

Package: truncSP (via r-universe)

May 10, 2026

Title Semi-Parametric Estimators of Truncated Regression Models

Version 1.2.4

Description Estimators for semi-parametric linear regression models with truncated response variables (fixed truncation point). The estimators implemented are the Symmetrically Trimmed Least Squares (STLS) estimator introduced by Powell (1986) <doi:10.2307/1914308>, the Quadratic Mode (QME) estimator introduced by Lee (1993) <doi:10.1016/0304-4076(93)90056-B>, and the Left Truncated (LT) estimator introduced by Karlsson (2006) <doi:10.1007/s00184-005-0023-x>.

Depends R (>= 3.4.0)

Imports stats, methods, truncreg, boot

License GPL (>= 2)

Encoding UTF-8

LazyData true

Collate 'PM10-data.R' 'PM10trunc-data.R' 'bootconfint.R'
'covar.boot.R' 'funcval.LT.R' 'mlcoef.R' 'funcLT.R'
'funcval.QME.R' 'funcQME.R' 'funcval.STLS.R' 'funcSTLS.R'
'lt-class.R' 'lt.fit.R' 'lt.R' 'summary.lt-class.R'
'ltmethods.r' 'qme-class.R' 'qme.fit.R' 'qme.R'
'summary.qme-class.R' 'qmemethods.r' 'stls-class.R'
'stls.fit.R' 'stls.R' 'summary.stls-class.R' 'stlsmethods.r'
'truncSP-package.R'

RoxygenNote 7.3.3

NeedsCompilation no

Author Anita Lindmark [aut, cre, cph], Maria Karlsson [aut]

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

Repository <https://anitalindmark.r-universe.dev>

Date/Publication 2025-09-11 14:35:31 UTC

RemoteUrl <https://github.com/cran/truncSP>

RemoteRef HEAD

RemoteSha 26e933ccba37a3f532bb4a8ba85a67416c58c4ed

Contents

truncSP-package	3
bootconfint	3
coef.lt	4
coef.qme	4
coef.stls	5
covar.boot	5
fitted.lt	6
fitted.qme	6
fitted.stls	7
funcLT	7
funcQME	8
funcSTLS	9
funcval.LT	10
funcval.QME	11
funcval.STLS	11
lt	12
lt-class	16
lt.fit	16
mlcoef	18
PM10	19
PM10trunc	20
print.lt	20
print.qme	21
print.stls	21
print.summary.lt	22
print.summary.qme	22
print.summary.stls	23
qme	23
qme-class	27
qme.fit	27
residuals.lt	29
residuals.qme	29
residuals.stls	30
stls	30
stls-class	33
stls.fit	34
summary.lt	35
summary.lt-class	36
summary.qme	37
summary.qme-class	38
summary.stls	38
summary.stls-class	39
vcov.lt	40
vcov.qme	40
vcov.stls	41

`truncSP-package`*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).

Author(s)

Maintainer: Anita Lindmark <anita.lindmark@umu.se> [copyright holder]

Authors:

- Maria Karlsson <maria.karlsson@umu.se>

`bootconfint`*Function to obtain bootstrap confidence intervals (percentile)*

Description

Function to obtain bootstrap confidence intervals (percentile)

Usage

```
bootconfint(bootrepl, level)
```

Arguments

<code>bootrepl</code>	bootstrap replicates
<code>level</code>	confidence level

Value

The function returns the bootstrap confidence intervals as a two-column matrix with one row per parameter

`coef.lt`*Function to extract model coefficients for objects of class "lt"*

Description

Function to extract model coefficients for objects of class "lt"

Usage

```
## S3 method for class 'lt'  
coef(object, ...)
```

Arguments

<code>object</code>	object of class "lt"
<code>...</code>	additional arguments

`coef.qme`*Function to extract model coefficients for objects of class "qme"*

Description

Function to extract model coefficients for objects of class "qme"

Usage

```
## S3 method for class 'qme'  
coef(object, ...)
```

Arguments

<code>object</code>	object of class "qme"
<code>...</code>	additional arguments

coef.stls	<i>Function to extract model coefficients for objects of class "stls"</i>
-----------	---

Description

Function to extract model coefficients for objects of class "stls"

