Simulation Extrapolation

Description

Implementation of the SIMEX Algorithm for measurement error models according to Cook and Stefanski

Usage

simex(model, SIMEXvariable, measurement.error, lambda = c(0.5,1,1.5,2), B = 100, fitting.method = "quadratic", jackknife.estimation = "quad", asymptotic = TRUE)

Arguments

model
SIMEXvariable
measurement.error
lambda
B
fitting.method
jackknife.estimation
asymptotic

Arguments:

model
the naive model
SIMEXvariable
character or vector of characters containing the names of the variables with measurement error
measurement.error
vector of standard deviations of the known measurement errors
lambda
vector of lambdas for which the simulation step should be done (without 0)
B
number of iterations for each lambda
fitting.method
fitting method linear, quadratic, nonlinear are implemented. (first 4 letters are enough)
jackknife.estimation
specifying the extrapolation method for jackknife variance estimation. Can be set to FALSE if it should not be performed
asymptotic
logical, indicating if asymptotic variance estimation should be done, in the Naive model the option x = TRUE have to be set.

Details

nonlinear is implemented as described in Cook and Stefanski, but is numerically not stable. It is not advisable to use this feature. See fit.nls for details. If a nonlinear extrapolation is desired please use the refit function.

Asymptotic is only implemented for naive models of class lm or glm

Value

Returns an object of class SIMEX which contains:

coefficients
the corrected coefficients of the SIMEX model,
**Author(s)**

Wolfgang Lederer, wolfgang.lederer@googlemail.com

**References**


**See Also**

mcsimex for discreete data with misclassification, lm, glm, refit

**Examples**

```r
# to test nonlinear extrapolation set.seed(3)
x <- rnorm(200,0,100)
u <- rnorm(200,0,25)
w <- x+u
y <- x + rnorm(200,0,9)
true.model <- lm(y~x)  # True model
naive.model <- lm(y~w, x=TRUE)
simex.model <- simex(model = naive.model
                         , SIMEXvariable = "w"
                         , ...)```
plot(x,y)
abline(true.model,col="darkblue")
abline(simex.model,col ="red")
abline(naive.model,col = "green")
legend(min(x),max(y),legend=c("True Model","SIMEX model","Naive Model")
, col = c("darkblue","red","green"),lty=1)
plot(simex.model, mfrow = c(2,2))

[Package simex version 1.2 Index]
Print method for the SIMEX and MCSIMEX class

Description

Printing the most important values in a clear way

Usage

print.SIMEX(x, digits = max(3,getOption("digits") - 3), ...)

Arguments

x  object of class SIMEX
digits  number of digits to be printed
...  arguments passed to other functions

Author(s)

Wolfgang Lederer

[Package simex version 1.2 Index]
**Plot method for the SIMEX- and MCSIMEX-class**

**Description**

Creates plots for the SIMEX- or MCSIMEX-class. For each variable a plot is produced showing the results of the extrapolation step.

**Usage**

```r
plot.SIMEX(x, xlab = expression((1 + lambda)), ylab = colnames(b[, -1]),
            ask = FALSE, show = rep(TRUE, NCOL(b) - 1), ...)
```

**Arguments**

- `x` object of class SIMEX
- `xlab` optional name for the X-Axis
- `ylab` vector containing the names for the Y-Axis
- `ask` logical. If TRUE, the user is asked for input, before a new figure is drawn.
- `show` vector of logicals indicating for which variables a plot should be produced
- `...` optional, passed to `par()`

**Author(s)**

Wolfgang Lederer

**See Also**

`simex`, `mcsimex`, `par`

**Examples**

```r
a <- rnorm(100)
b <- a*20 + rnorm(100,100,10)
e <- a + rnorm(100,1,5)
naive.model <- lm(a~ b + e, x= TRUE,y =TRUE)

w<- simex(model = naive.model
          , SIMEXvariable = c("b","e")
          , measurement.error = c(10,5)
          , lambda=c(0.5,1,1.5,1.75,2.25,2.5,3))

plot(w, ask = FALSE, mfrow = c(2,2))
```

[Package simex version 1.2 Index]
summary.SIMEX(simex)

**Summarizing Model Fits for the SIMEX method**

**Description**

summary method for class SIMEX.

**Usage**

summary.SIMEX(object,...)

**Arguments**

- `object`: Object of class SIMEX
- `...`: further arguments

**Value**

- `coefficients`: a p x 4 matrix with columns for the estimated coefficient, its standard error, t-statistic and corresponding (two-sided) p-value.
- `residuals`: residuals of the corrected model
- `call`: the function call
- `B`: number of iterations
- `naive.model`: the naive model
- `SIMEXvariable`: character vector of the SIMEXvariable
- `measurement.error`: vector of the variances of the known error measurement error

**Author(s)**

Wolfgang Lederer

**See Also**

`simex`

**Examples**

```
# continues example from simex()
# summary(simex.model)
```