Package ‘BGPhazard’

January 20, 2025

Title Markov Beta and Gamma Processes for Modeling Hazard Rates

Version 2.1.1

Author L. E. Nieto-Barajas, J. A. Garcia Bueno, E.A. Morones Ishikawa and J. Pliego

Maintainer Emilio Akira Morones Ishikawa <`emiliomorones@gmail.com`>

Description Computes the hazard rate estimate as described by Nieto-Barajas & Walker (2002), Nieto-Barajas (2003), Nieto-Barajas & Walker (2007) and Nieto-Barajas & Yin (2008).

License GPL (>= 2)

URL `https://github.com/EAMI91/BGPhazard`

BugReports `https://github.com/EAMI91/BGPhazard/issues`

Depends R (>= 3.5.0)

Imports Brobdingnag, dplyr, ggplot2, ggthemes, gridExtra, magrittr, progress, purrr, readr, rlang, stats, stringr, survival, tibble, tidyr, utils

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

Date/Publication 2023-09-03 14:20:05 UTC

Encoding UTF-8

LazyData true

NeedsCompilation no

Repository CRAN

RoxygenNote 7.2.3

Contents

`BeMRes` *Markov Beta Model*

Description

Posterior inference for the Bayesian non-parametric Markov beta model for discrete survival times.

Usage

```
BeMRes(
  times,
  delta = rep(1, length(times)),
  alpha = rep(1e-04, K),
  beta = rep(1e-04, K),
  c.r = rep(0, K - 1),
  a.eps = 0.1,
  b.eps = 0.1,
  type.c = 4,
  epsilon = 1,
  iterations = 2000,
```

```
  burn.in = floor(iterations * 0.2),
  thinning = 5,
  printtime = TRUE
)
```

Arguments

`times`	Numeric positive vector. Failure times.
`delta`	Logical vector. Status indicator. `TRUE` (1) indicates exact lifetime is known, `FALSE` (0) indicates that the corresponding failure time is right censored.
`alpha`	Nonnegative vector. Small entries are recommended in order to specify a non-informative prior distribution.
`beta`	Nonnegative vector. Small entries are recommended in order to specify a non-informative prior distribution.
`c.r`	Nonnegative vector. The higher the entries, the higher the correlation of two consecutive failure times.
`a.eps`	Numeric. Shape parameter for the prior gamma distribution of epsilon when `type.c = 4`.
`b.eps`	Numeric. Scale parameter for the prior gamma distribution of epsilon when `type.c = 4`.
`type.c`	Integer. 1=defines `c.r` as a zero-entry vector; 2=lets the user define `c.r` freely; 3=assigns `c.r` an exponential prior distribution with mean `epsilon`; 4=assigns `c.r` an exponential hierarchical distribution with mean `epsilon` which in turn has a a Ga(a.eps, b.eps) distribution.
`epsilon`	Double. Mean of the exponential distribution assigned to `c.r`
`iterations`	Integer. Number of iterations including the `burn.in` and `thining` to be computed for the Markov chain.
`burn.in`	Integer. Length of the burn-in period for the Markov chain.
`thinning`	Integer. Factor by which the chain will be thinned. Thinning the Markov chain is to reduces autocorrelation.
`printtime`	Logical. If `TRUE`, prints out the execution time.

Details

Computes the Gibbs sampler given by the full conditional distributions of u and Pi (Nieto-Barajas & Walker, 2002) and arranges the resulting Markov chain into a tibble which can be used to obtain posterior summaries.

Note

It is recommended to verify chain's stationarity. This can be done by checking each partition element individually. See BePlotDiag.

References

- Nieto-Barajas, L. E. & Walker, S. G. (2002). Markov beta and gamma processes for modelling hazard rates. *Scandinavian Journal of Statistics* **29**: 413-424.

See Also

BePlotDiag, BePloth

Examples

```
## Simulations may be time intensive. Be patient.

## Example 1
#  data(psych)
#  timesP <- psych$time
#  deltaP <- psych$death
#  BEX1 <- BeMRes(timesP, deltaP, iterations = 3000, burn.in = 300, thinning = 1)

## Example 2
#  data(gehan)
#  timesG <- gehan$time[gehan$treat == "control"]
#  deltaG <- gehan$cens[gehan$treat == "control"]
#  BEX2 <- BeMRes(timesG, deltaG, type.c = 2, c.r = rep(50, 22))
```

`BePlotDiag` *Diagnosis plots for PI, U, C and Epsilon*

Description

Diagnostic plots for hazard rate (PI), latent variable (U), dependence parameter (C) and parameter of the hierarchical model (Epsilon).

Usage

```
BePlotDiag(M, variable = "PI", pos = 1)
```

Arguments

`M`	Tibble. Contains the output by `BeMRes`
`variable`	Either "PI", "U", "C" or "Epsilon". Variable for which diagnostic plot will be shown.
`pos`	Positive integer. Position of the selected `variable` to be plotted.

Details

This function returns a diagnostics plot for the chain of the selected variable. The diagnostics includes trace, ergodic mean, autocorrelation function and histogram.

References

- Nieto-Barajas, L. E. & Walker, S. G. (2002). Markov beta and gamma processes for modelling hazard rates. *Scandinavian Journal of Statistics* **29**: 413-424.

See Also

BeMRes

Examples

```
## Simulations may be time intensive. Be patient.

## Example 1
#  data(psych)
#  timesP <- psych$time
#  deltaP <- psych$death
#  BEX1 <- BeMRes(timesP, deltaP, iterations = 3000, burn.in = 300, thinning = 1)
#  BePlotDiag(BEX1, variable = "PI", pos = 2)
#  BePlotDiag(BEX1, variable = "U", pos = 3)

## Example 2
#  data(gehan)
#  timesG <- gehan$time[gehan$treat == "control"]
#  deltaG <- gehan$cens[gehan$treat == "control"]
#  BEX2 <- BeMRes(timesG, deltaG, type.c = 2, c.r = rep(50, 22))
#  BePlotDiag(BEX2, variable = "PI", pos = 5)
#  BePlotDiag(BEX2, variable = "U", pos = 4)
```

`BePloth` *Plots for the discrete Hazard and Survival Function Estimates*

Description

Plots the resulting hazard function along with the survival function estimates defined by the Markov beta process (Nieto-Barajas and Walker, 2002).

Usage

```
BePloth(
  M,
  type.h = "dot",
  add.survival = T,
  intervals = T,
  confidence = 0.95,
```

```
  summary = FALSE
)
```

Arguments

`M`	tibble. Contains the output generated by `BeMRres`.
`type.h`	character, "line" = plots the hazard rate of each interval joined by a line, "dot" = plots the hazard rate of each interval with a dot.
`add.survival`	logical, If `TRUE`, plots the Nelson-Alen based estimate in the same graphic of the hazard rate and the Kaplan-Meier estimates of the survival function.
`intervals`	logical. If TRUE, plots confidence bands for the selected functions including Nelson-Aalen and/or Kaplan-Meier estimate.
`confidence`	Numeric. Confidence band width.
`summary`	Logical. If `TRUE`, a summary for hazard and survival functions is returned as a tibble.

Details

This function returns estimators plots for the hazard rate as computed by `BeMRes` together with the Nelson-Aalen estimate along with their confidence intervals for the data set given. Additionally, it plots the survival function and the Kaplan-Meier estimate with their corresponding credible intervals.

Value

`SUM.h`	Numeric tibble. Summary for the mean, median, and a `confint / 100` confidence interval for each failure time of the hazard function.
`SUM.S`	Numeric tibble. Summary for the mean, median, and a `confint / 100` confidence interval for each failure time of the survival function.

