Package 'ARCensReg'

May 6, 2026

Type Package

Title Fitting Univariate Censored Linear Regression Model with Autoregressive Errors

Version 3.0.2

Description It fits a univariate left, right, or interval censored linear regression model with autoregressive errors, considering the normal or the Student-t distribution for the innovations. It provides estimates and standard errors of the parameters, predicts future observations, and supports missing values on the dependent variable.
References used for this package:
Schumacher, F. L., Lachos, V. H., & Dey, D. K. (2017). Censored regression models with autoregressive errors: A likelihood-based perspective. Canadian Journal of Statistics, 45(4), 375-392 <doi:10.1002/cjs.11338>.
Schumacher, F. L., Lachos, V. H., Vilca-Labra, F. E., & Castro, L. M. (2018). Influence diagnostics for censored regression models with autoregressive errors. Australian & New Zealand Journal of Statistics, 60(2), 209-229 <doi:10.1111/anzs.12229>.
Valeriano, K. A., Schumacher, F. L., Galarza, C. E., & Matos, L. A. (2024). Censored autoregressive regression models with Student-t innovations. Canadian Journal of Statistics, 52(3), 804-828 <doi:10.1002/cjs.11804>.

License GPL (>= 2)

Encoding UTF-8

Imports ggplot2, gridExtra, matrixcalc, methods, msm, mvtnorm, numDeriv, qqplotr, Rcpp, Rdpack, stats, tmvtnorm, utils

RdMacros Rdpack

LinkingTo RcppArmadillo, Rcpp

Suggests SMNCensReg

NeedsCompilation yes

Repository CRAN

Author Fernanda L. Schumacher [aut, cre] (ORCID: <https://orcid.org/0000-0002-5724-8918>),
Katherine A. L. Valeriano [ctb] (ORCID: <https://orcid.org/0000-0001-6388-4753>),
Victor H. Lachos [ctb] (ORCID: <https://orcid.org/0000-0002-7239-2459>),

Christian G. Morales [ctb] (ORCID:
<https://orcid.org/0000-0002-4818-6006>),
Larissa A. Matos [ctb] (ORCID: <https://orcid.org/0000-0002-2635-0901>)

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

Date/Publication 2025-08-26 09:30:09 UTC

Contents

`ARCensReg` *Censored linear regression model with autoregressive errors*

Description

It fits a univariate left, right, or interval censored linear regression model with autoregressive errors under the normal distribution, using the SAEM algorithm. It provides estimates and standard errors of the parameters, supporting missing values on the dependent variable.

Usage

```
ARCensReg(cc, lcl = NULL, ucl = NULL, y, x, p = 1, M = 10,
  perc = 0.25, MaxIter = 400, pc = 0.18, tol = 1e-04,
  show_se = TRUE, quiet = FALSE)
```

Arguments

`cc`	Vector of censoring indicators of length n, where n is the total observations. For each observation: 0 if non-censored, 1 if censored/missing.
`lcl`, `ucl`	Vectors of length n that represent the lower and upper bounds of the interval, which contains the observade value of the censored observation. Default=`NULL`, indicating no-censored data. See details for more information.

y	Vector of responses of length n.
x	Matrix of covariates of dimension $n \times l$, 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).
p	Order of the autoregressive process. It must be a positive integer value.
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 with no memory. It is recommended that 50<MaxIter*pc<100. Default=0.18.
tol	The convergence maximum error permitted.
show_se	TRUE or FALSE. Indicates if the standard errors should be estimated. Default=TRUE.
quiet	TRUE or FALSE. Indicates if printing information should be suppressed. Default=FALSE.

Details

The linear regression model with autocorrelated errors, defined as a discrete-time autoregressive (AR) process of order p, at time t is given by

$Y_t = x_t^T \beta + \xi_t,$

$\xi_t = \phi_1 \xi_{t-1} + ... + \phi_p \xi_{t-p} + \eta_t, t = 1, ..., n,$

where Y_t is the response variable, $\beta = (\beta_1, ..., \beta_l)^T$ is a vector of regression parameters of dimension l, and $x_t = (x_{t1}, ..., x_{tl})^T$ is a vector of non-stochastic regressor variables values; ξ_t is the AR error with Gaussian disturbance η_t, $\phi = (\phi_1, ..., \phi_p)^T$ is the vector of AR coefficients, and n is the sample size.

