, key.axes, las=1, conf.int=FALSE, legend=TRUE, curvlab,
contour=TRUE, nlevels=20, levels=pretty(zlim, nlevels), ...)
```

## Arguments

| | |
|---|---|
| `x` | An object of class 'TPCmsm'. |
| `image.type` | A character string specifying the type of image. If "tc" the image with time in the x axis, covariate in the y axis and transition probability in the z axis is drawn. If "ct" the image with covariate in the x axis, time in the y axis and transition probability in the z axis is drawn. Defaults to "tc". |
| `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. |
| `xlim` | Limits of x-axis for the plot. |
| `ylim` | Limits of y-axis for the plot. |
| `zlim` | Limits of z-axis for the plot. Defaults to `c(0,1)`. |
| `col` | Vector of colour. Defaults to `heat.colors(nlevels)[nlevels:1]`. |
| `xlab` | x-axis label. If `image.type="tc"` defaults to "Time". If `image.type="ct"` defaults to "Covariate". |
| `ylab` | y-axis label. If `image.type="tc"` defaults to "Covariate". If `image.type="ct"` defaults to "Time". |
| `main` | The main title for the plot. By default no main title is added. |
| `sub` | A sub title for the plot. By default no sub title is added. |
| `key.title` | Statements which add titles for the plot key. |
| `key.axes` | Statements which draw axes on the plot key. This overrides the default axis. |
| `las` | The style of labeling to be used. The default is to use horizontal labeling. |
| `conf.int` | Logical. Whether to display images of confidence regions. Default is FALSE. |
| `legend` | A logical specifying if a legend should be added. |
| `curvlab` | A character or expression vector to appear in the legend. Default is the name of the transitions. |
| `contour` | If `TRUE` contour lines are added to the image. Default is `TRUE`. |

| | |
|---|---|
| nlevels | The number of levels to divide the range of z. Defaults to 20 levels. |
| levels | Numeric vector of levels at which to draw contour lines. Defaults to pretty(zlim,nlevels). The breaks of image are set equal to levels. |
| ... | Further arguments for image. |

## Value

No value is returned.

## Note

The device is divided by the number of transitions specified by argument tr.choice. Being the number of columns equal to the number of transitions. If argument conf.int=TRUE the device is further divided to make room for the confidence regions. In this case two rows are added, one for each side of the confidence region. So if conf.int=TRUE the center row provides the image of the estimates. The upper row provides the upper side of the confidence region. And the lower row provides the lower side of the confidence region.

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## References

Meira-Machado L., de Uña-Álvarez J., Datta S. Conditional Transition Probabilities in a non-Markov Illness-death Model. Discussion Papers in Statistics and Operation Research n 11/03, 2011. Department of Statistics and Operations Research, University of Vigo (ISSN: 1888-5756, Deposito Legal VG 1402 - 2007). This file can be downloaded from: http://webs.uvigo.es/depc05/reports/12_05.pdf

## See Also

contour, image.

## Examples

```
# Set the number of threads
nth <- setThreadsTP(2)

# Create survTP object
data(heartTP)
heartTP_obj <- with( heartTP, survTP(time1, event1, Stime, event, age=age) )

# Compute LIN conditional transition probabilities with confidence band
TPCmsm_obj <- transLIN(heartTP_obj, s=57, t=310, x=0, conf=TRUE, n.boot=100,
method.boot="basic")

# Plot image with Time in the x-axis
image(TPCmsm_obj, image.type="tc", tr.choice=c("1 1", "1 2", "2 2"), conf.int=TRUE,
ylab="Age")
```

```
# Plot image with Time in the y-axis
image(TPCmsm_obj, image.type="ct", tr.choice=c("1 1", "1 2", "1 3"), conf.int=TRUE,
xlab="Age")

# Restore the number of threads
setThreadsTP(nth)
```

---

`lines.TPCmsm` *lines method for a TPCmsm object*

---

## Description

lines method for an object of class 'TPCmsm'.

## Usage

```
## S3 method for class 'TPCmsm'
lines(x, plot.type="t", tr.choice, col, lty, conf.int=FALSE,
ci.col, ci.lty, legend=FALSE, legend.pos, curvlab, legend.bty="n", ...)
```

## Arguments

| | |
|---|---|
| `x` | An object of class 'TPCmsm'. |
| `plot.type` | A character string specifying the type of plot. If "t" the scatterplot of transition probability versus time is plotted. If "c" the scatterplot of transition probability versus covariate is plotted. |
| `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. |
| `col` | Vector of colour. |
| `lty` | Vector of line type. Default is 1:number of transitions. |
| `conf.int` | Logical. Whether to display pointwise confidence bands. Default is FALSE. |
| `ci.col` | Colour of the confidence bands. Default is `col`. |
| `ci.lty` | Line type of the confidence bands. 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. By default no box is drawn. |
| `...` | Further arguments for lines. |

## Value

No value is returned.

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## References

Meira-Machado L., de Uña-Álvarez J., Datta S. Conditional Transition Probabilities in a non-Markov Illness-death Model. Discussion Papers in Statistics and Operation Research n 11/03, 2011. Department of Statistics and Operations Research, University of Vigo (ISSN: 1888-5756, Deposito Legal VG 1402 - 2007). This file can be downloaded from: http://webs.uvigo.es/depc05/reports/12_05.pdf

## See Also

legend, lines, plot.default, plot.TPCmsm.

## Examples

```
# Set the number of threads
nth <- setThreadsTP(2)

# Create survTP object
data(heartTP)
heartTP_obj <- with( heartTP, survTP(time1, event1, Stime, event, age=age) )

# Compute IPCW1 conditional transition probabilities without confidence band
TPC_IPCW1 <- transIPCW(heartTP_obj, s=57, t=310, x=15, conf=FALSE, method.est=1)

# Compute IPCW2 conditional transition probabilities without confidence band
TPC_IPCW2 <- transIPCW(heartTP_obj, s=57, t=310, x=15, conf=FALSE, method.est=2)

# Compute LIN conditional transition probabilities without confidence band
TPC_LIN <- transLIN(heartTP_obj, s=57, t=310, x=15, conf=FALSE)

# Build covariate plots
tr.choice <- dimnames(TPC_LIN$est)[[3]]
par.orig <- par( c("mfrow", "cex") )
par( mfrow=c(2,3) )
for ( i in seq_len( length(tr.choice) ) ) {
plot(TPC_IPCW1, plot.type="c", tr.choice=tr.choice[i], legend=FALSE,
main=tr.choice[i], col=1, lty=1, xlab="", ylab="")
lines(TPC_IPCW2, plot.type="c", tr.choice=tr.choice[i], legend=FALSE, col=2, lty=1)
lines(TPC_LIN, plot.type="c", tr.choice=tr.choice[i], legend=FALSE, col=3, lty=1)
}
plot.new()
legend(x="center", legend=c("IPCW1", "IPCW2", "LIN"), col=1:3, lty=1, bty="n", cex=1.5)
par(mfrow=c(1, 1), cex=1.2)
title(xlab="Age", ylab="Transition probability", line=3)
par(par.orig)

# Restore the number of threads
setThreadsTP(nth)
```

`lines.TPmsm` *lines method for a TPmsm object*

## Description

lines method for an object of class 'TPmsm'.

## Usage

```
## S3 method for class 'TPmsm'
lines(x, tr.choice, col, lty, conf.int=FALSE, ci.col, ci.lty,
legend=FALSE, legend.pos, curvlab, legend.bty="n", ...)
```

## Arguments

| | |
|---|---|
| `x` | An object of class 'TPmsm'. |
| `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. |
| `col` | Vector of colour. Default is black. |
| `lty` | Vector of line type. Default is 1:number of transitions. |
| `conf.int` | Logical. Whether to display pointwise confidence bands. Default is FALSE. |
| `ci.col` | Colour of the confidence bands. Default is `col`. |
| `ci.lty` | Line type of the confidence bands. 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. By default no box is drawn. |
| `...` | Further arguments for lines. |

## Value

No value is returned.

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## See Also

`legend`, `lines`, `plot.default`, `plot.TPmsm`.

## Examples