References

- Nieto-Barajas, L. E. & Walker, S. G. (2002). Markov beta and gamma processes for modelling hazard rates. *Scandinavian Journal of Statistics* **29**: 413-424.

See Also

BeMRes, BePlotDiag

Examples

```
## Simulations may be time intensive. Be patient.

## Example 1
#  data(psych)
#  timesP <- psych$time
#  deltaP <- psych$death
```

```
#  BEX1 <- BeMRes(timesP, deltaP, iterations = 3000, burn.in = 300, thinning = 1)
#  BePloth(BEX1)
#  sum <- BePloth(BEX1, type.h = "line", summary = T)

## Example 2
#  data(gehan)
#  timesG <- gehan$time[gehan$treat == "control"]
#  deltaG <- gehan$cens[gehan$treat == "control"]
#  BEX2 <- BeMRes(timesG, deltaG, type.c = 2, c.r = rep(50, 22))
#  BePloth(BEX2)
```

`BGPHazard` *BGPHazard: A package bayesian nonparametric inference in survival analysis.*

Description

The BGPHazard package provides three categories of important functions: simulating, diagnostic and result.

Simulating functions

The simulating functions are used to make posterior inference for the bayesian survival semiparametric models as described by Nieto-Barajas and Walker (2002), Nieto-Barajas (2003) and Nieto-Barajas, L. E., & Yin, G. (2008)

Diagnostic functions

The diagnostic functions are used to make convergence diagnosics plots about the simulations of the parameters/variables.

Result functions

The result functions are used to produce estimators plots of the hazard function along with the survival function defined by the model.

`BMTKleinbook` *Times to death or relapse (in days) for patients with bone marrow transplants for Hodgkin and non-Hodgkin lymphoma*

Description

Was collected on 43 bone marrow transplant patients at The Ohio State University Bone Marrow Transplant Unit. Details of this study can be found in Avalos et al. (1993).

Usage

```
data(BMTKleinbook)
```

Format

A data frame with 43 observations containing:

`times` time to death or relapse in days

`delta` Status indicator: 1 = death or relapse; 0 = otherwise

`tTransplant` Allogeneic transplant from an HLA match sibling donor (1) or an autogeneic transplant (0)

`hodgkin` Hodgkin disease (1), or non-Hodgkin lymphoma (0)

`karnofsky` The pretransplant Karnofsky score

`waiting` Waiting time to transplant

Source

Klein, J. P., and Moeschberger, M. L. (2003). Survival analysis: techniques for censored and truncated data. Springer Science & Business Media.

References

Copelan, E. A., Biggs, J. C., Thompson, J. M., Crilley, P., Szer, J., Klein, J. P., Kapoor, N., Avalos, B. R., Cunningham, I., Atkinson, K., Downs, K., Harmon, G. S., Daly, M. B., Brodsky, I., Bulova, S. I., and Tutschka, P. J. Treatment for Acute Myelocytic Leukemia with Allogeneic Bone Marrow Transplantation Following Preparation with Bu/Cy. Blood 78 (1991): 838-843.

Examples

```
## Cox Cure Gama Process Example 1
# data(BMTKleinbook)
   # res <- CCuMRes(BMTKleinbook, covs.x = c("tTransplant","hodgkin","karnofsky","waiting"),
    #                covs.y = c("tTransplant","hodgkin","karnofsky","waiting"),
    #                        type.t = 2, K = 72, length = 30,
   #                  alpha = rep(2,72), beta = rep(2,72), c.r = rep(50, 71), type.c = 2,
    #                        var.delta.str = .1, var.theta.str = 1,
    #                        var.delta.ini = 100, var.theta.ini = 100,
    #                        iterations = 100, burn.in = 10, thinning = 1)
```

`BSBHaz` *BSBHaz posterior samples using Gibbs Sampler*

Description

`BSBHaz` samples posterior observations from the bivariate survival model (BSBHaz model) proposed by Nieto-Barajas & Walker (2007).

Usage

```
BSBHaz(
  bsb_init,
  iter,
  burn_in = 0,
  omega_d = NULL,
  gamma_d = NULL,
  theta_d = NULL,
  seed = 42
)
```

Arguments

`bsb_init`	An object of class 'BSBinit' created by `BSBInit`.
`iter`	A positive integer. Number of samples generated by the Gibbs Sampler.
`burn_in`	A positive integer. Number of iterations that should be discarded as burn in period.
`omega_d`	A positive double. This parameter defines the interval used in the Metropolis-Hastings algorithm to sample proposals for omega. See details.
`gamma_d`	A positive double. This parameter defines the interval used in the Metropolis-Hastings algorithm to sample proposals for gamma. See details.
`theta_d`	A positive double. This parameter defines the interval used in the Metropolis-Hastings algorithm to sample proposals for theta. See details.
`seed`	Random seed used in sampling.

Details

BSBHaz (Nieto-Barajas & Walker, 2007) is a bayesian semiparametric model for bivariate survival data. The marginal densities are nonparametric survival models and the joint density is constructed via a mixture. Dependence between failure times is modeled using two frailties, and the dependence between these frailties is modeled with a copula.

This command obtains posterior samples from model parameters. The samples from omega, gamma, and theta are obtained using the Metropolis-Hastings algorithm. The proposal distributions are uniform for the three parameters. The parameters `omega_d`, `gamma_d` and `theta_d` modify the intervals from which the uniform proposals are sampled. If these parameters are too large, the acceptance rates will decrease and the chains will get stuck. On the other hand, if these parameters are small, the acceptance rates will be too high and the chains will not explore the posterior support effectively.

Value

An object of class '`BSBHaz`' containing the samples from the variables of interest.

Examples

```
t1 <- survival::Surv(c(1, 2, 3))
t2 <- survival::Surv(c(1, 2, 3))

init <- BSBInit(t1 = t1, t2 = t2, seed = 0)
samples <- BSBHaz(init, iter = 10, omega_d = 2,
gamma_d = 10, seed = 10)
```

`BSBInit` *Initial setup for BSBHaz model*

Description

`BSBInit` creates the necessary data structure for use in `BSBHaz`.

Usage

```
BSBInit(
  df = NULL,
  t1 = NULL,
  t2 = NULL,
  alpha = 0.001,
  beta = 0.001,
  c = 1000,
  part_len = 1,
  seed = 42
)
```

Arguments

`df`	A data frame with columns 't1', 't2', 'delta1', 'delta2'. Any other columns not named 'id' are taken to be predictors. These predictors must be numeric, i.e., **categorical predictors must be one-hot encoded**.
`t1, t2`	Objects of class 'Surv' as created by `Surv`.
`alpha, beta, c`	Doubles. Parameters for Markov gamma hazard priors.
`part_len`	A double that gives the length of time partition intervals.
`seed`	Random seed for variable initialization.

Details

This function reads and formats censored bivariate survival data in the following way. If `df` is provided, failure times and censoring indicadors are assumed to be columns named 't1', 't2', 'delta1', and 'delta2'. Other columns not named 'id' (ignoring case) are taken to be predictors. If `df` has no columns 'delta1' or 'delta2', observations are taken as exact.

If `df` is not provided, then `t1` and `t2` are expected to be objects of class 'Surv' created by `Surv` and the model does not use predictors. Only right-censored observations are supported. Only `df` or `t1` and `t2` must be supplied. `df` argument comes first for use in pipes.

