Package ‘truncSP’

February 20, 2015

Type Package
Title Semi-parametric estimators of truncated regression models
Version 1.2.2
Date 2014-05-05
Author Anita Lindmark and Maria Karlsson, Department of Statistics, Umea University
Maintainer Anita Lindmark <anita.lindmark@stat.umu.se>
Description Semi-parametric estimation of truncated linear regression models
License GPL (&gt;= 2)
LazyLoad yes
Depends R(&gt;= 2.10), stats, methods, truncreg, boot
NeedsCompilation no
Repository CRAN
Date/Publication 2014-05-05 11:43:44

R topics documented:

  trun SP-package .................................................... 2
  lt ................................................................. 4
  lt-class ......................................................... 7
  lt.fit ............................................................. 9
  PM10 ............................................................. 10
  PM10trunc ....................................................... 11
  qme ............................................................. 12
  qme-class ...................................................... 15
  qme.fit .......................................................... 16
  stls ............................................................ 18
  stls-class ..................................................... 21
  stls.fit ......................................................... 22
  summary.lt-class ............................................. 24
  summary.qme-class ........................................... 25
  summary.stls-class .......................................... 26

Index 28
Estimators of semi-parametric truncated regression models

Description

Functions for estimation of semi-parametric linear regression models with truncated response variables (fixed truncation point). Estimation using the Symmetrically Trimmed Least Squares (STLS) estimator (Powell 1986), Quadratic Mode (QME) estimator (Lee 1993) and Left Truncated (LT) estimator (Karlsson 2006).

Details

Package: truncSP
Type: Package
Version: 1.2.2
Date: 2014-05-05
License: GPL (>=2)
LazyLoad: yes
Depends: R(>= 2.10), methods, truncreg, boot

These semi-parametric estimators provide an alternative to maximum likelihood estimators, which are sensitive to distributional misspecification (Davidson and MacKinnon, 1993, p 536). All three estimators use trimming of the conditional density of the error terms. STLS assumes symmetrically distributed error terms, while QME and LT have been shown to be consistent for estimation of the slope parameters under asymmetrically distributed errors as well (Laitila 2001 and Karlsson 2006). The functions in the package (qme, lt and stls), all use optim to maximize or minimize objective functions wrt the vector of regression coefficients in order to find estimates (Karlsson and Lindmark, 2014). As the covariance matrices of the estimators depend on the density of the error distribution, the estimation of these is complicated and bootstrap (as described in Karlsson 2004 and Karlsson and Lindmark 2014) is used in all three functions.

Author(s)

Anita Lindmark and Maria Karlsson, Department of Statistics, Umea University

Maintainer: Anita Lindmark <anita.lindmark@stat.umu.se>

References


See Also

*truncreg*, function for estimating models with truncated response variables by maximum likelihood assuming Gaussian errors

Examples

```r
## Simulate a data.frame (model with asymmetrically distributed errors)
n <- 10000
x1 <- runif(n,0,10)
x2 <- runif(n,0,10)
x3 <- runif(n,-5,5)
eps <- rexp(n,0.2)- 5
y <- 2-2*x1+x2+2*x3+eps
d <- data.frame(y=y,x1=x1,x2=x2,x3=x3)

## Use a truncated subsample
dtrunc <- subset(d, y>0)

## Use qme or lt to consistently estimate the slope parameters
qme(y~x1+x2+x3, dtrunc, point=0, direction="left", cval="ols", const=1, beta="ols", covar=FALSE)
lt(y~x1+x2+x3, dtrunc, point=0, direction="left", clower="ols", const=1, cupper=2, beta="ols", covar=FALSE)

## Simulate a data.frame (symmetrically distributed errors)
n <- 10000
x1 <- runif(n,0,10)
x2 <- runif(n,0,10)
x3 <- runif(n,-5,5)
y <- 1-2*x1+x2+2*x3+rnorm(n,0,2)
d <- data.frame(y=y,x1=x1,x2=x2,x3=x3)
```
## Use a truncated subsample

dtrunc <- subset(d, y>0)

## Use stls to estimate the model

stls(y~x1+x2+x3, dtrunc, point=0, direction="left", beta="ols", covar=FALSE)

---

**lt**  

**Estimation of truncated regression models using the Left Truncated (LT) estimator**

---

### Description

Estimates linear regression models with truncated response variables (fixed truncation point), using the LT estimator (Karlsson 2006).