It is assumed that Y_t is not fully observed for all t. For left censored observations, we have lcl=-Inf and ucl=V_t, such that the true value $Y_t \leq V_t$. For right censoring, lcl=V_t and ucl=Inf, such that $Y_t \geq V_t$. For interval censoring, lcl and ucl must be finite values, such that $V_{1t} \leq Y_t \leq V_{2t}$. Missing data can be defined by setting lcl=-Inf and ucl=Inf.

The initial values are obtained by ignoring censoring and applying maximum likelihood estimation with the censored data replaced by their censoring limits. Furthermore, just set cc as a vector of zeros to fit a regression model with autoregressive errors for non-censored data.

Value

An object of class "ARpCRM", representing the AR(p) censored regression normal fit. Generic functions such as print and summary have methods to show the results of the fit. The function plot provides convergence graphics for the parameters when at least one censored observation exists.

Specifically, the following components are returned:

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 (β, σ^2, ϕ).
SE	Vector of the standard errors of (β, σ^2, ϕ).
loglik	Log-likelihood value.
AIC	Akaike information criterion.
BIC	Bayesian information criterion.
AICcorr	Corrected Akaike information criterion.
yest	Augmented response variable based on the fitted model.
yyest	Final estimative of `E(Y%*%t(Y))`.
x	Matrix of covariates of dimension $n \times l$.
iter	Number of iterations until convergence.
criteria	Attained criteria value.
call	The `ARCensReg` call that produced the object.
tab	Table of estimates.
critFin	Selection criteria.
cens	"left", "right", or "interval" for left, right, or interval censoring, respectively.
nmiss	Number of missing observations.
ncens	Number of censored observations.
converge	Logical indicating convergence of the estimation algorithm.
MaxIter	The maximum number of iterations used for the SAEM algorithm.
M	Size of the Monte Carlo sample generated in each step of the SAEM algorithm.
pc	Percentage of initial iterations of the SAEM algorithm with no memory.
time	Time elapsed in processing.
plot	A list containing convergence information.

Author(s)

Fernanda L. Schumacher, Katherine L. Valeriano, Victor H. Lachos, Christian E. Galarza, and Larissa A. Matos

References

Delyon B, Lavielle M, Moulines E (1999). "Convergence of a stochastic approximation version of the EM algorithm." *Annals of statistics*, 94–128.

Schumacher FL, Lachos VH, Dey DK (2017). "Censored regression models with autoregressive errors: A likelihood-based perspective." *Canadian Journal of Statistics*, **45**(4), 375–392.

See Also

arima, ARtCensReg, InfDiag

Examples

```
## Example 1: (p = l = 1)
# Generating a sample
set.seed(23451)
n = 50
x = rep(1, n)
dat = rARCens(n=n, beta=2, phi=.5, sig2=.3, x=x, cens='left', pcens=.1)

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

## Example 2: (p = l = 2)
# Generating a sample
n = 100
x = cbind(1, runif(n))
dat = rARCens(n=n, beta=c(2,1), phi=c(.48,-.2), sig2=.5, x=x, cens='left',
              pcens=.05)

# Fitting the model
fit1 = ARCensReg(dat$data$cc, dat$data$lcl, dat$data$ucl, dat$data$y, x,
                 p=2, tol=0.0001)
summary(fit1)
plot(fit1)

# Plotting the augmented variable
library(ggplot2)
data.plot = data.frame(yobs=dat$data$y, yest=fit1$yest)
ggplot(data.plot) + theme_bw() +
  geom_line(aes(x=1:nrow(data.plot), y=yest), color=4, linetype="dashed") +
  geom_line(aes(x=1:nrow(data.plot), y=yobs)) + labs(x="Time", y="y")

## Example 3: Simulating missing values
miss = sample(1:n, 3)
yMISS = dat$data$y
yMISS[miss] = NA
cc = dat$data$cc
cc[miss] = 1
lcl = dat$data$lcl
ucl = dat$data$ucl
ucl[miss] = Inf

fit2 = ARCensReg(cc, lcl, ucl, yMISS, x, p=2)
plot(fit2)

# Imputed missing values
data.frame(yobs=dat$data$y[miss], yest=fit2$yest[miss])
```

ARtCensReg *Censored autoregressive regression model with Student-t innovations*

Description

It fits a univariate left, right, or interval censored linear regression model with autoregressive errors considering Student-t innovations, through the SAEM algorithm. It provides estimates and standard errors of the parameters, supporting missing values on the dependent variable.