Value

An object of class '`BSBinit`'

Examples

```
t1 <- survival::Surv(c(1, 2, 3))
t2 <- survival::Surv(c(1, 2, 3))

init <- BSBInit(t1 = t1, t2 = t2, seed = 0)
```

`BSBPlotDiag` *Plot diagnostics for BSBHaz model*

Description

Plot diagnostics for BSBHaz model

Usage

```
BSBPlotDiag(
  bsbhaz,
  variable = c("omega1", "omega2", "lambda1", "lambda2", "gamma", "theta"),
  type = c("traceplot", "ergodic_means")
)
```

Arguments

`bsbhaz`	An object of class 'BSBHaz' created by `BSBHaz`.
`variable`	A character indicating which variable to get the plot from.
`type`	A character indicating if the plot should be a traceplot or plot the ergodic means.

Examples

```
t1 <- survival::Surv(c(1, 2, 3))
t2 <- survival::Surv(c(1, 2, 3))

init <- BSBInit(t1 = t1, t2 = t2, seed = 0)
samples <- BSBHaz(init, iter = 10, omega_d = 2,
gamma_d = 10, seed = 10)

BSBPlotDiag(samples, variable = "omega1", type = "traceplot")
```

`BSBPlotSumm` *Plot summaries for BSBHaz model*

Description

Plot summaries for BSBHaz model

Usage

```
BSBPlotSumm(bsbhaz, variable = c("lambda1", "lambda2", "s1", "s2"))
```

Arguments

`bsbhaz`	An object of class 'BSBHaz' created by `BSBHaz`.
`variable`	A character indicating the variable to plot.

Examples

```
t1 <- survival::Surv(c(1, 2, 3))
t2 <- survival::Surv(c(1, 2, 3))

init <- BSBInit(t1 = t1, t2 = t2, seed = 0)
samples <- BSBHaz(init, iter = 10, omega_d = 2,
gamma_d = 10, seed = 10)

BSBPlotSumm(samples, "s1")
```

`BSBSumm` *Get posterior summaries for BSBHaz model*

Description

Get posterior summaries for BSBHaz model

Usage

```
BSBSumm(
  bsbhaz,
 variable = c("omega1", "omega2", "lambda1", "lambda2", "gamma", "theta", "s1", "s2")
)
```

Arguments

`bsbhaz`	An object of class 'BSBHaz' created by `BSBHaz`.
`variable`	A character indicating which variable to get summaries from.

Value

A data frame with posterior sample means and a 95 % probability interval. For `omega1`, `omega2`, `gamma`, and `theta` also includes a column with the acceptance rates for the Metropolis-Hastings algorithm.

Examples

```
t1 <- survival::Surv(c(1, 2, 3))
t2 <- survival::Surv(c(1, 2, 3))

init <- BSBInit(t1 = t1, t2 = t2, seed = 0)
samples <- BSBHaz(init, iter = 10, omega_d = 2,
gamma_d = 10, seed = 10)

BSBSumm(samples, variable = "gamma")
BSBSumm(samples, variable = "omega1")
BSBSumm(samples, variable = "lambda1")
```

`CCuMRes` *Bayesian Semiparametric Cure Rate Model with an Unknown Threshold and Covariate Information*

Description

Posterior inference for the bayesian semiparmetric cure rate model with covariates in survival analysis.

Usage

```
CCuMRes(
  data,
  covs.x = names(data)[seq.int(3, ncol(data))],
  covs.y = names(data)[seq.int(3, ncol(data))],
  type.t = 3,
  K = 50,
  utao = NULL,
  alpha = rep(0.01, K),
  beta = rep(0.01, K),
  c.r = rep(0, K - 1),
  c.nu = 1,
  var.theta.str = 25,
  var.delta.str = 25,
  var.theta.ini = 100,
  var.delta.ini = 100,
  type.c = 4,
  a.eps = 0.1,
  b.eps = 0.1,
  epsilon = 1,
  iterations = 5000,
  burn.in = floor(iterations * 0.2),
  thinning = 3,
  printtime = TRUE
)
```

Arguments

`data`	Double tibble. Contains failure times in the first column, status indicator in the second, and, from the third to the last column, the covariate(s).
`covs.x`	Character. Names of covariables to be part of the multiplicative part of the hazard
`covs.y`	Character. Names of covariables to determine the cure threshold por each patient.
`type.t`	Integer. 1=computes uniformly-dense intervals; 2= partition arbitrarily defined by the user with parameter utao and 3=same length intervals.
`K`	Integer. Partition length for the hazard function.
`utao`	vector. Partition specified by the user when type.t = 2. The first value of the vector has to be 0 and the last one the maximum observed time, either censored or uncensored.
`alpha`	Nonnegative entry vector. Small entries are recommended in order to specify a non-informative prior distribution.
`beta`	Nonnegative entry vector. Small entries are recommended in order to specify a non-informative prior distribution.
`c.r`	Nonnegative vector. The higher the entries, the higher the correlation of two consective intervals.

`c.nu`	Tuning parameter for the proposal distribution for c. Only when `type.c` is 3 or 4.
`var.theta.str`	Double. Variance of the proposal normal distribution for theta in the Metropolis-Hastings step.
`var.delta.str`	Double. Variance of the proposal normal distribution for delta in the Metropolis-Hastings step.
`var.theta.ini`	Double. Variance of the prior normal distribution for theta.
`var.delta.ini`	Double. Variance of the prior normal distribution for delta. from the acceptance ratio in the Metropolis-Hastings algorithm for delta*.
`type.c`	1=defines `c.r` as a zero-entry vector; 2=lets the user define `c.r` freely; 3=assigns `c.r` an exponential prior distribution with mean 1; 4=assigns `c.r` an exponential hierarchical distribution with mean `epsilon` which in turn has a Ga(a.eps, b.eps) distribution.
`a.eps`	Double. Shape parameter for the prior gamma distribution of epsilon when `type.c = 4`.
`b.eps`	Double. Scale parameter for the prior gamma distribution of epsilon when `type.c = 4`.
`epsilon`	Double. Mean of the exponencial distribution assigned to `c.r` when `type.c = 3`.
`iterations`	Integer. Number of iterations including the `burn.in` to be computed for the Markov chain.
`burn.in`	Integer. Length of the burn-in period for the Markov chain.
`thinning`	Integer. Factor by which the chain will be thinned. Thinning the Markov chain reduces autocorrelation.
`printtime`	Logical. If `TRUE`, prints out the execution time.

Details

Computes the Gibbs sampler with the full conditional distributions of all model parameters (Nieto-Barajas & Yin, 2008) and arranges the resulting Markov chain into a tibble which can be used to obtain posterior summaries. Prior distributions for the regression coefficients Theta and Delta are assumend independent normals with zero mean and variance `var.theta.ini`, `var.delta.ini`, respectively.

Note

It is recommended to verify chain's stationarity. This can be done by checking each element individually. See `CCuPlotDiag`.

References

- Nieto-Barajas, L. E., & Yin, G. (2008). Bayesian semiparametric cure rate model with an unknown threshold. Scandinavian Journal of Statistics, 35(3), 540-556. https://doi.org/10.1111/j.1467-9469.2007.00589.x
- Nieto-barajas, L. E. (2002). Discrete time Markov gamma processes and time dependent covariates in survival analysis. Statistics, 2-5.

