Package 'CARMAgeddon'

March 28, 2018

Version 0.3.7

Type Package

Date 2018-01-08

Title MONOCAR and other CARMA models

Maintainer Alexander Tahk <tahk@alum.mit.edu>

Depends R (>= 3.0.0)

Imports stats, methods, Rcpp (>= 0.11.0), RcppParallel, nloptr (>= 1.0.0)

LinkingTo Rcpp, RcppArmadillo (>= 0.6.000.0.0), RcppParallel

Suggests mvtnorm

SystemRequirements GNU make

Description
Functions for fitting and working with multivariate overlapping noisy observation continuous-time autoregression (MONOCAR) models and other CARMA models.

LazyLoad true

LazyData true

License GPL-3

URL http://monocar.tahk.us/

R topics documented:

create.ctdata 2
monocar.estimate 4
monocar.hist 7
PresUnemploy 9
simulate.monocar 9
summary.monocar 11

Index **13**

`create.ctdata` *Create and combine MONOCAR data sets*

Description

Create and combine data sets for estimation of MONOCAR models by `monocar.estimate` and `monocar.hist`

Usage

```
create.ctdata(x, v, t1, t2 = NULL,
              data = NULL,
              series.name = NULL,
              house.name = NULL,
              process.name = NULL,
              transform = NULL,
              transform.var = NULL,
              time.origin = NULL,
              time.unit = NULL,
              exo.data = NULL,
              inclusive.end.date = FALSE)

join.ctdata(...)
```

Arguments

`x`	A numeric vector giving the value of each observation.
`v`	A numeric vectorgiving the (untransformed) variance of each observation.
`t1`	A numeric, date-time, or date vector giving the start time for each observation.
`t2`	An optional numeric, date-time, or date vector giving the end time for each observation. Must be of the same class as `t1`. The default is to assume `t2` equals `t1`, meaning instantaneous observations.
`data`	An optional data frame in which to search for variables.
`series.name`	An optional character, factor, or integer vector identifying the series for each observation. The default value of `NULL` uses the quoted `x` argument.
`house.name`	An optional character, factor, or integer vector identifying the house for each observation. The default is to use the series.
`process.name`	An optional character, factor, or integer vector identifying the process for each observation. The default is to assume all observations belong to the same process.
`transform`	An optional function specifying a transformation to apply to each observation. The function must take two inputs corresponding to vectors of means and variances. The default is leave the observations untransformed. Alternatively, a character object giving the name of a common variance-stabilizing transformation. The two implemented transformations are "arcsin" for binomial-distributed data (which uses $arcsin\left(\left(1-\frac{1}{5000}\right)\left(2\frac{x}{v}-1\right)\right)$ and implied variance $\frac{1}{v}$; this assumes `x` gives the number of successes and `v` gives the sample sizes for each observation) and "sqrt" for Poisson-distributed data (which uses $\sqrt{x+\frac{3}{8}}$ and implied variance $\frac{1}{4}$; `v` is ignored).

`transform.var` An optional function specifying a transformation to apply to the variance of each observation. The function must take two inputs corresponding to vectors of means and variances. The default is leave the variances untransformed or, if `transform` is a character object, the corresponding transformation of the variances for the named variance-stabilizing transformation.

`time.origin` An optional numeric, date-time, or date object of length one specifying origin time to be subtracted from `t1` and `t2`.

`time.unit` An optional numeric (if `t1` is a numeric vector) or difftime (if `t1` is a date-time or date vector) object specifying the units for `t1` and `t2` (i.e., the amount of time that should be considered to be length one.

`exo.data` A vector, data frame, matrix, or formula of possible exogenous variables to be included in the data.

`inclusive.end.date` If `t1` and `t2` are date vectors, whether the end date means the start or end of the day. Thus, for an observation which starts and ends on January 1, a `FALSE` value for `inclusive.end.date` would imply a length of zero hours while a `TRUE` value would imply a length of 24 hours.

`...` Multiple arguments giving objects of class `ct.data.frame`, usually each created by `create.ctdata`, to be joined together. Alternatively, a single list containing multiple objects of class `ct.data.frame`.

Details

See `monocar.estimate` for more details about the MONOCAR model.

Observations are grouped into series, houses, and processes. Series correspond to different elements of the latent process, houses allow for house-specific effects, and processes allow for multiple, independent latent processes that share common parameters (specifically, Θ and Σ as well as, optionally, μ and/or δ; whether the latter are included is controlled by `mu.by.process` and `delta.by.process`). Series are the most fundamental of the three. Houses are usually nested within series or identical to series. In most cases, all observations belong to the same process.

