Package ‘smicd’

January 10, 2021

Type  Package
Title  Statistical Methods for Interval-Censored Data
Version  1.1.2
Author  Paul Walter
Maintainer  Paul Walter <paul.walter@fu-berlin.de>

Description  
Functions that provide statistical methods for interval-censored (grouped) data. The package supports the estimation of linear and linear mixed regression models with interval-censored dependent variables. Parameter estimates are obtained by a stochastic expectation maximization algorithm. Furthermore, the package enables the direct (without covariates) estimation of statistical indicators from interval-censored data via an iterative kernel density algorithm. Survey and Organisation for Economic Co-operation and Development (OECD) weights can be included into the direct estimation (see, Walter, P. (2019) <doi:10.17169/refubium-1621>).

License  GPL-2
Encoding  UTF-8
LazyData  true
Suggests  knitr, rmarkdown, mlmRev, R.rsp, Kernelheaping
RoxygenNote  7.1.1
Imports  ineq, truncnorm, lme4, MuMIn, formula.tools, mvtnorm, Hmisc, laeken, weights, graphics
VignetteBuilder  R.rsp
NeedsCompilation  no
Repository  CRAN
Date/Publication  2021-01-10 20:30:02 UTC

R topics documented:

Exam ......................................................... 2
kdeAlgo ..................................................... 3
kdeAlgoObject ............................................. 5
plot.kdeAlgo ............................................. 6
Exam scores from inner London

Description

Exam scores of 4,059 students from 65 schools in Inner London, as in Exam.

Format

A data frame with 4059 observations with the following 10 variables:

- **school**: School ID - a factor.
- **examsc**: Exam score.
- **schgend**: School gender - a factor. Levels are mixed, boys, and girls.
- **schavg**: School average of intake score.
- **vr**: Student level Verbal Reasoning (VR) score band at intake - a factor. Levels are bottom 25%, mid 50%, and top 25%
- **intake**: Band of student’s intake score - a factor. Levels are bottom 25%, mid 50% and top 25%
- **standLRT**: Standardised LR test score.
- **sex**: Sex of the student - levels are F and M.
- **type**: School type - levels are Mxd and Sngl.
- **student**: Student id (within school) - a factor

References

**Estimation of Statistical Indicators from Interval-Censored Data**

**Description**

The function applies an iterative kernel density algorithm for the estimation of a variety of statistical indicators (e.g. mean, median, quantiles, gini) from interval-censored data. The estimation of the standard errors is facilitated by a non-parametric bootstrap.

**Usage**

```r
defense USAGE
  defense kdeAlgo(
    xclass,
    classes,
    threshold = 0.6,
    burnin = 80,
    samples = 400,
    bootstrap.se = FALSE,
    b = 100,
    bw = "nrd0",
    evalpoints = 4000,
    adjust = 1,
    custom_indicator = NULL,
    upper = 3,
    weights = NULL,
    oecd = NULL
  )
```

**Arguments**

- `xclass`: interval-censored values; factor with ordered factor values, as in `dclass`
- `classes`: numeric vector of classes; Inf as last value is allowed, as in `dclass`
- `threshold`: used for the Head-Count Ratio and Poverty Gap, default is 60% of the median e.g. `threshold=0.6`
- `burnin`: burn-in sample size, as in `dclass`
- `samples`: sampling iteration size, as in `dclass`
- `bootstrap.se`: if `TRUE` standard errors for the statistical indicators are estimated
- `b`: number of bootstrap iterations for the estimation of the standard errors
- `bw`: bandwidth selector method, defaults to "nrd0", as in `density`
- `evalpoints`: number of evaluation grid points, as in `dclass`
- `adjust`: the user can multiply the bandwidth by a certain factor such that `bw=adjust*bw` as in `density`
custom_indicator

A list of functions containing the indicators to be additionally calculated. Such functions must only depend on the target variable \( y \) and the threshold. For the estimation of weighted custom indicators the function must also depend on weights. Defaults to NULL.

upper

If the upper bound of the upper interval is \( \text{Inf} \) e.g. \((15000, \text{Inf})\), then \( \text{Inf} \) is replaced by \( 15000 \times \text{upper} \).

weights

Any kind of survey or design weights that will be used for the weighted estimation of the statistical indicators.

oeccd

Weights for equivalized household size.

