# Package 'cents'

February 19, 2015

**Version** 0.1-41

**Date** 2014-08-26

**Title** Censored time series

**Author** A.I. McLeod, Nagham M. Mohammad, Justin Veenstra and Abdel El-Shaarawi

**Maintainer** A.I. McLeod <aim@stats.uwo.ca>

**Depends** R (>= 2.1.0)

**Description** Fit censored time series

**LazyLoad** yes

**LazyData** yes

**License** GPL (>= 2)

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2014-08-26 22:14:53

## R topics documented:

---

`ar1est0` *Exact mle ar-parameter in AR(1) with known mean zero.*

---

## Description

An efficient exact algorithm.

## Usage

```
ar1est0(z)
```

## Arguments

| | |
|---|---|
| z | the time series |

## Details

More details later.

## Value

MLE estimate of phi

## Note

Modification of the algorithm in the mleur package.

## Author(s)

A. I. McLeod

## References

Zhang, Yu and McLeod (2013)

## Examples

```
z <- arima.sim(model=list(ar=0.8), n=100)
ar1est0(z)
```

## boot.Cenarma *parametric bootstrap for Cenarma object*

### Description

Simulates a time series from the model fit by cenarma() and produces an object of class 'cents'.

### Usage

```
boot.Cenarma(obj, ...)
```

### Arguments

| | |
|---|---|
| obj | the output from cenarma() |
| ... | optional arguments, not used |

### Value

Simulated time series as a cents object

### See Also

cenarma, rcarma, plot.cents

### Examples

```
set.seed(43137)
n <- 200
z <- arima.sim(model=list(ar=0.8), n)
i <- sample(1:n, size=floor(0.25*n))
z[i] <- NA
ANS <- cenarma(y=z, p=1)
out <- boot.Cenarma(ANS)
cenarma(out$y, p=1)
```

## cenarma *Censored arma estimation*

### Description

A quasi-EM algorithm is implemented. The R function arima() is used in the maximization step. The Durbin-Levinson recursions are used to compute conditional expectations.

### Usage

```
cenarma(y, iy, p=0, q=0, include.mean = TRUE, verbose = FALSE,
  MaxIter = 100, ETOL = 1e-05, algorithm = c("exact", "approx"), ...)
```

## Arguments

| | |
|---|---|
| y | time series as a vector of length n |
| iy | indicator with entries: "o","L","R","na". If missing, it is assumed there is no censoring and iy entries are set to "o" or "na" according to whether the corresponding value in y is numeric or NA. |
| p | ar order |
| q | ma order |
| include.mean | Default is to estimate the mean. FALSE means we assume the mean is zero. |
| verbose | If true, show successive log-likelihoods |
| MaxIter | maximum number of iterations |
| ETOL | error tolerance |
| algorithm | "exact" uses our tacvfARMA() and approximate uses acfARMA() |
| ... | options passed to arima |

## Value

fitted model out is a list:

| | |
|---|---|
| Arima | the output for the function arima() |
| v0 | covariance matrix of the parameters |
| dataSummary | number of data values in each class |
| exitStatus | "converged" or "Maxit iterations reached" |

## Examples

```
#Default example
#Example. Left-censoring, 10%
## Not run:
set.seed(313177)
n <- 500
out <- rcarma(n, ar=0.8, ma=-0.6, mu=100, siga=15, rates=c(0.1, NA))
y <- out$y
iy <- out$iy
ans <- cenarma(y, iy, p=1, q=1)
ans[[1]]
#
#Example ARMA(1,1) with missing values.
#Fit using arima() and cenarma()
#compare final relative likelihood and difference log-likelihoods
set.seed(313177)
n <- 500
out <- rcarma(n, ar=0.8, ma=-0.6, mu=100, siga=15, rates=c(NA, NA), Mrate=0.25)
y <- out$y
iy <- out$iy
(ans0 <- arima(y, order=c(1,0,1)))
(ans1 <- cenarma(y, iy, p=1, q=1))[[1]]
logL0 <- ans0$loglik
```

```
betaHat <- coef(ans1[[1]])
arHat <- betaHat[1]
maHat <- betaHat[2]
muHat <- betaHat[3]
ans1B <- arima(y, order=c(1,0,1), fixed=c(arHat,maHat,muHat),transform.pars=FALSE)
logL1 <- ans1B$loglik
RL <- exp(logL1-logL0)
RL
logL1-logL0

## End(Not run)
```

