Package ‘lm.br’

September 27, 2024

Type Package

Title Linear Model with Breakpoint

Version 2.9.8

Date 2024-09-26

Description Exact significance tests for a changepoint in linear or multiple linear regression. Confidence regions with exact coverage probabilities for the changepoint. Based on Knowles, Siegmund and Zhang (1991) <doi:10.1093/biomet/78.1.15>.

License GPL (>= 2)

Depends R(>= 3.0.1), Rcpp (>= 0.11.0)

Imports stats, methods, graphics, datasets

LinkingTo Rcpp

NeedsCompilation yes

Author Marc Adams [aut, cre],
authors of R function 'lm' [ctb] (general interface),
authors of 'lm.gls' [ctb] (interface and R code for covariate weights),
U.S. NIST [ctb] (C++ code for TNT::Vector template)

Maintainer Marc Adams <lm.br.pkg@gmail.com>

Repository CRAN

Date/Publication 2024-09-27 19:50:02 UTC

Contents

'lm.br' *Fit a Linear Model with a Breakpoint*

Description

Exact significance tests for a changepoint in linear or multiple linear regression. Confidence intervals and confidence regions with exact coverage probabilities for the changepoint.

Usage

```
lm.br(formula, type ="LL", data, subset, weights, inverse =FALSE,
      var.known =FALSE, na.action, contrasts, offset, ...)
```

Arguments

`formula` a formula expression of the form `response ~ predictors`, the same as for regression models; see `formula`

`type` "LL", "LT" or "TL" which stand for line-line, line-threshold or threshold-line, defined below

`data` an optional data-frame that assigns values in `formula`

`subset` expression saying which subset of the data to use

`weights` vector or matrix

`inverse` if TRUE then 'weights' specifies the inverse of the weights vector or matrix, as for a covariance matrix

`var.known` is the variance known?

`na.action` a function to filter missing data

`contrasts` an optional list; see 'contrasts.arg' in `model.matrix`

`offset` a constant vector to be subtracted from the responses vector

`...` other arguments to `lm.fit` or `lm.wfit`

Details

A broken-line model consists of two straight lines joined at a changepoint. Three versions are

```
LL   y = alpha + B * min(x - theta, 0) + Bp * max(x - theta, 0) + e

LT   y = alpha + B * min(x - theta, 0) + e

TL   y = alpha + Bp * max(x - theta, 0) + e
```

where e ~ Normal(0, var * inv(weights)). The LT and TL versions omit 'alpha' if the formula is without intercept, such as 'y~x+0'. Parameters 'theta', 'alpha', 'B', 'Bp', 'var' are unknown, but 'weights' is known.

The same models apply for a multiple-regression formula such as 'y ~ x1 + x2 + ... + xn' where 'alpha' becomes the coefficient of the "1"-vector and 'theta' the changepoint for the coefficient of the first predictor term, 'x1'.

The test for the presence of a changepoint is by a postulate value outside the range of 'x'-values. Thus, in the LL model 'sl(min(x1) - 1)' would give the exact significance level of the null hypothesis "single line" versus the alternate hypothesis "broken line."

Exact inferences about the changepoint 'theta' or '(theta,alpha)' are based on the distribution of its likelihood-ratio statistic, conditional on sufficient statistics for the other parameters. This method is called conditional likelihood-ratio (CLR) for short.

Value

'lm.br' returns a list that includes a C++ object with accessor functions. Functions `sl`, `ci` and `cr` get significance levels, confidence intervals, and confidence regions for the changepoint's x-coordinate or (x,y)-coordinates. Other functions are `mle` to get maximum likelihood estimates and `sety` to set new y-values. The returned object also lists 'coefficients', 'fitted.values' and 'residuals', the same as for an 'lm' output list.

Note

Data can include more than one 'y' value for a repeat 'x' value. If variance is known, then 'var' = 1 and 'weights' is the inverse of the variances vector or variance-covariance matrix.

References

Knowles, M., Siegmund, D. and Zhang, H.P. (1991) Confidence regions in semilinear regression, _Biometrika_, *78*, 15-31.

Siegmund, D. and Zhang, H.P. (1994), Confidence regions in broken line regression, in "Changepoint Problems", _IMS Lecture Notes – Monograph Series_, *23*, eds. E. Carlstein, H. Muller and D. Siegmund, Hayward, CA: Institute of Mathematical Statistics, 292-316.