Usage

```
ARtCensReg(cc, lcl = NULL, ucl = NULL, y, x, p = 1, M = 10,
  perc = 0.25, MaxIter = 400, pc = 0.18, nufix = NULL, tol = 1e-04,
  show_se = TRUE, quiet = FALSE)
```

Arguments

cc	Vector of censoring indicators of length n, where n is the total observations. For each observation: 0 if non-censored, 1 if censored/missing.
lcl, ucl	Vectors of length n that represent the lower and upper bounds of the interval, which contains the observade value of the censored observation. Default=NULL, indicating no-censored data. See details for more information.
y	Vector of responses of length n.
x	Matrix of covariates of dimension $n \times l$, 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).
p	Order of the autoregressive process. It must be a positive integer value.
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 with no memory. It is recommended that 50<MaxIter*pc<100. Default=0.18.
nufix	If the degrees of freedom (ν) are unknown, nufix should be equal to NULL; otherwise, it must be a number greater than 2.
tol	The convergence maximum error permitted.
show_se	TRUE or FALSE. Indicates if the standard errors should be estimated. Default=TRUE.
quiet	TRUE or FALSE. Indicates if printing information should be suppressed. Default=FALSE.

Details

The linear regression model with autocorrelated errors, defined as a discrete-time autoregressive (AR) process of order p, at time t is given by

$Y_t = x_t^T \beta + \xi_t,$

$\xi_t = \phi_1 \xi_{t-1} + ... + \phi_p \xi_{t-p} + \eta_t, t = 1, ..., n,$

where Y_t is the response variable, $\beta = (\beta_1, ..., \beta_l)^T$ is a vector of regression parameters of dimension l, $x_t = (x_{t1}, ..., x_{tl})^T$ is a vector of non-stochastic regressor variables values, and ξ_t is the AR

error with η_t being a shock of disturbance following the Student-t distribution with ν degrees of freedom, $\phi = (\phi_1, ..., \phi_p)^T$ being the vector of AR coefficients, and n denoting the sample size.

It is assumed that Y_t is not fully observed for all t. For left censored observations, we have `lcl=-Inf` and `ucl=`V_t, such that the true value $Y_t \leq V_t$. For right censoring, `lcl=`V_t and `ucl=Inf`, such that $Y_t \geq V_t$. For interval censoring, `lcl` and `ucl` must be finite values, such that $V_{1t} \leq Y_t \leq V_{2t}$. Missing data can be defined by setting `lcl=-Inf` and `ucl=Inf`.

The initial values are obtained by ignoring censoring and applying maximum likelihood estimation with the censored data replaced by their censoring limits. Moreover, just set `cc` as a vector of zeros to fit a regression model with autoregressive errors for non-censored data.

Value

An object of class "ARtpCRM" representing the AR(p) censored regression Student-t fit. Generic functions such as print and summary have methods to show the results of the fit. The function plot provides convergence graphics for the parameter estimates.

Specifically, the following components are returned:

`beta`	Estimate of the regression parameters.
`sigma2`	Estimated scale parameter of the innovation.
`phi`	Estimate of the autoregressive parameters.
`nu`	Estimated degrees of freedom.
`theta`	Vector of parameters estimate $(\beta, \sigma^2, \phi, \nu)$.
`SE`	Vector of the standard errors of $(\beta, \sigma^2, \phi, \nu)$.
`yest`	Augmented response variable based on the fitted model.
`uest`	Final estimated weight variables.
`x`	Matrix of covariates of dimension $n \times l$.
`iter`	Number of iterations until convergence.
`criteria`	Attained criteria value.
`call`	The `ARtCensReg` call that produced the object.
`tab`	Table of estimates.
`cens`	"left", "right", or "interval" for left, right, or interval censoring, respectively.
`nmiss`	Number of missing observations.
`ncens`	Number of censored observations.
`converge`	Logical indicating convergence of the estimation algorithm.
`MaxIter`	The maximum number of iterations used for the SAEM algorithm.
`M`	Size of the Monte Carlo sample generated in each step of the SAEM algorithm.
`pc`	Percentage of initial iterations of the SAEM algorithm with no memory.
`time`	Time elapsed in processing.
`plot`	A list containing convergence information.

Warning

This algorithm assumes that the first p values in the response vector are completely observed.

Author(s)

Katherine L. Valeriano, Fernanda L. Schumacher, and Larissa A. Matos

References

Delyon B, Lavielle M, Moulines E (1999). "Convergence of a stochastic approximation version of the EM algorithm." *Annals of statistics*, 94–128.

Valeriano KL, Schumacher FL, Galarza CE, Matos LA (2021). "Censored autoregressive regression models with Student-t innovations." *arXiv preprint arXiv:2110.00224*.