```
# Set the number of threads
nth <- setThreadsTP(2)

# Create survTP object
data(bladderTP)
bladderTP_obj <- with( bladderTP, survTP(time1, event1, Stime, event) )

# Compute transition probabilities without confidence band
KMW <- transKMW(object=bladderTP_obj, s=5, t=59, conf=FALSE, method.est=1)
KMPW <- transKMPW(object=bladderTP_obj, s=5, t=59, conf=FALSE, method.est=1)
AJ <- transAJ(object=bladderTP_obj, s=5, t=59, conf=FALSE)
PAJ <- transPAJ(object=bladderTP_obj, s=5, t=59, conf=FALSE)
LIN <- transLIN(object=bladderTP_obj, s=5, t=59, conf=FALSE)
LS <- transLS(object=bladderTP_obj, s=5, t=59, h=c(0.25, 2.5),
nh=25, ncv=50, conf=FALSE)

# Plot '1 2' KMW transition probability estimate
par( mfrow=c(1, 1) )
plot(KMW, tr.choice="1 2", ylab="P12(5, Time)", xlab="Time",
col=1, lty=1, legend=FALSE)

# Add other '1 2' transition probability estimates
lines(KMPW, tr.choice="1 2", col=2, lty=1)
lines(AJ, tr.choice="1 2", col=3, lty=1)
lines(PAJ, tr.choice="1 2", col=4, lty=1)
lines(LIN, tr.choice="1 2", col=5, lty=1)
lines(LS, tr.choice="1 2", col=6, lty=1)

# Add legend
legend(x="topleft", legend=c("KMW", "KMPW", "AJ", "PAJ", "LIN", "LS"),
col=1:6, lty=1, bty="n")

# Plot all the transitions
tr.choice <- colnames(KMW$est)
par.orig <- par( c("mfrow", "cex") )
par( mfrow=c(2, 3) )
for ( i in seq_len( length(tr.choice) ) ) {
plot(KMW, tr.choice=tr.choice[i], col=1, lty=1, legend=FALSE,
main=tr.choice[i], xlab="", ylab="")
lines(KMPW, tr.choice=tr.choice[i], col=2, lty=1)
lines(AJ, tr.choice=tr.choice[i], col=3, lty=1)
lines(PAJ, tr.choice=tr.choice[i], col=4, lty=1)
lines(LIN, tr.choice=tr.choice[i], col=5, lty=1)
lines(LS, tr.choice=tr.choice[i], col=6, lty=1)
}
plot.new()
legend(x="center", legend=c("KMW", "KMPW", "AJ", "PAJ", "LIN", "LS"),
col=1:6, lty=1, bty="n", cex=1.5)
par(mfrow=c(1, 1), cex=1.2)
title(xlab="Time", ylab="Transition probability", line=3)
par(par.orig)
```

```
# Restore the number of threads
setThreadsTP(nth)
```

---

`plot.TPCmsm` *plot method for a TPCmsm object*

---

## Description

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

## Usage

```
## S3 method for class 'TPCmsm'
plot(x, plot.type="t", tr.choice, xlab, ylab, col, lty, xlim, ylim,
conf.int=FALSE, ci.col, ci.lty, legend=TRUE, legend.pos, curvlab,
legend.bty="n", ...)
```

## Arguments

| | |
|---|---|
| `x` | An object of class 'TPCmsm'. |
| `plot.type` | A character string specifying the type of plot. If "t" the scatterplot of transition probability versus time is plotted. If "c" the scatterplot of transition probability versus covariate is plotted. |
| `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. |
| `ylab` | y-axis label. |
| `col` | Vector of colour. |
| `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 bands. Default is FALSE. |
| `ci.col` | Colour of the confidence bands. Default is `col`. |
| `ci.lty` | Line type of the confidence bands. 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. By default no box is drawn. |
| `...` | Further arguments for plot. |

## Value

No value is returned.

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## References

Meira-Machado L., de Uña-Álvarez J., Datta S. Conditional Transition Probabilities in a non-Markov Illness-death Model. Discussion Papers in Statistics and Operation Research n 11/03, 2011. Department of Statistics and Operations Research, University of Vigo (ISSN: 1888-5756, Deposito Legal VG 1402 - 2007). This file can be downloaded from: http://webs.uvigo.es/depc05/reports/12_05.pdf

## See Also

legend, plot.default.

## Examples

```
# Set the number of threads
nth <- setThreadsTP(2)

# Create survTP object
data(heartTP)
heartTP_obj <- with( heartTP, survTP(time1, event1, Stime, event, age=age) )

# Compute IPCW conditional transition probabilities with confidence band
TPCmsm_obj <- transIPCW(heartTP_obj, s=57, t=310, x=c(0, 15), conf=TRUE, n.boot=100,
method.boot="percentile", method.est=2)

# Build time plots
tr.choice <- dimnames(TPCmsm_obj$est)[[3]]
par.orig <- par( c("mfrow", "cex") )
par( mfrow=c(2,3) )
for ( i in seq_len( length(tr.choice) ) ) {
plot(TPCmsm_obj, plot.type="t", tr.choice=tr.choice[i], conf.int=TRUE, legend=TRUE,
main=tr.choice[i], col=seq_len( length(TPCmsm_obj$x) ), lty=1, xlab="", ylab="",
curvlab=c("Age = 0", "Age = 15"))
}
par(mfrow=c(1, 1), cex=1.2)
title(xlab="Time", ylab="Transition probability", line=3)
par(par.orig)

# Build covariate plots without colors and without confidence band
plot(TPCmsm_obj, plot.type="c", xlab="Age")

# Build covariate plots with colors and without confidence band
plot(TPCmsm_obj, plot.type="c", col=seq_len(5), lty=1, xlab="Age")
```

```
# Build covariate plots with confidence band
tr.choice <- dimnames(TPCmsm_obj$est)[[3]]
par.orig <- par( c("mfrow", "cex") )
par( mfrow=c(2,3) )
for ( i in seq_len( length(tr.choice) ) ) {
plot(TPCmsm_obj, plot.type="c", tr.choice=tr.choice[i], conf.int=TRUE, legend=FALSE,
main=tr.choice[i], xlab="", ylab="")
}
par(mfrow=c(1, 1), cex=1.2)
title( xlab="Age", ylab=paste("P(", TPCmsm_obj$s, ", ", TPCmsm_obj$t, " | Age)", sep=""), line=3)
par(par.orig)

# Restore the number of threads
setThreadsTP(nth)
```

---

`plot.TPmsm` *plot method for a TPmsm object*

---

## Description

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

## Usage

```
## S3 method for class 'TPmsm'
plot(x, tr.choice, xlab = "Time", ylab="Transition probability",
col, lty, xlim, ylim, conf.int=FALSE, ci.col, ci.lty,
legend=TRUE, legend.pos, curvlab, legend.bty="n", ...)
```

## Arguments

| | |
|---|---|
| `x` | An object of class 'TPmsm'. |
| `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 bands. Default is FALSE. |
| `ci.col` | Colour of the confidence bands. Default is `col`. |
| `ci.lty` | Line type of the confidence bands. 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. By default no box is drawn. |
| ... | Further arguments for plot. |

## Value

No value is returned.

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## See Also

legend, plot.default.

## Examples

```
# Set the number of threads
nth <- setThreadsTP(2)

# Create survTP object
data(bladderTP)
bladderTP_obj <- with(bladderTP, survTP(time1, event1, Stime, event))

# Compute KMW transition probabilities with confidence band
TPmsm_obj <- transKMW(object=bladderTP_obj, s=5, t=59, conf=TRUE, conf.level=0.95,
method.boot="basic", method.est=2)