If series are not specified, all observations will be assumed to belong to a single series. Usually this will not be the case. However, even when it is not the case, it may be easier to create separate `ct.data.frame` objects with `create.ctdata` for each series and combine them with `joint.ctdata`.

If houses are not specified, they will be assumed to equal the series (implying series are not further divided into houses). If processes are not specified, all observations will be assumed to be based on a single (usually multivariate) latent process, as is typically the case.

Value

An object of class `ct.data.frame` containing the data.

See Also

`monocar.estimate` and `monocar.hist`.

Examples

```
## Create a simple data set with instantaneous observations every
## day for the first month of 2000
obs <- c(-0.25, 1.91, 0.06, 2.24, -0.48, -1.84, -1, -0.07,
```

```
-0.5, -1.17,  0.32, 0.67, 0.99, 0.39, 1.2, 0.12, -0.7, -0.65,
-1.52, -2.28,  0.98, 1.62, -0.36, 0.94, -1.61, -0.25, -0.76,
-1.81, -1.09, 0.9, -0.79)
obs.time <- seq(as.Date("2000-1-1"), by=1, length.out=length(obs))
obs.var <- rep(1, length(obs))
new.data <- create.ctdata(obs, obs.var, obs.time)
```

`monocar.estimate` *Estimate MONOCAR model*

Description

Estimate a Multivariate overlapping noisy observation continuous-time autoregressive (MONOCAR) model

Usage

```
monocar.estimate(data, init=list(), restrict=list(),
    subset=NULL, exovars=NULL, byexo="series",
    enforce.bounds=TRUE, lower.bounds=NULL, upper.bounds=NULL,
    delta.group=TRUE, verbose=1, mu.by.process=FALSE,
    delta.by.process=FALSE, var.by.house=FALSE, remove.unused=TRUE,
    ...)
```

Arguments

`data`	A data file of class `ct.data.frame` (see `create.ctdata`)
`init`	A list containing initial values for parameters. Omitted items will be guessed. The default is for `theta`, `sigma`, and `alpha` to be identity matrices, one of `mu` or `delta` to be the mean of each series or house (depending on whether `delta.group` is used), and variance-transformation parameters to be `0` for `var.add` and `1` for `var.mult` and `var.pow` (i.e., no transformation).
`restrict`	A list of which parameters to be restricted to their initial values. `TRUE` means restricted to initial value, `FALSE` means unrestricted, `NA` means restricted to equal the previous parameter (useful, for example, to set the variance-transformation parameters to be equal for multiple series. The parameter `sigma` can also be specified as one of the following strings: "diagonal" (meaning only the off-diagonal elements of `sigma` are restricted), "restricted" (meaning all elements of `sigma` are restricted), or "unrestricted" (meaning all elements of `sigma` are unrestricted). The default is for `theta` to be unrestricted, `sigma` to be restricted to be diagonal, `alpha` to be fully restricted, one of `mu` or `delta` to be unrestricted (depending on `delta.group` or whether initial values are explicitly specified), and variance-transformation parameters to be restricted unless initial values are explicitly specified.
`subset`	An optional vector specifying a subset of observations to be used in the fitting process.
`exovars`	An optional formula giving the exogenous variables to be included, a character vector giving the names of exogenous variables to be included from `data`, or an integer vector giving the location of the exogenous variables to be included from `data`. The default is to include no exogenous variables.

`byexo` A string or null value specifying whether exogenous variables are to have a single coefficient, separate coefficients for each series, or separate coefficients for each house. Possible values are `series`, `house` or `NULL`. Default is `NULL`, meaning a single coefficient, common to all series and houses, for each exogenous variable.

`enforce.bounds` A boolean or integer specifying whether to enforce bounds on the variance-transformation parameters during optimization. If equal to `FALSE` or `0`, no bounds are imposed. If equal to `TRUE` or `1`, `var.add` and `var.mult` are restricted to be non-negative, avoiding negative transformed variances. If equal to `2`, `var.pow` is also restricted to be non-negative, ensuring the variance transformation is non-decreasing. The default is `TRUE`.

`lower.bounds` An optional vector of lower bounds for parameters if desired. The default is NULL, meaning no lower bounds other than those implied by `enforce.bounds`.