See Also

arima, ARCensReg

Examples

```
## Example 1: (p = l = 1)
# Generating a sample
set.seed(1234)
n = 80
x = rep(1, n)
dat = rARCens(n=n, beta=2, phi=.6, sig2=.3, x=x, cens='right', pcens=.05,
              innov='t', nu=4)

# Fitting the model (quick convergence)
fit0 = ARtCensReg(dat$data$cc, dat$data$lcl, dat$data$ucl, dat$data$y, x,
                  M=5, pc=.12, tol=0.001)
fit0

## Example 2: (p = l = 2)
# Generating a sample
set.seed(783796)
n = 200
x = cbind(1, runif(n))
dat = rARCens(n=n, beta=c(2,1), phi=c(.48,-.2), sig2=.5, x=x, cens='left',
              pcens=.05, innov='t', nu=5)

# Fitting the model with nu known
fit1 = ARtCensReg(dat$data$cc, dat$data$lcl, dat$data$ucl, dat$data$y, x,
                   p=2, M=15, pc=.20, nufix=5)
summary(fit1)
plot(fit1)

# Fitting the model with nu unknown
fit2 = ARtCensReg(dat$data$cc, dat$data$lcl, dat$data$ucl, dat$data$y, x,
                  p=2, M=15, pc=.20)
summary(fit2)
plot(fit2)
```

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 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 JW, Genton MG, Ghosh SK (2007). "Censored time series analysis with autoregressive moving average models." *Canadian Journal of Statistics*, **35**(1), 151–168.

See Also

ARCensReg, ARtCensReg

Examples

```
library(ggplot2)

data(CloudCeiling)
ggplot(CloudCeiling) + geom_line(aes(x=1:length(y), y=y)) +
  labs(x="Time") + theme_bw()

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

## Not run:
# 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$y)))
cc  = CloudCeiling$cc
lcl = CloudCeiling$y
ucl = rep(Inf, length(CloudCeiling$y))
```

```
miss =  which(is.na(CloudCeiling$y))
cc[miss]  = 1
lcl[miss] = -Inf
AR_reg = ARCensReg(cc, lcl, ucl, CloudCeiling$y, x, 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 perturbation schemes: response perturbation (y), scale matrix perturbation (Sigma), or explanatory variable perturbation (x). A benchmark value is calculated that depends on k.

Usage

```
InfDiag(object, k = 3, indpar = rep(1, length(object$theta)),
  indcolx = rep(1, ncol(object$x)), perturbation = "y")
```

Arguments

`object`	Object of class `'ARpCRM'` given as an output of function `ARCensReg`.
`k`	Constant to be used in the benchmark calculation: `M0+k*sd(M0)`.
`indpar`	Vector of length equal to the number of parameters, with each element 0 or 1 indicating if the respective parameter should be considered in the influence calculation.
`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.
`perturbation`	Perturbation scheme. Possible values: "y" for response perturbation, "Sigma" for scale matrix perturbation, or "x" for explanatory variable perturbation.

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. (2018).

Value

An object of class "DiagARpCRM" with the following components is returned:

`M0`	Vector of length n with the aggregated contribution of all eigenvectors of the matrix associated with the normal curvature.
`perturbation`	Perturbation scheme.
`benchmark`	`M0 + k*sd(M0)`.

Author(s)

Fernanda L. Schumacher, Katherine L. Valeriano, Victor H. Lachos, Christian E. Galarza, and Larissa A. Matos

References

Cook RD (1986). "Assessment of local influence." *Journal of the Royal Statistical Society: Series B (Methodological)*, **48**(2), 133–155.

Schumacher FL, Lachos VH, Vilca-Labra FE, Castro LM (2018). "Influence diagnostics for censored regression models with autoregressive errors." *Australian & New Zealand Journal of Statistics*, **60**(2), 209–229.

Zhu H, Lee S (2001). "Local influence for incomplete data models." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **63**(1), 111–126.

See Also

ARCensReg

Examples

```
library(ggplot2)

# Generating the data
set.seed(12341)
x = cbind(1,runif(100))
dat = rARCens(n=100, beta=c(1,-1), phi=c(.48,-.2), sig2=.5, x=x,
              cens='left', pcens=.05)

# Creating an outlier
dat$data$y[40] = 5
ggplot(dat$data) + geom_line(aes(x=1:100, y=y)) + theme_bw() +
  labs(x="Time")

# Fitting the model
fit = ARCensReg(dat$data$cc, dat$data$lcl, dat$data$ucl, dat$data$y, x,
                p=2, tol=0.001, show_se=FALSE)