### Usage

```r
lt(formula, data, point = 0, direction = "left", clower = "ml", const = 1, cupper = 2,
    beta = "ml", covar = FALSE, na.action, ...)
```

### Arguments

- `x, object` an object of class "lt"
- `formula` a symbolic description of the model to be estimated
- `data` an optional data frame
- `point` the value of truncation (the default is 0)
- `direction` the direction of truncation, either "left" (the default) or "right"
- `clower` the lower threshold value to be used when trimming the conditional density of the errors from below. The default is "ml" meaning that the residual standard deviation from fitting a maximum likelihood model for truncated regression, using `truncreg`, is used. Method "ols" uses the estimated residual standard deviation from a linear model fitted by `lm`. It is also possible to manually supply the threshold value by setting `clower` to be equal to a number or numeric vector of length one.
const a number that can be used to alter the size of the lower threshold. const=0.5 would give a lower threshold value that is half the original size. The default value is 1.

cupper number indicating what upper threshold to use when trimming the conditional density of the errors from above. The number is used to multiply the lower threshold value, i.e. if cupper=2 (the default value) the upper threshold value is two times larger than the lower threshold value.

beta the method of determining the starting values of the regression coefficients (See Details for more information):

• The default method is "ml", meaning that the estimated regression coefficients from fitting a maximum likelihood model for truncated regression, assuming Gaussian errors, are used. The maximum likelihood model is fitted using truncreg.

• Method "ols" means that the estimated regression coefficients from fitting a linear model with lm.

• The third option is to manually provide starting values as either a vector, column matrix or row matrix.

covar logical. Indicates whether or not the covariance matrix should be estimated. If TRUE the covariance matrix is estimated using bootstrap. The default number of replicates is 2000 but this can be adjusted (see argument ...). However, since the bootstrap procedure is time-consuming the default is covar=FALSE.

na.action a function which indicates what should happen when the data contain NAs.

digits the number of digits to be printed

level the desired level of confidence, for confidence intervals provided by summary.lt. A number between 0 and 1. The default value is 0.95.

... additional arguments. For lt the number of bootstrap replicates can be adjusted by setting R=the desired number of replicates. Also the control argument of optim can be set by control=list() (see Details for more information).

Details

Minimizes the objective function described in Karlsson (2006) wrt the vector of regression coefficients, in order to find the LT estimates. The minimization is performed by optim using the "Nelder–Mead" method, and a maximum number of iterations of 2000. The maximum number of iterations can be adjusted by setting control=list(maxit=...) (for more information see the documentation for optim).

It is recommended to use one of the methods for generating the starting values of the regression coefficients (see argument beta) rather than supplying these manually, unless one is confident that one has a good idea of what these should be. This because the starting values can have a great impact on the result of the minimization.

Note that setting cupper=1 means that the LT estimates will coincide with the estimates from the Quadratic Mode Estimator (see function qme). For more detailed information see Karlsson and Lindmark (2014).
Value

lt returns an object of class "lt".

The function summary prints a summary of the results, including two types of confidence intervals (normal approximation and percentile method). The generic accessor functions coef, fitted, residuals and vcov extract various useful features of the value returned by lt.

An object of class "lt", a list with elements:

- coefficients: the named vector of coefficients
- startcoef: the starting values of the regression coefficients used by optim
- cvalues: information about the thresholds used. The method and constant used and the resulting lower and upper threshold values.
- value: the value of the objective function corresponding to coefficients
- counts: number of iterations used by optim. See the documentation for optim for further details
- convergence: from optim. An integer code. 0 indicates successful completion. Possible error codes are 1 indicating that the iteration limit maxit had been reached. 10 indicating degeneracy of the Nelder–Mead simplex.
- message: from optim. A character string giving any additional information returned by the optimizer, or NULL.
- residuals: the residuals of the model
- fitted.values: the fitted values
- df.residual: the residual degrees of freedom
- call: the matched call
- covariance: if covar=TRUE, the estimated covariance matrix
- R: if covar=TRUE, the number of bootstrap replicates
- bootrepl: if covar=TRUE, the bootstrap replicates

Author(s)

Anita Lindmark and Maria Karlsson

References


See Also

- `lt.fit`, the function that does the actual fitting
- `qme`, for estimation of models with truncated response variables using the QME estimator
- `stls`, for estimation of models with truncated response variables using the STLS estimator
- `truncreg` for estimating models with truncated response variables by maximum likelihood, assuming Gaussian errors

Examples

```r
##Simulate a data frame (model with asymmetrically distributed errors)
n <- 10000
x1 <- runif(n,0,10)
x2 <- runif(n,0,10)
x3 <- runif(n,-5,5)
eps <- rexp(n,0.2)- 5
y <- 2-2*x1+x2+2*x3+eps
d <- data.frame(y=y,x1=x1,x2=x2,x3=x3)

## Use a truncated subsample
dtrunc <- subset(d, y>0)

## Use lt to consistently estimate the slope parameters
lt(y~x1+x2+x3, dtrunc, point=0, direction="left", clower="ml", const=1,
cupper=2, beta="ml", covar=FALSE)

## Example using data "PM10trunc"
data(PM10trunc)
ltpm10 <- lt(PM10~cars+temp+wind.speed+temp.diff+wind.dir+hour+day,
data=PM10trunc, point=2, control=list(maxit=2500))

summary(ltpm10)
```

Description

Documentation on S4 class "lt".

