

Package ‘gof’

April 17, 2009

Type Package

Title Model-diagnostics based on cumulative residuals

Version 0.6-4

Date 2009-08-04

Author Klaus K. Holst

Maintainer Klaus. K. Holst <kkho@biostat.ku.dk>

Description Implementation of model-checking technique based on cumulative residuals

Depends R (>= 2.5), survival

License GPL

LazyLoad yes

Repository CRAN

Date/Publication 2009-04-16 08:42:12

R topics documented:

gof-package	1
confint.cumres	2
cumres	3
cumres.coxph	5
plot.cumres	6
Index	8

`gof-package`*Model-diagnostics based on cumulative residuals*

Description

Implementation of model-checking technique based on cumulative residuals

Details

Package: `gof`
Type: `Package`
Version: `0.6-4`
Date: `2009-08-04`
License: `GPL`
LazyLoad: `yes`

Author(s)

Klaus. K. Holst <kkho@biostat.ku.dk>

References

D.Y. Lin and L.J. Wei and Z. Ying (2002) *Model-Checking Techniques Based on Cumulative Residuals*. Biometrics, Volume 58, pp 1-12.

John Q. Su and L.J. Wei (1991) *A lack-of-fit test for the mean function in a generalized linear model*. Journal. Amer. Statist. Assoc., Volume 86, Number 414, pp 420-426.

See Also

`cox.aalen` in the `timereg`-package for similar GOF-methods for survival-data.

Examples

```
example(cumres)
```

`confint.cumres`*Returns prediction bands for 'cumres' object*

Description

`confint` returns prediction bands for the cumulative residual process under the null.

Usage

```
## S3 method for class 'cumres':  
confint (object, parm=1, level=0.95, cval=NULL, ...)
```

Arguments

object	Object produced by the function <code>cumres</code>
parm	vector of numbers indicating which processes from the <code>x</code> to calculate prediction bands for.
level	The required prediction level.
cval	Overrules the level-parameter by calculating symmetric prediction bands defined by the standard error multiplied by <code>cval</code> .
...	Additional arguments.

Value

list with the following members:

t	Ordered values of variable that is used to cumulate residuals after
yu	Upper simultaneous confidence limit.

Author(s)

Klaus K. Holst <kkho@biostat.ku.dk>

See Also

[cumres](#)

Examples

```
n <- 500; x <- abs(rnorm(n, sd=0.2))+0.01; y <- sqrt(x) + rnorm(n, sd=0.2)  
l <- lm(y ~ x)  
g <- cumres(l, R=1000)  
confint(g, 1)
```

cumres

Calculates GOF measures for for generalized linear models

Description

Currently linear regression models (lm) and logistic and poisson regression models are supported.

Usage

```
## S3 method for class 'lm':
cumres (model, ...)
## S3 method for class 'glm':
cumres (model,
        variable=c("predicted", colnames(model.matrix(model))),
        data=data.frame(model.matrix(model)),
        R=500, b=0, plots=min(R,50),
        seed=round(runif(1,1,1e9)), ...)
```

Arguments

model	Model object (lm or glm)
variable	List of variable to order the residuals after
data	data.frame used to fit model (complete cases)
R	Number of samples used in simulation
b	Moving average bandwidth (0 corresponds to infinity = standard cumulated residuals)
plots	Number of realizations to save for use in the plot-routine
seed	Random seed
...	additional arguments

Value

Returns an object of class 'cumres'.

Author(s)

Klaus K. Holst

See Also

[cox.aalen](#) in the `timereg`-package for similar GOF-methods for survival-data.

Examples

```
sim1 <- function(n=100, f=function(x1,x2) {10+x1+x2^2}, sd=1, seed=1) {
  if (!is.null(seed))
    set.seed(seed)
  x1 <- rnorm(n);
  x2 <- rnorm(n)
  X <- cbind(1, x1, x2)
  y <- f(x1, x2) + rnorm(n, sd=sd)
  d <- data.frame(y, x1, x2)
  return(d)
}
d <- sim1(100); l <- lm(y ~ x1 + x2, d)
system.time(g <- cumres(l, R=100, plots=50))
```

```

g
## Not run: plot(g)
g1 <- cumres(l, c("y"), R=100, plots=50)
g1
g2 <- cumres(l, c("y"), R=100, plots=50, b=0.5)
g2

```

cumres.coxph	<i>Calculates GOF measures for Cox's propoportional hazard model for right censored survival times</i>
--------------	--

Description

Currently only implemented for score processes which serves as tests for proportionality of hazards.