# Influence diagnostic
M0y = InfDiag(fit, k=3.5, perturbation="y")
plot(M0y)
M0Sigma = InfDiag(fit, k=3.5, perturbation="Sigma")
plot(M0Sigma)
M0x = InfDiag(fit, k=3.5, indcolx=c(0,1), perturbation="x")
plot(M0x)

# Perturbation on a subset of parameters
M0y1 = InfDiag(fit, k=3.5, indpar=c(1,1,0,0,0), perturbation="y")$M0
M0y2 = InfDiag(fit, k=3.5, indpar=c(0,0,1,1,1), perturbation="y")$M0
#
ggplot(data.frame(M0y1,M0y2)) + geom_point(aes(x=M0y1, y=M0y2)) +
```

```
geom_hline(yintercept=mean(M0y2)+3.5*sd(M0y2), linetype="dashed") +
geom_vline(xintercept=mean(M0y1)+3.5*sd(M0y1), linetype="dashed") +
theme_bw()
```

---

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

The water discharge dataset was obtained from the website of the 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` Censoring limit.

`time` Year-Month.

Source

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

https://CRAN.R-project.org/package=carx

See Also

ARCensReg, ARtCensReg

Examples

```
library(ggplot2)

data(phosphorus)
n = nrow(phosphorus)

ggplot(phosphorus) + geom_line(aes(x=1:n, y=lP)) +
  geom_line(aes(x=1:n, y=lcl), color="red", linetype="dashed") +
  labs(x="Time") + theme_bw()

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

# A censored regression model
x   = cbind(1, phosphorus$lQ)
cc  = phosphorus$cc
lcl = rep(-Inf, n)
ucl = phosphorus$lcl
miss =  which(is.na(phosphorus$lP))
cc[miss]  = 1
ucl[miss] = Inf

# Fitting a model with normal innovations
set.seed(8765)
mod1 = ARCensReg(cc, lcl, ucl, phosphorus$lP, x, p=1, tol=.001)

# Fitting a model with Student-t innovations
set.seed(287399)
mod2 = ARtCensReg(cc, lcl, ucl, phosphorus$lP, x, p=1, tol=.001)

# Plotting observed and imputed values
data.plot = data.frame(y=phosphorus$lP, ynorm=mod1$yest, yt=mod2$yest)
#
ggplot(data.plot) + geom_line(aes(x=1:n, y=ynorm), color=4) +
  geom_line(aes(x=1:n, y=yt), color="deeppink", linetype="dashed") +
  geom_line(aes(x=1:n, y=y)) + labs(x="Time", y="lP") + theme_bw()

# Imputed values
data.plot[cc==1,]
```

`plot` *Plot an ARpCRM or ARtpCRM object*

Description

It displays convergence graphs 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.

Usage

```
  ## S3 method for class 'ARpCRM'
plot(x, ...)

  ## S3 method for class 'ARtpCRM'
plot(x, ...)
```

Arguments

x	An object inheriting from class ARpCRM or ARtpCRM, representing a fitted censored autoregressive model of order p, with normal and Student-t innovations, respectively.
...	Additional arguments.

Value

A ggplot object.

Author(s)

Fernanda L. Schumacher, Katherine L. Valeriano, Victor H. Lachos, Christian E. Galarza, and Larissa A. Matos

See Also

ggplot, ARCensReg, ARtCensReg

Examples

```
n = 50; x = rep(1, n)
dat = rARCens(n=n, beta=2, phi=.5, sig2=.3, x=x, cens='left', pcens=.1)

fit = ARCensReg(dat$data$cc, dat$data$lcl, dat$data$ucl, dat$data$y, x,
                M=5, pc=.12, tol=0.001, show_se=FALSE)
plot(fit)
```

---

plot.DiagARpCRM *Plot influence diagnostic measures*

Description

Plot method for objects of class "DiagARpCRM".

Usage

```
  ## S3 method for class 'DiagARpCRM'
plot(x, ...)
```

Arguments

x	An object inheriting from class DiagARpCRM. The influence diagnostic measures are calculated by function InfDiag, with three possible perturbation schemes: response perturbation (y), scale matrix perturbation (Sigma), or explanatory variable perturbation (x).
...	Additional arguments.

Value

A ggplot object, plotting the index versus the influence diagnostic measure.

Author(s)

Fernanda L. Schumacher, Katherine L. Valeriano, Victor H. Lachos, Christian E. Galarza, and Larissa A. Matos

See Also

ggplot, InfDiag, ARCensReg

Examples