See Also

CCuPlotDiag, CCuPloth

Examples

```
# data(BMTKleinbook)
   # res <- CCuMRes(BMTKleinbook, covs.x = c("tTransplant","hodgkin","karnofsky","waiting"),
    #                covs.y = c("tTransplant","hodgkin","karnofsky","waiting"),
    #                        type.t = 2, K = 72, length = 30,
   #                   alpha = rep(2,72), beta = rep(2,72), c.r = rep(50, 71), type.c = 2,
    #                        var.delta.str = .1, var.theta.str = 1,
    #                        var.delta.ini = 100, var.theta.ini = 100,
    #                        iterations = 100, burn.in = 10, thinning = 1)
```

`CCuPlotDiag` *Diagnostics plots for Lambda, Theta, Delta, U, C, Pi, Z and Epsilon. Hazard function, cure proportion and cure time for the median observation.*

Description

Diagnostic plots for hazard rate (Lambda), regression parameters for the hazard (Theta), regression parameters for the cure rate (Delta), latent variable (U), dependence parameter (C), mean of cure threshold (Mu), cure proportion (Pi), cure threshold (Z) and the parameter of the hierarchical prior (Epsilon).

Usage

```
CCuPlotDiag(M, variable = "Lambda", pos = 1)
```

Arguments

`M`	tibble. Contains the output by `CCuMRes`.
`variable`	Either "Lambda", "U", "C", "Mu", "Pi", "Z" or "Epsilon". Variable for which diagnostic plot will be shown.
`pos`	Positive integer. Position of the selected `variable` to be plotted.

Details

This function returns a diagnosyics plot for which the chain for the selected variable can be monitored. Diagnostics includes trace, ergodic mean, autocorrelation function and histogram.

References

Nieto-Barajas, L. E., & Yin, G. (2008). Bayesian semiparametric cure rate model with an unknown threshold. *Scandinavian Journal of Statistics*, **35(3)**, 540-556. https://doi.org/10.1111/j.1467-9469.2007.00589.x

See Also

CCuMRes

Examples

```
## Simulations may be time intensive. Be patient.

## Example 1
# data(BMTKleinbook)
# res <- CCuMRes(BMTKleinbook, covs.x = c("tTransplant","hodgkin","karnofsky","waiting"),
#                 covs.y = c("tTransplant","hodgkin","karnofsky","waiting"),
#                         type.t = 2, K = 72, length = 30,
#                      alpha = rep(2,72), beta = rep(2,72), c.r = rep(50, 71), type.c = 2,
#                         var.delta.str = .1, var.theta.str = 1,
#                         var.delta.ini = 100, var.theta.ini = 100,
#                         iterations = 100, burn.in = 10, thinning = 1)
# CCuPlotDiag(M = res, variable = "Z")
# CCuPlotDiag(M = res, variable = "Pi.m")
# CCuPlotDiag(M = res, variable = "Lambda", pos = 2)
# CCuPlotDiag(M = res, variable = "U", pos = 4)
```

`CCuPloth` *Plots for the Hazard and Survival Funcion Estimates*

Description

Plots the resulting hazard function and the survival function estimates defined by the bayesian semiparametric cure rate model with an unknown threshold (Nieto-Barajas & Yin, 2008).

Usage

```
CCuPloth(
  M,
  new_obs = NULL,
  type.h = "segment",
  qn = 0.5,
  intervals = T,
  confidence = 0.95,
```

```
  summary = FALSE
)
```

Arguments

`M`	tibble. Contains the output generated by `CuMRres`.
`new_obs`	tibble. Contains the covariate information for new observations.
`type.h`	character. "segment"= use segments to plot hazard rates, "line" = link hazard rates by a line
`qn`	Numeric. Quantile for Tao (cure time) that should be visualized on the plot.
`intervals`	logical. If TRUE, plots credible intervals.
`confidence`	Numeric. Confidence level.
`summary`	Logical. If `TRUE`, a summary for the hazard and survival functions is returned as a tibble.

Details

This function returns estimators plots for the hazard rate as it is computed by CCuMRes and the cure time (quantile of Tao specified by the user) together with credible intervals. Additionally, it plots the survival function and the cure proportion estimates with their corresponding credible intervals.

Value

`SUM.h`	Numeric tibble. Summary for the mean, median, and a `confint / 100` confidence interval for each segment of the hazard function. If `summary = TRUE`
`SUM.S`	Numeric tibble. Summary for the mean, median, and a `confint / 100` confidence interval for each segment of the survival function. If `summary = TRUE`

References

- Nieto-Barajas, L. E. (2003). Discrete time Markov gamma processes and time dependent covariates in survival analysis. *Bulletin of the International Statistical Institute 54th Session*. Berlin. (CD-ROM).

-Nieto-Barajas, L. E., & Yin, G. (2008). Bayesian semiparametric cure rate model with an unknown threshold. *Scandinavian Journal of Statistics*, **35(3)**, 540-556. https://doi.org/10.1111/j.1467-9469.2007.00589.x

See Also

CCuMRes,

Examples

```
## Simulations may be time intensive. Be patient.

## Example 1
# data(BMTKleinbook)
```

```
# res <- CCuMRes(BMTKleinbook, covs.x = c("tTransplant","hodgkin","karnofsky","waiting"),
 #                covs.y = c("tTransplant","hodgkin","karnofsky","waiting"),
 #                        type.t = 2, K = 72, length = 30,
#                   alpha = rep(2,72), beta = rep(2,72), c.r = rep(50, 71), type.c = 2,
 #                        var.delta.str = .1, var.theta.str = 1,
 #                        var.delta.ini = 100, var.theta.ini = 100,
 #                        iterations = 100, burn.in = 10, thinning = 1)
 #
 # CCuPloth(res, type.h = "segment",qn=.5, summary = T)
 #
 # new_obs <- tibble(tTransplant=c(0,0,0,0),
 #                        hodgkin=c(0,1,0,1),
 #                        karnofsky=c(90,90,60,60),
 #                        waiting=c(36,36,36,36)
 # )
 #
 # ind <- CCuPloth(res, new_obs, qn = .5)
 # ind
```

`CGaMRes` *Markov Gamma Model with Covariates*

Description

Posterior inference for the Bayesian non-parametric Markov gamma model with covariates in survival analysis.

Usage

```
CGaMRes(
  data,
  type.t = 2,
  length = 1,
  K = 5,
  alpha = rep(0.01, K),
  beta = rep(0.01, K),
  c.r = rep(1, K - 1),
  c.nu = 1,
  var.theta.str = 25,
  var.theta.ini = 100,
  a.eps = 0.1,
  b.eps = 0.1,
  type.c = 4,
  epsilon = 1,
  iterations = 1000,
  burn.in = floor(iterations * 0.2),
```

```
    thinning = 3,
    printtime = TRUE
  )
```

Arguments

`data`	Double tibble. Contains failure times in the first column, status indicator in the second, and, from the third to the last column, the covariate(s).
`type.t`	Integer. 1=computes uniformly-dense intervals; 2=length intervals defined by user and 3=same length intervals.
`length`	Integer. Interval length of the partition.
`K`	Integer. Partition length for the hazard function.
`alpha`	Nonnegative entry vector. Small entries are recommended in order to specify a non-informative prior distribution.
`beta`	Nonnegative entry vector. Small entries are recommended in order to specify a non-informative prior distribution.
`c.r`	Nonnegative vector. The higher the entries, the higher the correlation of two consecutive intervals.
`c.nu`	Tuning parameter for the proposal distribution for c.
`var.theta.str`	Double. Variance of the proposal normal distribution for theta in the Metropolis-Hastings step.
`var.theta.ini`	Double. Variance of the prior normal distribution for theta.
`a.eps`	Double. Shape parameter for the prior gamma distribution of epsilon when `type.c = 4`.
`b.eps`	Double. Scale parameter for the prior gamma distribution of epsilon when `type.c = 4`.
`type.c`	1=defines `c.r` as a zero-entry vector; 2=lets the user define `c.r` freely; 3=assigns `c.r` by computing an exponential prior distribution with mean 1; 4=assigns `c.r` an exponential hierarchical distribution with mean `epsilon` which in turn has a Ga(a.eps, b.eps) distribution.
`epsilon`	Double. Mean of the exponential distribution assigned to `c.r` when `type.c = 3`.
`iterations`	Integer. Number of iterations including the `burn.in` to be computed for the Markov chain.
`burn.in`	Integer. Length of the burn-in period for the Markov chain.
`thinning`	Integer. Factor by which the chain will be thinned. Thinning the Markov chain reduces autocorrelation.
`printtime`	Logical. If `TRUE`, prints out the execution time.