---

CM *Censored mean and standard deviation in normal samples*

---

## Description

An exact algorithm is used.

## Usage

```
CM(y, iy)
```

## Arguments

| | |
|---|---|
| y | data vector |
| iy | indicator with entries: "o","L","R","na". If missing, it is assumed there is no censoring and iy entries are set to "o" or "na" according to whether the corresponding value in y is numeric or NA. |

## Details

More details later.

## Value

a list

## Author(s)

A. I. McLeod

## References

later

## See Also

cenarma

## Examples

```
z <- log(NiagaraToxic$toxic)
iz <- c("o", "L")[1+NiagaraToxic$cQ]
CM(z,iz)
cenarma(z, iz, p=0, q=0)
```

---

Erf *Error functions and its complement*

---

## Description

Needed for compatibilty with Mathematica

## Usage

```
Erf(z)
Erfc(z)
```

## Arguments

| | |
|---|---|
| z | standard normal variable |

## Value

probability that a standard normal random variable is less than z or its complementary probability

## See Also

EZL

## Examples

```
Erf(1.96)
Erfc(1.96)
```

## EZL *Expected value in left or right truncated normal distribution*

### Description

The formula was obtained using Mathematica.

### Usage

```
EZL(zmu, zsig, c)
EZR(zmu, zsig, c)
```

### Arguments

| | |
|---|---|
| zmu | mean |
| zsig | standard deviation |
| c | detection limit (DL) |

### Value

expected value from a left/right truncated normal

### Examples

```
EZR(100,15,80)
EZL(100,15,105)
```

## fitar1 *Exact MLE for mean and AR-parmeter in AR(1)*

### Description

later

### Usage

```
fitar1(z, meanZeroQ = FALSE)
```

### Arguments

| | |
|---|---|
| z | time series data vector |
| meanZeroQ | default assumes mean is not zero. |

### Details

later

## Value

vector of length 3 with the estimates of mean, ar-parameter and the optimized log-likelihood

## Note

later

## Author(s)

A.I. McLeod

## References

later

## See Also

ar1est0

## Examples

```
z <- arima.sim(model=list(ar=0.8), n=100)
fitar1(z)
```

---

# fitcar1 *Fit AR(1) model with censoring and missing values*

---

## Description

later

## Usage

```
fitcar1(z, cL = -Inf, cU = Inf, verboseQ = FALSE)
```

## Arguments

| | |
|---|---|
| z | a time series vector including possible NA values |
| cL | lower censor point |
| cU | upper censor point |
| verboseQ | show iterations |

## Details

later

## Value

vector of length 3 with the estimates of mean, ar-parameter and the optimized log-likelihood

## Note

later

## Author(s)

A. I. McLeod

## References

later

## See Also

fitar1

## Examples

```
z <- arima.sim(model=list(ar=0.8), n=100)
z[50] <- NA
fitcar1(z)
```

---

# NiagaraToxic — *Successive readings of a toxic substance in the Niagara River near Fort Erie, Ontario.*

---

## Description

Niagara River at Fort Erie, successive readings of 12-Dichloro in units of ng/L measured approximately biweekly.

## Usage

```
data(NiagaraToxic)
```

## Format

A data frame with 144 observations on the following 4 variables.