See Also

vignette("lm.br")
demo(testscript)

Examples

```
#  Smith & Cook (1980), "Straight Lines with a Change-point: A Bayesian
#  Analysis of some Renal Transplant Data", Appl Stat, *29*, 180-189,
#  reciprocal of blood creatinine L/micromol  vs  day after transplant.
creatinine <- c(37.3, 47.1, 51.5, 67.6, 75.9, 73.3, 69.4, 61.5, 31.8, 19.4)
day <- 1:10
sc <- lm.br( creatinine ~ day )
sc $ mle()
sc $ ci()
sc $ sl( day[1] - 1.5 )      # test for the presence of a changepoint
plot( sc$residuals )


#  A 'TL' example, data from figure 1 in Chiu et al. (2006), "Bent-cable
#  regression theory and applications”, J Am Stat Assoc, *101*, 542-553,
#  log(salmon abundance) vs year.
salmon <- c( 2.50, 2.93, 2.94, 2.83, 2.43, 2.84, 3.06, 2.97, 2.94, 2.65,
  2.92, 2.71, 2.93, 2.60, 2.12, 2.08, 1.81, 2.45, 1.71, 0.55, 1.30 )
year <- 1980 : 2000
chiu <- lm.br( salmon ~ year, 'tl' )
chiu $ ci()


#  A multiple regression example, using an R dataset,
#  automobile miles-per-gallon  versus  weight and horsepower.
lm.br( mpg ~ wt + hp,  data = mtcars )


#  An example with variance known, for the Normal approximations of binomial
#  random variables using formula 2.28 of Cox and Snell (1989).
#    Ex. 3.4 of Freeman (2010) "Inference for binomial changepoint data” in
# _Advances in Data Analysis_, ed. C Skiadas, Boston: Birkhauser, 345-352.
trials <- c( 15, 82, 82, 77, 38, 81, 12, 97, 33, 75,
  85, 37, 44, 96, 76, 26, 91, 47, 41, 35 )
successes <- c( 8, 44, 47, 39, 24, 38, 3, 51, 16, 43,
  47, 27, 33, 64, 41, 18, 61, 32, 33, 24 )
log_odds <- log( (successes - 0.5)/(trials - successes - 0.5) )
variances <- (trials-1)/( successes*(trials-successes) )
group <- 1 : 20
lm.br( log_odds ~ group, 'TL', w= variances, inv= TRUE, var.known= TRUE )


#  An example that shows different confidence regions from inference by
#  conditional likelihood-ratio (CLR)  versus  approximate-F (AF).
y <- c( 1.6, 3.2, 6.3, 4.8, 4.3, 4.0, 3.5, 1.8 )
x <- 1:8
eg <- lm.br( y ~ x )
eg$cr( output='t' )
eg$cr( method = 'aF', output='t' )
```

`ci` *Confidence Interval for the Changepoint*

Description

Confidence interval for ’theta’, the changepoint’s x-coordinate.

Usage

```
## S4 method for signature 'Cpp_Clmbr'
ci(  CL =0.95, method ="CLR", output ="T" )
```

Arguments

`CL`	confidence level, between 0 and 1.
`method`	"CLR" or "AF" which stand for conditional likelihood-ratio or approximate-F, see `sl` for details.
`output`	"T", "V" or "B" which stand for text, value or both.

Details

This subroutine scans to determine the postulate values of 'theta' that have significance level greater than 1-CL.

Value

'ci' prints-out the confidence interval for 'theta' but does not return a value if 'output' is "T". 'sl' returns a numeric vector of boudaries for the contiguous segments of the confidence interval if 'output' is "V" or "B".

Examples

```
#  Data for Patient B from Smith and Cook (1980)
y <- c(37.3, 47.1, 51.5, 67.6, 75.9, 73.3, 69.4, 61.5, 31.8, 19.4)
x <- 1:10
sc <- lm.br( y ~ x )
sc$ci()
sc $ ci( 0.90 )
sc $ ci( .99, 'af' )
sc $ ci( out= 'v' )
```

---

cr *Confidence Region for the Changepoint*

Description

Joint confidence region for (theta, alpha), the changepoint's (x,y)-coordinates.

Usage

```
## S4 method for signature 'Cpp_Clmbr'
cr(  CL =0.95 ,  method ="CLR",  incr,  output ="G"  )
```

Arguments

CL confidence level, between 0 and 1.

method "CLR" or "AF" which stand for conditional likelihood-ratio or approximate-F (rapid), see sl for details.

incr increment of theta values for the confidence region's boundary-points.

output "G", "T" or "V" which stand for graph, text or value.

Details

This subroutine scans to determine the postulate values of (theta, alpha) that have significance level greater than 1-CL. It scans first along the (theta, alpha-MLE) ridge to determine the 'theta' boundary limits.

Value

If 'output' is "G" or "T" then 'cr' graphs or prints-out the confidence region but does not return a value. If 'output' is "V" then 'cr' returns an N x 3 matrix of boundary points (theta, min-alpha, max-alpha).

Examples

```
#  A quick example
y <- c( 2, 0, 2.001, 4, 6 )
x <- 1:5
t <- lm.br( y ~ x )
t $ cr()
t$cr( .9, 'af', incr = 0.1, out='t' )
```

mle *Maximum Likelihood Estimates*

Description

Maximum-likelihood estimates of parameters. Estimates are without bias correction except for the variance.

