Package: pcensmix (via r-universe)

June 2, 2026

Type Package

Title Model Fitting to Progressively Censored Mixture Data

Version 1.2-1

Depends R (>= 3.3.3), stats

Imports utils

Description Functions for generating progressively Type-II censored data in a mixture structure and fitting models using a constrained EM algorithm. It can also create a progressive Type-II censored version of a given real dataset to be considered for model fitting.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Author Lida Fallah [aut, cre], John Hinde [aut]

Maintainer Lida Fallah <l.fallah22@gmail.com>

Repository https://lida-fallah.r-universe.dev

Date/Publication 2017-07-24 18:01:10 UTC

RemoteUrl https://github.com/cran/pcensmix

RemoteRef HEAD

RemoteSha 27679508fc3ec7609742d8a6ce01a15ba0c306c4

Contents

`pcensmix-package` *Generate and fit a model to censored mixture data*

Description

This package provides functions to generate two-component mixture data from various different mixture distribution, generate progressive Type-II censored data in a mixture structure and fit a normal mixture model using a constrained EM algorithm. In addition, it can create a progressive Type-II censored version of a given real dataset and fit a normal mixture model to. Main functions are pcgen, pcensmixSim and pcensmixR. Example datasets are included for illustration.

Details

Package: pcensmix

Type: Package

Version: 1.2-1

Date: 2017-07-24

License: GPL (>= 2)

Author(s)

Lida Fallah <l.fallah22@gmail.com> and John Hinde

Maintainer: Lida Fallah <l.fallah22@gmail.com>

References

Aitkin, M., Francis, B., Hinde, J. and Darnell, R., (2009). Statistical Modelling in R. Oxford: Oxford University Press.

Balakrishnan, N. and Aggarwala, R., (2000). Progressive Censoring: Theory, Methods, and Applications. Springer Science & Business Media.

Hathaway, R.J., (1985). A constrained formulation of maximum-likelihood estimation for normal mixture distributions. The Annals of Statistics, 795-800.

McLachlan, G. and Krishnan, T., (2007). The EM Algorithm and Extensions. John Wiley & Sons.

McLachlan, G. and Peel, D., (2004). Finite Mixture Models. John Wiley & Sons.

blood *Blood pressure of mine workers.*

Description

A dataframe containing the blood pressure of a population of mine workers in Ghana.

Usage

```
blood
```

Format

A data frame with 495 rows and 3 variables

- Systolic.BP. Systolic blood pressure
- Diastolic.BP. Diastolic blood pressure
- Amplitude. Amplitude

References

Boehning, D., (2000). Computer-assisted Analysis of Mixtures and Applications: Meta-analysis, Disease mapping and Others. CRC press.

Gunga, H.C., Forson, K., Amegby, N. and Kirsch, K., (1991). Lebensbedingungen und gesundheitszustand von berg-und fabrikarbeitern im tropischen regenwald von Ghana. Arbeitsmedizin Sozialmedizin Praventivmediz in, 17-25.

insulate *Failure times of insulating fluid.*

Description

A dataset containing measurements of 19 failure times (in minutes) for an insulating fluid between two electrodes subject to a voltage of 34 KV, see Nelson (1982).

Usage

```
insulate
```

Format

A vector of length 19.

References

Nelson, W., (1982). Applied Life Data Analysis. Wiley, New York.

`mixgen` *Generating Mixture Datasets*

Description

This function generates two-component mixture data from a various different mixture distributions.

Usage

```
mixgen(N, dist1, dist2, control)
```

Arguments

`N`	population size.
`dist1, dist2`	respective distributions of the first and second mixture components to be simulated from. For Normal, Log-normal, Weibull, Gamma, Cauchy and Beta distributions, they must be provided as `'norm'`, `'lnorm'`, `'weibull'`, `'gamma'`, `'cauchy'` and `'beta'` respectively. The Exponential distribution can be included as a Gamma distribution with scale parameter one.
`control`	a list of parameters for controlling the simulation process. This includes parameters of the first and second mixture component distributions and the mixture proportion π which should be provided in the order mentioned, i.e., parameters for the first component come first, those of the second component come after and then the value of the mixing proportion. All values should be provided in a list of numerics. Note parameters for each component distribution should be added in the order as required in their generator functions, see `rnorm`, `rlnorm`, `rweibull`, `rgamma`, `rcauchy` and `rbeta`.