Details

Computes the Gibbs sampler with the full conditional distributions of Lambda and Theta (Nieto-Barajas, 2003) and arranges the resulting Markov chain into a matrix which can be used to obtain posterior summaries. Prior distributions for the re gression coefficients (Theta) are assumed independent normals with zero mean and variance `var.theta.ini`.

Note

It is recommended to verify chain's stationarity. This can be done by checking each element individually. See CGaPlotDiag To obtain posterior summaries of the coefficients use function CGaPloth.

References

- Nieto-Barajas, L. E. (2003). Discrete time Markov gamma processes and time dependent covariates in survival analysis. *Bulletin of the International Statistical Institute 54th Session*. Berlin. (CD-ROM).

- Nieto-Barajas, L. E. & Walker, S. G. (2002). Markov beta and gamma processes for modelling hazard rates. *Scandinavian Journal of Statistics* **29**: 413-424.

See Also

CGaPlotDiag, CGaPloth

Examples

```
## Simulations may be time intensive. Be patient.

## Example 1
#  data(leukemiaFZ)
#  leukemia1 <- leukemiaFZ
#  leukemia1$wbc <- log(leukemiaFZ$wbc)
#  CGEX1 <- CGaMRes(data = leukemia1, K = 10, iterations = 100, thinning = 1)

## Example 2. Refer to "Cox-gamma model example" section in package vignette for details.
# SampWeibull <- function(n, a = 10, b = 1, beta = c(1, 1)) {
#   M <- tibble(i = seq(n), x_i1 = runif(n), x_i2 = runif(n),
#               t_i = rweibull(n, shape = b,
#                                  scale = 1 / (a * exp(x_i1*beta[1] + x_i2*beta[2]))),
#               c_i = rexp(n), delta = t_i > c_i,
#               `min{c_i, d_i}` = min(t_i, c_i))
#   return(M)
# }
#  dat <- SampWeibull(100, 0.1, 1, c(1, 1))
#  dat <- dat %>% select(4,6,2,3)
#  CG <- CGaMRes(data = leukemia1, K = 10, iterations = 100, thinning = 1)
#  CGaPloth(CG)
```

`CGaPlotDiag` *Diagnostics plots for lambda, U, C, Epsilon and Theta*

Description

Diagnostics plots for hazard rate (Lambda), latent variable (U), dependence variable (C), parameter of the hierarchical model (Epsilon) and regression coefficients (Theta).

Usage

```
CGaPlotDiag(M, variable = "Lambda", pos = 1)
```

Arguments

`M`	Tibble. Contains the output by `CGaMRes`
`variable`	Either "Lambda", "U", "C", "Epsilon" or "Theta". Variable for which diagnostics plot will be shown.
`pos`	Positive integer. Position of the selected `variable` to be plotted.

Details

This function returns a diagnostics plot for the chain of the selected variable. The diagnostics includes trace, ergodic mean, autocorrelation function and histogram.

References

- Nieto-Barajas, L. E. (2003). Discrete time Markov gamma processes and time dependent covariates in survival analysis. *Bulletin of the International Statistical Institute 54th Session*. Berlin. (CD-ROM).

- Nieto-Barajas, L. E. & Walker, S. G. (2002). Markov beta and gamma processes for modelling hazard rates. *Scandinavian Journal of Statistics* **29**: 413-424.

See Also

CGaMRes

Examples

```
## Simulations may be time intensive. Be patient.

## Example 1
#  data(leukemiaFZ)
#  leukemia1 <- leukemiaFZ
#  leukemia1$wbc <- log(leukemiaFZ$wbc)
#  CGEX1 <- CGaMRes(data = leukemia1, K = 10, iterations = 1000, thinning = 1)
#  CGaPlotDiag(CGEX1,variable="Theta",pos=1)
```

`CGaPloth` *Plots for the Hazard and Survival Funcion Estimates for the Bayesian non-parametric Markov gamma model with covariates in survival analysis.*

Description

Plots the resulting hazard function along with the survival function estimate defined by the Markov gamma process with covariates (Nieto-Barajas, 2003).

Usage

```
CGaPloth(
  M,
  new_obs = NULL,
  type.h = "segment",
  coxSurv = T,
  intervals = T,
  confidence = 0.95,
  summary = FALSE
)
```

Arguments

`M`	tibble. Contains the output generated by `CuMRres`.
`new_obs`	tibble. The function calculates the hazard rates and survival function estimates for specific individuals expressed in a tibble, the names of the columns have to be the same as the data input.
`type.h`	character. "segment"= use segments to plot hazard rates, "line" = link hazard rates by a line
`coxSurv`	logical. Add estimated Survival function with the Cox-Model
`intervals`	logical. If TRUE, plots confidence bands for the selected functions including Cox-Model.
`confidence`	Numeric. Confidence level.
`summary`	logical. If `TRUE`, a summary for hazard and survival functions is returned as a tibble.

Details

This function return plots for the resulting hazard rate as it is computed by `CGaMRes` `and the quantile of Tao specified by the user aswell as an annotation`. In the same plot the credible intervals for both variables are plotted; The mean of Pi is also annotated. Additionally, it plots the survival function with their corresponding credible intervals.

Value

`SUM.h`	Numeric tibble. Summary for the mean, median, and a `confint / 100` confidence interval for each segment of the hazard function. If `summary = TRUE`
`SUM.S`	Numeric tibble. Summary for the mean, median, and a `confint / 100` confidence interval for each segment of the survival function. If `summary = TRUE`

References

- Nieto-Barajas, L. E. (2003). Discrete time Markov gamma processes and time dependent covariates in survival analysis. *Bulletin of the International Statistical Institute 54th Session*. Berlin. (CD-ROM).

- Nieto-Barajas, L. E. & Walker, S. G. (2002). Markov beta and gamma processes for modelling hazard rates. *Scandinavian Journal of Statistics* **29**: 413-424.

See Also

CGaMRes,

Examples

```
## Simulations may be time intensive. Be patient.

   # ## Example 1
   #  data(leukemiaFZ)
   #  leukemia1 <- leukemiaFZ
   #  leukemia1$wbc <- log(leukemiaFZ$wbc)
   #  CGEX1 <- CGaMRes(data = leukemia1, K = 10, iterations = 100, thinning = 1)
   #  CGaPloth(CGEX1)
```

---

cpo *Conditional Predictive Ordinate (CPO) Statistic*

Description

Makes the CPO Plot and calculates the logarithm of the Pseudomarginal likelihood (LPML).