`upper.bounds` An optional vector of lower bounds for parameters if desired. The default is NULL, meaning no lower bounds.

`delta.group` An optional vector of integers specifying which houses are associated with which series. Alternatively, the logical value `TRUE`, in which case the data will be used to determine which series corresponds to each house, or `NULL`, in which case no attempt to recenter δ will be taken. `delta.group` is used to recenter δ such that the mean of $delta$ within each series is zero with μ adjusted to compensate (thus providing an equivalent parameterization). This only makes sense if houses are nested within series. The value `TRUE` will be treated as `NULL` if houses are not nested within series. The default is `TRUE`.

`verbose` An integer specifying the amount of output. `0` means no output, `1` means a dot (`.`) for each gradient computation and a star (`*`) for each Hessian evaluation, and a `2` means a dot for each likelihood evaluation, a colon (`:`) for each gradient evaluation, and a star for each Hessian evaluation. A `+` indicates an optimization algorithm converged while `-` means it failed to do so. The default is `1`.

`mu.by.process` If multiple processes are specified, a boolean indicating whether the series means should be identical across processes. The default is `FALSE`.

`delta.by.process` If multiple processes are specified, a boolean indicating whether the house means should be identical across processes. The default is `FALSE`.

`var.by.house` Whether variance-transformation parameters should be applied by house or by series. The default is to apply by series.

`remove.unused` Whether to remove unused series and houses, thereby reducing the number of parameters. This can be particularly useful when using a subset of the data that does not contain all houses. Note that the size of the parameters specified in `init` and `restrict` may change as a result.

`...` Additional parameters to be passed to the `monocar.cal`, which carries out the optimization.

Details

Observations are grouped into series, houses, and processes. Series correspond to different elements of the latent process, houses allow for house-specific effects, and processes allow for multiple, independent latent processes that share common parameters (specifically, Θ and Σ as well as, optionally, μ and/or δ; whether the latter are included is controlled by `mu.by.process` and `delta.by.process`). Series are the most fundamental of the three. Houses are usually nested within series or identical to series. In most cases, all observations belong to a single process.

The latent process, x_t is an Ornstein-Uhlenbeck process defined by the stochastic differential equation

$$x_t = \Theta\left(\mu - x_t\right) dt + \Sigma^{\frac{1}{2}} dW_t$$

where Θ describes how each element of the latent process affects itself and other elements of the latent process, Σ is the instantaneous variance-covariance matrix for the error process, μ is the long-run mean of μ, and W_t is a multivariate Wiener process.

The observation equations are defined by

$$y_i = \int_{r_i}^{s_i} x_{s_i,t} dt + \delta_{h_i} + \epsilon_i$$

when $q_i < r_i$ or

$$y_i = x_{s_i,q_i} + \delta_{h_i} + \epsilon_i$$

when $q_i = r_i$ with

$$\epsilon_i \sim \mathcal{N}\left(0, \gamma_{s_i} + \nu_{s_i}\left(\frac{v_i}{\bar{v}_{s_i}}\right)^{\xi_{s_i}}\right)$$

and independent across observations. Here, y_i is the value of observation i, v_i is the untransformed (or naive) variance of the error term for the observation, δ_j is a house-specific effect for house j, h_i is the house of equation i, q_i is the start time for observation i, r_i is the end time for observation i, $\bar{v}_j$ is the variance center for house j, and s_i and h_i index the series and house of observation i. y_i, v_i, q_i, r_i, s_i, and h_i are all known (entered as `x`, `v`, `t1`, and `t2` in `create.ctdata`). If `var.by.house` is `TRUE`, then γ, ν, ξ, and $\bar{v}$ are subscripted by h_i rather than s_i.

The variance centers ($\bar{v}$) are used to improve parameter estimation and interpretation. By default, they are the mean of the untransformed variances for observations in each house. These default to the mean variance of each series or house, but other values can be specified (see `monocar.cal`). Aside from possible computational issues, a change of variance centers does not affect the model once ν and ξ are adjusted appropriately.

For multiple processes, the latent process is replaced by multiple latex processes with common parameters Θ and Σ defined by

$$x_{k,t} = \Theta\left(\mu_k - x_{k,t}\right) dt + \Sigma^{\frac{1}{2}} dW_{k,t}$$

for all k, where k indexes the latent process. The observation equations also likewise adjusted by replacing x_t by $x_{k_i,t}$, where k_i is the latent process observed by observation eqni.