Usage

```
## S3 method for class 'stls'  
coef(object, ...)
```

Arguments

object	object of class "stls"
...	additional arguments

covar.boot	<i>Function to obtain bootstrap covariance matrix</i>
------------	---

Description

Function to obtain bootstrap covariance matrix

Usage

```
covar.boot(bootrepl)
```

Arguments

bootrepl	bootstrap replicates
----------	----------------------

fitted.lt	<i>Function to obtain fitted values from objects of class "lt"</i>
-----------	--

Description

Function to obtain fitted values from objects of class "lt"

Usage

```
## S3 method for class 'lt'  
fitted(object, ...)
```

Arguments

object	object of class "lt"
...	additional arguments

fitted.qme	<i>Function to obtain fitted values from objects of class "qme"</i>
------------	---

Description

Function to obtain fitted values from objects of class "qme"

Usage

```
## S3 method for class 'qme'  
fitted(object, ...)
```

Arguments

object	object of class "qme"
...	additional arguments

fitted.stls	<i>Function to obtain fitted values from objects of class "stls"</i>
-------------	--

Description

Function to obtain fitted values from objects of class "stls"

Usage

```
## S3 method for class 'stls'
fitted(object, ...)
```

Arguments

object	object of class "stls"
...	additional arguments

funcLT	<i>Function to be used by boot to obtain bootstrap replicates of the parameters estimated with QME</i>
--------	--

Description

Function to be used by boot to obtain bootstrap replicates of the parameters estimated with QME

Usage

```
funcLT(
  mf,
  mf_orig,
  d,
  formula,
  beta,
  bet,
  clower,
  const,
  cupper,
  point,
  direction,
  control
)
```

Arguments

mf	the <code>model.frame</code> containing the variables to be used when fitting the model.
mf_orig	original model frame (before possible manipulations due to non-default truncation point or direction) used to ensure correct order of attributes in the call to <code>truncreg</code>
d	vector of indices to define the bootstrap samples
formula	a symbolic description of the model to be estimated
beta	the method of determining the starting values of the regression coefficients
bet	starting values to be used by <code>optim</code> . Only used if beta is numeric.
c_lower	the lower threshold value to be used when trimming the conditional density of the errors from below.
const	a number that can be used to alter the size of the lower threshold.
c_upper	number indicating what upper threshold to use when trimming the conditional density of the errors from above.
point	the point of truncation
direction	the direction of truncation
control	list of control parameters to be used by <code>optim</code>

Author(s)

Anita Lindmark and Maria Karlsson

funcQME	<i>Function to be used by boot to obtain bootstrap replicates of the parameters estimated with QME</i>
---------	--

Description

Function to be used by boot to obtain bootstrap replicates of the parameters estimated with QME

Usage

```
funcQME(
  mf,
  mf_orig,
  d,
  formula,
  beta,
  bet,
  cval,
  const,
  point,
  direction,
  control
)
```

Arguments

mf	the <code>model.frame</code> containing the variables to be used when fitting the model.
mf_orig	original model frame (before possible manipulations due to non-default truncation point or direction) used to ensure correct order of attributes in the call to truncreg
d	vector of indices to define the bootstrap samples
formula	a symbolic description of the model to be estimated
beta	the method of determining the starting values of the regression coefficients
bet	starting values to be used by optim . Only used if beta is numeric.
cval	the threshold value to be used when trimming the conditional density of the errors.
const	a number that can be used to alter the size of the threshold value.
point	the point of truncation
direction	the direction of truncation
control	list of control parameters to be used by optim

Author(s)

Anita Lindmark and Maria Karlsson

funcSTLS	<i>Function to be used by boot to obtain bootstrap replicates of the parameters estimated with STLS</i>
----------	---

Description

Function to be used by `boot` to obtain bootstrap replicates of the parameters estimated with STLS

Usage

```
funcSTLS(mf, mf_orig, d, formula, beta, bet, point, direction, control)
```

Arguments

mf	the <code>model.frame</code> containing the variables to be used when fitting the model.
mf_orig	original model frame (before possible manipulations due to non-default truncation point or direction) used to ensure correct order of attributes in the call to truncreg
d	vector of indices to define the bootstrap samples
formula	a symbolic description of the model to be estimated
beta	the method of determining the starting values of the regression coefficients
bet	starting values to be used by optim . Only used if beta is numeric.
point	the point of truncation
direction	the direction of truncation
control	list of control parameters to be used by optim

Author(s)

Anita Lindmark and Maria Karlsson

funcval.LT

LT objective function

Description

LT objective function

Usage

```
funcval.LT(bet, x, y, cl, cu)
```

Arguments

bet	parameter values. Column matrix with p rows.
x	model matrix
y	response variable, column matrix
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).

