Package 'ARCensReg'

September 11, 2016

Type Package

Title Fitting Univariate Censored Linear Regression Model with Autoregressive Errors

Version 2.1

Date 2016-09-10

Author Fernanda L. Schumacher, Victor H. Lachos and Christian E. Galarza

Maintainer Fernanda L. Schumacher <fernandalschumacher@gmail.com>

Imports mvtnorm, tmvtnorm, msm, numDeriv, utils, graphics, stats

Suggests SMNCensReg

Description It fits an univariate left or right censored linear regression model with autoregressive errors under the normal distribution. It provides estimates and standard errors of the parameters, prediction of future observations and it supports missing values on the dependent variable.
It also performs influence diagnostic through local influence for three possible perturbation schemes.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2016-09-11 00:54:34

R topics documented:

ARCensReg 2
CloudCeiling 4
InfDiag 5
phosphorus 7
rARCens 9

Index **11**

ARCensReg *Censored Linear Regression Model with Autoregressive Errors*

Description

It fits an univariate left or right censored linear regression model with autoregressive errors under the normal distribution using the SAEM algorithm. It provides estimates and standard errors of the parameters, prediction of future observations and it supports missing values on the dependent variable. It also provides convergence plots when exists at least one censored observation.

Usage

```
ARCensReg(cc,y,x,p=1,cens='left',x_pred=NULL,miss=NULL,
  tol=0.0001,show.convergence=TRUE,M=10,perc=0.25,MaxIter=400,
  pc=0.18,show_se=TRUE)
```

Arguments

cc	Vector of censoring indicators of length n, where n is the total of observations. For each observation: 0 if non-censored, 1 if censored.
x	Matrix of covariates of dimension n x l, where l is the number of fixed effects including the intercept, if considered (in models which include an intercept x should contain a column of ones).
y	Vector of responses of length n.
p	Order of the autoregressive process. Must be a positive integer value. For p equal to 0 we suggest to use the function CensReg.SMN from SMNCensReg package.
cens	"left" for left censoring, "right" for right censoring.
x_pred	Matrix of covariates for responses to be predicted. If x_pred = NULL no responses are predicted.
miss	Vector containing the index of missing observations. miss = NULL indicates that no observations are missing.
tol	The convergence maximum error permitted.
show.convergence	TRUE or FALSE. Indicates if convergence graphs should be built for the parameters estimates (for the case with at least one censored observation). The dashed line indicates the iteration of the SAEM algorithm that simulations start being smoothed. Default=TRUE.
M	Size of the Monte Carlo sample generated in each step of the SAEM algorithm. Default=10.
perc	Percentage of burn-in on the Monte Carlo sample. Default=0.25.
MaxIter	The maximum number of iterations of the SAEM algorithm. Default=400.
pc	Percentage of initial iterations of the SAEM algorithm. It is recommended that 50<MaxIter*pc<100. Default=0.18.
show_se	TRUE or FALSE. Indicates if the standard errors should be estimated. Default=TRUE.

Details

The initial values are obtained by ignoring censoring and applying maximum likelihood estimation with the censored data simply replaced by their censoring limits. If you want to fit a regression model with autoregressive errors for non-censored data, just set "cc" as a vector of zeros and "cens" as either "right" or "left".

Value

`beta`	Estimate of the regression parameters.
`sigma2`	Estimated variance of the white noise process.
`phi`	Estimate of the autoregressive parameters.
`pi1`	Estimate of the first p partial autocorrelations.
`theta`	Vector of parameters estimate (beta, sigma2, phi).
`SE`	Vector of the standard errors of (beta, sigma2, phi).
`loglik`	Log-likelihood value.
`AIC`	Akaike information criterion.
`BIC`	Bayesian information criterion.
`AICcorr`	Corrected Akaike information criterion.
`time`	Processing time.
`pred`	Predicted values (if x_pred is not `NULL`).
`criteria`	Attained criteria value.
`yest`	Augmented response variable based on the fitted model.
`yyest`	Final estimative of `E(Y%*%t(Y))`.
`iter`	Number of iterations until convergence.

Author(s)

Fernanda L. Schumacher <<fernandalschumacher@gmail.com>>, Victor H. Lachos <<hlachos@ime.unicamp.br>> and Christian E. Galarza <<cgalarza88@gmail.com>>

Maintainer: Fernanda L. Schumacher <<fernandalschumacher@gmail.com>>

References

Delyon, B., Lavielle, M. & Moulines, E. (1999) Convergence of a stochastic approximation version of the EM algorithm. Journal of the Royal Statistical Society, Series B, 39, 1-38.

Schumacher, F. L., Lachos, V. H. & Dey, D. K. (2016) Censored regression models with autoregressive errors: A likelihood-based perspective. Submitted.