Details

It generates a two-component mixture dataset from a density function

$$\pi f_1 + (1 - \pi) f_2,$$

where f_1 and f_2 are the first and the second mixture component distributions respectively.

Value

An object of class `data.frame` containing the following information:

`z`	mixture data
`label`	component indicator

Author(s)

Lida Fallah, John Hinde

Maintainer: Lida Fallah <l.fallah22@gmail.com>

See Also

pcgen

Examples

```
## Generate a sample from a two component Normal-Weibull mixture distribution
## with mixing components as N(12, 2) and Weibull(15, 4), mixing proportion 0.3
## and size of N = 20.

mixture<- mixgen(N = 20, dist1 = 'norm', dist2 = 'weibull', control = list(12, 2, 15, 4, 0.3))
```

---

pcensmixR *Fitting a Normal Mixture Model to a Real Progressive Type-II Censored Mixture Data Using EM Algorithm*

Description

This function uses a two-layer EM algorithm to fit a mixture model to progressive Type-II censored mixture data by estimating the latent mixture components and the censored data.

Usage

```
pcensmixR(Pdat, ...)

## S3 method for class 'pcgen'
pcensmixR(Pdat, start, iteration = 1e+05, INERiter = 20,
  warn = FALSE, ...)
```

Arguments

Pdat	an object of class "pcgen" created by function pcgen or a two-column matrix (or data.frame) with first column giving a vector of censored version of a two-component mixed normal data, and the other one indicating the censoring status associated with them (1 if not censored, otherwise zero).
...	additinal arguments to pass by.
start	a numeric vector; used as starting values for the EM algorithm.
iteration	the maximum number of required iteration for the EM algorithm until convergence– default value is 1e+05.
INERiter	the maximum number of required iteration for the second EM algorithm– default is 20.
warn	logical. shows warning messages if TRUE, if there is any– default is FALSE.

Details

This function fits a two-component normal mixture model to a given progressive Type-II censored data.

It uses a two-layer EM algorithm for fitting the model. Generally speaking, the first layer estimates the mixture component latent variables, in the E-step, by finding their conditional expected values given the current parameter estimates and the data; and the second layer consists of another EM algorithm to estimate the missing censored data and eventually the parameters of interest. The layers are repeated until convergence achieved.

Value

`pcensmixR` gives an object of class `data.frame` containing the following components:

`muhat1`, `sigmahat1`
: component one parameter estimates ($\hat{\mu}_1$, $\hat{\sigma}_1$)

`muhat2`, `sigmahat2`
: component two parameter estimates ($\hat{\mu}_2$, $\hat{\sigma}_2$)

`pihat`
: estimation of mixture proportion $\hat{\pi}$

`se.muhat1`, `se.sigmahat1`
: standard errors of $\hat{\mu}_1$ and $\hat{\sigma}_1$

`se.muhat2`, `se.sigmahat2`
: standard errors of $\hat{\mu}_2$ and $\hat{\sigma}_2$

`se.pihat`
: standard error of $\hat{\pi}$

`no.fails.comp1`, `no.fails.comp2`
: number of failures from each mixture component

`no.cens.comp1`, `no.cens.comp2`
: number of censored observations from each mixture component

`ll`
: log-likelihood value

Note

See `pcgen` for the definition of censored version of data.

Author(s)

Lida Fallah, John Hinde

Maintainer: Lida Fallah <l.fallah22@gmail.com>

See Also

`pcgen`, `pcensmixSim`

Examples

```
## Example 1: fit a mixture model to 'insulate' data
set.seed(107)
Pdat<- pcgen(r = 15, p = 0.6, data = insulate)
pcensmixR(Pdat, start = c(5, 3, 35, 20, 0.6))

## Not run:
## Example 2: fit a mixture model to 'Systolic blood pressure' data
set.seed(1010)
pcensmixR(Pdat = pcgen(360, 0.35, blood$Systolic.BP),
               start = c(120, 15, 150, 20, 0.6))
## End(Not run)
```

`pcensmixSim` *Fitting a Normal Mixture Model to a Simulated Progressive Type-II Censored Data Using EM Algorithm*

Description

This function fits a normal mixture model to progressive Type-II censored mixture data by dealing with the two aspects of missing data, latent mixture components and the censored data, using a maximum likelihood estimation through a constrained two-layer EM algorithm.