# Plot all the transitions without confidence band
plot(TPmsm_obj, conf.int=FALSE, col=seq_len(5), lty=1)

# Plot all the transitions with confidence band
tr.choice <- colnames(TPmsm_obj$est)
par.orig <- par( c("mfrow", "cex") )
par( mfrow=c(2,3) )
for ( i in seq_len( length(tr.choice) ) ) {
plot(TPmsm_obj, tr.choice=tr.choice[i], conf.int=TRUE, legend=FALSE, main=tr.choice[i],
xlab="", ylab="")
}
par(mfrow=c(1, 1), cex=1.2)
title(xlab="Time", ylab="Transition probability", line=3)
par(par.orig)

# Restore the number of threads
setThreadsTP(nth)
```

`setPackageSeedTP` *Set the initial package seed*

## Description

The random number generator (RNG) with multiple independent streams developed by L'Ecuyer et al. (2002) is used for parallel computation of uniform pseudorandom numbers. Package **TPmsm** makes extensive use of uniform pseudorandom numbers, particularly for the bootstrapping statistical techniques and for the generation of univariate and multivariate pseudorandom data. This function defines the seed for the creation of RNG streams.

## Usage

```
setPackageSeedTP(seed=12345)
```

## Arguments

| | |
|---|---|
| `seed` | A vector of one to six integers. Defaults to `rep(x=12345,times=6)`. |

## Details

If the user defines a vector with length lower than six as seed, then the seed is internally defined as a vector of length six with the first elements equal to the user defined values, and the leaving elements equal to `12345`. If a vector with more than six elements is provided as seed, then only the first six elements are used.

## Value

Invisibly returns NULL.

## Note

When package **TPmsm** loads, an initial set of RNG streams is created, one stream for each thread available for parallel computation. The initial set of RNG streams is created from the package seed `c(12345,12345,12345,12345,12345,12345)`. Every time this function is called, the old set of RNG streams is deleted, and a new set of RNG streams is created from the user defined package seed. After the creation of each new RNG stream, the internally stored package seed changes. So each RNG stream is created from a different package seed, and yields different sets of pseudorandom numbers.

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## References

Karl A. T., Eubank R., Milovanovic J., Reiser M., Young D. (2014) Using RngStreams for parallel random number generation in C++ and R. *Computational Statistics* **29(5)**, 1301–1320.

L'Ecuyer, P. (1999) Good parameters and implementations for combined multiple recursive random number generators. *Operations Research* **47(1)**, 159—164.

L'Ecuyer P., Simard R., Chen E. J., Kelton W. D. (2002) An object-oriented random-number package with many long streams and substreams. *Operations Research* **50(6)**, 1073—1075.

## See Also

setSeedTP.

## Examples

```
# Set the number of threads
nth <- setThreadsTP(2)

# Define package seed
seed <- rep(x=1, times=6)

# Set package seed
setPackageSeedTP(seed)

# Create survTP object
data(heartTP)
heartTP_obj <- with(heartTP, survTP(time1, event1, Stime, event))

# Compute transition probabilities with confidence band
TPmsm0 <- transAJ(object=heartTP_obj, s=33, t=412, conf=TRUE,
conf.level=0.9, method.boot="percentile")

# Compute transition probabilities with confidence band
TPmsm1 <- transAJ(object=heartTP_obj, s=33, t=412, conf=TRUE,
conf.level=0.9, method.boot="percentile")

# The objects should be different
all.equal(TPmsm0, TPmsm1)

# Set package seed
setPackageSeedTP(seed)

# Compute transition probabilities with confidence band
TPmsm2 <- transAJ(object=heartTP_obj, s=33, t=412, conf=TRUE,
conf.level=0.9, method.boot="percentile")

# Both objects were computed from the same seed and should be equal
all.equal(TPmsm0, TPmsm2)

# Restore the number of threads
setThreadsTP(nth)
```

setSeedTP *Save and restore RNG stream seeds*

## Description

The random number generator (RNG) with multiple independent streams developed by L'Ecuyer et al. (2002) is used for parallel computation of uniform pseudorandom numbers. Package **TPmsm** makes extensive use of uniform pseudorandom numbers, particularly for the bootstrapping statistical techniques and for the generation of univariate and multivariate pseudorandom data. This function permits saving and restoring the seed of each individual RNG stream.

## Usage

```
setSeedTP(x)
```

## Arguments

| | |
|---|---|
| x | either a NULL object or an object of class 'TPmsmSeed'. |

## Details

An object of class 'TPmsmSeed' can be obtained by a previous call to function `setSeedTP`, usually `setSeedTP(NULL)` or `setSeedTP()`. The object can be saved and used as input on a later call to function `setSeedTP` effectively restoring the seed of each individual RNG stream. An object of class 'TPmsmSeed' can be manipulated or defined with arbitrary seeds, however such procedure is not recommended. It is strongly recommended to input objects of class 'TPmsmSeed' obtained from previous calls to function `setSeedTP`. A seed of choice can be defined by calling function `setPackageSeedTP`.

## Value

Invisibly returns an object of class 'TPmsmSeed'. 'TPmsmSeed' objects are implemented as a list of RNG stream seeds.

## Note

Unlike function `setPackageSeedTP` this function doesn't recreate the RNG streams each time it is called.

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## References

Karl A. T., Eubank R., Milovanovic J., Reiser M., Young D. (2014) Using RngStreams for parallel random number generation in C++ and R. *Computational Statistics* **29(5)**, 1301–1320.

L'Ecuyer, P. (1999) Good parameters and implementations for combined multiple recursive random number generators. *Operations Research* **47(1)**, 159—164.

L'Ecuyer P., Simard R., Chen E. J., Kelton W. D. (2002) An object-oriented random-number package with many long streams and substreams. *Operations Research* **50(6)**, 1073—1075.

## See Also

setPackageSeedTP.

## Examples

```
# Set the number of threads
nth <- setThreadsTP(2)

# Generate bivariate survival data
survTP0 <- dgpTP(n=100, corr=1, dist="weibull", dist.par=c(2, 7, 2, 7),
model.cens="exponential", cens.par = 6, state2.prob=0.6)

# Save seed
seed <- setSeedTP()

# Generate bivariate survival data
survTP1 <- dgpTP(n=100, corr=1, dist="weibull", dist.par=c(2, 7, 2, 7),
model.cens="exponential", cens.par = 6, state2.prob=0.6)

# The objects should be different
all.equal(survTP0, survTP1)

# Restore seed
setSeedTP(seed)

# Generate bivariate survival data
survTP2 <- dgpTP(n=100, corr=1, dist="weibull", dist.par=c(2, 7, 2, 7),
model.cens="exponential", cens.par = 6, state2.prob=0.6)

# Both objects were computed from the same seed and should be equal
all.equal(survTP1, survTP2)

# Restore the number of threads
setThreadsTP(nth)
```

---

# setThreadsTP *Set the number of threads*

---

## Description

Specifies the number of threads used by default in parallel sections.

## Usage

```
setThreadsTP(num_threads=NULL)
```

## Arguments

| | |
|---|---|
| num_threads | the number of threads to use. |

## Details

If num_threads is greater than the number of processors/cores then the number of processors/cores is used. If package **TPmsm** was compiled without OpenMP support then this function returns 1 regardless of the number of processors/cores available. If num_threads=NULL the number of threads is not defined. This is useful when the current number of threads is desired without defining a new thread number.

## Value

Invisibly returns the previous number of threads.

## Note

The given thread number is stored in a global variable. This global variable is then passed to the num_threads clause defined on all parallel sections of underlying C code. By specifying the number of threads in this way instead of specifying with a call to omp_set_num_threads we are certain that there is no interference with the R process. Every time this function is called the RNG streams are recreated. For more details see setPackageSeedTP.

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## References

OpenMP Architecture Review Board, OpenMP Application Program Interface Version 3.0, May 2008, p110 (http://www.openmp.org/mp-documents/spec30.pdf)
"Runtime Library Routines", Summary of OpenMP 3.0 C/C++ Syntax, p5 (http://www.openmp.org/mp-documents/OpenMP3.0-SummarySpec.pdf)

## Examples

```
# Set the number of threads
nth <- setThreadsTP(2)