Details

The statistical indicators are estimated using pseudo samples as proxy for the interval-censored variable. The object \( \text{resultX} \) returns the pseudo samples for each iteration step of the KDE-algorithm.

Value

An object of class "kdeAlgo" that provides estimates for statistical indicators and optionally, corresponding standard error estimates. Generic functions such as, \text{print}, and \text{plot} have methods that can be used to obtain further information. See \text{kdeAlgoObject} for a description of components of objects of class "kdeAlgo".

References


See Also

dclass, print.kdeAlgo, plot.kdeAlgo

Examples

```r
## Not run:
# Generate data
x <- rlnorm(500, meanlog = 8, sdlog = 1)
classes <- c(0, 500, 1000, 1500, 2000, 2500, 3000, 4000, 5000, 6000, 8000, 10000, 15000, Inf)
xclass <- cut(x, breaks = classes)
weights <- abs(rnorm(500, 0, 1))
oecd <- rep(seq(1, 6.9, 0.3), 25)

# Estimate statistical indicators with default settings
Indicator <- kdeAlgo(xclass = xclass, classes = classes)
```
# Include custom indicators
Indicator_custom <- kdeAlgo(
  xclass = xclass, classes = classes,
  custom_indicator = list(quant5 = function(y, threshold) {
    quantile(y, probs = 0.05)
  })
)

# Include survey and oecd weights
Indicator_weights <- kdeAlgo(
  xclass = xclass, classes = classes,
  weights = weights, oecd = oecd
)

## End(Not run)

---

### kdeAlgoObject

- **Fitted kdeAlgoObject**

---

**Description**

An object of class "kdeAlgo" that represents the estimated statistical indicators and the estimated standard errors. Objects of this class have methods for the generic functions `print` and `plot`.

**Value**

An object of class "kdeAlgo" is a list containing at least the following components.

- **Point_estimate**
  - the estimated statistical indicators: Mean, Gini, Head-Count Ratio, Quantiles (10%, 25%, 50%, 75%, 90%), Poverty-Gap, Quintile-Share Ratio and if specified the selected custom indicators.

- **Standard_Error**
  - if bootstrap.se = TRUE, the standard errors for the statistical indicator are estimated

- **Mestimates**
  - kde object containing the corrected density estimate, as in `dclass`

- **resultDensity**
  - estimated density for each iteration, as in `dclass`

- **resultX**
  - true latent values X estimates, as in `dclass`

- **xclass**
  - classified values; factor with ordered factor values, as in `dclass`

- **gridx**
  - grid on which density is evaluated, as in `dclass`

- **classes**
  - classes; Inf as last value is allowed, as in `dclass`

- **burnin**
  - burn-in sample size, as in `dclass`

- **samples**
  - sampling iteration size, as in `dclass`

- **Point_estimates.run**
  - the estimated statistical indicators: Mean, Gini, Head-Count Ratio, Quantiles (10%, 25%, 50%, 75%, 90%), Poverty-Gap, Quintile-Share Ratio and if specified the selected custom indicators for each iteration run of the KDE-algorithm
weights any kind of survey or design weights that will be used for the weighted estimation of the statistical indicators

upper if the upper bound of the upper interval is \( \text{Inf} \) e.g. \((15000, \text{Inf})\), then \( \text{Inf} \) is replaced by \( 15000 \times \text{upper} \)

References


See Also

smicd, dclass

plot.kdeAlgo

Plot Diagnostics for a kdeAlgo Object

Description

Plots the estimated density of the interval-censored variable. Also, convergence plots are given for all estimated statistical indicators. The estimated indicator is plotted for each iteration step of the KDE-algorithm. Furthermore, the average up to iteration step \( M \) is plotted (without the burn-in iterations). A vertical line indicates the end of the burn-in period. A horizontal line marks the value of the estimated statistical indicator

Usage

## S3 method for class 'kdeAlgo'
plot(x, indicator = NULL, ...)

Arguments

- \( x \) an object of type "kdeAlgo", typical result of kdeAlgo
- \( \text{indicator} \) a vector of indicator names specifying for which indicators convergence plots are plotted, e.g. \( \text{c("mean","gini")} \)
- \( \ldots \) optional arguments passed to generic function.