Usage

```
cpo(res)
```

Arguments

`res`	tibble. The output from the *Res functions, where * could either be BeM, GaM, CGaM, CuM, CCuM

Details

Computes de CPO as a goodness of fit measure

Value

`LPML`	The value of the logarithm of the Pseudomarginal likelihood
`plot`	CPO Plot

References

See Geisser (1993); Gelfand, Dey, and Chang (1992); Dey, Chen, and Chang (1997); and Sinha and Dey (1997)

Examples

```
## Example 1
#  data(gehan)
#  timesG <- gehan$time[gehan$treat == "6-MP"]
#  deltaG <- gehan$cens[gehan$treat == "6-MP"]
#  GEX1 <- GaMRes(timesG, deltaG, K = 8, iterations = 3000)
#  cpo(GEX1)
```

`crm3` *Simulated data from the proportional hazards cure rate model of Yakolev & Tsodikov (1996) by taking a baseline density with a bounded support.*

Description

Triangular distribution Tri(a, c, b) as the baseline density, which puts a probability of one to the interval [a, b] and the mode at c.

Usage

```
data("crm3")
```

Format

A data frame with 100 observations with the following 2 variables.

`times`	Simulated time
`delta`	Simulated censoring

Details

In particular we took, a = 0, c = 1 and b = 4. The censoring time was independently generated from a uniform distribution to yield a 30% censoring rate. Sample size n = 100 and the cure proportion exp{-theta}=0.2.

Source

Nieto-Barajas, L. E., & Yin, G. (2008)

Examples

```
## Cure Gama model Example 1
# data(crm3)
# times<-crm3$times
# delta<-crm3$delta
# res <- CuMRes(times, delta, type.t = 2,
#                  K = 100, length = .1, alpha = rep(1, 100  ),
#                  beta = rep(1, 100),c.r = rep(50, 99),
#                  iterations = 100, burn.in = 10, thinning = 1, type.c = 2)
```

`CuMRes` *Bayesian Semiparametric Cure Rate Model with an Unknown Threshold*

Description

Posterior inference for the bayesian semiparametric cure rate model in survival analysis.

Usage

```
CuMRes(
  times,
  delta = rep(1, length(times)),
  type.t = 3,
  K = 5,
  utao = NULL,
  alpha = rep(0.01, K),
  beta = rep(0.01, K),
  c.r = rep(1, (K - 1)),
  type.c = 4,
  epsilon = 1,
  c.nu = 1,
  a.eps = 0.1,
  b.eps = 0.1,
  a.mu = 0.01,
  b.mu = 0.01,
  iterations = 1000,
  burn.in = floor(iterations * 0.2),
```

```
    thinning = 5,
    printtime = TRUE
  )
```

Arguments

`times`	Numeric positive vector. Failure times.
`delta`	Logical vector. Status indicator. `TRUE` (1) indicates exact lifetime is known, `FALSE` (0) indicates that the corresponding failure time is right censored.
`type.t`	Integer. 1=computes uniformly-dense intervals; 2= partition arbitrarily defined by the user with parameter utao and 3=same length intervals.
`K`	Integer. Partition length for the hazard function if `type.t`=1 or `type.t`=3.
`utao`	vector. Partition specified by the user when type.t = 2. The first value of the vector has to be 0 and the last one the maximum observed time, either censored or uncensored.
`alpha`	Nonnegative entry vector. Small entries are recommended in order to specify a non-informative prior distribution.
`beta`	Nonnegative entry vector. Small entries are recommended in order to specify a non-informative prior distribution.
`c.r`	Nonnegative vector. The higher the entries, the higher the correlation of two consecutive intervals.
`type.c`	1=defines `c.r` as a zero-entry vector; 2=lets the user define `c.r` freely; 3=assigns `c.r` by computing an exponential prior distribution with mean epsilon; 4=assigns `c.r` by computing an exponential hierarchical distribution with mean `epsilon` which in turn has a Ga(a.eps, b.eps) distribution.
`epsilon`	Double. Mean of the exponential distribution assigned to `c.r` when `type.c = 3`. When `type.c = 4`, `epsilon` is assigned a Ga(a.eps,b.eps) distribution.
`c.nu`	Tuning parameter for the proposal distribution for c.
`a.eps`	Numeric. Shape parameter for the prior gamma distribution of epsilon when `type.c = 4`.
`b.eps`	Numeric. Scale parameter for the prior gamma distribution of epsilon when `type.c = 4`.
`a.mu`	Numeric. Shape parameter for the prior gamma distribution of mu
`b.mu`	Numeric. Scale parameter for the prior gamma distribution of mu
`iterations`	Integer. Number of iterations including the `burn.in` to be computed for the Markov Chain.
`burn.in`	Integer. Length of the burn-in period for the Markov chain.
`thinning`	Integer. Factor by which the chain will be thinned. Thinning the Markov chain is to reduces autocorrelation.
`printtime`	Logical. If `TRUE`, prints out the execution time.

Details

Computes the Gibbs sampler with the full conditional distributions of all model parameters (Nieto-Barajas & Yin 2008) and arranges the resulting Markov chain into a tibble which can be used to obtain posterior summaries.

Note

It is recommended to verify chain's stationarity. This can be done by checking each element individually. See `CuPlotDiag`.

Examples

```
## Simulations may be time intensive. Be patient.
## Example 1
# data(crm3)
# times<-crm3$times
# delta<-crm3$delta
# res <- CuMRes(times, delta, type.t = 2,
#                   K = 100, length = .1, alpha = rep(1, 100  ),
#                   beta = rep(1, 100),c.r = rep(50, 99),
#                   iterations = 100, burn.in = 10, thinning = 1, type.c = 2)
```

`CuPlotDiag` *Diagnosis plots for Lambda, U, C, Mu, Pi, Z and Epsilon*

Description

Diagnostics plots for hazard rate (Lambda), latent variable (U), dependence variable (C), mean of cure threshold (Mu), cure proportion (Pi), cure threshold (Z) and the parameter of the hierarchical prior (Epsilon).

Usage

```
CuPlotDiag(M, variable = "Lambda", pos = 1)
```

Arguments

`M`	List. Contains the output by `CuMRes`.
`variable`	Either "Lambda", "U", "C", "Mu", "Pi", "Z" or "Epsilon". Variable for which diagnostic plot will be shown.
`pos`	Positive integer. Position of the selected `variable` to be plotted.

Details

This function returns a diagnostics plot for which the chain for the selected variable can be monitored. Diagnostics includes trace, ergodic mean, autocorrelation function and histogram.

References

Nieto-Barajas, L. E., & Yin, G. (2008). Bayesian semiparametric cure rate model with an unknown threshold. *Scandinavian Journal of Statistics*, **35(3)**, 540-556. https://doi.org/10.1111/j.1467-9469.2007.00589.x

See Also

CuMRes

Examples

```
## Simulations may be time intensive. Be patient.

## Example 1
# data(crm3)
# times<-crm3$times
# delta<-crm3$delta
# res <- CuMRes(times, delta, type.t = 2,
#                    K = 100, length = .1, alpha = rep(1, 100  ),
#                    beta = rep(1, 100),c.r = rep(50, 99),
#                    iterations = 100, burn.in = 10, thinning = 1, type.c = 2)
# CuPlotDiag(M = res, variable = "Mu")
# CuPlotDiag(M = res, variable = "Z")
# CuPlotDiag(M = res, variable = "Pi")
# CuPlotDiag(M = res, variable = "Lambda", pos = 2)
# CuPlotDiag(M = res, variable = "U", pos = 4)
# CuPlotDiag(M = res, variable = "C", pos = 3)
```

`CuPloth` *Plots for the Hazard and Survival Funcion Estimates*

Description

Plots the hazard function and the survival function estimates defined by the bayesian semiparametric cure rate model with an unknown threshold (Nieto-Barajas & Yin, 2008).