Usage

```
pcensmixSim(Pdat, ...)

## S3 method for class 'pcgen'
pcensmixSim(Pdat, r, p, param, iteration = 1e+05,
  INERiter = 20, ...)
```

Arguments

`Pdat`	an object of class "pcgen" created by function pcgen or a two-column matrix (or data.frame) with first column giving a vector of censored version of a two-component mixed normal data, and the other one indicating the censoring status associated with them (1 if not censored, otherwise zero).
`...`	additinal arguments to pass by.
`r`	desired number of failures to observe.
`p`	a parameter controlling the amount of censoring. The action of censoring individuals after each failure occurs with probabilty p from binomial distribution at each stage. If `p = 0`, there will be no censoring.
`param`	a numeric vector; used as starting values for the EM and simulating a new data to replace in case of happening singularity in the likelihood.

`iteration` the maximum number of required iteration for the EM algorithm until convergence– default value is 1e+05.

`INERiter` the maximum number of required iteration for the second EM algorithm– default is 20.

Details

This function fits a two-component normal mixture model to simulated progressive Type-II censored data with density function

$$\pi(\frac{1}{\sigma_1})\,\phi\Big[\frac{(z-\mu_1)}{\sigma_1}\Big] + (1-\pi)(\frac{1}{\sigma_2})\,\phi\Big[\frac{(z-\mu_2)}{\sigma_2}\Big]$$

where ϕ is the standard normal density.

It uses a constrained two-layer EM algorithm to deal with the two forms of missing data: the censored survival times and the mixture component labels. Given the EM algorithm is at a particular iteration: (i) first, in the E-step it obtains the mixture component indicator estimates given the current parameter estimates and the observed data. (ii) Next, for re-estimation of the unknown parameters, a new EM algorithm is nested in the M-step of the initial EM algorithm to deal with the estimation of the missing censored survival times and consequently building the maximum likelihood equations. These steps are repeated until the model converges.

Value

`pcensmixSim` gives an object of class `data.frame` containing the following components:

`muhat1`, `sigmahat1`
: component one parameter estimates ($\hat{\mu}_1$, $\hat{\sigma}_1$)

`muhat2`, `sigmahat2`
: component two parameter estimates ($\hat{\mu}_2$, $\hat{\sigma_2}$

`pihat` estimation of mixture proportion $\hat{\pi}$

`se.muhat1`, `se.sigmahat1`
: standard errors of $\hat{\mu}_1$ and $\hat{\sigma_1}$

`se.muhat2`, `se.sigmahat2`
: standard errors of $\hat{\mu}_2$ and $\hat{\sigma}_2$

`se.pihat` standard error of $\hat{\pi}$

`no.fails.comp1`, `no.fails.comp2`
: number of failures from each mixture component

`no.cens.comp1`, `no.cens.comp2`
: number of censored observations from each mixture component

`ll` log-likelihood value

`datachange_flag`
: `TRUE` if data has been replaced by a newly generated one

Note

- In fitting the model, to overcome the problem of singularity and model non-identifiability that might happen in some cases depending on the true means and standard deviations of the components, we use the constraint proposed by Hathaway (1985). Based on this, the ratios of the standard deviations are considered to be greater than a pre-fixed constant. We consider the constraints $\sigma_1/\sigma_2 > 0.1$ and $\sigma_2/\sigma_1 > 0.1$ which lead to obtain a parameter space with no singularities. In case this conditions doesn't hold, the data will be replaced by a new simulated one and `datachange_flag` will appear as `TRUE` in the output.
- See `pcgen` for the definition of censored version of data.

Author(s)

Lida Fallah, John Hinde

Maintainer: Lida Fallah <l.fallah22@gmail.com>

See Also

`pcgen`, `run_pcensmix`

Examples

```
## Not run:
set.seed(100)

Pdat<- pcgen(r = 60, p = 0.3, data = mixgen(N = 100, dist1 = 'norm',
                 dist2 = 'norm', control = list(12, 2, 14, 5, 0.35)))
pcensmixSim(Pdat, r = 60, p = 0.3, param=c(12, 2, 14, 5, 0.35))
## End(Not run)
```

---

pcgen *Creating a Progressively Type-II Censored Version of a Given Dataset*

Description

This function implements an algorithm for generating a progressive Type-II censored version of a specified dataset.