`toxic` a numeric vector

`jday` a numeric vector

`cQ` a numeric vector

`cL` a numeric vector

## Details

Dr. Abdel El-Shaarwai provided through Environment Canada some a special water quality time series that is of great practical interest. The time series is from Station ON02HA0019 (Fort Erie) on the water quality of the Niagara River. There are more than 500 water quality parameters or variables of interest in this river. The water quality in this river is montiored by a joint U.S./Canada committee. One important toxic variable of great interest is a chemical known as 12-Dichloro which when dissolved in water is measured in units of ng/L. We use a portion of the recent data on this variable that was measured approximately every two weeks over the period from March 1, 2001. This period was chosen because it is the most recent period over which we have a time series of approximately biweekly observations. The time series plot below plots the Julian day number defined so that Julian day number 1 corresponds to the date of the first observation (March 1, 2001).

In total there are 144 values and the data are left censored. The observed censoring rate is r=21/144=14 After March 24, 2005 the detection level for 12 Dichloro dropped from 0.214 to 0.0878. After this change there was only one censored value at Julian day number 1807. Before the change in censoring there were 75 complete observations and 20 censored ones while from March 24, 2005 to the last observation on March 22, 2007 there were 48 complete observations and only one censored observation.

## Source

Dr. Abdel El-Shaarwai, Environment Canada

## References

N. M. Mohammad (2014). Censored time series analysis. Ph.D. Thesis, Western University.

## Examples

```
data(NiagaraToxic)
head(NiagaraToxic)

#Example from thesis
## Not run:
#Diagnostic checks and bootstrap confidence intervals
Zdf <- NiagaraTmeans <- apply(A, MARGIN=2, fun=mean)oxic
z <- log(Zdf$toxic)
iz <- c("o", "L")[1+Zdf$cQ]
#
#CENARMA(1,1)
cenarma(z, iz, p=1, q=1)
#fit CENAR(1)
cenarma(z, iz, p=1)
#
#diagnostic checks########
#test CENARMA(1,1)
SimModel <- function(OUTCENARMA){
  outSim <- boot.Cenarma(OUTCENARMA)
}
FitModel <- function(outSim){
  z <- outSim$y
```

```
  iz <- outSim$iy
  ans <- cenarma(z, iz, p=1, q=1)
  res <- resid(ans$outarima)
  list(res=res)
}
OUTCENARMA <- cenarma(y=NiagaraToxic$toxic, iy=c("o", "L")[1+NiagaraToxic$cQ], p=1, q=1)
func <- list(SimModel=SimModel, FitModel=FitModel)
start.time <- proc.time()[3]
outp <- portest(OUTCENARMA$outarima, lags=5:25, nslaves=8, NREP=10^3, func=func, test="LjungBox")
total.time <- proc.time()[3]-start.time
total.time
plot(outp[,1], outp[,4], xlab="lag", ylab="P-Value", cex=1.5, col="blue", pch=18, ylim=c(0,1))
abline(col="red", h=0.05)
#
#test CENAR(1)
SimModel <- function(OUTCENARMA){
  boot.Cenarma(OUTCENARMA)
}
FitModel <- function(outSim){
  z <- outSim$y
  iz <- outSim$iy
  ans <- cenarma(z, iz, p=1)
  res <- resid(ans$outarima)
  list(res=res)
}
OUTCENARMA <- cenarma(y=log(NiagaraToxic$toxic), iy=c("o", "L")[1+NiagaraToxic$cQ], p=1)
func <- list(SimModel=SimModel, FitModel=FitModel)
start.time <- proc.time()[3]
outp <- portest(OUTCENARMA$outarima, lags=5:25, nslaves=8, NREP=10^3, func=func, test="LjungBox")
total.time <- proc.time()[3]-start.time
total.time
plot(outp[,1], outp[,4], xlab="lag", ylab="P-Value", cex=1.5, col="blue", pch=18, ylim=c(0,1))
abline(col="red", h=0.05)
#
#bootstrap confidence intervals
#CENARMA(1,1)
OUTCENARMA <- cenarma(y=log(NiagaraToxic$toxic), iy=c("o", "L")[1+NiagaraToxic$cQ], p=1, q=1)
nboot <- 1000
A <- matrix(numeric(nboot*3), ncol=3)
start.time <- proc.time()[3]
for (iboot in 1:nboot){
  out <- boot.Cenarma(OUTCENARMA)
  A[iboot, ] <- coef(cenarma(y=out$y, iy=out$iy, p=1, q=1)$outarima)
}
total.time <- proc.time()[3]-start.time
total.time
means <- apply(A, MARGIN=2, FUN=mean)
means
LO <- apply(A, MARGIN=2, function(x) quantile(x, 0.025))
HI <- apply(A, MARGIN=2, function(x) quantile(x, 0.975))
ansARMA11 <- matrix(c(LO,HI), nrow=3, dimnames=list(c("phi","theta","mu"), c("lo", "hi")))
#CEAR(1)
OUTCENARMA <- cenarma(y=log(NiagaraToxic$toxic), iy=c("o", "L")[1+NiagaraToxic$cQ], p=1)
```

```
nboot <- 1000
A <- matrix(numeric(nboot*2), ncol=2)
start.time <- proc.time()[3]
for (iboot in 1:nboot){
  out <- boot.Cenarma(OUTCENARMA)
  A[iboot, ] <- coef(cenarma(y=out$y, iy=out$iy, p=1)$outarima)
}
total.time <- proc.time()[3]-start.time
total.time
means <- apply(A, MARGIN=2, FUN=mean)
means
LO <- apply(A, MARGIN=2, function(x) quantile(x, 0.025))
HI <- apply(A, MARGIN=2, function(x) quantile(x, 0.975))
ansAR2 <- matrix(c(LO,HI), nrow=2, dimnames=list(c("phi","mu"), c("lo", "hi")))
#summary
ansARMA11
ansAR2

## End(Not run)
```