See Also

`arima`, `CensReg.SMN`, `InfDiag`

Examples

```
##simple example (p = l = 1)
#generating a sample
set.seed(23451)
n=50
x=rep(1,n)
dat = rARCens(n=n,beta=2,pit=.5,sig2=.3,x=x,
                cens='left',pcens=.1)

#fitting the model (quick convergence)
fit0 = ARCensReg(dat$data$cc,dat$data$y,x,tol=0.001,
                  pc=.12,M=5,show_se=FALSE)

## Not run:

##another example (p = l = 2)
#generating a sample
n=100
x=cbind(1,runif(n))
dat = rARCens(n=n,beta=c(2,1),pit=c(.4,-.2),sig2=.5,
                    x=x,cens='left',pcens=.05)

#fitting the model
fit1 = ARCensReg(dat$data$cc,dat$data$y,x,p=2,
                      cens="left",tol=0.0001)

#plotting the augmented variable
par(mfrow=c(1,1))
plot.ts(fit1$yest,lty='dashed',col=4)
lines(dat$data$y)

#simulating missing values
miss = sample(1:n,3)
yMISS = dat$data$y
yMISS[miss] = NA
fit2 = ARCensReg(dat$data$cc,yMISS,x,p=2,miss=miss,
                cens="left",tol=0.0001)


## End(Not run)
```

`CloudCeiling` *Cloud Ceiling Height*

Description

The cloud ceiling heights, collected by the National Center for Atmospheric Research (NCAR), were observed hourly in San Francisco during the month of March 1989, consisting of n=716 observations (Park et al., 2007).

Usage

```
data(CloudCeiling)
```

Format

This data frame contains the following columns:

`y` logarithm of the cloud ceiling heights.

`cc` right censoring indicator (1 if the observation is right censored and 0 otherwise).

Source

Park, J. W., Genton, M. G. & Ghosh, S. K. (2007). Censored time series analysis with autoregressive moving average models. Journal of Computational and Graphical Statistics, 18(4), 797-817.

See Also

ARCensReg

Examples

```
## Not run:
data(CloudCeiling)

plot.ts(CloudCeiling$y)

#Proportion of censoring
prop.table(table(CloudCeiling$cc))

#A censored regression model
##this may take a long time due to the number of censored observations.
##For other examples see help(ARCensReg).

x = as.matrix(rep(1,length(CloudCeiling$y)))
miss =  which(is.na(CloudCeiling$y))
AR_reg = ARCensReg(CloudCeiling$cc,CloudCeiling$y,x,cens='right',miss=miss,p=1,tol=.001)

## End(Not run)
```

InfDiag — *Influence Diagnostic in Censored Linear Regression Model with Autoregressive Errors*

Description

It performs influence diagnostic by a local influence approach (Cook, 1986) with three possible perturbations schemes: response perturbation (y), scale matrix perturbation (Sigma) or explanatory variable perturbation (x). A benchmark value is calculated that depends on k.

Usage

```
InfDiag(theta,yest,yyest,x,k=3,plots=T,indpar=rep(1,length(theta)),
                  perturbation ='y',indcolx = rep(1,ncol(x)))
```

Arguments

theta	Vector of estimated parameters.
yest	Vector of responses of length n with agmented data. Should be the value yest of the ARCensReg function in the case that at least one observation is censored.
yyest	Should be the value yyest of the ARCensReg function in the case that at least one observation is censored. Otherwise, must be y%*%t(y).
x	Matrix of covariates of dimension n x l, where l is the number of fixed effects including the intercept, if considered (in models which include an intercept x should contain a column of ones).
k	Constant to be used in the benchmark calculation: M0+k*sd(M0).
plots	TRUE or FALSE. Indicates if a graph should be plotted.
indpar	Vector of length equal to the number of parameters, with each element 0 or 1 indicating if the respective parameter should be taking into account in the influence calculation.
perturbation	Perturbation scheme. Possible values: "y" for response perturbation, "Sigma" for scale matrix perturbation or "x" for explanatory variable perturbation.
indcolx	If perturbation="x", indcolx must be a vector of length equal to the number of columns of x, with each element 0 or 1 indicating if the respective column of x should be perturbed. All columns are perturbed by default.

Details

The function returns a vector of length n with the aggregated contribution (M0) of all eigenvectors of the matrix associated with the normal curvature. For details see (Schumacher et. al., 2016).

Value

M0

Author(s)

Fernanda L. Schumacher <<fernandalschumacher@gmail.com>>, Victor H. Lachos <<hlachos@ime.unicamp.br>> and Christian E. Galarza <<cgalarza88@gmail.com>>

Maintainer: Fernanda L. Schumacher <<fernandalschumacher@gmail.com>>

References

Cook, R. D. (1986). Assessment of local influence. Journal of the Royal Statistical Society, Series B, 48, 133-169.

Schumacher, F. L., Lachos, V. H. & Vilca-Labra, F. E. (2016) Influence diagnostics for censored regression models with autoregressive errors. Submitted.

Zhu, H. & Lee, S. (2001). Local influence for incomplete-data models. Journal of the Royal Statistical Society, Series B, 63, 111-126.