Usage

```
CuPloth(
  M,
  type.h = "segment",
  intervals = T,
  confidence = 0.95,
  qn = 0.5,
  summary = FALSE,
```

```
  position_label = "right"
)
```

Arguments

`M`	tibble. Contains the output generated by `CuMRres`.
`type.h`	character. "segment"= use segments to plot hazard rates, "line" = link hazard rates by a line
`intervals`	logical. If TRUE, plots credible intervals.
`confidence`	Numeric. Confidence level.
`qn`	Numeric. Quantile for Tao that should be visualized on the plot.
`summary`	Logical. If `TRUE`, a summary for hazard and survival functions is returned as a tibble.
`position_label`	character. Labels on the right or left side of the plot.

Details

This function return estimators plots for the resulting hazard rate as it is computed by CuMRes and the cure time (quantile of Tao specified by the user), together with credible intervals. Additionally, it plots the survival function and the cure proportion estimates with their corresponding credible intervals.

Value

`SUM.h`	Numeric tibble. Summary for the mean, median, and a `confint / 100` confidence interval for each segment of the hazard function. If `summary = TRUE`
`SUM.S`	Numeric tibble. Summary for the mean, median, and a `confint / 100` confidence interval for a grid of the survival function. If `summary = TRUE`

References

- Nieto-Barajas, L. E. (2003). Discrete time Markov gamma processes and time dependent covariates in survival analysis. *Bulletin of the International Statistical Institute 54th Session*. Berlin. (CD-ROM).

-Nieto-Barajas, L. E., & Yin, G. (2008). Bayesian semiparametric cure rate model with an unknown threshold. *Scandinavian Journal of Statistics*, **35(3)**, 540-556. https://doi.org/10.1111/j.1467-9469.2007.00589.x

See Also

CuMRes,

Examples

```
## Simulations may be time intensive. Be patient.

## Example 1
```

```
# data(crm3)
    # times<-crm3$times
    # delta<-crm3$delta
    # res <- CuMRes(times, delta, type.t = 2, length = .1,
    #                   K = 100, alpha = rep(1, 100  ),
    #                   beta = rep(1, 100),c.r = rep(50, 99),
    #                   iterations = 100, burn.in = 10, thinning = 1, type.c = 2)
    # CuPloth(res, type.h = "segment",qn=.5, summary = T)
    # CuPloth(res, type.h = "line",qn=.5)
```

`GaMRes` *Markov Gamma Model*

Description

Computes the Gibbs sampler given by the full conditional distributions of U, Lambda, C and Epsilon (Nieto-Barajas & Walker, 2002) and arranges the resulting Markov chain into a tibble which can be used to obtain posterior summaries.

Usage

```
GaMRes(
  times,
  delta = rep(1, length(times)),
  type.t = 3,
  K = 5,
  utao = NULL,
  alpha = rep(0.01, K),
  beta = rep(0.01, K),
  c.r = rep(1, (K - 1)),
  c.nu = 1,
  a.eps = 0.1,
  b.eps = 0.1,
  type.c = 4,
  epsilon = 1,
  iterations = 1000,
  burn.in = floor(iterations * 0.2),
  thinning = 5,
  printtime = TRUE
)
```

Arguments

`times`	Numeric positive vector. Failure times.

`delta`	Logical vector. Status indicator. `TRUE` (1) indicates exact lifetime is known, `FALSE` (0) indicates that the corresponding failure time is right censored.
`type.t`	Integer. 1=computes uniformly-dense intervals; 2= partition arbitrarily defined by the user with parameter utao and 3=same length intervals.
`K`	Integer. Partition length for the hazard function if `type.t`=1 or `type.t`=3.
`utao`	vector. Partition specified by the user when type.t = 2. The first value of the vector has to be 0 and the last one the maximum observed time, either censored or uncensored.
`alpha`	Nonnegative entry vector. Small entries are recommended in order to specify a non-informative prior distribution.
`beta`	Nonnegative entry vector. Small entries are recommended in order to specify a non-informative prior distribution.
`c.r`	Nonnegative vector. The higher the entries, the higher the correlation of two consecutive intervals.
`c.nu`	Tuning parameter for the proposal distribution for c.
`a.eps`	Numeric. Shape parameter for the prior gamma distribution of epsilon when `type.c = 4`.
`b.eps`	Numeric. Scale parameter for the prior gamma distribution of epsilon when `type.c = 4`.
`type.c`	1=assigns `c.r` a zero-entry vector; 2=lets the user define `c.r` freely; 3=assigns `c.r` an exponential prior distribution with mean 1; 4=assigns `c.r` an exponential hierarchical distribution with mean `epsilon` which in turn has a Ga(a.eps, b.eps) distribution.
`epsilon`	Double. Mean of the exponential distribution assigned to `c.r` when `type.c = 3`
`iterations`	Integer. Number of iterations including the `burn.in` to be computed for the Markov chain.
`burn.in`	Integer. Length of the burn-in period for the Markov chain.
`thinning`	Integer. Factor by which the chain will be thinned. Thinning the Markov chain is to reducec autocorrelation.
`printtime`	Logical. If `TRUE`, prints out the execution time.

Details

Posterior inference for the Bayesian non-parametric Markov gamma model in survival analysis.

Examples

```
## Simulations may be time intensive. Be patient.

## Example 1
data(gehan)
timesG <- gehan$time[gehan$treat == "6-MP"]
deltaG <- gehan$cens[gehan$treat == "6-MP"]
 GEX1 <- GaMRes(timesG, deltaG, K = 8, iterations = 3000)
```

```
## Example 2
data(leukemiaFZ)
timesFZ <- leukemiaFZ$time
deltaFZ <- leukemiaFZ$delta
GEX2 <- GaMRes(timesFZ, deltaFZ, type.c = 4)
```

`GaPlotDiag` *Diagnosis plots for Lambda, U, C and Epsilon*

Description

Diagnostics plots for hazard rate (Lambda), latent variable (U), dependence parameter (C) and the parameter of the hierarchical prior (Epsilon).

Usage

```
GaPlotDiag(M, variable = "Lambda", pos = 1)
```

Arguments

`M`	List. Contains the output by `GaMRes`.
`variable`	Either "Lambda", "U", "C" or "Epsilon". Variable for which informative plot will be shown.
`pos`	Positive integer. Position of the selected `variable` to be plotted.

Details

This function returns a diagnostics plot for which the chain of the selected variable can be monitored. Diagnostics includes trace, ergodic mean, autocorrelation function and histogram.

References

- Nieto-Barajas, L. E. & Walker, S. G. (2002). Markov beta and gamma processes for modelling hazard rates. *Scandinavian Journal of Statistics* **29**: 413-424.

See Also

GaMRes

Examples

```
## Simulations may be time intensive. Be patient.

## Example 1
#  data(gehan)
#  timesG <- gehan$time[gehan$treat == "6-MP"]
#  deltaG <- gehan$cens[gehan$treat == "6-MP"]
#  GEX1 <- GaMRes(timesG, deltaG, K = 8, iterations = 3000)
#  GaPlotDiag(GEX1, variable = "Lambda", pos = 2)
#  GaPlotDiag(GEX1, variable = "U", pos = 5)

## Example 2
#  data(leukemiaFZ)
#  timesFZ <- leukemiaFZ$time
#  deltaFZ <- leukemiaFZ$delta
#  GEX2 <- GaMRes(timesFZ, deltaFZ, type.c = 4)
#  GaPlotDiag(GEX2, variable = "Lambda", pos = 2)
#  GaPlotDiag(GEX2, variable = "U", pos = 3)
```

`GaPloth` *Plots for the Hazard and Survival Function Estimates*

Description

Plots the hazard function and with the survival function estimates defined by the Markov gamma process with and without covariates (Nieto-Barajas & Walker, 2002).