Value

Returns the value of the objective function for given parameter values

Author(s)

Anita Lindmark and Maria Karlsson

See Also

[lt](#)

funcval.QME	<i>QME objective function</i>
-------------	-------------------------------

Description

QME objective function

Usage

```
funcval.QME(bet, x, y, cv)
```

Arguments

bet	parameter values. Column matrix with p rows.
x	model matrix
y	response variable, column matrix
cv	threshold value (see qme , argument cval, for more information).

Value

Returns the value of the objective function for given parameter values

Author(s)

Anita Lindmark and Maria Karlsson

See Also

[qme](#)

funcval.STLS	<i>STLS objective function</i>
--------------	--------------------------------

Description

STLS objective function

Usage

```
funcval.STLS(bet, x, y)
```

Arguments

bet	parameter values. Column matrix with p rows.
x	model matrix
y	response variable, column matrix

Value

Returns the value of the objective function for given parameter values

Author(s)

Anita Lindmark and Maria Karlsson

See Also

[stls](#)

lt	<i>Estimation of truncated regression models using the Left Truncated (LT) estimator</i>
----	--

Description

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

Usage

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

Arguments

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"

c.lower	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 c.lower 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): <ul style="list-style-type: none"> • 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. 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.
...	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:

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

Author(s)

Anita Lindmark and Maria Karlsson

References

Karlsson, M. (2006) Estimators of regression parameters for truncated and censored data, *Metrika*, **63**, pp 329–341

Karlsson, M., Lindmark, A. (2014) `truncSP`: An R Package for Estimation of Semi-Parametric Truncated Linear Regression Models, *Journal of Statistical Software*, **57(14)**, pp 1–19, <https://www.jstatsoft.org/article/view/v057i14>

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

```
#####
#           Simulated data           #
#####
#Simulated data (asymmetrically distributed errors):
n <- 1000
set.seed(319993)
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.sim <- lt(y ~ x1 + x2 + x3, dtrunc, point = 0, direction = "left", clower = "ml",
            const = 1, cupper = 2, beta = "ml", covar = FALSE)

summary(lt.sim)

#####
# 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)
```

lt-class	<i>An S4 class for objects from the <code>lt</code> function</i>
----------	--

Description

An S4 class for objects from the `lt` function

Slots

`call` The matched call
`coefficients` A matrix
`startcoef` A matrix
`cvalues` A data frame
`value` Numeric
`counts` Integer
`convergence` Integer
`message` A character vector
`residuals` A matrix
`fitted.values` A matrix
`df.residual` Integer
`covariance` A matrix
`bootrepl` A matrix

lt.fit	<i>Function for fitting LT</i>
--------	--------------------------------

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

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

Value

A list with components:

startcoef	the starting values of the regression coefficients used by <code>optim</code>
coefficients	the named vector of coefficients
counts	number of iterations used by <code>optim</code> . See the documentation for <code>optim</code> for further details
convergence	from <code>optim</code> . An integer code. 0 indicates successful completion. Possible error codes are 1 indicating that the iteration limit <code>maxit</code> had been reached. 10 indicating degeneracy of the Nelder–Mead simplex.
message	from <code>optim</code> . 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

```

require(utils)
##Model frame
n <- 1000
x <- rnorm(n,0,2)
y <- 2+x+4*rnorm(n)

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

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

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

```

mlcoef	<i>Gives the starting coefficients from fitting a maximum likelihood model for truncated regression, assuming Gaussian errors</i>
--------	---

Description

Gives the starting coefficients from fitting a maximum likelihood model for truncated regression, assuming Gaussian errors. Uses the [truncreg](#) function from the `truncreg` package.