Usage

```

## S3 method for class 'coxph':
cumres (model,
        variable=c(colnames(model.matrix(model))),
        type=c("score", "residual"),
        R=500, plots=min(R, 50), seed=round(runif(1, 1, 1e9)), ...)

```

Arguments

model	Model object (lm or glm)
variable	List of variable to order the residuals after
R	Number of samples used in simulation
type	Type of GOF-procedure
plots	Number of realizations to save for use in the plot-routine
seed	Random seed
...	additional arguments

Value

Returns an object of class 'cumres'.

Author(s)

Klaus K. Holst and Thomas Scheike

References

Lin, D. Y. and Wei, L. J. and Ying, Z. (1993) *Checking the Cox model with cumulative sums of martingale-based residuals* Biometrika, Volume 80, No 3, p. 557-572.

Martinussen, Torben and Scheike, Thomas H. *Dynamic regression models for survival data* (2006), Springer, New York.

See Also

`cumres.glm`, `coxph`, and `cox.aalen` in the `timereg`-package for similar GOF-methods for survival-data.

Examples

```
library(survival)

simcox <- function(n=100, seed=1) {
  if (!is.null(seed))
    set.seed(seed)
  require(survival)
  time<-rexp(n); cen<-2*rexp(n);
  status<-(time<cen);
  time[status==0]<-cen[status==0];
  X<-matrix(rnorm(2*n),n,2)
  return(data.frame(time=time, status=status, X))
}
n <- 100; d <- simcox(n); model <- coxph(Surv(time,status)~ X1 + X2, data=d, robust=TRUE)

dtimes <- sort(d$time[d$status==1])
system.time(a <- cumres(model))
summary(a)

## PBC example
data(pbc)
fit.cox <- coxph(Surv(time,status) ~ age + edema + bili + protime, data=pbc)
system.time(pbc.gof <- cumres(fit.cox,R=2000))
## Not run:
par(mfrow=c(2,2))
plot(pbc.gof, ci=TRUE, legend=NULL)
## End(Not run)
```

plot.cumres

Plot cumulative residuals from a 'cumres' object

Description

plot displays the observed cumulative residual process with realizations under the null. 95% prediction bands

Usage

```
## S3 method for class 'cumres':
plot(x, idx=1:length(x$variable), col=c("grey"),
      ci=TRUE,
      col.ci="darkblue", col.alpha=0.3, lty.ci=0, level=0.95,
      legend=c("type1", "type2", "none"), ylim=NULL,
      ...)
```

Arguments

<code>x</code>	Object produced by the function <code>cumres</code> .
<code>idx</code>	vector of numbers (or variable names) indicating which processes from the <code>x</code> to plot.
<code>col</code>	Color of the sample processes. By setting this parameter to "none" or <code>NULL</code> no realizations will be drawn. The number of realizations is determined by the <code>cumres</code> -object.
<code>ci</code>	Type of prediction bands to plot. Defaults to none. Set to <code>TRUE</code> to obtain simultaneous prediction bands under the null (pointwise can be obtained by setting to "pointwise").
<code>col.ci</code>	Color of prediction band.
<code>col.alpha</code>	Degree of transparency (0-1) of the prediction bands.
<code>lty.ci</code>	Line type of prediction band.
<code>level</code>	The required prediction level.
<code>legend</code>	Type of legend where "type1" gives p-values of GOF-tests and "type2" gives usual type of legends.
<code>ylim</code>	Range of y axis.
<code>...</code>	Additional arguments passed to the plot-routine.

Author(s)

Klaus K. Holst <kkho@biostat.ku.dk>

See Also

[cumres](#)

Examples

```
n <- 500; x <- abs(rnorm(n,sd=0.2))+0.01; y <- sqrt(x) + rnorm(n,sd=0.2)
l <- lm(y ~ x)
g <- cumres(l, R=1000)
plot(g, idx=1, ci="sim", col=NULL, col.ci="purple", legend="type2")
```

Index

*Topic **hplot**

plot.cumres, 6

*Topic **models**

confint.cumres, 2

cumres, 3

cumres.coxph, 5

*Topic **package**

gof-package, 1

*Topic **regression**

confint.cumres, 2

cumres, 3

cumres.coxph, 5

plot.cumres, 6

confint.cumres, 2

cox.aalen, 2, 4, 5

coxph, 5

cumres, 3, 3, 7

cumres.coxph, 5

cumres.glm, 5

gof (*gof-package*), 1

gof-package, 1

plot.cumres, 6

timereg-package, 5