Usage

```
GaPloth(
  M,
  type.h = "segment",
  addSurvival = T,
  intervals = T,
  confidence = 0.95,
  summary = FALSE
)
```

Arguments

`M`	tibble. Contains the output by `CGaMRres` and `GaMRes`.
`type.h`	character. "segment"= use segments to plot hazard rates, "line" = link hazard rates by a line

`addSurvival`	Logical. If `TRUE`, Nelson-Aalen estimate is plotted over the hazard function and Kaplan-Meier estimate is plotted over the survival function.
`intervals`	logical. If TRUE, plots confidence bands for the selected functions including Nelson-Aalen and/or Kaplan-Meier estimate.
`confidence`	Numeric. Confidence level.
`summary`	Logical. If `TRUE`, a summary for hazard and survival functions is returned as a tibble.

Details

This function returns estimators plots for the resulting hazard rate as it is computed by GaMRes and CGaMRes and the Nelson-Aalen estimate along with their confidence intervals for the data set given. Additionally, it plots the survival function and the Kaplan-Meier estimate with their corresponding credible/confidence intervals.

Value

`SUM.h`	Numeric tibble. Summary for the mean, median, and a `confint / 100` confidence interval for each segment of the hazard function. If `summary = TRUE`
`SUM.S`	Numeric tibble. Summary for the mean, median, and a `confint / 100` confidence interval for a grid of the survival function. If `summary = TRUE`

References

- Nieto-Barajas, L. E. (2003). Discrete time Markov gamma processes and time dependent covariates in survival analysis. *Bulletin of the International Statistical Institute 54th Session*. Berlin. (CD-ROM).

- Nieto-Barajas, L. E. & Walker, S. G. (2002). Markov beta and gamma processes for modelling hazard rates. *Scandinavian Journal of Statistics* **29**: 413-424.

See Also

GaMRes, CGaMRes, CGaPlotDiag, GaPlotDiag

Examples

```
## Simulations may be time intensive. Be patient.

## Example 1
#  data(gehan)
#  timesG <- gehan$time[gehan$treat == "6-MP"]
#  deltaG <- gehan$cens[gehan$treat == "6-MP"]
#  GEX1 <- GaMRes(timesG, deltaG, K = 8, iterations = 3000)
#  GaPloth(GEX1)


## Example 2
```

```
# data(leukemiaFZ)
# timesFZ <- leukemiaFZ$time
# deltaFZ <- leukemiaFZ$delta
# GEX2 <- GaMRes(timesFZ, deltaFZ, type.c = 4)
# GaPloth(GEX2)
```

`gehan` *Times of Remission of Leukemia Patients (Gehan)*

Description

Freireich et al. (1963) report the results of a clinical trial of a drug 6-mercaptopurine (6-MP) versus a placebo (control) in 42 children with acute leukemia. The trial was conducted at 11 American hospitals. The trial was conducted by matching pairs of patients at a given hospital by remission status (complete or partial) and randomizing within the pair to either a 6-MP or placebo maintenance therapy. Patients were followed until their leukemia returned (relapse) or until the end of the study (in weeks). The data was taken from Klein & Moeschberger (2003) and is contained in the `MASS` package.

Usage

```
data(gehan)
```

Format

A data frame with 42 observations containing:

`pair` Pair index.

`time` Remission time (weeks).

`cens` Status: 0=censored.

`treat` Treatment: control or 6-MP.

Source

Klein, J. P., & Moeschberger, M. L. (2003). Survival analysis: techniques for censored and truncated data. Springer Science & Business Media.

References

Freireich, E. J., et al. (1963). The effect of 6-mercaptopurine on the duration of steroid-induced remissions in acute leukemia: A model for evaluation of other potentially useful therapy. *Blood*, **21(6)**, 699-716.

Examples

```
## Gamma Process Example 1
#  data(gehan)
#  timesG <- gehan$time[gehan$treat == "6-MP"]
#  deltaG <- gehan$cens[gehan$treat == "6-MP"]
#  GEX1 <- GaMRes(timesG, deltaG, K = 8, iterations = 3000)
#  GaPloth(GEX1)
```

`KIDNEY` *Recurrent infection of kidney catheters*

Description

Data on the recurrent times to infection, at the point of insertion of the catheter, for kidney patients using portable dialysis equipment. Catheters may be removed for reasons other than infection, in which case the observation is censored. Each patient has exactly 2 observations. Only sex was kept as an explanatory variable.

Usage

```
KIDNEY
```

Format

A data frame with 38 rows and 6 variables:

id patient ID

t1,t2 times to infection

delta1,delta2 censorship indicators (1 = exact, 0 = right-censored)

sex 0 = female, 1 = male

Source

https://www.mayo.edu/research/documents/kidneyhtml/doc-10027569

`leukemiaFZ` *Survival Times of Patients with Leukemia (Feigl and Zelen)*

Description

Survival times of 33 patients with leukemia (Feigl and Zeelen, 1965). Times are measured in weeks from diagnosis. Reported covariates are white blood cell counts (WBC) and a binary variable AG that indicates a positive or negative test related to the white blood cell characteristics. Three of the observations were censored. The data was taken from Lawless (2003).

Usage

```
data(leukemiaFZ)
```

Format

A data frame with 33 observations on the following 4 variables.

`time` Weeks from diagnosis.

`delta` Status indicator: 0=censored.

`AG` Indicates a positive or negative test related to the white blood cell characteristics. (1=AG-positive, 2=AG-negative).

`wbc` White blood cell counts in thousands (reported covariates).

Source

Lawless, J.F. (2003). Statistical Models and Methods for Lifetime Data. Wiley: New Jersey.

References

Feigl, P. and Zelen, M. (1965). Estimation of Exponential Survival Probabilities with Concomitant Information. *Biometrics* **21**, 826-838.

Examples

```
## Cox-Gamma Process Example 1
#  data(leukemiaFZ)
#  leukemia1 <- leukemiaFZ
#  leukemia1$wbc <- log(leukemiaFZ$wbc)
#  CGEX1 <- CGaMRes(data = leukemia1, K = 10, iterations = 100, thinning = 1)
```

`psych` *Death Times of Psychiatric Patients (Woolson)*

Description

Woolson (1981) has reported survival data on 26 psychiatric inpatients admitted to the University of Iowa hospitals during the years 1935-1948. This sample is part of a larger study of psychiatric inpatients discussed by Tsuang and Woolson (1977) and it is contained in the `KMsurv` package.

Usage

```
data(psych)
```

Format

A data frame with 26 observations containing:

`sex` Patient sex: 1=male, 2=female.

`age` Age at first admission to the hospital.

`time` Number of years of follow-up.

`death` Patient status at the follow-up time: 0=alive, 1=dead.

Source

Klein, J. P., and Moeschberger, M. L. (2003). Survival analysis: techniques for censored and truncated data. Springer Science & Business Media.

References

Tsuang, M. T. and Woolson, R. F. (1977). Mortality in Patients with Schizophrenia, Mania and Depression. *British Journal of Psychiatry*, **130**: 162-166.

Woolson, R. F. (1981). Rank Tests and a One-Sample Log Rank Test for Comparing Observed Survival Data to a Standard Population. *Biometrics* **37**: 687-696.

Examples

```
## Beta Process Example 1
## Example 1
#  data(psych)
#  timesP <- psych$time
#  deltaP <- psych$death
#  BEX1 <- BeMRes(timesP, deltaP, iterations = 3000, burn.in = 300, thinning = 1)
```

Index