For a structural MONOCAR model, the latent process becomes

$$Ax_t = \Theta\left(\mu - x_t\right) dt + \Sigma^{\frac{1}{2}} dW_t$$

where A is a matrix describing instantaneous, non-gradual effects and is assumed to be the identity matrix by default.

In specifying `init` and `restrict`, the possible parameters to list are `theta` (Θ, an $N \times N$ matrix), `sigma` (Σ, an $n \times n$ matrix), `mu` (μ, an N-element vector), `delta` (δ, an H-element vector), `alpha` (A, an $N \times N$ matrix), `var.add` (γ, an H-element vector), `var.mult` (ν, an H-element vector), and `var.pow` (ξ, an H-element vector), where N is the number of series and H is the number of houses. Parameters can always be omitted from `init` and/or `restrict`, in which case reasonable guesses are taken. If `mu.by.process` is `TRUE`, then `mu` is a $K \times N$ matrix, where K is the number of processes. If `delta.by.process` is `TRUE`, then `delta` should be a $K \times H$ matrix.

The actual estimation is carried out by the internal function `monocar.cal`. See `monocar.cal` for additional arguments which control the estimation process. The arguments `optimizers`, `maxiters`, `tolerances`, `epsilon`, `lower.bounds`, `upper.bounds`, and `var.centers` can be included as arguments to `monocar.estimate`, which passes them to `monocar.cal`.

Value

An object of class monocar.

References

Tahk, A. M. (2015) "A Continuous-Time, Latent-Variable Model of Time-Series Data," *Political Analysis* 23 (2): 278–298.

See Also

summary.monocar, monocar.estimate, and create.ctdata.

Examples

```
## Run a simple MONOCAR model
car.model <- monocar.estimate(data = PresUnemploy)
summary(car.model)

## Run a MONOCAR model with a Bush administration
##   dummy variable added
car.model.bush <-
    monocar.estimate(data = PresUnemploy, exo="Bush")
summary(car.model.bush)

## Run a MONOCAR model on Gallup polls only
car.model.gallup <-
    monocar.estimate(data = PresUnemploy,
                     subset = house == "Gallup" |
                         house == "unemployment")
summary(car.model.gallup)

## Run a MONOCAR model with correlated error processes
car.model.cor <-
    monocar.estimate(data = PresUnemploy,
                     restrict=list(sigma="unrestricted"))
summary(car.model.cor)

## Run a MONOCAR model with correlated error processes and transformed variances
car.model.cor.var <-
    monocar.estimate(data = PresUnemploy,
                     restrict=list(sigma="unrestricted",
                         var.add=c(TRUE,FALSE),
                         var.mult=c(FALSE,TRUE),
                         var.pow=c(FALSE,TRUE)))
summary(car.model.cor.var)
```

---

monocar.hist *Estimate a series history from a MONOCAR model*

Description

Estimate the history of a latent process from a Multivariate overlapping noisy observation continuous-time autoregressive (MONOCAR) model

Usage

```
monocar.hist(data, model, estimates=NULL, exovars=NULL, byexo="series",
  delta.group=TRUE, verbose=1, mu.by.process=FALSE,
  delta.by.process=FALSE, var.by.house=FALSE, ...)
```

Arguments

data	A data file of class ct.data.frame (see create.ctdata)
model	A model of class monocar generated by monocar.estimate. Can be NULL, in which case the specification is taken from estimates and exovars.
estimates	If model is NULL, a list of parameters.
exovars	If model is NULL, an optional formula giving the exogenous variables to be included, a character vector giving the names of exogenous variables to be included from data, or an integer vector giving the location of the exogenous variables to be included from data. The default is to include no exogenous variables.
byexo	If model is NULL, a string or null value specifying whether exogenous variables are to have a single coefficient, separate coefficients for each series, or separate coefficients for each house. Possible values are series, house or NULL. Default is NULL, meaning a single coefficient, common to all series and houses, for each exogenous variable.
delta.group	An optional vector of integers specifying which houses are associated with which series.
verbose	An integer specifying the amount of output. 0 means no output, 1 means a dot (.) for each gradient computation and a star (*) for each hessian evaluation, and a 2 means a dot for each likelihood evaluation, a colon (:) for each gradient evaluation, and a star for each hessian evaluation. The default is 1.
mu.by.process	If multiple processes are specified, a boolean indicating whether the series means should be identical across processes. The default is FALSE.
delta.by.process	If multiple processes are specified, a boolean indicating whether the house means should be identical across processes. The default is FALSE.
var.by.house	Whether variance-transformation parameters should be applied by house or by series. The default is to apply by series.
...	Additional parameters to be passed to the monocar.cal, which carries out the estimation.