# Restore the number of threads
setThreadsTP(nth)
```

`survTP` *Create a survTP object*

## Description

Creates a 'survTP' object, usually used as input to other functions.

## Usage

```
survTP(time1, event1, Stime, event, ...)
is.survTP(x)
```

## Arguments

| | |
|---|---|
| `time1` | Time of the transition into state 2, state 3 or censoring time. |
| `event1` | Indicator of transition into state 2 or state 3; 0 if the transition time is censored and 1 otherwise. |
| `Stime` | The total time of the process. |
| `event` | Censoring indicator of the total time of the process; 0 if the total time is censored and 1 otherwise. |
| `...` | Any number of covariates can be specified. |
| `x` | Any R object. |

## Value

An object of class 'survTP'.

'survTP' objects are implemented as a single element list

| | |
|---|---|
| `data` | a data.frame with `time1`, `event1`, `Stime`, `event` and covariates as columns. |

In the case of `is.survTP`, a logical value `TRUE` if `x` inherits from class 'survTP', otherwise `FALSE`.

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## Examples

```
data(bladderTP)
bladderTP_obj <- with(bladderTP, survTP(time1, event1, Stime, event))
#or
bladderTP_obj <- survTP(bladderTP$time1, bladderTP$event1, bladderTP$Stime,
bladderTP$event)
data(heartTP)
heartTP_obj <- with(heartTP, survTP(time1, event1, Stime, event, age=age))
#or
heartTP_obj <- survTP(heartTP$time1, heartTP$event1, heartTP$Stime,
heartTP$event, age=heartTP$age)
```

TPmsmOut *Convert a data.frame in the* **TPmsm** *format to other formats*

## Description

Converts a data.frame in the **TPmsm** format to formats supported by external packages.

## Usage

```
TPmsmOut(data, names, package="p3state.msm")
```

## Arguments

| | |
|---|---|
| data | A data.frame in the **TPmsm** format. |
| names | A character vector of lenght 4, indicating the variable names equivalent to variable names "time1", "event1", "Stime", "event" in the **TPmsm** format, in this order. |
| package | The format of the data.frame. Possible options are "p3state.msm" and "etm". Defaults to "p3state.msm". |

## Value

A data.frame in the format specified by argument package.

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## References

Allignol A., Schumacher M., Beyersmann J. (2011) Empirical Transition Matrix of Multi-State Models: The etm Package. *Journal of Statistical Software* **38(4)**, 1–15.

Meira-Machado L., Roca-Pardiñas J. (2011) p3state.msm: Analyzing Survival Data from an Illness-Death Model. *Journal of Statistical Software* **38(3)**, 1–18.

## See Also

as.data.frame.survTP, survTP.

## Examples

```
data(heartTP)
heartP3 <- TPmsmOut( heartTP, c("time1", "event1", "Stime", "event") )
head(heartP3)
```

transAJ *Aalen-Johansen transition probabilities*

## Description

Provides estimates for the transition probabilities based on the Aalen-Johansen estimator, AJ.

## Usage

```
transAJ(object, s, t, state.names=c("1", "2", "3"), conf=FALSE, n.boot=1000,
conf.level=0.95, method.boot="percentile")
```

## Arguments

| | |
|---|---|
| object | An object of class 'survTP'. |
| s | The first time for obtaining estimates for the transition probabilities. If missing, 0 will be used. |
| t | The second time for obtaining estimates for the transition probabilities. If missing, the maximum of Stime will be used. |
| state.names | A vector of characters giving the state names. |
| conf | Provides pointwise confidence bands. Defaults to FALSE. |
| n.boot | The number of bootstrap samples. Defaults to 1000 samples. |
| conf.level | Level of confidence. Defaults to 0.95 (corresponding to 95%). |
| method.boot | The method used to compute bootstrap confidence bands. Possible options are "percentile" and "basic". Defaults to "percentile". |

## Value

An object of class 'TPmsm'. There are methods for contour, image, print and plot. 'TPmsm' objects are implemented as a list with elements:

| | |
|---|---|
| method | A string indicating the type of estimator used in the computation. |
| est | A matrix with transition probability estimates. The rows being the event times and the columns the 5 possible transitions. |
| inf | A matrix with the lower transition probabilities of the confidence band. The rows being the event times and the columns the 5 possible transitions. |
| sup | A matrix with the upper transition probabilities of the confidence band. The rows being the event times and the columns the 5 possible transitions. |
| time | Vector of times where the transition probabilities are computed. |
| s | Start of the time interval. |
| t | End of the time interval. |
| h | The bandwidth used. If the estimator doesn't require a bandwidth, it's set to NULL. |

| | |
|---|---|
| state.names | A vector of characters giving the states names. |
| n.boot | Number of bootstrap samples used in the computation of the confidence band. |
| conf.level | Level of confidence used to compute the confidence band. |

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## References

Aalen O. O., Johansen S. (1978) An Empirical Transition Matrix for Nonhomogeneous Markov Chains Based on Censored Observations. *Scandinavian Journal of Statistics* **5(3)**, 141–150.

Davison A. C., Hinkley D. V. (1997) *Bootstrap Methods and their Application* Chapter 5, Cambridge University Press.

## See Also

transIPCW, transKMPW, transKMW, transLIN, transLS, transPAJ.

## Examples

```
# Set the number of threads
nth <- setThreadsTP(2)

# Create survTP object
data(heartTP)
heartTP_obj <- with(heartTP, survTP(time1, event1, Stime, event))

# Compute transition probabilities
transAJ(object=heartTP_obj, s=33, t=412)

# Compute transition probabilities with confidence band
transAJ(object=heartTP_obj, s=33, t=412, conf=TRUE, conf.level=0.9,
method.boot="percentile")

# Restore the number of threads
setThreadsTP(nth)
```

---

# transIPCW *Inverse probability of censoring weighted transition probabilities*

---

## Description

Provides estimates for the transition probabilities based on inverse probability of censoring weighted estimators, IPCW.

## Usage