Usage

```
mlcoef(mf, mf_orig)
```

Arguments

mf	model frame
mf_orig	original model frame (before possible manipulations due to non-default truncation point or direction) used to ensure correct order of attributes in the call to truncreg

Author(s)

Anita Lindmark and Maria Karlsson

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.

`PM10` Hourly values of the logarithm of the concentration of PM10 (particles)

`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

Source

Aldrin, M. (2006) Improved predictions penalizing both slope and curvature in additive models, *Computational Statistics & Data Analysis*, **50**, pp 267–284

References

Aldrin, M. (2006) Improved predictions penalizing both slope and curvature in additive models, *Computational Statistics & Data Analysis*, **50**, pp 267–284

Examples

`data(PM10)`

PM10trunc	<i>Air pollution data (Truncated)</i>
-----------	---------------------------------------

Description

Dataset [PM10](#), truncated from the left at variable value $PM10 = 2$ (8 percent truncation).

Usage

```
data(PM10)
```

Format

A data frame with 460 observations on the following 8 variables.

PM10 Hourly values of the logarithm of the concentration of PM10 (particles)
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)
```

print.lt	<i>Print function for objects of class "lt"</i>
----------	---

Description

Print function for objects of class "lt"

Usage

```
## S3 method for class 'lt'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	object of class "lt"
digits	number of digits to be printed.
...	additional arguments

print.qme *Print function for objects of class "qme"*

Description

Print function for objects of class "qme"

Usage

```
## S3 method for class 'qme'  
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	object of class "qme"
digits	number of digits to be printed.
...	additional arguments

print.stls *Print function for objects of class "stls"*

Description

Print function for objects of class "stls"

Usage

```
## S3 method for class 'stls'  
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	object of class "stls"
digits	number of digits to be printed.
...	additional arguments

print.summary.lt *Print function for objects of class "summary.lt"*

Description

Print function for objects of class "summary.lt"

Usage

```
## S3 method for class 'summary.lt'  
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	object of class "summary.lt"
digits	number of digits to be printed.
...	additional arguments

print.summary.qme *Print function for objects of class "summary.qme"*

Description

Print function for objects of class "summary.qme"

Usage

```
## S3 method for class 'summary.qme'  
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	object of class "summary.qme"
digits	number of digits to be printed.
...	additional arguments

```
print.summary.stls      Print function for objects of class "summary.stls"
```

Description

Print function for objects of class "summary.stls"

Usage

```
## S3 method for class 'summary.stls'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	object of class "summary.stls"
digits	number of digits to be printed.
...	additional arguments

```
qme      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

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

Arguments

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 <code>truncreg</code> , is used. Method "ols" uses the residual standard deviation from fitting a linear model using <code>lm</code> . It is also possible to manually supply the threshold by setting <code>cval</code> 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. <code>const=0.5</code> 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): <ul style="list-style-type: none"> • 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 <code>truncreg</code>. • Method "ols" means that the estimated regression coefficients from fitting a linear model with <code>lm</code> 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 <code>covar=FALSE</code> .
na.action	a function which indicates what should happen when the data contain NAs.
...	additional arguments. For <code>qme</code> the number of bootstrap replicates can be adjusted by setting <code>R=</code> the desired number of replicates. Also the control argument of <code>optim</code> can be set by <code>control=list()</code> (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 <code>optim</code>
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 <code>optim</code> . See the documentation for <code>optim</code> for further details
convergence	from <code>optim</code> . 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 <code>optim</code> . 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

- Karlsson, M. (2004) Finite sample properties of the QME, *Communications in Statistics - Simulation and Computation*, **5**, pp 567–583
- Karlsson, M., Lindmark, A. (2014) truncSP: An R Package for Estimation of Semi-Parametric Truncated Linear Regression Models, *Journal of Statistical Software*, **57(14)**, pp 1–19, <https://www.jstatsoft.org/article/view/v057i14>
- Laitila, T. (2001) Properties of the QME under asymmetrically distributed disturbances, *Statistics & Probability Letters*, **52**, pp 347–352

Lee, M. (1993) Quadratic mode regression, *Journal of Econometrics*, **57**, pp 1-19