Objects from the Class

Objects from the class are usually obtained by a call to the function `lt`. 
Slots

call: Object of class "call" the function call
coefficients: Object of class "matrix" the estimated coefficients from fitting a model for truncated regression using the Quadratic Mode Estimator (QME)
startcoef: Object of class "matrix" the starting coefficients used when fitting the model
cvalues: Object of class "data.frame" containing information about the thresholds used
value: Object of class "numeric" the value of the objective function corresponding to coefficients
counts: Object of class "integer" number of iterations until convergence
convergence: Object of class "integer" indicating whether convergence was achieved
message: Object of class "character" a character string giving any additional information returned by the optimizer
residuals: Object of class "matrix" the residuals of the model
fitted.values: Object of class "matrix" the fitted values
df.residual: Object of class "integer" the residual degrees of freedom
covariance: Object of class "matrix" the estimated covariance matrix
bootrepl: Object of class "matrix" bootstrap replicates used to estimate the covariance matrix

Methods

coeff signature(object = "lt") extracts the coefficients of the model fitted using lt
fitted signature(object = "lt") extracts the fitted values of the model fitted using lt
print signature(x = "lt") print method
residuals signature(object = "lt") extracts the residuals of the model fitted using lt
summary signature(object = "lt") summary method
vcov signature(object = "lt") extracts the covariance matrix of the model fitted using lt

Author(s)
Anita Lindmark and Maria Karlsson

See Also
Function lt and class "summary.lt"

Examples
showClass("lt")
**lt.fit**

*Function for fitting LT*

**Description**

Function to find LT estimates of the regression coefficients for regression models with truncated response variables. Uses `optim`. Intended to be called through `lt`, not on its own, since `lt` also transforms data into the correct form etc.

**Usage**

```r
lt.fit(formula, mf, point, direction, bet, cl, cu, ...)```

**Arguments**

- `formula` a symbolic description of the model to be estimated
- `mf` the `model.frame` containing the variables to be used when fitting the model. `lt` transforms the model frame to the correct form before calling `lt.fit`. If `lt.fit` is called on its own the model frame needs to be transformed manually.
- `point` point of truncation
- `direction` direction of truncation
- `bet` starting values to be used by `optim`. Column matrix with p rows.
- `cl` lower threshold value to be used, number or numeric vector of length 1. (See `lt`, argument `clower`, for more information).
- `cu` upper threshold value to be used, number or numeric vector of length 1. (See `lt`, argument `cupper`, for more information).
- `...` additional arguments to be passed to `optim` (see the documentation for `lt` for further details).

**Value**

A list with components:

- `startcoef` the starting values of the regression coefficients used by `optim`
- `coefficients` the named vector of coefficients
- `counts` number of iterations used by `optim`. See the documentation for `optim` for further details
- `convergence` from `optim`. An integer code. 0 indicates successful completion. Possible error codes are 1 indicating that the iteration limit `maxit` had been reached. 10 indicating degeneracy of the Nelder–Mead simplex.
- `message` from `optim`. A character string giving any additional information returned by the optimizer, or `NULL`.
- `residuals` the residuals of the model
- `df.residual` the residual degrees of freedom
- `fitted.values` the fitted values
Author(s)

Anita Lindmark and Maria Karlsson

See Also

lt

Examples

```r
require(utils)
## Model frame
n <- 10000
x <- rnorm(n,0,2)
y <- 2+x+4*xnorm(n)
d <- data.frame(y=x, x=x)
dl0 <- subset(d, y>0)
mf <- model.frame(y=x, data=dl0)

## Starting values and threshold values
lmmod <- lm(data=mf)
bet <- lmmod$coef
bet <- matrix(bet)
c1 <- sqrt(deviance(lmmod)/df.residual(lmmod))
cu <- 2*c1