Value

Convergence and density plots.
plot.sem

See Also

kdeAlgoObject, kdeAlgo

plot.sem  Plot Diagnostics for sem Objects

Description

Available are convergence plots for the estimated fixed effects model parameters and the residual variance of the linear or linear mixed regression model. If the Box-Cox transformation is used for the transformation of the dependent variable, a convergence plot of the transformation parameter lambda is also available. In each of the convergence plots, the estimated parameter is plotted for each iteration step of the SEM-algorithm. Furthermore, the average up to iteration step M is plotted (without the burn-in iterations). A vertical line indicates the end of the burn-in period. A horizontal line marks the value of the estimated statistical indicator Furthermore, the estimated density of the simulated dependent variable from the last iteration step is plotted with a histogram of the interval-censored true dependent variable in the back.

Usage

## S3 method for class 'sem'
plot(x, ...)

Arguments

x  an object of type "sem", typical result of semLm or semLme.

...  optional arguments passed to generic function.

Value

Convergence and density plots.

References


See Also

semObject, semLm, semLme
Description

Basic information of a kdeAlgo object is printed.

Usage

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

Arguments

x an object of class "kdeAlgo"
...

optional arguments passed to generic function

See Also

kdeAlgoObject, kdeAlgo

Description

Basic information of a sem object is printed.

Usage

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

Arguments

x an object of class "sem".
...

optional arguments passed to generic function

See Also

semObject, semLm, semLme
**print.summary.sem**

*Prints a summary.sem Object*

**Description**

The elements described in summary.sem are printed.

**Usage**

```r
## S3 method for class 'summary.sem'
print(x, ...)
```

**Arguments**

- `x`: an object of class "summary.sem".
- `...`: additional arguments that are not used in this method.

**semLm**

*Linear Regression with Interval-Censored Dependent Variable*

**Description**

This function estimates the linear regression model when the dependent variable is interval-censored. The estimation of the standard errors is facilitated by a non-parametric bootstrap.

**Usage**

```r
semLm(
  formula,
  data,
  classes,
  burnin = 40,
  samples = 200,
  trafo = "None",
  adjust = 2,
  bootstrap.se = FALSE,
  b = 100
)
```
Arguments

formula an object of class formula, as in \texttt{lm}. The dependent variable is measured as interval-censored values; factor with ordered factor values.
data a data frame containing the variables of the model.
classes numeric vector of classes: -\texttt{Inf} as lower interval bound and \texttt{Inf} as upper interval bound is allowed. If the Box-Cox or logarithmic transformation is chosen, the minimum interval bound must be $\geq 0$.
burnin the number of burn-in iterations of the SEM-algorithm.
samples the number of additional iterations of the SEM-algorithm for parameter estimation.
trafo transformation of the dependent variable to fulfill the model assumptions
  \begin{itemize}
    \item "log" for Logarithmic transformation
    \item "bc" for Box-Cox transformation
  \end{itemize}
default is "None". Transformations can only be used if the minimum interval bound is $\geq 0$.
adjust extends the number of iteration steps of the SEM-algorithm for finding the optimal lambda of the Box-Cox transformation. The number of iterations is extended in the following way: \((\text{burnin}+\text{samples}) \times \text{adjust}\).
bootstrap.se if \texttt{TRUE} standard errors of the regression parameters are estimated.
b number of bootstrap iterations for the estimation of the standard errors.

Details

The model parameters are estimated using pseudo samples as a proxy for the interval-censored dependent variable. The object \texttt{pseudo.y} returns the pseudo samples of each iteration step of the SEM-algorithm.

Value

An object of class "sem" that provides parameter estimates for linear regression models with interval-censored dependent variable. Generic functions such as, \texttt{print}, \texttt{plot}, and \texttt{summary} have methods that can be used to obtain further information. See \texttt{semObject} for a description of the components of objects of class "sem".