Usage

```
pcgen(r, p, data)
```

Arguments

r	desired number of failures to observe.
p	a parameter controlling the amount of censoring. The action of censoring individuals after each failure occurs with probabilty p from binomial distribution at each stage. If p = 0, no censoring will happen.
data	a numeric vector of a real dataset (mixture/not mixture) or an object of class data.frame generated by mixgen.

Details

It creates a progressive Type-II censored version of a given real dataset or a simulated dataset from mixgen. The output of this function can be passed as an argument to pcensmixR or pcensmixSim for the purpose of fitting a normal mixture model to the progressively censored dataset.

Value

An object of class "pcgen" containing the following information:

original_data	original mixture data
label	component membership indicator for the original simulated mixture data. This will not be returned if a real data has been used.
censored_version_of_data	progressive Type-II censored version of data, i.e., each observation is equal to the actual observed survival time in the event of failure and is equal to the latest observe failure time if it is associated to an unobserved censored observation. Notice that they are order statistics.
component_indicator	component indicator associated with the censored_verison_of_data. This will not be returned if a real data has been used.
censoring_indicator	censoring indicator associated with the censored_verison_of_data.

Note

See print.pcgen for printing data of class "pcgen".

Author(s)

Lida Fallah, John Hinde

Maintainer: Lida Fallah <l.fallah22@gmail.com>

See Also

mixgen, print.pcgen.

Examples

```
## 1. Generate a progressive Type-II censored data from a simulated mixture data with
## allowing for censoring with controlling parameters p = 0.3 and r = 12.
set.seed(0)
mixture <- mixgen(N = 20, dist1 = 'norm', dist2 = 'weibull', control = list(12, 2, 15, 4, 0.3))
Pdat0 <- pcgen(r = 12, p = 0.3, data = mixture)
print(Pdat0)


## 2. Examples of generating a progresively Type-II censored data

set.seed(0)
Pdat1 <- pcgen(r = 6, p = 0.3, data = insulate)
print(Pdat1)

set.seed(100)
Pdat2 <- pcgen(r = 260, p = 0.35, data = blood$Systolic.BP)
print(Pdat2)
```

`print.pcgen` *Print Method for pcgen Objects*

Description

This function prints the progressive censored data generated by the S3 class `pcgen`.

Usage

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

Arguments

`x`	object of class pcgen.
`...`	optional arguments to pass by.

Value

This function uses the generic function `print` to print the dataset of class "pcgen" in a nice format.

Examples

```
## Generate a two component normal mixture data,
Pdat <- pcgen(r = 80, p = 0.3, data = mixgen(N = 100, dist1 = 'norm',
             dist2 = 'norm', control = list(12, 2, 14, 4, 0.3)))
# and print it.
print(Pdat)
```

run_pcensmix *Generating Progressively Type-II Censored Mixture Data and Fitting a Model*

Description

This function implements an algorithm using the mixgen, pcgen and pcensmixSim functions to generate data and fit a model using EM algorithm with a specified number of iterations.

Usage

```
run_pcensmix(N, r, p, param, repetition = 100)
```

Arguments

N	population size.
r	required number of failures to observe.
p	a parameter controlling the amount of censoring. The action of censoring individuals after each failure occurs with probabilty p from a binomial distribution at each stage. If p = 0, there will be no censoring.
param	a numeric vector; used as starting values for the EM and simulating a new data to replace in case of happening singularity in the likelihood.
repetition	the required number of repetition of the algorithm– default is 100.

Value

It returns the parameter estimates given by pcensmixSim with the desired number of repetitions. In each repetition it generates a new normal mixture progressive Type-II censored dataset from the same true parameter values and fits a model.

Author(s)

Lida Fallah, John Hinde

Maintainer: Lida Fallah <l.fallah22@gmail.com>

See Also

pcgen, pcensmixSim, mixgen

Examples

```
## Not run:

## Example 1: with very well separated mixture components
set.seed(3)
f1 <- run_pcensmix(N = 160, r = 120, p = 0.3, param = c(10, 2, 25, 4, 0.3), repetition = 100)
colMeans(f1)
```

```
## Example 2.
set.seed(150)
f2 <- run_pcensmix(N = 160, r = 130, p = 0.35, param = c(10, 2, 17, 4, 0.3), repetition = 100)
colMeans(f2)

## Example 3.
set.seed(20)
f3 <- run_pcensmix(N = 160, r = 130, p = 0.3, param = c(20, 6, 22, 12, 0.6), repetition = 100)
colMeans(f3)


## End(Not run)
```

Index