```
transIPCW(object, s, t, x, bw="dpik", window="normal", method.weights="NW",
state.names=c("1", "2", "3"), conf=FALSE, n.boot=1000, conf.level=0.95,
method.boot="percentile", method.est=1, ...)
```

## Arguments

| | |
|---|---|
| object | An object of class 'survTP'. |
| s | The first time for obtaining estimates for the transition probabilities. If missing, 0 will be used. |
| t | The second time for obtaining estimates for the transition probabilities. If missing, the maximum of Stime will be used. |
| x | Covariate values for obtaining estimates for the conditional transition probabilities. If missing, unconditioned transition probabilities will be computed. |
| bw | A character string indicating a function to compute a kernel density bandwidth. Defaults to "dpik" from package **KernSmooth**. Alternatively a single numeric value can be specified. |
| window | A character string specifying the desired kernel. See details below for possible options. Defaults to "normal" where the gaussian density kernel will be used. |
| method.weights | A character string specifying the desired weights method. Possible options are "NW" for the Nadaraya-Watson weights and "LL" for local linear weights. Defaults to "NW". |
| state.names | A vector of characters giving the state names. |
| conf | Provides pointwise confidence bands. Defaults to FALSE. |
| n.boot | The number of bootstrap samples. Defaults to 1000 samples. |
| conf.level | Level of confidence. Defaults to 0.95 (corresponding to 95%). |
| method.boot | The method used to compute bootstrap confidence bands. Possible options are "percentile" and "basic". Defaults to "percentile". |
| method.est | The method used to compute the estimate. Possible options are 1 or 2. |
| ... | Further arguments. Typically these arguments are passed to the function specified by argument bw. |

## Details

If bw="dpik" then possible options for argument window are "normal", "box", "epanech", "biweight" or "triweight". When argument bw is numeric then argument window accepts the same options as when bw="dpik" plus one of "tricube", "triangular" or "cosine".

If method.est=1 then $p_{11}(s,t|X)$, $p_{12}(s,t|X)$ and $p_{22}(s,t|X)$ are estimated according to the following expressions:

$$p_{11}(s,t|X) = \frac{1-P(Z\leq t|X)}{1-P(Z\leq s|X)},$$

$$p_{12}(s,t|X) = \frac{P(Z\leq t|X)-P(Z\leq s|X)-P(s<Z\leq t, T\leq t|X)}{1-P(Z\leq s|X)},$$

$p_{22}(s,t|X) = \frac{P(Z \leq s|X) - P(Z \leq s, T \leq t|X)}{P(Z \leq s|X) - P(T \leq s|X)}$.

Then, $p_{13}(s,t|X) = 1 - p_{11}(s,t|X) - p_{12}(s,t|X)$ and $p_{23}(s,t|X) = 1 - p_{22}(s,t|X)$.

If `method.est=2` then $p_{11}(s,t|X)$, $p_{12}(s,t|X)$ and $p_{22}(s,t|X)$ are estimated according to the following expressions:

$p_{11}(s,t|X) = \frac{P(Z>t|X)}{P(Z>s|X)}$,

$p_{12}(s,t|X) = \frac{P(s<Z\leq t, T>t|X)}{P(Z>s|X)}$,

$p_{22}(s,t|X) = \frac{P(Z\leq s, T>t|X)}{P(Z\leq s, T>s|X)}$.

Then, $p_{13}(s,t|X) = 1 - p_{11}(s,t|X) - p_{12}(s,t|X)$ and $p_{23}(s,t|X) = 1 - p_{22}(s,t|X)$.

## Value

If argument `x` is missing or if argument `object` doesn't contain a covariate, an object of class 'TPmsm' is returned. There are methods for `contour`, `image`, `print` and `plot`. 'TPmsm' objects are implemented as a list with elements:

| | |
|---|---|
| `method` | A string indicating the type of estimator used in the computation. |
| `est` | A matrix with transition probability estimates. The rows being the event times and the columns the 5 possible transitions. |
| `inf` | A matrix with the lower transition probabilities of the confidence band. The rows being the event times and the columns the 5 possible transitions. |
| `sup` | A matrix with the upper transition probabilities of the confidence band. The rows being the event times and the columns the 5 possible transitions. |
| `time` | Vector of times where the transition probabilities are computed. |
| `s` | Start of the time interval. |
| `t` | End of the time interval. |
| `h` | The bandwidth used. If the estimator doesn't require a bandwidth, it's set to `NULL`. |
| `state.names` | A vector of characters giving the states names. |
| `n.boot` | Number of bootstrap samples used in the computation of the confidence band. |
| `conf.level` | Level of confidence used to compute the confidence band. |

If argument `x` is specified and argument `object` contains a covariate, an object of class 'TPCmsm' is returned. There are methods for `print` and `plot`. 'TPCmsm' objects are implemented as a list with elements:

| | |
|---|---|
| `method` | A string indicating the type of estimator used in the computation. |

| | |
|---|---|
| est | A 3 dimensional array with transition probability estimates. The first dimension being the event times, the second the covariate values and the last one the 5 possible transitions. |
| inf | A 3 dimensional array with the lower transition probabilities of the confidence band. The first dimension being the event times, the second the covariate values and the last one the 5 possible transitions. |
| sup | A 3 dimensional array with the upper transition probabilities of the confidence band. The first dimension being the event times, the second the covariate values and the last one the 5 possible transitions. |
| time | Vector of times where the transition probabilities are computed. |
| covariate | Vector of covariate values where the conditional transition probabilities are computed. |
| s | Start of the time interval. |
| t | End of the time interval. |
| x | Additional covariate values where the conditional transition probabilities are computed, which may or may not be present in the sample. |
| h | The bandwidth used. |
| state.names | A vector of characters giving the states names. |
| n.boot | Number of bootstrap samples used in the computation of the confidence band. |
| conf.level | Level of confidence used to compute the confidence band. |

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## References

Meira-Machado L., de Uña-Álvarez J., Datta S. Conditional Transition Probabilities in a non-Markov Illness-death Model. Discussion Papers in Statistics and Operation Research n 11/03, 2011. Department of Statistics and Operations Research, University of Vigo (ISSN: 1888-5756, Deposito Legal VG 1402 - 2007). This file can be downloaded from: http://webs.uvigo.es/depc05/reports/12_05.pdf

Meira Machado L. F., de Uña-Álvarez J., Cadarso-Suárez C. (2006) Nonparametric estimation of transition probabilities in a non-Markov illness-death model. *Lifetime Data Anal* **12(3)**, 325–344.

Davison, A. C., Hinkley, D. V. (1997) *Bootstrap Methods and their Application* Chapter 5, Cambridge University Press.

## See Also

transAJ, transKMPW, transKMW, transLIN, transLS, transPAJ.

## Examples

```
# Set the number of threads
nth <- setThreadsTP(2)

# Create survTP object with age as covariate
data(heartTP)
heartTP_obj <- with(heartTP, survTP(time1, event1, Stime, event, age=age))

# Compute unconditioned transition probabilities
transIPCW(object=heartTP_obj, s=33, t=412)

# Compute unconditioned transition probabilities with confidence band
transIPCW(object=heartTP_obj, s=33, t=412, conf=TRUE, conf.level=0.9,
method.boot="basic", method.est=2)

# Compute conditional transition probabilities
transIPCW(object=heartTP_obj, s=33, t=412, x=0)

# Compute conditional transition probabilities with confidence band
transIPCW(object=heartTP_obj, s=33, t=412, x=0, conf=TRUE, conf.level=0.95,
n.boot=100, method.boot="percentile", method.est=2)

# Restore the number of threads
setThreadsTP(nth)
```

---

# transKMPW *Presmoothed Kaplan-Meier weighted transition probabilities*

---

## Description

Provides estimates for the transition probabilities based on presmoothed Kaplan-Meier weighted estimators, KMPW.

## Usage