References


See Also

\texttt{lm, print.sem, plot.sem, summary.sem}
semLme

Examples

```r
## Not run:
# Load and prepare data
data <- Exam
classes <- c(1, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.7, 8.5, Inf)
data$exams$class <- cut(data$exams$eb, classes)

# Run model with default settings
model <- semLm(
  formula = exams$class ~ standLRT + schavg, data = data,
  classes = classes
)
summary(model)

## End(Not run)
```

semLme

Linear Mixed Regression with Interval-Censored Dependent Variable

Description

This function estimates the linear mixed regression model when the dependent variable is interval-censored. The estimation of the standard errors is facilitated by a parametric bootstrap.

Usage

```r
semLme(
  formula,
  data,
  classes,
  burnin = 40,
  samples = 200,
  trafo = "None",
  adjust = 2,
  bootstrap.se = FALSE,
  b = 100
)
```

Arguments

- `formula` a two-sided linear formula object describing both the fixed-effects and random-effects part of the model, with the response on the left of a ~ operator and the terms, separated by + operators, on the right. Random-effects terms are distinguished by vertical bars (|) separating expressions for design matrices from grouping factors, as in `lmer`. Note: Only models with a maximum of one random intercept and one random slope are implemented at this point (e.g. `y ~ x`
+ (1| ID), or y ~ x + (x| ID). The dependent variable is measured as interval-censored values; factor with ordered factor values.

data: a data frame containing the variables of the model.

classes: numeric vector of classes; -Inf as lower interval bound and Inf as upper interval bound is allowed. If the Box-Cox or logarithmic transformation is chosen, the minimum interval bound must be \( \geq 0 \).

burnin: the number of burn-in iterations of the SEM-algorithm.
samples: the number of additional iterations of the SEM-algorithm for parameter estimation.

trafo: transformation of the dependent variable to fulfil the model assumptions.
   - "log" for Logarithmic transformation
   - "bc" for Box-Cox transformation
   default is "None". Transformations can only be used if the minimum interval bound is \( \geq 0 \).

adjust: extends the number of iteration steps of the SEM-algorithm for finding the optimal lambda of the Box-Cox transformation. The number of iterations is extended in the following way: (burnin+samples)*adjust

bootstrap.se: if TRUE standard errors of the regression parameters are estimated.
b: number of bootstrap iterations for the estimation of the standard errors.

Details
The model parameters are estimated using pseudo samples of the interval-censored dependent variable. The object \( \text{pseudo.y} \) returns the pseudo samples of each iteration step of the SEM-algorithm.

Value
An object of class "sem" that provides parameter estimated for linear regression models with interval-censored dependent variable. Generic functions such as, print, plot, and summary have methods that can be used to obtain further information. See \( \text{semObject} \) for descriptions of components of objects of class "sem".

References

See Also
lmer, print.sem, plot.sem, summary.sem

Examples
```r
## Not run:
# Load and prepare data
data <- Exam
classes <- c(1, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.7, 8.5, Inf)
```
data$examsc.class <- cut(data$examsc, classes)

# Run model with random intercept and default settings
model1 <- semLme(
  formula = examsc.class ~ standLRT + schavg + (1 | school),
  data = data, classes = classes
)
summary(model1)

## End(Not run)

---

**semObject**

**Fitted semObject**

**Description**

An object of class "sem" that represents the estimated model parameters and standard errors. Objects of this class have methods for the generic functions print, plot and summary.

**Value**

An object of class "sem" is a list containing the following components. Some parameters are only estimated for linear mixed regression models (and vice versa).