Usage

```
## S4 method for signature 'Cpp_Clmbr'
mle( output ="T" )
```

Arguments

output "T", "V" or "B" which stand for text, value or both.

Value

'mle' prints-out the maximum-likelihood estimates but does not return a value if 'output' is "T". 'mle' returns a numeric vector of maximum-likelihood estimates if 'output' is "V" or "B".

Examples

```
#  Data for Patient B from Smith and Cook (1980)
y <- c(37.3, 47.1, 51.5, 67.6, 75.9, 73.3, 69.4, 61.5, 31.8, 19.4)
x <- 1:10
sc <- lm.br(y~x)
sc$mle()
estimates <- sc$mle( 'v' )
estimates
```

---

sety *Set y-Values*

Description

Reset the response values in the C++ object.

Usage

```
## S4 method for signature 'Cpp_Clmbr'
sety( rWy )
```

Arguments

rWy	vector of 'y' values, pre-multiplied by the square-root of 'weights'.

Details

The 'rWy' vector is simply the y-vector if the model does not specify weights. The square-root of a vector 'W' is the vector 'rW' of the square-roots of the elements of 'W'. The square-root of a matrix 'W' here is the matrix 'rW' such that rW*rW = W (a stricter definition than rW*transpose(rW) = W).

Note

The pre-multiplied vector is more convenient as input during simulation tests. 'sety' changes the y-values only for the accessor functions 'sl', 'ci', 'cr' and 'mle'. 'rW' is the inverse square-root if 'inverse' was TRUE in the 'lm.br' call.

Examples

```
#  A simulation test
x <- c( 1.0, 1.1, 1.3, 1.7, 2.4, 3.9, 5.7, 7.6, 8.4, 8.6 )
y <- x
LLmodel <- lm.br( y ~ x )
countCLR <- countAF <- 0
theta <- 3
for( i in 1:10000 )  {
  y <- 0 + (-1.)*pmin(x-theta,0) + (0.5)*pmax(x-theta,0) + rnorm(10)
  LLmodel$sety( y )
  stest <- LLmodel$sl( theta, 'clr', .0001, ”V” )
  if( stest > 0.05 )  countCLR <- countCLR + 1
  stest <- LLmodel$sl( theta, 'af', .0001, ”V” )
  if( stest > 0.05 )  countAF <- countAF + 1
  if( floor(i/1000) - i/1000 == 0 ) cat(i, countCLR/i, countAF/i, ”\n”)
}
```

sl *Significance Level for Changepoint*

Description

Significance level of a postulate value for the changepoint's x- or (x,y)-coordinates.

Usage

```
## S4 method for signature 'Cpp_Clmbr'
sl( theta0,  method =”CLR”, tolerance =0.001, output =”T” )
## S4 method for signature 'Cpp_Clmbr'
sl( theta0, alpha0,  method =”CLR”, tolerance =0.001, output =”T” )
```

Arguments

`theta0` postulate value for ’theta’, the changepoint's x-coordinate.

`alpha0` postulate value for ’alpha’, the changepoint's y-coordinate.

`method` "CLR", "MC" or "AF" which stand for conditional likelihood-ratio, conditional likelihood-ratio by Monte Carlo or approximate-F, details below.

`tolerance` maximum absolute error in numerical integration for the "CLR" method or in Monte-Carlo evaluation for the "MC" method, not referenced for the "AF" method.

`output` "T", "V" or "B" which stand for text, value or both.

Details

Knowles, Siegmund and Zhang (1991) reduced the conditional likelihood-ratio significance test to a probability expression based on a generic random variable.

The default method "CLR" evaluates this probability using a geometric-expectation formula that Knowles et al. also derived. This formula slightly over-estimates, but the error is negligible for significance levels below 0.20.

Method "MC" evaluates that probability expression directly by Monte Carlo simulation, which avoids the over-estimate of the "CLR" method.

Method "AF" estimates the distribution of the likelihood-ratio statistic by the related F-distribution (or chi-squared if variance is known) which would be exact for a linear model. This method is not exact, but it is common for non-linear regression.

Value

'sl' prints-out the result but does not return a value if 'output' is "T". 'sl' returns a numeric value if 'output' is "V" or "B".

Note

The 'tolerance' error-limit does not include the slight over-estimate that is inherent in the "CLR" method, nor the approximation inherent in the "AF" method.

Examples

```
#  Data for Patient B from Smith and Cook (1980)
y <- c(37.3, 47.1, 51.5, 67.6, 75.9, 73.3, 69.4, 61.5, 31.8, 19.4)
x <- 1:10
sc <- lm.br( y ~ x )

sc $ sl( 6.1 )
sc $ sl( 6.1, 'mc' )
sc $ sl( 6.1, 'mc', 0.00001 )
sc $ sl( 6.1, 88.2, 'clr' )
sc $ sl( 6.1, 88.2, 'af' )
tmp <- sc $ sl( 6.1, 88.2, 'mc', 0.001, "B" )
tmp
```

Index