```
transKMPW(object, s, t, state.names=c("1", "2", "3"), conf=FALSE, n.boot=1000,
conf.level=0.95, method.boot="percentile", method.est=3)
```

## Arguments

| | |
|---|---|
| object | An object of class 'survTP'. |
| s | The first time for obtaining estimates for the transition probabilities. If missing, 0 will be used. |
| t | The second time for obtaining estimates for the transition probabilities. If missing, the maximum of Stime will be used. |
| state.names | A vector of characters giving the state names. |
| conf | Provides pointwise confidence bands. Defaults to FALSE. |

| | |
|---|---|
| `n.boot` | The number of bootstrap samples. Defaults to 1000 samples. |
| `conf.level` | Level of confidence. Defaults to 0.95 (corresponding to 95%). |
| `method.boot` | The method used to compute bootstrap confidence bands. Possible options are "percentile" and "basic". Defaults to "percentile". |
| `method.est` | The method used to compute the estimate. Possible options are 1, 2, 3 or 4. |

## Details

If `method.est=1` then $p_{11}(s,t)$, $p_{12}(s,t)$ and $p_{22}(s,t)$ are estimated according to the following expressions:

$p_{11}(s,t) = \frac{1-P(Z\leq t)}{1-P(Z\leq s)}$,

$p_{12}(s,t) = \frac{P(Z\leq t)-P(Z\leq s)-P(s<Z\leq t,T\leq t)}{1-P(Z\leq s)}$,

$p_{22}(s,t) = \frac{P(Z\leq s)-P(Z\leq s,T\leq t)}{P(Z\leq s)-P(T\leq s)}$.

Then, $p_{13}(s,t) = 1 - p_{11}(s,t) - p_{12}(s,t)$ and $p_{23}(s,t) = 1 - p_{22}(s,t)$.

If `method.est=2` then $p_{11}(s,t)$, $p_{12}(s,t)$ and $p_{22}(s,t)$ are estimated according to the following expressions:

$p_{11}(s,t) = \frac{P(Z>t)}{P(Z>s)}$,

$p_{12}(s,t) = \frac{P(s<Z\leq t,T>t)}{P(Z>s)}$,

$p_{22}(s,t) = \frac{P(Z\leq s,T>t)}{P(Z\leq s,T>s)}$.

Then, $p_{13}(s,t) = 1 - p_{11}(s,t) - p_{12}(s,t)$ and $p_{23}(s,t) = 1 - p_{22}(s,t)$.

If `method.est=3` then $p_{11}(s,t)$, $p_{13}(s,t)$ and $p_{23}(s,t)$ are estimated according to the following expressions:

$p_{11}(s,t) = \frac{1-P(Z\leq t)}{1-P(Z\leq s)}$,

$p_{13}(s,t) = \frac{P(Z>s,T\leq t)}{1-P(Z\leq s)}$,

$p_{23}(s,t) = \frac{P(Z\leq s,s<T\leq t)}{P(Z\leq s)-P(T\leq s)}$.

Then, $p_{12}(s,t) = 1 - p_{11}(s,t) - p_{13}(s,t)$ and $p_{22}(s,t) = 1 - p_{23}(s,t)$.

If `method.est=4` then $p_{11}(s,t)$, $p_{13}(s,t)$ and $p_{23}(s,t)$ are estimated according to the following expressions:

$p_{11}(s,t) = \frac{P(Z>t)}{P(Z>s)}$,

$p_{13}(s,t) = \frac{P(Z>s,T\leq t)}{P(Z>s)}$,

$p_{23}(s,t) = \frac{P(Z \leq s, s < T \leq t)}{P(Z \leq s, T > s)}$.

Then, $p_{12}(s,t) = 1 - p_{11}(s,t) - p_{13}(s,t)$ and $p_{22}(s,t) = 1 - p_{23}(s,t)$.

## Value

An object of class 'TPmsm'. There are methods for `contour`, `image`, `print` and `plot`. 'TPmsm' objects are implemented as a list with elements:

| | |
|---|---|
| `method` | A string indicating the type of estimator used in the computation. |
| `est` | A matrix with transition probability estimates. The rows being the event times and the columns the 5 possible transitions. |
| `inf` | A matrix with the lower transition probabilities of the confidence band. The rows being the event times and the columns the 5 possible transitions. |
| `sup` | A matrix with the upper transition probabilities of the confidence band. The rows being the event times and the columns the 5 possible transitions. |
| `time` | Vector of times where the transition probabilities are computed. |
| `s` | Start of the time interval. |
| `t` | End of the time interval. |
| `h` | The bandwidth used. If the estimator doesn't require a bandwidth, it's set to `NULL`. |
| `state.names` | A vector of characters giving the states names. |
| `n.boot` | Number of bootstrap samples used in the computation of the confidence band. |
| `conf.level` | Level of confidence used to compute the confidence band. |

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## References

Amorim A. P., de Uña-Álvarez J., Meira Machado L. F. (2011) Presmoothing the transition probabilities in the illness-death model. *Statistics and Probability Letters* **81**, 797–806.

Davison, A. C., Hinkley, D. V. (1997) *Bootstrap Methods and their Application* Chapter 5, Cambridge University Press.

## See Also

`transAJ`, `transIPCW`, `transKMW`, `transLIN`, `transLS`, `transPAJ`.

## Examples

```
# Set the number of threads
nth <- setThreadsTP(2)

# Create survTP object
data(heartTP)
```

```
heartTP_obj <- with(heartTP, survTP(time1, event1, Stime, event))

# Compute transition probabilities
transKMPW(object=heartTP_obj, s=33, t=412)

# Compute transition probabilities with confidence band
transKMPW(object=heartTP_obj, s=33, t=412, conf=TRUE, conf.level=0.9,
method.boot="percentile", method.est=4)

# Restore the number of threads
setThreadsTP(nth)
```

---

`transKMW` *Kaplan-Meier weighted transition probabilities*

---

## Description

Provides estimates for the transition probabilities based on Kaplan-Meier weighted estimators, KMW.

## Usage

```
transKMW(object, s, t, state.names=c("1", "2", "3"), conf=FALSE, n.boot=1000,
conf.level=0.95, method.boot="percentile", method.est=3)
```

## Arguments

| | |
|---|---|
| `object` | An object of class 'survTP'. |
| `s` | The first time for obtaining estimates for the transition probabilities. If missing, 0 will be used. |
| `t` | The second time for obtaining estimates for the transition probabilities. If missing, the maximum of `Stime` will be used. |
| `state.names` | A vector of characters giving the state names. |
| `conf` | Provides pointwise confidence bands. Defaults to `FALSE`. |
| `n.boot` | The number of bootstrap samples. Defaults to 1000 samples. |
| `conf.level` | Level of confidence. Defaults to 0.95 (corresponding to 95%). |
| `method.boot` | The method used to compute bootstrap confidence bands. Possible options are "percentile" and "basic". Defaults to "percentile". |
| `method.est` | The method used to compute the estimate. Possible options are 1, 2, 3 or 4. |

## Details

If `method.est=1` then $p_{11}(s,t)$, $p_{12}(s,t)$ and $p_{22}(s,t)$ are estimated according to the following expressions:

$p_{11}(s,t) = \frac{1-P(Z\leq t)}{1-P(Z\leq s)}$,

$p_{12}(s,t) = \frac{P(Z\leq t)-P(Z\leq s)-P(s<Z\leq t,T\leq t)}{1-P(Z\leq s)}$,

$p_{22}(s,t) = \frac{P(Z\leq s)-P(Z\leq s,T\leq t)}{P(Z\leq s)-P(T\leq s)}$.

Then, $p_{13}(s,t) = 1 - p_{11}(s,t) - p_{12}(s,t)$ and $p_{23}(s,t) = 1 - p_{22}(s,t)$.

If `method.est=2` then $p_{11}(s,t)$, $p_{12}(s,t)$ and $p_{22}(s,t)$ are estimated according to the following expressions:

$p_{11}(s,t) = \frac{P(Z>t)}{P(Z>s)}$,

$p_{12}(s,t) = \frac{P(s<Z\leq t,T>t)}{P(Z>s)}$,

$p_{22}(s,t) = \frac{P(Z\leq s,T>t)}{P(Z\leq s,T>s)}$.

Then, $p_{13}(s,t) = 1 - p_{11}(s,t) - p_{12}(s,t)$ and $p_{23}(s,t) = 1 - p_{22}(s,t)$.

If `method.est=3` then $p_{11}(s,t)$, $p_{13}(s,t)$ and $p_{23}(s,t)$ are estimated according to the following expressions:

$p_{11}(s,t) = \frac{1-P(Z\leq t)}{1-P(Z\leq s)}$,

$p_{13}(s,t) = \frac{P(Z>s,T\leq t)}{1-P(Z\leq s)}$,

$p_{23}(s,t) = \frac{P(Z\leq s,s<T\leq t)}{P(Z\leq s)-P(T\leq s)}$.

Then, $p_{12}(s,t) = 1 - p_{11}(s,t) - p_{13}(s,t)$ and $p_{22}(s,t) = 1 - p_{23}(s,t)$.

If `method.est=4` then $p_{11}(s,t)$, $p_{13}(s,t)$ and $p_{23}(s,t)$ are estimated according to the following expressions:

$p_{11}(s,t) = \frac{P(Z>t)}{P(Z>s)}$,

$p_{13}(s,t) = \frac{P(Z>s,T\leq t)}{P(Z>s)}$,

$p_{23}(s,t) = \frac{P(Z\leq s,s<T\leq t)}{P(Z\leq s,T>s)}$.

Then, $p_{12}(s,t) = 1 - p_{11}(s,t) - p_{13}(s,t)$ and $p_{22}(s,t) = 1 - p_{23}(s,t)$.

## Value

An object of class 'TPmsm'. There are methods for `contour`, `image`, `print` and `plot`. 'TPmsm' objects are implemented as a list with elements:

| | |
|---|---|
| `method` | A string indicating the type of estimator used in the computation. |
| `est` | A matrix with transition probability estimates. The rows being the event times and the columns the 5 possible transitions. |
| `inf` | A matrix with the lower transition probabilities of the confidence band. The rows being the event times and the columns the 5 possible transitions. |
| `sup` | A matrix with the upper transition probabilities of the confidence band. The rows being the event times and the columns the 5 possible transitions. |
| `time` | Vector of times where the transition probabilities are computed. |
| `s` | Start of the time interval. |
| `t` | End of the time interval. |
| `h` | The bandwidth used. If the estimator doesn't require a bandwidth, it's set to `NULL`. |
| `state.names` | A vector of characters giving the states names. |
| `n.boot` | Number of bootstrap samples used in the computation of the confidence band. |
| `conf.level` | Level of confidence used to compute the confidence band. |

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## References

Meira Machado L. F., de Uña-Álvarez J., Cadarso-Suárez C. (2006) Nonparametric estimation of transition probabilities in a non-Markov illness-death model. *Lifetime Data Anal* **12(3)**, 325–344.

Davison, A. C., Hinkley, D. V. (1997) *Bootstrap Methods and their Application* Chapter 5, Cambridge University Press.

## See Also

`transAJ`, `transIPCW`, `transKMPW`, `transLIN`, `transLS`, `transPAJ`.

## Examples