Lee, M. & Kim, H. (1998) Semiparametric econometric estimators for a truncated regression model: a review with an extension, *Statistica Neerlandica*, **52(2)**, pp 200–225

See Also

`qme.fit`, the function that does the actual fitting

`stls`, for estimation of models with truncated response variables using the STLS 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

```
#####
#           Simulated data           #
#####
#Simulated data (asymmetrically distributed errors):
n <- 1000
set.seed(319993)
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 to consistently estimate the slope parameters
qme.sim <- qme(y ~ x1 + x2 + x3, dtrunc, point = 0, direction = "left",
              cval = "ml", const = 1, beta = "ml", covar = FALSE)

summary(qme.sim)

#####
# Example using data "PM10trunc" #
#####

data(PM10trunc)

qmepm10 <- qme(PM10 ~ cars + temp + wind.speed + temp.diff + wind.dir + hour + day,
              data = PM10trunc, point = 2, control = list(maxit=4500))

summary(qmepm10)
```

qme-class	<i>An S4 class for objects from the qme function</i>
-----------	--

Description

An S4 class for objects from the [qme](#) function

Slots

call The matched call
coefficients A matrix
startcoef A matrix
cval A data frame
value Numeric
counts Integer
convergence Integer
message A character vector
residuals A matrix
fitted.values A matrix
df.residual Integer
covariance A matrix
bootrepl A matrix

qme.fit	<i>Function for fitting QME</i>
---------	---------------------------------

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

Value

A list with components:

startcoef	the starting values of the regression coefficients used by <code>optim</code>
coefficients	the named vector of coefficients
counts	number of iterations used by <code>optim</code> . See the documentation for <code>optim</code> for further details
convergence	from <code>optim</code> . An integer code. 0 indicates successful completion. Possible error codes are 1 indicating that the iteration limit <code>maxit</code> had been reached. 10 indicating degeneracy of the Nelder–Mead simplex.
message	from <code>optim</code> . A character string giving any additional information returned by the optimizer, or <code>NULL</code> .
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 <- 1000
x <- rnorm(n,0,2)
```

```

y <- 2+x+4*rnorm(n)

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

##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))

```

residuals.lt *Function to extract model residuals from objects of class "lt"*

Description

Function to extract model residuals from objects of class "lt"

Usage

```
## S3 method for class 'lt'
residuals(object, ...)
```

Arguments

object	object of class "lt"
...	additional arguments

residuals.qme *Function to extract model residuals from objects of class "stls"*

Description

Function to extract model residuals from objects of class "stls"

Usage

```
## S3 method for class 'qme'
residuals(object, ...)
```

Arguments

object	object of class "stls"
...	additional arguments

residuals.stls	<i>Function to extract model residuals from objects of class "stls"</i>
----------------	---

Description

Function to extract model residuals from objects of class "stls"

Usage

```
## S3 method for class 'stls'  
residuals(object, ...)
```

Arguments

object	object of class "stls"
...	additional arguments

stls	<i>Estimation of truncated regression models using the Symmetrically Trimmed Least Squares (STLS) estimator</i>
------	---

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,  
  ...  
)
```

Arguments

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): <ul style="list-style-type: none"> • 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 <code>truncreg</code>. • Method "ols" means that the estimated regression coefficients from fitting a linear model with <code>lm</code> 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. 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.
...	additional arguments. For <code>stls</code> the number of bootstrap replicates can be adjusted by setting <code>R</code> =the desired number of replicates. Also the <code>control</code> argument of <code>optim</code> can be set by <code>control=list()</code> (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 <code>optim</code>
value	the value of the objective function corresponding to coefficients
counts	number of iterations used by <code>optim</code> . See the documentation for <code>optim</code> for further details
convergence	from <code>optim</code> . An integer code. 0 indicates successful completion. Possible error codes are 1 indicating that the iteration limit <code>maxit</code> had been reached. 10 indicating degeneracy of the Nelder–Mead simplex.
message	from <code>optim</code> . A character string giving any additional information returned by the optimizer, or <code>NULL</code> .
residuals	the residuals of the model
fitted.values	the fitted values
df.residual	the residual degrees of freedom
call	the matched call
covariance	if <code>covar = TRUE</code> , the estimated covariance matrix
R	if <code>covar = TRUE</code> , the number of bootstrap replicates
bootrepl	if <code>covar = TRUE</code> , the bootstrap replicates