str(lt. <- lt.fit(y=x, mf, point=0, direction="left", bet, c1, cu))
```

PM10 Air pollution data

Description

The data are a subsample of 500 observations from a data set that originates in a study where air pollution at a road is related to traffic volume and meteorological variables, collected by the Norwegian Public Roads Administration. The response variable consists of hourly values of the logarithm of the concentration of PM10 (particles), measured at Alnabru in Oslo, Norway, between October 2001 and August 2003. (Source: Statlib)

Usage

`data(PM10)`

Format

A data frame with 500 observations on the following 8 variables.

<table>
<thead>
<tr>
<th>PM10</th>
<th>Hourly values of the logarithm of the concentration of PM10 (particles)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cars</td>
<td>The logarithm of the number of cars per hour</td>
</tr>
</tbody>
</table>
temp  Temperature 2 meters above ground (degree C)
wind.speed  Wind speed (meters/second)
temp.diff  The temperature difference between 25 and 2 meters above ground (degree C)
wind.dir  Wind direction (degrees between 0 and 360)
hour  Hour of day
day  Day number from October 1, 2001

Source
http://lib.stat.cmu.edu/, dataset PM10, submitted by Magne Aldrin on July 28, 2004

References

Examples
data(PM10)

PM10trunc  Air pollution data (Truncated)

Description
Dataset PM10, truncated from the left at variable value PM10 = 2 (8 percent truncation).

Usage
data(PM10trunc)

Format
A data frame with 460 observations on the following 8 variables.
PM10  Hourly values of the logarithm of the concentration of PM10 (particles). Left-truncated at point 2.
cars  The logarithm of the number of cars per hour
temp  Temperature 2 meters above ground (degree C)
wind.speed  Wind speed (meters/second)
temp.diff  The temperature difference between 25 and 2 meters above ground (degree C)
wind.dir  Wind direction (degrees between 0 and 360)
hour  Hour of day
day  Day number from October 1, 2001

Examples
data(PM10trunc)
Estimation of truncated regression models using the Quadratic Mode Estimator (QME)

Description
Estimation of linear regression models with truncated response variables (fixed truncation point), using the Quadratic Mode Estimator (QME) (Lee 1993 and Laitila 2001)

Usage
```r
qme(formula, data, point = 0, direction = "left", cval = "ml",
    const = 1, beta = "ml", covar = FALSE, na.action, ...)
```

Arguments
- `x, object` an object of class "qme"
- `formula` a symbolic description of the model to be estimated
- `data` an optional data frame
- `point` the value of truncation (the default is 0)
- `direction` the direction of truncation, either "left" (the default) or "right"
- `cval` the threshold value to be used when trimming the conditional density of the errors. The default is "ml" meaning that the estimated residual standard deviation from a maximum likelihood model for truncated regression, fitted using `truncreg`, is used. Method "ols" uses the residual standard deviation from fitting a linear model using `lm`. It is also possible to manually supply the threshold by setting `cval` to be equal to a number or numeric vector of length one.
- `const` a number that can be used to alter the size of the threshold value. `const=0.5` would give a threshold value that is half the original size. The default value is 1.
- `beta` the method of determining the starting values of the regression coefficients (See Details for more information):
- The default method is "ml", meaning that the estimated regression coefficients from fitting a maximum likelihood model for truncated regression, assuming Gaussian errors, are used. The maximum likelihood model is fitted using `truncreg`.
- Method "ols" means that the estimated regression coefficients from fitting a linear model with `lm` are used.
- The third option is to manually provide starting values as either a vector, column matrix or row matrix.

`covar` logical. Indicates whether or not the covariance matrix should be estimated. If TRUE the covariance matrix is estimated using bootstrap, as described in Karlsson (2004). The default number of replicates is 2000 but this can be adjusted (see argument ...). However, since the bootstrap procedure is time-consuming the default is `covar=FALSE`.

`na.action` a function which indicates what should happen when the data contain NAs.

`digits` the number of digits to be printed

`level` the desired level of confidence, for confidence intervals provided by `summary.qme`. A number between 0 and 1. The default value is \( \pi \times 0.95 \).

`...` additional arguments. For `qme` the number of bootstrap replicates can be adjusted by setting \( R=\) the desired number of replicates. Also the `control` argument of `optim` can be set by `control=list()` (for more information on this see Details).

**Details**

Finds the QME estimates of the regression coefficients by maximizing the objective function described in Lee (1993) wrt the vector of regression coefficients. The maximization is performed by `optim` using the "Nelder–Mead" method. The maximum number of iterations is set at 2000, but this can be adjusted by setting `control=list(maxit=...)` (for more information see the documentation for `optim`).

The starting values of the regression coefficients can have a great impact on the result of the maximization. For this reason it is recommended to use one of the methods for generating these rather than supplying the values manually, unless one is confident that one has a good idea of what the starting values should be. For more detailed information see Karlsson and Lindmark (2014).

**Value**

`qme` returns an object of class "qme".

The function `summary` prints a summary of the results, including two types of confidence intervals (normal approximation and percentile method). The generic accessor functions `coef`, `fitted`, `residuals` and `vcov` extract various useful features of the value returned by `qme`.

An object of class "qme", a list with elements:

- `coefficients` the named vector of coefficients
- `startcoef` the starting values of the regression coefficients used by `optim`
cval  information about the threshold value used. The method and constant value used and the resulting threshold value.

value  the value of the objective function corresponding to coefficients

counts  number of iterations used by \texttt{optim}. See the documentation for \texttt{optim} for further details

convergence  from \texttt{optim}. An integer code. 0 indicates successful completion. Possible error codes are 1 indicating that the iteration limit maxit had been reached. 10 indicating degeneracy of the Nelder–Mead simplex.

message  from \texttt{optim}. A character string giving any additional information returned by the optimizer, or NULL.

residuals  the residuals of the model

fitted.values  the fitted values

df.residual  the residual degrees of freedom

call  the matched call

covariance  if covar=TRUE, the estimated covariance matrix

R  if covar=TRUE, the number of bootstrap replicates

bootrepl  if covar=TRUE, the bootstrap replicates

Author(s)

Anita Lindmark and Maria Karlsson

References


See Also

\texttt{qme.fit}, the function that does the actual fitting

\texttt{lt}, for estimation of models with truncated response variables using the LT estimator
stls, for estimation of models with truncated response variables using the STLS estimator.

truncreg for estimating models with truncated response variables by maximum likelihood, assuming Gaussian errors.

Examples