```
# Set the number of threads
nth <- setThreadsTP(2)

# Create survTP object
data(heartTP)
heartTP_obj <- with(heartTP, survTP(time1, event1, Stime, event))

# Compute transition probabilities
transKMW(object=heartTP_obj, s=33, t=412)
```

```
# Compute transition probabilities with confidence band
transKMW(object=heartTP_obj, s=33, t=412, conf=TRUE, conf.level=0.9,
method.boot="basic", method.est=2)

# Restore the number of threads
setThreadsTP(nth)
```

---

`transLIN` *LIN based transition probabilities*

---

## Description

Provides estimates for the transition probabilities based on LIN estimators, LIN.

## Usage

```
transLIN(object, s, t, x, bw="dpik", window="normal", method.weights="NW",
state.names=c("1", "2", "3"), conf=FALSE, n.boot=1000, conf.level=0.95,
method.boot="percentile", ...)
```

## Arguments

| | |
|---|---|
| `object` | An object of class 'survTP'. |
| `s` | The first time for obtaining estimates for the transition probabilities. If missing, 0 will be used. |
| `t` | The second time for obtaining estimates for the transition probabilities. If missing, the maximum of `Stime` will be used. |
| `x` | Covariate values for obtaining estimates for the conditional transition probabilities. If missing, unconditioned transition probabilities will be computed. |
| `bw` | A character string indicating a function to compute a kernel density bandwidth. Defaults to "dpik" from package **KernSmooth**. Alternatively a single numeric value can be specified. |
| `window` | A character string specifying the desired kernel. See details below for possible options. Defaults to "normal" where the gaussian density kernel will be used. |
| `method.weights` | A character string specifying the desired weights method. Possible options are "NW" for the Nadaraya-Watson weights and "LL" for local linear weights. Defaults to "NW". |
| `state.names` | A vector of characters giving the state names. |
| `conf` | Provides pointwise confidence bands. Defaults to `FALSE`. |
| `n.boot` | The number of bootstrap samples. Defaults to 1000 samples. |
| `conf.level` | Level of confidence. Defaults to 0.95 (corresponding to 95%). |
| `method.boot` | The method used to compute bootstrap confidence bands. Possible options are "percentile" and "basic". Defaults to "percentile". |
| `...` | Further arguments. Typically these arguments are passed to the function specified by argument `bw`. |

## Details

If bw="dpik" then possible options for argument window are "normal", "box", "epanech", "biweight" or "triweight". When argument bw is numeric then argument window accepts the same options as when bw="dpik" plus one of "tricube", "triangular" or "cosine".

## Value

If argument x is missing or if argument object doesn't contain a covariate, an object of class 'TPmsm' is returned. There are methods for contour, image, print and plot. 'TPmsm' objects are implemented as a list with elements:

| | |
|---|---|
| method | A string indicating the type of estimator used in the computation. |
| est | A matrix with transition probability estimates. The rows being the event times and the columns the 5 possible transitions. |
| inf | A matrix with the lower transition probabilities of the confidence band. The rows being the event times and the columns the 5 possible transitions. |
| sup | A matrix with the upper transition probabilities of the confidence band. The rows being the event times and the columns the 5 possible transitions. |
| time | Vector of times where the transition probabilities are computed. |
| s | Start of the time interval. |
| t | End of the time interval. |
| h | The bandwidth used. If the estimator doesn't require a bandwidth, it's set to NULL. |
| state.names | A vector of characters giving the states names. |
| n.boot | Number of bootstrap samples used in the computation of the confidence band. |
| conf.level | Level of confidence used to compute the confidence band. |

If argument x is specified and argument object contains a covariate, an object of class 'TPCmsm' is returned. There are methods for print and plot. 'TPCmsm' objects are implemented as a list with elements:

| | |
|---|---|
| method | A string indicating the type of estimator used in the computation. |
| est | A 3 dimensional array with transition probability estimates. The first dimension being the event times, the second the covariate values and the last one the 5 possible transitions. |
| inf | A 3 dimensional array with the lower transition probabilities of the confidence band. The first dimension being the event times, the second the covariate values and the last one the 5 possible transitions. |
| sup | A 3 dimensional array with the upper transition probabilities of the confidence band. The first dimension being the event times, the second the covariate values and the last one the 5 possible transitions. |
| time | Vector of times where the transition probabilities are computed. |
| covariate | Vector of covariate values where the conditional transition probabilities are computed. |

| | |
|---|---|
| s | Start of the time interval. |
| t | End of the time interval. |
| x | Additional covariate values where the conditional transition probabilities are computed, which may or may not be present in the sample. |
| h | The bandwidth used. |
| state.names | A vector of characters giving the states names. |
| n.boot | Number of bootstrap samples used in the computation of the confidence band. |
| conf.level | Level of confidence used to compute the confidence band. |

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## References

Meira-Machado L., de Uña-Álvarez J., Datta S. Conditional Transition Probabilities in a non-Markov Illness-death Model. Discussion Papers in Statistics and Operation Research n 11/03, 2011. Department of Statistics and Operations Research, University of Vigo (ISSN: 1888-5756, Deposito Legal VG 1402 - 2007). This file can be downloaded from: http://webs.uvigo.es/depc05/reports/12_05.pdf

Meira Machado L. F., de Uña-Álvarez J., Cadarso-Suárez C. (2006) Nonparametric estimation of transition probabilities in a non-Markov illness-death model. *Lifetime Data Anal* **12(3)**, 325–344.

Davison, A. C., Hinkley, D. V. (1997) *Bootstrap Methods and their Application* Chapter 5, Cambridge University Press.

## See Also

transAJ, transIPCW, transKMPW, transKMW, transLS, transPAJ.

## Examples

```
# Set the number of threads
nth <- setThreadsTP(2)

# Create survTP object with age as covariate
data(heartTP)
heartTP_obj <- with(heartTP, survTP(time1, event1, Stime, event, age=age))

# Compute unconditioned transition probabilities
transLIN(object=heartTP_obj, s=33, t=412)

# Compute unconditioned transition probabilities with confidence band
transLIN(object=heartTP_obj, s=33, t=412, conf=TRUE, conf.level=0.9,
method.boot="basic")

# Compute conditional transition probabilities
transLIN(object=heartTP_obj, s=33, t=412, x=0)
```

```
# Compute conditional transition probabilities with confidence band
transLIN(object=heartTP_obj, s=33, t=412, x=0, conf=TRUE, conf.level=0.95,
n.boot=100, method.boot="percentile")

# Restore the number of threads
setThreadsTP(nth)
```

---

`transLS` *Location-Scale transition probabilities*

---

## Description

Provides estimates for the transition probabilities based on the Location-Scale estimator, LS.

## Usage