Author(s)

Anita Lindmark and Maria Karlsson

References

- Karlsson, M., Lindmark, A. (2014) `truncSP`: An R Package for Estimation of Semi-Parametric Truncated Linear Regression Models, *Journal of Statistical Software*, **57(14)**, pp 1–19, <https://www.jstatsoft.org/article/view/v057i14>
- Powell, J. (1986) Symmetrically Trimmed Least Squares Estimation for Tobit Models, *Econometrika*, **54(6)**, pp 1435–1460

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

```
#####
#           Simulated data           #
#####
##Simulated data
n <- 1000
set.seed(319993)
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.sim <- stls(y ~ x1 + x2 + x3, dtrunc, point = 0,
                direction = "left", beta = "ml", covar = FALSE)

summary(stls.sim)

#####
# 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)
```

stls-class

An S4 class for objects from the [stls](#) function

Description

An S4 class for objects from the [stls](#) function

Slots

call The matched call
coefficients A matrix
startcoef A matrix
value Numeric
counts Integer

convergence Integer
 message A character vector
 residuals A matrix
 fitted.values A matrix
 df.residual Integer
 covariance A matrix
 bootrepl A matrix

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	the point of truncation
direction	the 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 <code>optim</code> . An integer code. 0 indicates successful completion. Possible error codes are 1 indicating that the iteration limit <code>maxit</code> had been reached. 10 indicating degeneracy of the Nelder–Mead simplex.
message	from <code>optim</code> . A character string giving any additional information returned by the optimizer, or <code>NULL</code> .
residuals	the residuals of the model
<code>df.residual</code>	the residual degrees of freedom
<code>fitted.values</code>	the fitted values

Author(s)

Anita Lindmark and Maria Karlsson

See Also

[stls](#)

Examples

```
require(utils)

##Model frame
n <- 1000
x <- rnorm(n,0,2)
y <- 2+x+4*rnorm(n)

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

##Starting values
lmmod <- lm(data=mf)
bet <- lmmod$coef
bet <- matrix(bet)

str(stls. <- stls.fit(y~x,mf,point=0,direction="left",bet))
```

summary.lt

Summary function for objects of class "lt"

Description

Summary function for objects of class "lt"

Usage

```
## S3 method for class 'lt'
summary(object, level = 0.95, ...)
```

Arguments

object	object of class "lt"
level	confidence level. A number between 0 and 1. The default value is 0.95.
...	additional arguments

Value

A list with values:

coefficients	the named vector of coefficients
cvalues	information about the thresholds used. The method and constant used and the resulting lower and upper threshold values.
counts	number of iterations used by optim . See the documentation for optim for further details
call	the matched call
covariance	if covar = TRUE, the estimated covariance matrix
level	confidence level
confint	confidence intervals, based on normal distribution
bootconfint	bootstrap confidence intervals, percentile method

summary.lt-class	<i>An S4 class for objects from the summary.lt function</i>
------------------	---

Description

An S4 class for objects from the [summary.lt](#) function

Slots

call	The matched call
coefficients	A matrix
startcoef	A matrix
cvalues	A data frame
value	Numeric
counts	Integer
convergence	Integer
message	A character vector

residuals A matrix
 fitted.values A matrix
 df.residual Integer
 covariance A matrix
 bootrepl A matrix
 level Numeric
 confint A matrix
 bootconfint A matrix

 summary.qme

Summary function for objects of class "qme"

Description

Summary function for objects of class "qme"

Usage

```
## S3 method for class 'qme'
summary(object, level = 0.95, ...)
```

Arguments

object	object of class "qme"
level	confidence level. A number between 0 and 1. The default value is 0.95.
...	additional arguments

Value

A list with values:

coefficients	the named vector of coefficients
cval	information about the threshold used. The method and constant used and the resulting lower and upper threshold value.
counts	number of iterations used by optim . See the documentation for optim for further details
call	the matched call
covariance	if covar = TRUE, the estimated covariance matrix
level	confidence level
confint	confidence intervals, based on normal distribution
bootconfint	bootstrap confidence intervals, percentile method

summary.qme-class	<i>An S4 class for objects from the summary.qme function</i>
-------------------	--