```
library(ggplot2)

# Generating the data
set.seed(12341)
x = cbind(1,runif(100))
dat = rARCens(n=100, beta=c(1,-1), phi=c(.48,-.2), sig2=.5, x=x,
              cens='left', pcens=.05)

# Creating an outlier
dat$data$y[40] = 5
ggplot(dat$data) + geom_line(aes(x=1:100, y=y)) + theme_bw() +
  labs(x="Time")

# Fitting the model
fit = ARCensReg(dat$data$cc, dat$data$lcl, dat$data$ucl, dat$data$y, x,
                p=2, tol=0.001, show_se=FALSE)

# Influence diagnostic
M0y = InfDiag(fit, k=3.5, perturbation="y")
plot(M0y)
M0Sigma = InfDiag(fit, k=3.5, perturbation="Sigma")
plot(M0Sigma)
M0x = InfDiag(fit, k=3.5, indcolx=c(0,1), perturbation="x")
plot(M0x)

# Perturbation on a subset of parameters
M0y1 = InfDiag(fit, k=3.5, indpar=c(1,1,0,0,0), perturbation="y")$M0
M0y2 = InfDiag(fit, k=3.5, indpar=c(0,0,1,1,1), perturbation="y")$M0
#
```

```
ggplot(data.frame(M0y1,M0y2)) + geom_point(aes(x=M0y1, y=M0y2)) +
  geom_hline(yintercept=mean(M0y2)+3.5*sd(M0y2), linetype="dashed") +
  geom_vline(xintercept=mean(M0y1)+3.5*sd(M0y1), linetype="dashed") +
  theme_bw()
```

`plot.residARpCRM` *Show diagnostic residual plots*

Description

It returns four plots for the quantile residuals: the time series plot of the residuals, the quantile-quantile plot, the histogram, and the ACF plot of the residuals.

Usage

```
  ## S3 method for class 'residARpCRM'
plot(x, ...)
```

Arguments

`x`	An object inheriting from class `residARpCRM` obtained as an output of function `residuals`.
`...`	Additional arguments.

Value

A ggplot object.

Author(s)

Fernanda L. Schumacher, Katherine L. Valeriano, Victor H. Lachos, Christian E. Galarza, and Larissa A. Matos

See Also

`ggplot`, `ARCensReg`, `ARtCensReg`, `residuals.ARpCRM`, `residuals.ARtpCRM`

Examples

```
## Example 1: Generating data with normal innovations
set.seed(93899)
x = cbind(1, runif(300))
dat1 = rARCens(n=300, beta=c(1,-1), phi=c(.48,-.2), sig2=.5, x=x,
               cens='left', pcens=.05, innov="norm")

# Fitting the model with normal innovations
mod1 = ARCensReg(dat1$data$cc, dat1$data$lcl, dat1$data$ucl, dat1$data$y,
                 x, p=2, tol=0.001)
r1 = residuals(mod1)
```

```
class(r1)
plot(r1)

# Fitting the model with Student-t innovations
mod2 = ARtCensReg(dat1$data$cc, dat1$data$lcl, dat1$data$ucl, dat1$data$y,
                  x, p=2, tol=0.001)
r2 = residuals(mod2)
plot(r2)


## Example 2: Generating heavy-tailed data
set.seed(12341)
x = cbind(1, runif(300))
dat2 = rARCens(n=300, beta=c(1,-1), phi=c(.48,-.2), sig2=.5, x=x,
               cens='left', pcens=.05, innov="t", nu=3)

# Fitting the model with normal innovations
mod3 = ARCensReg(dat2$data$cc, dat2$data$lcl, dat2$data$ucl, dat2$data$y,
                 x, p=2, tol=0.001)
r3 = residuals(mod3)
plot(r3)

# Fitting the model with Student-t innovations
mod4 = ARtCensReg(dat2$data$cc, dat2$data$lcl, dat2$data$ucl, dat2$data$y,
                  x, p=2, tol=0.001)
r4 = residuals(mod4)
plot(r4)
```

`predict` *Forecast for Autoregressive censored models with Normal and Student-t innovations*

Description

Forecast from models fitted by `ARCensReg` and `ARtCensReg`.

Usage

```
  ## S3 method for class 'ARpCRM'
predict(object, x_pred, ...)