```r
# Simulate a data.frame (model with asymmetrically distributed errors)
set.seed(1)
n <- 10000
x1 <- runif(n, 0, 10)
x2 <- runif(n, 0, 10)
x3 <- runif(n, -5, 5)
eps <- rexp(n, 0.2) - 5
y <- 2 - 2*x1 + x2 + 2*x3 + eps
d <- data.frame(y = y, x1 = x1, x2 = x2, x3 = x3)

dtrunc <- subset(d, y > 0)

# Use qme to consistently estimate the slope parameters
qme(y ~ x1 + x2 + x3, dtrunc, point = 0, direction = "left", cval = "ml", const = 1, beta = "ml", covar = FALSE)

# Example using data "PM10trunc"
data(PM10trunc)

qmePM10 <- qme(PM10 ~ cars + temp + wind + wind.dir + hour + day, data = PM10trunc, point = 2, control = list(maxit = 5000))

summary(qmePM10)
```

---

**qme-class**

**Class** "qme"

**Description**

Documentation on S4 class "qme".

**Objects from the Class**

Objects from the class are usually obtained by a call to the function `qme`.

**Slots**

- `call`: Object of class "call" the function call
- `coefficients`: Object of class "matrix" the estimated coefficients from fitting a model for truncated regression using the Quadratic Mode Estimator (QME)
- `startcoef`: Object of class "matrix" the starting coefficients used when fitting the model
cval: Object of class "data.frame" containing information about the threshold value used
value: Object of class "numeric" the value of the objective function corresponding to coefficients
counts: Object of class "integer" number of iterations until convergence
convergence: Object of class "integer" indicating whether convergence was achieved
message: Object of class "character" a character string giving any additional information re-
turned by the optimizer
residuals: Object of class "matrix" the residuals of the model
fitted.values: Object of class "matrix" the fitted values
df.residual: Object of class "integer" the residual degrees of freedom
covariance: Object of class "matrix" the estimated covariance matrix
bootrepl: Object of class "matrix" bootstrap replicates used to estimate the covariance matrix

Methods

coef signature(object = "qme"): extracts the coefficients of the model fitted using qme
fitted signature(object = "qme"): extracts the fitted values of the model fitted using qme
print signature(x = "qme"): print method
residuals signature(object = "qme"): extracts the residuals of the model fitted using qme
summary signature(object = "qme"): summary method
vcov signature(object = "qme"): extracts the covariance matrix of the model fitted using qme

Author(s)

Anita Lindmark and Maria Karlsson

See Also

Function qme and class "summary.qme"

Examples

showClass("qme")

qme.fit Function for fitting QME

Description

Function to find QME estimates of the regression coefficients for regression models with truncated
response variables. Uses optim. Intended to be called through qme, not on its own, since qme also
transforms data into the correct form etc.
Usage

qme.fit(formula, mf, point, direction, bet, cv, ...)

Arguments

formula a symbolic description of the model to be estimated
mf the model.frame containing the variables to be used when fitting the model. qme transforms the model frame to the correct form before calling qme.fit. If qme.fit is called on its own the model frame needs to be transformed manually.
point point of truncation
direction direction of truncation
bet starting values to be used by optim. Column matrix with p rows.
cv threshold value to be used, number or numeric vector of length 1. (See qme, argument cval, for more information).
... additional arguments to be passed to optim (see the documentation for qme for further details).

Value

a list with components:

startcoef the starting values of the regression coefficients used by optim
coefficients the named vector of coefficients
counts number of iterations used by optim. See the documentation for optim for further details
convergence from optim. An integer code. 0 indicates successful completion. Possible error codes are 1 indicating that the iteration limit maxit had been reached. 10 indicating degeneracy of the Nelder–Mead simplex.
message from optim. A character string giving any additional information returned by the optimizer, or NULL.
residuals the residuals of the model
df.residual the residual degrees of freedom
fitted.values the fitted values

Author(s)

Anita Lindmark and Maria Karlsson

See Also

qme
Examples

require(utils)
# Model frame
n <- 10000
x <- rnorm(n, 0, 2)
y <- 2 + x + x + rnorm(n)
d <- data.frame(y = y, x = x)
dl0 <- subset(d, y > 0)
mf <- model.frame(y = x, data = dl0)

# Starting values and threshold value
lmmod <- lm(data = mf)
bet <- lmmod$coef
bet <- matrix(bet)
cv <- sqrt(deviance(lmmod)/df.residual(lmmod))
str(qme. <- qme.fit(y = x, mf, point = 0, direction = "left", bet, cv))

stls

Estimation of truncated regression models using the Symmetrically Trimmed Least Squares (STLS) estimator

Description

Function for estimation of linear regression models with truncated response variables (fixed truncation point), using the STLS estimator (Powell 1986)

Usage

stls(formula, data, point = 0, direction = "left", beta = "ml",
     covar = FALSE, na.action, ...)