See Also

ARCensReg

Examples

```
## Not run:
#generating the data
set.seed(12341)
x = cbind(1,runif(100))
dat = rARCens(n=100,beta = c(1,-1),pit = c(.4,-.2),sig2=.5,
            x=x,cens='left',pcens=.05)
#creating an outlier
dat$data$y[40] = 5
plot.ts(dat$data$y)

#fitting the model
fit = ARCensReg(cc=dat$data$cc,y=dat$data$y,x,p=2,cens='left',
      tol=0.001,show_se=F)

#influence diagnostic
M0y = InfDiag(theta=fit$res$theta, yest=fit$yest, yyest=fit$yyest,
        x=x, k = 3.5, perturbation = "y")
M0Sigma = InfDiag(theta=fit$res$theta, yest=fit$yest, yyest=fit$yyest,
        x=x, k = 3.5, perturbation = "Sigma")
M0x = InfDiag(theta=fit$res$theta, yest=fit$yest, yyest=fit$yyest,
        x=x, k = 3.5, perturbation = "x",indcolx =c(0,1))

#perturbation on a subset of parameters
M0y1 = InfDiag(theta=fit$res$theta, yest=fit$yest, yyest=fit$yyest,
        x=x, k = 3.5, perturbation = "y",indpar=c(1,1,0,0,0))
M0y2 = InfDiag(theta=fit$res$theta, yest=fit$yest, yyest=fit$yyest,
        x=x, k = 3.5, perturbation = "y",indpar=c(0,0,1,1,1))
plot(M0y1,M0y2)
abline(v = mean(M0y1)+3.5*sd(M0y1),h = mean(M0y2)+3.5*sd(M0y2),lty=2)

## End(Not run)
```

---

phosphorus *Phosphorus Concentration data*

Description

The phosphorus concentration (P) data of West Fork Cedar River at Finchford, Iowa, USA, collected under the ambient water quality program conducted by the Iowa Department of Natural Resources (Iowa DNR) were observed monthly from 10/1998 to 10/2013 (n=181) and the dataset was first

available in the R package carx. The phosphorus concentration measurement was subject to a limit of detection (lcl), thereby the P data are left censored.

The water discharge dataset were obtained from the website of U.S. Geological Survey (site number 05458900), and it is measured in cubic feet per second.

Usage

```
data(phosphorus)
```

Format

This data frame contains the following columns:

`lP` logarithm of the phosphorus concentration.

`cc` left censoring indicator (1 if the observation is left censored and 0 otherwise).

`lQ` logarithm of the water discharge.

`lcl` lower censoring limit.

`time` Year-Month.

Source

http://waterdata.usgs.gov/ia/nwis/monthly/

https://cran.r-project.org/web/packages/carx/carx.pdf

See Also

ARCensReg

Examples

```
## Not run:
data(phosphorus)

plot.ts(phosphorus$lP)
lines(phosphorus$lcl,col=2,lty=2)

#Proportion of censoring
prop.table(table(phosphorus$cc))

#A censored regression model
x = cbind(1,phosphorus$lQ)
miss =  which(is.na(phosphorus$lP))
AR_reg = ARCensReg(phosphorus$cc,phosphorus$lP,x,cens='left',miss=miss,p=1,tol=.001)

## End(Not run)
```

rARCens *Generating Censored Autoregressive Data*

Description

It simulates a censored response variable with autoregressive errors of order p, with a established censoring rate.

Usage

```
rARCens(n,beta,pit,sig2,x,cens,pcens)
```

Arguments

n	Length of the desired time serie.
beta	Vector of theoretical regression parameters of length l.
pit	Vector of theoretical partial autocorrelations. Each element must be in (-1,1).
sig2	Theoretical variance of the error.
x	Matrix of covariates of dimension n x l (in models that include an intercept x should contain a column of ones).
cens	"left" for left censoring, "right" for right censoring.
pcens	Desired censoring rate.

Value

data	Generated response (y) and censoring indicator (cc).
param	Theoretical parameters (beta, sig2, phi).

Author(s)

Fernanda L. Schumacher <<fernandalschumacher@gmail.com>>, Victor H. Lachos <<hlachos@ime.unicamp.br>> and Christian E. Galarza <<cgalarza88@gmail.com>>

Maintainer: Fernanda L. Schumacher <<fernandalschumacher@gmail.com>>

See Also

ARCensReg

Examples

```
#generating a sample
dat = rARCens(n=100,beta = c(1,-1),pit = c(.4,-.2),
     sig2=.5,x=cbind(1,runif(100)),cens='left',pcens=.05)
#
plot.ts(dat$data$y)
table(dat$data$cc)

dat$param
#[1]  1.00 -1.00  0.50  0.48 -0.20
```

Index

∗Topic **autoregressive errors**
ARCensReg, 2
InfDiag, 5
rARCens, 9
∗Topic **censored regression**
ARCensReg, 2
InfDiag, 5
rARCens, 9
∗Topic **datasets**
CloudCeiling, 4
phosphorus, 7
∗Topic **influence diagnostic**
InfDiag, 5
∗Topic **package**
ARCensReg, 2
InfDiag, 5
rARCens, 9

ARCensReg, 2, *5*, *7–9*
arima, *3*

CensReg.SMN, *2, 3*
CloudCeiling, 4

InfDiag, *3*, 5

phosphorus, 7

rARCens, 9