Description

An S4 class for objects from the [summary.qme](#) function

Slots

call The matched call
 coefficients A matrix
 startcoef A matrix
 cval A data frame
 value Numeric
 counts Integer
 convergence Integer
 message A character vector
 residuals A matrix
 fitted.values A matrix
 df.residual Integer
 covariance A matrix
 bootrepl A matrix
 level Numeric
 confint A matrix
 bootconfint A matrix

summary.stls	<i>Summary function for objects of class "stls"</i>
--------------	---

Description

Summary function for objects of class "stls"

Usage

```
## S3 method for class 'stls'
summary(object, level = 0.95, ...)
```

Arguments

object	object of class "stls"
level	confidence level. A number between 0 and 1. The default value is 0.95.
...	additional arguments

Value

A list with values:

coefficients	the named vector of coefficients
counts	number of iterations used by optim . See the documentation for optim for further details
call	the matched call
covariance	if covar = TRUE, the estimated covariance matrix
level	confidence level
confint	confidence intervals, based on normal distribution
bootconfint	bootstrap confidence intervals, percentile method

summary.stls-class *An S4 class for objects from the [summary.stls](#) function*

Description

An S4 class for objects from the [summary.stls](#) function

Slots

call	The matched call
coefficients	A matrix
startcoef	A matrix
value	Numeric
counts	Integer
convergence	Integer
message	A character vector
residuals	A matrix
fitted.values	A matrix
df.residual	Integer
covariance	A matrix
bootrepl	A matrix
level	Numeric
confint	A matrix
bootconfint	A matrix

vcov.lt	<i>Function to obtain the variance-covariance matrix for objects of class "lt"</i>
---------	--

Description

Function to obtain the variance-covariance matrix for objects of class "lt"

Usage

```
## S3 method for class 'lt'  
vcov(object, ...)
```

Arguments

object	object of class "lt"
...	additional arguments

vcov.qme	<i>Function to obtain the variance-covariance matrix for objects of class "qme"</i>
----------	---

Description

Function to obtain the variance-covariance matrix for objects of class "qme"

Usage

```
## S3 method for class 'qme'  
vcov(object, ...)
```

Arguments

object	object of class "qme"
...	additional arguments

vcov.stls	<i>Function to obtain the variance-covariance matrix for objects of class "stls"</i>
-----------	--

Description

Function to obtain the variance-covariance matrix for objects of class "stls"

Usage

```
## S3 method for class 'stls'  
vcov(object, ...)
```

Arguments

object	object of class "stls"
...	additional arguments

Index

* datasets

PM10, 19
PM10trunc, 20

bootconfint, 3

coef.lt, 4
coef.qme, 4
coef.stls, 5
covar.boot, 5

fitted.lt, 6
fitted.qme, 6
fitted.stls, 7
funcLT, 7
funcQME, 8
funcSTLS, 9
funcval.LT, 10
funcval.QME, 11
funcval.STLS, 11

lm, 13, 24, 31
lt, 10, 12, 16, 17, 26, 32
lt-class, 16
lt.fit, 15, 16

mlcoef, 18

optim, 8, 9, 13, 14, 16, 17, 24, 25, 27, 28, 31,
32, 34–37, 39

PM10, 19, 20
PM10trunc, 20
print.lt, 20
print.qme, 21
print.stls, 21
print.summary.lt, 22
print.summary.qme, 22
print.summary.stls, 23

qme, 11, 14, 15, 23, 27, 28, 32

qme-class, 27
qme.fit, 26, 27

residuals.lt, 29
residuals.qme, 29
residuals.stls, 30

stls, 12, 15, 26, 30, 33–35
stls-class, 33
stls.fit, 32, 34
summary.lt, 35, 36
summary.lt-class, 36
summary.qme, 37, 38
summary.qme-class, 38
summary.stls, 38, 39
summary.stls-class, 39

truncreg, 8, 9, 13, 15, 18, 24, 26, 31, 32
truncSP (truncSP-package), 3
truncSP-package, 3

vcov.lt, 40
vcov.qme, 40
vcov.stls, 41