# S4 method for signature 'stls'
print(x, digits = max(3, getOption("digits") - 3), ...)

# S4 method for signature 'stls'
summary(object, level = 0.95, ...)

# S4 method for signature 'summary.stls'
print(x, digits = max(3, getOption("digits") - 3), ...)

# S4 method for signature 'stls'
coef(object, ...)

# S4 method for signature 'stls'
vcov(object, ...)

# S4 method for signature 'stls'
residuals(object, ...)

# S4 method for signature 'stls'
fitted(object, ...)
Arguments

- **x, object** an object of class "stls"
- **formula** a symbolic description of the model to be estimated
- **data** an optional data frame
- **point** the value of truncation (the default is 0)
- **direction** the direction of truncation, either "left" (the default) or "right"
- **beta** the method of determining the starting values of the regression coefficients (See Details for more information):
  - The default method is "ml", meaning that the estimated regression coefficients from fitting a maximum likelihood model for truncated regression, assuming Gaussian errors, are used. The maximum likelihood model is fitted using `truncreg`.
  - Method "ols" means that the estimated regression coefficients from fitting a linear model with `lm`.
  - The third option is to manually provide starting values as either a vector, column matrix or row matrix.
- **covar** logical. Indicates whether or not the covariance matrix should be estimated. If TRUE the covariance matrix is estimated using bootstrap. The default number of replicates is 2000 but this can be adjusted (see argument ...). However, since the bootstrap procedure is time-consuming the default is covar=FALSE.
- **na.action** a function which indicates what should happen when the data contain NAs.
- **digits** the number of digits to be printed
- **level** the desired level of confidence, for confidence intervals provided by summary.stls. A number between 0 and 1. The default value is 0.95.
- **...** additional arguments. For stls the number of bootstrap replicates can be adjusted by setting R=the desired number of replicates. Also the control argument of optim can be set by control=list() (for more information, see Details).

Details

Uses `optim` ("Nelder–Mead" method) to minimize the objective function described in Powell (1986) wrt the vector of regression coefficients in order to find the STLS estimates (see Karlsson and Lindmark 2014 for more detailed information and background). The maximum number of iterations is set at 2000, but this can be adjusted by setting control=list(maxit=...) (for more information see the documentation for optim).

As the starting values of the regression coefficients can have a great impact on the result of the minimization it is recommended to use one of the methods for generating these rather than supplying the values manually (unless one is confident that one has a good idea of what the starting values should be).

Value

**stls** returns an object of class "stls".
The function `summary` prints a summary of the results, including two types of confidence intervals (normal approximation and percentile method). The generic accessor functions `coef`, `fitted`, `residuals` and `vcov` extract various useful features of the value returned by `stls`.

An object of class "stls", a list with elements:

- `coefficients` - the named vector of coefficients
- `startcoef` - the starting values of the regression coefficients used by `optim`
- `value` - the value of the objective function corresponding to `coefficients`
- `counts` - number of iterations used by `optim`. See the documentation for `optim` for further details
- `convergence` - from `optim`. An integer code. 0 indicates successful completion. Possible error codes are 1 indicating that the iteration limit maxit had been reached. 10 indicating degeneracy of the Nelder–Mead simplex.
- `message` - from `optim`. A character string giving any additional information returned by the optimizer, or NULL.
- `residuals` - the residuals of the model
- `fitted.values` - the fitted values
- `df.residual` - the residual degrees of freedom
- `call` - the matched call
- `covariance` - if `covar=TRUE`, the estimated covariance matrix
- `R` - if `covar=TRUE`, the number of bootstrap replicates
- `bootrepl` - if `covar=TRUE`, the bootstrap replicates

**Author(s)**

Anita Lindmark and Maria Karlsson

**References**


**See Also**

- `stls.fit`, the function that does the actual fitting
- `qme`, for estimation of models with truncated response variables using the QME estimator
- `lt`, for estimation of models with truncated response variables using the LT estimator
- `truncreg` for estimating models with truncated response variables by maximum likelihood, assuming Gaussian errors
Examples

```r
# Simulate a data.frame
n <- 10000
x1 <- runif(n, 0, 10)
x2 <- runif(n, 0, 10)
x3 <- runif(n, -5, 5)
y <- 1 + 2 * x1 + x2 + 2 * x3 + rnorm(n, 0, 2)
d <- data.frame(y = y, x1 = x1, x2 = x2, x3 = x3)

# Use a truncated subsample
dtrunc <- subset(d, y > 0)

# Use stls to estimate the model
stls(y ~ x1 + x2 + x3, dtrunc, point = 0, direction = "left", beta = "ml", covar = FALSE)