- **pseudo.y**: a matrix containing the pseudo samples of the interval-censored variable from each iteration step
- **coef**: the estimated regression coefficients (fixed effects)
- **ranef**: the estimated regression random effects
- **sigmae**: estimated variance $\sigma_e$
- **VaVoc**: estimated covariance matrix of the random effects
- **se**: bootstrapped standard error of the coefficients
- **ci**: bootstrapped 95% confidence interval of the coefficients
- **lambda**: estimated lambda for the Box-Cox transformation
- **bootstraps**: number of bootstrap iterations for the estimation of the standard errors
- **r2**: estimated coefficient of determination
- **r2m**: estimated marginal coefficient of determination for generalized mixed-effect models, as in r.squaredGLMM
- **r2c**: estimated conditional coefficient of determination for generalized mixed-effect models, as in r.squaredGLMM
- **icc**: estimated interclass correlation coefficient
- **adj.r2**: estimated adjusted coefficient of determination
formula an object of class formula, as in `lm` or `lmer`
transformation the specified transformation "log" for logarithmic and "bc" for Box-Cox
n.classes the number of classes, the dependent variable is censored to
conv.coef estimated coefficients for each iteration step of the SEM-algorithm
conv.sigmoe estimated variance $\sigma_e$ for each iteration step of the SEM-algorothm
conv.VaCov estimated covariance matrix of the random effects for each iteration step of the SEM-algorithm
conv.lambda estimated lambda for the Box-Cox transformation for each iteration step of the SEM-algorithm
b.lambda the number of burn-in iteration the SEM-algorithm used to estimate lambda
m.lambda the number of additional iteration the SEM-algorithm used to estimate lambda
burnin the number of burn-in iterations of the SEM-algorithm
samples the number of additional iterations of the SEM-algorithm
classes specified intervals
original.y the dependent variable of the regression model measured on an interval-censored scale
call the function call

References


See Also

`smicd, lm, lmer, r.squaredGLMM`

**smicd**

*Statistical Methods for Interval Censored (Grouped) Data*

**Description**

The package **smicd** supports the estimation of linear and linear mixed regression models (random slope and random intercept models) with interval censored dependent variable. Parameter estimates are obtain by a stochastic expectation maximization (SEM) algorithm (Walter, 2019). Standard errors are estimated by a non-parametric bootstrap in the linear regression model and by a parametric bootstrap in the linear mixed regression model. To handle departures from the model assumptions transformations (log and Box-Cox) of the interval censored dependent variable are incorporated into the algorithm (Walter, 2019). Furthermore, the package **smicd** has implemented a non-parametric kernel density algorithm for the direct (without covariates) estimation of statistical indicators from interval censored data (Walter, 2019; Gross et al., 2017). The standard errors of the statistical indicators are estimated by a non-parametric bootstrap.
Details

The two estimation functions for the linear and linear mixed regression model are called `sem.Lm` and `sem.Lme`. So far, only random intercept and random slope models are implemented. For both functions, the following methods are available: `summary.sem`, `print.sem` and `plot.sem`.

The function for the direct estimation of statistical indicators is called `kdeAlgo`. The following methods are available: `print.kdeAlgo` and `plot.kdeAlgo`.

An overview of all currently provided functions can be requested by `library(help=smicd)`.

References


---

### summary.sem

**Summarizing Linear and Linear Mixed Models estimated with the SEM**

**Description**

summary method for class "sem".

**Usage**

```r
## S3 method for class 'sem'
summary(object, ...)
```

**Arguments**

- `object`: an object of class "sem".
- `...`: additional arguments that are not used in this method.

**Value**

an object of type "summary.sem" with following components:

- `call`: a list containing an image of the function call that produced the object.
- `coefficients`: a table that returns the estimation parameters and the standard errors and confidence intervals in case that the standard errors are estimated.
- `standard errors`: bootstrapped standard errors
confidence intervals
  bootstrapped confidence intervals

two R² measures  a multiple and adjusted R-squared in case of an object of class "sem","lm" and
a marginal and conditional R-squared in case of an object of class "sem","lme"
Index

dclass, 3–6
density, 3

Exam, 2, 2

kdeAlgo, 3, 6–8, 15
kdeAlgoObject, 4, 5, 7, 8

lm, 10, 14
lmer, 11, 12, 14

plot, 4, 5, 10, 12, 13
plot.kdeAlgo, 4, 6, 15
plot.sem, 7, 10, 12, 15
print, 4, 5, 10, 12, 13
print.kdeAlgo, 4, 8, 15
print.sem, 8, 10, 12, 15
print.summary.sem, 9

r.squaredGLMM, 13, 14

semLm, 7, 8, 9, 15
semLme, 7, 8, 11, 15
semObject, 7, 8, 10, 12, 13
smicd, 6, 14, 14
summary, 10, 12, 13
summary.sem, 10, 12, 15, 15