  ## S3 method for class 'ARtpCRM'
predict(object, x_pred, ...)
```

Arguments

`object`	An object inheriting from class `ARpCRM` or `ARtpCRM`, representing a fitted AR(p) censored linear model.
`x_pred`	Matrix of covariates for responses to be predicted.
`...`	Further arguments passed to or from other methods.

Value

A time series of predictions.

Author(s)

Fernanda L. Schumacher, Katherine L. Valeriano, Victor H. Lachos, Christian E. Galarza, and Larissa A. Matos

References

Schumacher FL, Lachos VH, Dey DK (2017). "Censored regression models with autoregressive errors: A likelihood-based perspective." *Canadian Journal of Statistics*, **45**(4), 375–392.

Valeriano KL, Schumacher FL, Galarza CE, Matos LA (2021). "Censored autoregressive regression models with Student-t innovations." *arXiv preprint arXiv:2110.00224*.

See Also

ARCensReg, ARtCensReg

Examples

```
# Generating a sample
set.seed(2839)
n = 210
x = cbind(1, rnorm(n))
dat = rARCens(n=n, beta=c(-1,2), phi=.5, sig2=.3, x=x, cens='left', pcens=.1)

# Fitting the model
data1 = dat$data[1:205,]
fit = ARCensReg(data1$cc, data1$lcl, data1$ucl, data1$y, x[1:205,],
                M=5, pc=.12, tol=0.001)

# Forecast
y_pred = predict(fit, x[206:n,])
mean((dat$data$y[206:n] - y_pred)^2) # MSPE
```

---

print *Print an ARpCRM or ARtpCRM object*

Description

Print an ARpCRM or ARtpCRM object.

Usage

```
  ## S3 method for class 'ARpCRM'
print(x, ...)

  ## S3 method for class 'ARtpCRM'
print(x, ...)
```

Arguments

x	An object inheriting from class ARpCRM or ARtpCRM, representing a fitted censored autoregressive model of order p.
...	Additional print arguments.

Author(s)

Fernanda L. Schumacher, Katherine L. Valeriano, Victor H. Lachos, Christian E. Galarza, and Larissa A. Matos

See Also

ARCensReg, ARtCensReg, summary, plot

Examples

```
n = 50; x = rep(1, n)
dat = rARCens(n=n, beta=2, phi=.5, sig2=.3, x=x, cens='left', pcens=.1)

fit = ARCensReg(dat$data$cc, dat$data$lcl, dat$data$ucl, dat$data$y, x,
                M=5, pc=.12, tol=0.001, show_se=FALSE)
fit
```

---

rARCens *Generating censored autoregressive data*

Description

It simulates a censored response variable with autoregressive errors of order p following normal or Student-t innovations, with an established censoring rate.

Usage

```
rARCens(n, beta, phi, sig2 = 1, x = rep(1, n), cens = "left",
  pcens = 0.1, innov = "norm", nu = NULL)
```

Arguments

n	Length of the desired time serie.
beta	Vector of theoretical regression parameters of length l.
phi	Vector of theoretical autoregressive coefficients of length p.
sig2	Theoretical variance of the error.
x	Matrix of covariates of dimension nxl (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.

innov	Distribution of the innovation variable. The values are 'norm' and 't' for normal and Student-t distribution, respectively.
nu	Degrees of freedom for Student-t innovations.

Value

data	Generated response (y), censoring indicator (cc), and lower (lcl) and upper (ucl) bounds of the interval, which contains the true value of the censored observation.
param	Theoretical parameters (beta, sig2, phi).

Note

For data generation with Student-t innovations, the first p observations are not censored.

Author(s)

Fernanda L. Schumacher, Katherine L. Valeriano, Victor H. Lachos, Christian E. Galarza, and Larissa A. Matos

See Also

ARCensReg, ARtCensReg

Examples

```
library(ggplot2)

## Example 1: Generating a sample with normal innovations
set.seed(1234)
dat = rARCens(n=100, beta=c(1,-1), phi=c(.48,-.2), sig2=.5,
              x=cbind(1,runif(100)), cens='left', pcens=.10)

# Plotting the time serie
ggplot(data.frame(dat$data$y), aes(x=1:100, y=dat$data$y)) + geom_line() +
  geom_line(aes(x=1:100, y=dat$data$ucl), color="red", linetype="twodash") +
  labs(x="Time", y=bquote(y["obs"])) + theme_bw()

table(dat$data$cc)

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

## Example 2: Generating a sample with Student-t innovations
set.seed(8278)
dat1 = rARCens(n=100, beta=c(1,-1), phi=c(.48,-.2), sig2=.5,
               x=cbind(1,rnorm(100)), cens='right', pcens=.10,
               innov='t', nu=3)