# Example using data "PM10trunc"
data(PM10trunc)

stlspm10 <- stls(PM10 = cars + temp + wind.speed + temp.diff + wind.dir + hour + day, data = PM10trunc, point = 2)
summary(stlspm10)
```

---

**Class "stls"**

**Description**

Documentation on S4 class "stls".

**Objects from the Class**

Objects from the class are usually obtained by a call to the function `stls`.

**Slots**

- `call`: Object of class "call" the function call
- `coefficients`: Object of class "matrix" the estimated coefficients from fitting a model for truncated regression using the Quadratic Mode Estimator (QME)
- `startcoef`: Object of class "matrix" the starting coefficients used when fitting the model
- `value`: Object of class "numeric" the value of the objective function corresponding to coefficients
- `counts`: Object of class "integer" number of iterations until convergence
- `convergence`: Object of class "integer" indicating whether convergence was achieved
- `message`: Object of class "character" a character string giving any additional information returned by the optimizer
residuals: Object of class "matrix" the residuals of the model
fitted.values: Object of class "matrix" the fitted values
df.residual: Object of class "integer" the residual degrees of freedom
covariance: Object of class "matrix" the estimated covariance matrix
bootrepl: Object of class "matrix" bootstrap replicates used to estimate the covariance matrix

Methods

- **coef** signature(object = "stls"): extracts the coefficients of the model fitted using stls
- **fitted** signature(object = "stls"): extracts the fitted values of the model fitted using stls
- **print** signature(x = "stls"): print method
- **residuals** signature(object = "stls"): extracts the residuals of the model fitted using stls
- **summary** signature(object = "stls"): summary method
- **vcov** signature(object = "stls"): extracts the covariance matrix of the model fitted using stls

Author(s)

Anita Lindmark and Maria Karlsson

See Also

Function stls and class "summary.stls"

Examples

showClass("stls")

---

**stls.fit**

*Function for fitting STLS*

---

Description

Function that utilizes optim to find STLS estimates of the regression coefficients for regression models with truncated response variables. Intended to be called through stls, not on its own, since stls also transforms data into the correct form etc.

Usage

stls.fit(formula,mf, point, direction, bet, ...)
Arguments

- **formula**: a symbolic description of the model to be estimated
- **mf**: the model.frame containing the variables to be used when fitting the model. `stls` transforms the model frame to the correct form before calling `stls.fit`. If `stls.fit` is called on its own the model frame needs to be transformed manually.
- **point**: point of truncation
- **direction**: direction of truncation
- **bet**: starting values to be used by `optim`. Column matrix with p rows.
- **...**: additional arguments to be passed to `optim` (see the documentation for `stls` for further details).

Value

- a list with components:
  - **startcoef**: the starting values of the regression coefficients used by `optim`
  - **coefficients**: the named vector of coefficients
  - **counts**: number of iterations used by `optim`. See the documentation for `optim` for further details
  - **convergence**: from `optim`. An integer code. 0 indicates successful completion. Possible error codes are 1 indicating that the iteration limit maxit had been reached. 10 indicating degeneracy of the Nelder–Mead simplex.
  - **message**: from `optim`. A character string giving any additional information returned by the optimizer, or NULL.
  - **residuals**: the residuals of the model
  - **df.residual**: the residual degrees of freedom
  - **fitted.values**: the fitted values

Author(s)

Anita Lindmark and Maria Karlsson

See Also

`stls`

Examples

```r
require(utils)
###Model frame
n <- 10000
x <- rnorm(n,0,2)
y <- 2+x+4*xnorm(n)
d <- data.frame(y=x, x=x)
```
Summary lt-class

```r
dl0 <- subset(d, y>0)
mf <- model.frame(y~x, data=d10)