---

`plot.cents` *Plot method for "cents" object*

---

## Description

a suitable time series plot

## Usage

```
## S3 method for class 'cents'
plot(x, y, xlab = "t", ylab = expression(y[t]), marko = TRUE, ...)
```

## Arguments

| | |
|---|---|
| `x` | cents object |
| `y` | is ignored |
| `xlab` | x-axis label |
| `ylab` | yaxis label |
| `marko` | mark each observation |
| `...` | options |

## Value

plot produced

## See Also

`rcarma`, `summary`

## Examples

```
#Default example
out <- rcarma()
plot(out)
#
#Example: Interval censoring and multiple censor points.
#double left-censoring
#first 100, rate 10% and second 100, rate is 5%
#right censoring, 20%
n <- 200
rates <- matrix(c(rep(0.1, 100), rep(0.05, 100), rep(0.2, 200)), ncol=2)
out <- rcarma(n, ar=0.7, ma=0.4, mu=100, siga=15, rates=rates)
summary(out)
plot(out)
```

---

# print.Cenarma *Plot method for "cents" object*

---

## Description

print method for cenarma() function

## Usage

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

## Arguments

| | |
|---|---|
| x | cents object |
| ... | options |

## Value

plot produced

## See Also

rcarma, summary

## Examples

```
set.seed(321)
n <- 200
z <- arima.sim(model=list(ar=0.8), n)
i <- sample(1:n, size=floor(0.25*n))
z[i] <- NA
cenarma(y=z, p=1)
```

rcarma *Simulate censored arma time series*

## Description

simulated censored time series

## Usage

```
rcarma(n=200, ar=0.9, ma=0.6, mu=100, siga=15, rates=c(0.15, NA), Mrate = 0)
```

## Arguments

| | |
|---|---|
| n | length of series, default 200. |
| ar | ar coefficients |
| ma | ma coefficients |
| mu | mean |
| siga | standard deviation of innovations |
| rates | either a vector of length 2 or a matrix with n rows and 2 columns. In the vector case, the first element indicates the left-censor rate and the second element indicates the right-censor rate. Set to NA is there is no censoring. Interval censored data corresponds to setting both a left-censor rate and right-censor rate. The default setting indicates a left-censor rate 0.15 with no right censoring. The vector case handles single censoring and the matrix case is for multiple censor points. In this case each column indicates the corresponding censoring for each observation. |
| Mrate | fraction of missing values. Default is 0. |