# Plotting the time serie
ggplot(data.frame(dat1$data$y), aes(x=1:100, y=dat1$data$y)) + geom_line() +
  geom_line(aes(x=1:100, y=dat1$data$lcl), color="red", linetype="twodash") +
```

```
  labs(x="Time", y=bquote(y["obs"])) + theme_bw()

dat1$param
#[1]  1.00 -1.00  0.50  0.48 -0.20  3.00
```

`residuals` *Extract model residuals from ARpCRM or ARtpCRM objects*

Description

The conditional residuals are obtained by subtracting the fitted values from the response vector, while the quantile residuals are obtained by inverting the estimated distribution function for each observation to obtain approximately normally distributed residuals. See, for instance, Dunn and Smyth (1996) and Kalliovirta (2012).

Usage

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

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

Arguments

`object`	An object inheriting from class `ARpCRM` or `ARtpCRM`, representing a fitted AR(p) censored linear model.
`...`	Further arguments passed to or from other methods.

Value

An object of class "residARpCRM", with the following components:

`residuals`	Vector with the conditional residuals of length n.
`quantile.resid`	Vector with the quantile residuals of length n.

Generic function `plot` has methods to show a graphic of residual vs. time, an autocorrelation plot, a histogram, and Quantile-Quantile (Q-Q) plot for the quantile residuals.

Author(s)

Fernanda L. Schumacher, Katherine L. Valeriano, Victor H. Lachos, Christian E. Galarza, and Larissa A. Matos

References

Dunn PK, Smyth GK (1996). "Randomized quantile residuals." *Journal of Computational and Graphical Statistics*, **5**(3), 236–244.

Kalliovirta L (2012). "Misspecification tests based on quantile residuals." *The Econometrics Journal*, **15**(2), 358–393.

See Also

ARCensReg, ARtCensReg

Examples

```
## Example 1: Generating data with normal innovations
set.seed(93899)
x = cbind(1, runif(300))
dat1 = rARCens(n=300, beta=c(1,-1), phi=c(.48,-.2), sig2=.5, x=x,
              cens='left', pcens=.05, innov="norm")

# Fitting the model with normal innovations
mod1 = ARCensReg(dat1$data$cc, dat1$data$lcl, dat1$data$ucl, dat1$data$y,
                 x, p=2, tol=0.001)
mod1$tab
plot(residuals(mod1))

# Fitting the model with Student-t innovations
mod2 = ARtCensReg(dat1$data$cc, dat1$data$lcl, dat1$data$ucl, dat1$data$y,
                  x, p=2, tol=0.001)
mod2$tab
plot(residuals(mod2))


## Example 2: Generating heavy-tailed data
set.seed(12341)
x = cbind(1, runif(300))
dat2 = rARCens(n=300, beta=c(1,-1), phi=c(.48,-.2), sig2=.5, x=x,
              cens='left', pcens=.05, innov="t", nu=3)

# Fitting the model with normal innovations
mod3 = ARCensReg(dat2$data$cc, dat2$data$lcl, dat2$data$ucl, dat2$data$y,
                 x, p=2, tol=0.001)
mod3$tab
plot(residuals(mod3))

# Fitting the model with Student-t innovations
mod4 = ARtCensReg(dat2$data$cc, dat2$data$lcl, dat2$data$ucl, dat2$data$y,
                  x, p=2, tol=0.001)
mod4$tab
plot(residuals(mod4))
```

---

summary *Summary of an ARpCRM or ARtpCRM object*

Description

summary method for class "ARpCRM" or "ARtpCRM".

Usage

```
  ## S3 method for class 'ARpCRM'
summary(object, ...)

  ## S3 method for class 'ARtpCRM'
summary(object, ...)
```

Arguments

object	An object inheriting from class ARpCRM or ARtpCRM, representing a fitted censored autoregressive model of order p.
...	Additional arguments.

Author(s)

Fernanda L. Schumacher, Katherine L. Valeriano, Victor H. Lachos, Christian E. Galarza, and Larissa A. Matos

See Also

ARCensReg, ARtCensReg, print, plot

Examples

```
n = 80; x = rep(1, n)
dat = rARCens(n=n, beta=2, phi=.6, sig2=.3, x=x, cens='right', pcens=.05,
              innov='t', nu=4)

fit = ARtCensReg(dat$data$cc, dat$data$lcl, dat$data$ucl, dat$data$y, x,
                 M=5, pc=.12, tol=0.001)
summary(fit)
```

Index