# Starting values
lmmmod <- lm(data=mf)
bet <- lmmmod$coef
bet <- matrix(bet)
str(stls <- stls.fit(y~x,mf,point=0,direction="left",bet))
```

---

Summary lt-class  

Class "summary.lt"

**Description**

Documentation on S4 class "summary.lt"

**Objects from the Class**

Objects from the class are usually obtained by a calling summary on an object of class "lt".

**Slots**

- `level`: Object of class "numeric" the level of confidence for confidence intervals
- `confint`: Object of class "matrix" confidence intervals for regression coefficients
- `bootconfint`: Object of class "matrix" bootstrap confidence intervals for regression coefficients
- `call`: Object of class "call" the function call
- `coefficients`: Object of class "matrix" the estimated coefficients from fitting a model for truncated regression using the Quadratic Mode Estimator (QME)
- `startcoef`: Object of class "matrix" the starting coefficients used when fitting the model
- `cvalues`: Object of class "data.frame" containing information about the threshold values used
- `value`: Object of class "numeric" the value of the objective function corresponding to coefficients
- `counts`: Object of class "integer" number of iterations until convergence
- `convergence`: Object of class "integer" indicating whether convergence was achieved
- `message`: Object of class "character" a character string giving any additional information returned by the optimizer
- `residuals`: Object of class "matrix" the residuals of the model
- `fitted.values`: Object of class "matrix" the fitted values
- `df.residual`: Object of class "integer" the residual degrees of freedom
- `covariance`: Object of class "matrix" the estimated covariance matrix
- `bootrepl`: Object of class "matrix" bootstrap replicates used to estimate the covariance matrix
summary.qme-class

Extends
Class "lt", directly.

Methods

  print signature(x = "summary.lt"): print method

Author(s)
Anita Lindmark and Maria Karlsson

See Also
Function lt and class "lt"

Examples

showClass("summary.lt")

summary.qme-class  Class "summary.qme"

Description
Documentation on S4 class "summary.qme"

Objects from the Class
Objects from the class are usually obtained by a calling summary on an object of class "qme".

Slots

  level: Object of class "numeric" the level of confidence for confidence intervals
  confint: Object of class "matrix" confidence intervals for regression coefficients
  bootconfint: Object of class "matrix" bootstrap confidence intervals for regression coefficients
  call: Object of class "call" the function call
  coefficients: Object of class "matrix" the estimated coefficients from fitting a model for truncated regression using the Quadratic Mode Estimator (QME)
  startcoef: Object of class "matrix" the starting coefficients used when fitting the model
  cval: Object of class "data.frame" containing information on the threshold value used
  value: Object of class "numeric" the value of the objective function corresponding to coefficients
  counts: Object of class "integer" number of iterations until convergence
  convergence: Object of class "integer" indicating whether convergence was achieved
  message: Object of class "character" a character string giving any additional information returned by the optimizer
Summary of Class "summary.stls"

Documentation on S4 class "summary.stls"

Objects from the Class

Objects from the class are usually obtained by a calling summary on an object of class "stls".

Slots

- **level**: Object of class "numeric" the level of confidence for confidence intervals
- **confint**: Object of class "matrix" confidence intervals for regression coefficients
- **bootconfint**: Object of class "matrix" bootstrap confidence intervals for regression coefficients
- **call**: Object of class "call" the function call
- **coefficients**: Object of class "matrix" the estimated coefficients from fitting a model for truncated regression using the Quadratic Mode Estimator (QME)
- **startcoef**: Object of class "matrix" the starting coefficients used when fitting the model
value: Object of class "numeric" the value of the objective function corresponding to coefficients
counts: Object of class "integer" number of iterations until convergence
convergence: Object of class "integer" indicating whether convergence was achieved
message: Object of class "character" a character string giving any additional information returned by the optimizer
residuals: Object of class "matrix" the residuals of the model
fitted.values: Object of class "matrix" the fitted values
df.residual: Object of class "integer" the residual degrees of freedom
covariance: Object of class "matrix" the estimated covariance matrix
bootrepl: Object of class "matrix" bootstrap replicates used to estimate the covariance matrix

Extends
Class "stls", directly.

Methods
print signature(x = "summary.stls") print method

Author(s)
Anita Lindmark and Maria Karlsson

See Also
Function stls and class "stls"

Examples
showClass("summary.stls")
Index

*Topic classes
  lt-class, 7
  qme-class, 15
  stls-class, 21
  summary.lt-class, 24
  summary.qme-class, 25
  summary.stls-class, 26

*Topic datasets
  PM10, 10
  PM10trunc, 11

*Topic package
  truncsp-package, 2

*Topic regression
  lt, 4
  lt.fit, 9
  qme, 12
  qme.fit, 16
  stls, 18
  stls.fit, 22
  coef,lt-method (lt), 4
  coef,qme-method (qme), 12
  coef,stls-method (stls), 18
  fitted,lt-method (lt), 4
  fitted,qme-method (qme), 12
  fitted,stls-method (stls), 18
  lm, 4, 5, 12, 13, 19
  lt, 2, 4, 7–10, 14, 20, 24, 25
  lt-class, 7
  lt.fit, 7, 9
  optim, 2, 5, 6, 9, 13, 14, 16, 17, 19, 20, 22, 23

  PM10, 10, 11
  PM10trunc, 11
  print,lt-method (lt), 4
  print,qme-method (qme), 12
  print,stls-method (stls), 18
  print,summary.lt-method (lt), 4
  print,summary.qme-method (qme), 12
  print,summary.stls-method (stls), 18
  qme, 2, 5, 7, 12, 15–17, 20, 25, 26
  qme-class, 15
  qme.fit, 14, 16
  residuals,lt-method (lt), 4
  residuals,qme-method (qme), 12
  residuals,stls-method (stls), 18
  stls, 2, 7, 15, 18, 21–23, 26, 27
  stls-class, 21
  stls.fit, 20, 22
  summary,lt-method (lt), 4
  summary,qme-method (qme), 12
  summary,stls-method (stls), 18
  summary.lt, 8
  summary.lt-class, 24
  summary.qme, 16
  summary,qme-class, 25
  summary,stls, 22
  summary,stls-class, 26
  truncreg, 3–5, 7, 12, 13, 15, 19, 20
  truncSP (truncSP-package), 2
  truncSP-package, 2
  vcov,lt-method (lt), 4
  vcov,qme-method (qme), 12
  vcov,stls-method (stls), 18