```
transLS(object, s, t, h, nh=40, ncv=10, window="normal", state.names=c("1", "2", "3"),
conf=FALSE, n.boot=1000, conf.level=0.95, method.boot="percentile", boot.cv=FALSE,
cv.full=TRUE)
```

## Arguments

| | |
|---|---|
| `object` | An object of class 'survTP'. |
| `s` | The first time for obtaining estimates for the transition probabilities. If missing, 0 will be used. |
| `t` | The second time for obtaining estimates for the transition probabilities. If missing, the maximum of `Stime` will be used. |
| `h` | A vector with 1 up to 4 values, indicating the minimum and maximum bandwidths to test by cross-validation. |
| `nh` | The number of bandwidth values to test by cross-validation. Defaults to 40. |
| `ncv` | The number of cross-validation samples. Defaults to 10. |
| `window` | A character string specifying the desired kernel. Possible options are "normal", "epanech", "biweight", "triweight", "box", "tricube", "triangular" or "cosine". Defaults to "normal" where the gaussian density kernel will be used. |
| `state.names` | A vector of characters giving the state names. |
| `conf` | Provides pointwise confidence bands. Defaults to `FALSE`. |
| `n.boot` | The number of bootstrap samples. Defaults to 1000 samples. |
| `conf.level` | Level of confidence. Defaults to 0.95 (corresponding to 95%). |
| `method.boot` | The method used to compute bootstrap confidence bands. Possible options are "percentile" and "basic". Defaults to "percentile". |
| `boot.cv` | If `TRUE` the bandwidth is computed by cross-validation for each bootstrap sample. If `FALSE` the bandwidth used to compute the estimates is used to compute each bootstrap estimate. Defaults to `FALSE`. |
| `cv.full` | If `TRUE` the bandwidth is computed by cross-validation for both the location and scale functions. If `FALSE` the bandwidth is computed by cross-validation only for the location function. And the bandwidth for the scale function is taken to be equal to the location one. Defaults to `TRUE`. |

## Value

An object of class 'TPmsm'. There are methods for `contour`, `image`, `print` and `plot`. 'TPmsm' objects are implemented as a list with elements:

| | |
|---|---|
| `method` | A string indicating the type of estimator used in the computation. |
| `est` | A matrix with transition probability estimates. The rows being the event times and the columns the 5 possible transitions. |
| `inf` | A matrix with the lower transition probabilities of the confidence band. The rows being the event times and the columns the 5 possible transitions. |
| `sup` | A matrix with the upper transition probabilities of the confidence band. The rows being the event times and the columns the 5 possible transitions. |
| `time` | Vector of times where the transition probabilities are computed. |
| `s` | Start of the time interval. |
| `t` | End of the time interval. |
| `h` | The bandwidth used. If the estimator doesn't require a bandwidth, it's set to `NULL`. |
| `state.names` | A vector of characters giving the states names. |
| `n.boot` | Number of bootstrap samples used in the computation of the confidence band. |
| `conf.level` | Level of confidence used to compute the confidence band. |

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## References

Meira-Machado L., Roca-Pardiñas J., Van Keilegom I., Cadarso-Suárez C. (2013) Bandwidth Selection for the Estimation of Transition Probabilities in the Location-Scale Progressive Three-State Model. *Computational Statistics* **28(5)**, 2185–2210.

Meira-Machado L., Roca-Pardiñas J., Van Keilegom I., Cadarso-Suárez C. Estimation of transition probabilities in a non-Markov model with successive survival times. Discussion paper 2010. This file can be downloaded from: http://sites.uclouvain.be/IAP-Stat-Phase-V-VI/ISBApub/dp2010/DP1053.pdf

Van Keilegom I., de Uña-Álvarez J., Meira-Machado L. (2011) Nonparametric location-scale models for successive survival times under dependent censoring. *Journal of Statistical Planning and Inference* **141(3)**, 1118–1131.

Davison, A. C., Hinkley, D. V. (1997) *Bootstrap Methods and their Application* Chapter 5, Cambridge University Press.

## See Also

`transAJ`, `transIPCW`, `transKMPW`, `transKMW`, `transLIN`, `transPAJ`.

## Examples

```
# Set the number of threads
nth <- setThreadsTP(2)

# Create survTP object
data(bladderTP)
bladderTP_obj <- with(bladderTP, survTP(time1, event1, Stime, event))

# Compute transition probabilities
LS0 <- transLS(object=bladderTP_obj, s=5, t=59, h=c(0.25, 2.5), nh=25, ncv=50, conf=FALSE)
print(LS0)

# Compute transition probabilities with confidence band
h <- with( LS0, c( rep(h[1], 2), rep(h[2], 2) ) )
transLS(object=bladderTP_obj, s=5, t=59, h=h, conf=TRUE,
conf.level=0.95, method.boot="percentile", boot.cv=FALSE)

# Restore the number of threads
setThreadsTP(nth)
```

---

# transPAJ *Presmoothed Aalen-Johansen transition probabilities*

---

## Description

Provides estimates for the transition probabilities based on the presmoothed Aalen-Johansen estimator, PAJ.

## Usage

```
transPAJ(object, s, t, state.names=c("1", "2", "3"), conf=FALSE, n.boot=1000,
conf.level=0.95, method.boot="percentile")
```

## Arguments

| | |
|---|---|
| object | An object of class 'survTP'. |
| s | The first time for obtaining estimates for the transition probabilities. If missing, 0 will be used. |
| t | The second time for obtaining estimates for the transition probabilities. If missing, the maximum of Stime will be used. |
| state.names | A vector of characters giving the state names. |
| conf | Provides pointwise confidence bands. Defaults to FALSE. |
| n.boot | The number of bootstrap samples. Defaults to 1000 samples. |
| conf.level | Level of confidence. Defaults to 0.95 (corresponding to 95%). |
| method.boot | The method used to compute bootstrap confidence bands. Possible options are "percentile" and "basic". Defaults to "percentile". |

## Value

An object of class 'TPmsm'. There are methods for `contour`, `image`, `print` and `plot`. 'TPmsm' objects are implemented as a list with elements:

| | |
|---|---|
| `method` | A string indicating the type of estimator used in the computation. |
| `est` | A matrix with transition probability estimates. The rows being the event times and the columns the 5 possible transitions. |
| `inf` | A matrix with the lower transition probabilities of the confidence band. The rows being the event times and the columns the 5 possible transitions. |
| `sup` | A matrix with the upper transition probabilities of the confidence band. The rows being the event times and the columns the 5 possible transitions. |
| `time` | Vector of times where the transition probabilities are computed. |
| `s` | Start of the time interval. |
| `t` | End of the time interval. |
| `h` | The bandwidth used. If the estimator doesn't require a bandwidth, it's set to `NULL`. |
| `state.names` | A vector of characters giving the states names. |
| `n.boot` | Number of bootstrap samples used in the computation of the confidence band. |
| `conf.level` | Level of confidence used to compute the confidence band. |

## Author(s)

Artur Araújo, Javier Roca-Pardiñas and Luís Meira-Machado

## References

Moreira A., de Uña-Álvarez J. and Meira-Machado L. Presmoothing the Aalen-Johansen estimator of transition probabilities. Discussion Papers in Statistics and Operation Research n 11/03, 2011. Department of Statistics and Operations Research, University of Vigo (ISSN: 1888-5756, Deposito Legal VG 1402 - 2007). This file can be downloaded from: http://webs.uvigo.es/depc05/reports/11_03.pdf

Davison A. C., Hinkley D. V. (1997) *Bootstrap Methods and their Application* Chapter 5, Cambridge University Press.

## See Also

`transAJ`, `transIPCW`, `transKMPW`, `transKMW`, `transLIN`, `transLS`.

## Examples

```
# Set the number of threads
nth <- setThreadsTP(2)

# Create survTP object
data(heartTP)
heartTP_obj <- with(heartTP, survTP(time1, event1, Stime, event))
```

```
# Compute transition probabilities
transPAJ(object=heartTP_obj, s=33, t=412)

# Compute transition probabilities with confidence band
transPAJ(object=heartTP_obj, s=33, t=412, conf=TRUE, conf.level=0.9,
method.boot="percentile")

# Restore the number of threads
setThreadsTP(nth)
```

# Index