## Value

an object of class 'cents' which is a list with three elements. First element, 'y', is the censored time series. Second element, 'iy', indicates for each observed valued "o", "L", "R", NA according to whether the value is fully observed, left-censored, right-censored, or missing. Third element, 'censorPts', is a matrix with 2 columns indicating the censor point or NA if no censoring is applicable. Note that censorPts does not indicate if the observation was actually censored since this depends on the unknown latent variable. An observation is censored if and only if the corresponding entry in iy is either "L" or "R". The component 'censorPts' is useful for plotting. See example below.

## Examples

```
#Default example
out <- rcarma()
plot(out)
```

scenNID *mle estimation of singly-left-censored data using AS 138*

## Description

mle estimation of mean and standard deviation

## Usage

```
scenNID(y, n, cz, type="L")
```

## Arguments

| | |
|---|---|
| y | fully observed values, less than n if there is censoring |
| n | sample size |
| cz | detection level |
| type | only left censoring at present |

## Value

mle estimates of mean and sd

## Author(s)

A. I. McLeod

## References

M. S. Wolynetz (1979). Algorithm AS 138: Maximum Likelihood Estimation from Confined and Censored Normal Data. Journal of the Royal Statistical Society. Series C (Applied Statistics), Vol. 28, No. 2, pp. 185-195

## Examples

```
n <- 50
cy <- (-1)
z <- rnorm(n)
ind <- z > cy
y <- z[ind]
scenNID(y, n, cy)
```

`se.Cenarma` *Bootstrap se for Cenarma object*

## Description

Bootstrap se for Cenarma object

## Usage

```
se.Cenarma(obj, nBoot = 250)
```

## Arguments

| | |
|---|---|
| `obj` | obj is the output produced by cenarma() |
| `nBoot` | number of bootstrap iterations |

## Value

se

## See Also

`cenarma`, `boot.Cenarma`

## Examples

```
#Example takes about 30 seconds
## Not run:
set.seed(43137)
n <- 200
z <- arima.sim(model=list(ar=0.8), n)
i <- sample(1:n, size=floor(0.25*n))
z[i] <- NA
ANS <- cenarma(y=z, p=1)
startTime <- proc.time()[3]
se.Cenarma(ANS)
totalTime <- proc.time()[3]-startTime
totalTime

## End(Not run)
```

## summary.cents *Summary method for 'cents'*

### Description

The summary indicates the number of observed time series values in each of the classes: "o", "L", "R", or NA.

### Usage

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

### Arguments

| | |
|---|---|
| object | cents object that is produced by rcarma() |
| ... | additional paramaters (ignored) |

### See Also

rcarma

### Examples

```
out <- rcarma()
plot(out)
```

## tacvfARMA *Computes TACVF*

### Description

Computes TACVF

### Usage

```
tacvfARMA(phi = numeric(0), theta = numeric(0), maxlag = 20, useCt = TRUE, sigma2 = 1)
```

### Arguments

| | |
|---|---|
| phi | AR parameter vector |
| theta | MA parameter vector |
| maxlag | computes at lags 0,1,...,maglag |
| useCt | default is to use C interface otherwise use R source |
| sigma2 | innovation variance |

## Details

See McLeod Why better

## Value

vector of length maxlag+1 containing autocovariances at lags 0 to maxlag

## Author(s)

A. I. McLeod

## References

McLeod (1975)

## See Also

ARMAacf

## Examples

```
2+2
```

# Index