References

Tahk, A. M. (2015) "A Continuous-Time, Latent-Variable Model of Time-Series Data," *Political Analysis* 23 (2): 278–298.

See Also

monocar.estimate and create.ctdata.

PresUnemploy — *Presidential Approval and Unemployment in the Reagan and Bush administrations*

Description

This data set gives the monthly unemployment rate and polls of presidential approval during the Reagan and Bush administrations.

Usage

```
PresUnemploy
```

Format

A ct.data.frame object containing 677 observations.

Source

Roper Center's iPoll DataBank and Federal Reserve Economic Data (FRED)

References

Tahk, A. M. (2015) "A Continuous-Time, Latent-Variable Model of Time-Series Data," *Political Analysis* 23 (2): 278–298.

simulate.monocar — *Simulate continuous-time data sets*

Description

Generate simulated continuous-time data sets

Usage

```
## S3 method for class 'monocar'
simulate(object, nsim=1, seed=NULL,
  var=NULL, t1=seq(0,100,1.0), t2=NULL, data=NULL,
  series.name = NULL, house.name = NULL, process.name = NULL,
  transform = function(x, v) x, transform.var = function(x, v) v,
  exo.data=NULL, exovars=NULL, byexo=NULL, var.centers=NULL, ...)
```

Arguments

object	A an object of class monocar or a list specifying the parameters of the model.
nsim	Number of data sets to simulate. Defaults to 1.
seed	An object specifying if and how the random number generator should be initialized. Either NULL or an integer that will be used in a call to set.seed before simulating the response vectors. If set, the value is saved as the seed attribute of the returned value. The default, NULL will not change the random generator state, and return .Random.seed as the seed attribute.

var	A numeric vector giving the variance of the measurement error for each observation. The default value of NULL means zero measurement error for each observation.
t1	A numeric vector giving the start time for each observation.
t2	An optional numeric vector giving the end time for each observation. By default, the end time of each observation equals the start time.
data	An optional data frame or object of class ct.data.frame.
series.name	An optional character, factor, or integer vector identifying the series for each observation. The default value of NULL uses the quoted x argument.
house.name	An optional character, factor, or integer vector identifying the house for each observation. The default is to use the series.
process.name	An optional character, factor, or integer vector identifying the process for each observation. The default is to assume all observations belong to the same process.
transform	An optional function specifying a transformation to apply to each observation. See monocar.data for more details.
transform.var	An optional function specifying a transformation to apply to the variance of each observation. See monocar.data for more details.
exo.data	A vector, data frame, matrix, or formula of possible exogenous variables to be included in the data.
exovars	An optional formula giving the exogenous variables to be included, a character vector giving the names of exogenous variables to be included from data, or an integer vector giving the location of the exogenous variables to be included from data. The default is to include no exogenous variables.
byexo	A string or null value specifying whether exogenous variables are to have a single coefficient, separate coefficients for each series, or separate coefficients for each house. Possible values are series, house or NULL. Default is NULL, meaning a single coefficient, common to all series and houses, for each exogenous variable.
var.centers	For each house, the value around which to recenter the variances. See monocar.cal for more details.
...	Additional optional arguments.

Value

If nsim==1, an object of class ct.data.frame containing the data. Otherwise, a list containing nsim objects of class ct.data.frame, each containing one simulated data set.

See Also

create.ctdata, monocar.estimate, and monocar.hist.

summary.monocar *Print and summarize MONOCAR models*

Description

Print and summary methods for MONOCAR models from monocar.estimate.

Usage

```
## S3 method for class 'monocar'
print(x, reverse.offdiag = TRUE,
    digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'monocar'
summary(object, reverse.offdiag = TRUE, ...)

## S3 method for class 'summary.monocar'
print(x,
     digits = max(3L, getOption("digits") - 3L),
     signif.stars = getOption("show.signif.stars"), ...)
```

Arguments

x	For print.monocar, an object of class monocar. For print.summary.monocar, an object of class summary.monocar.
object	An object of class monocar.
reverse.offdiag	A logical value indicating whether to reverse the sign of off-diagonal elements of Theta. In a MONOCAR model, a positive value of an off-diagonal element of Theta indicates that an increase in one latent process tends to *decrease* the level of another latent process. Reversing the sign of these parameters means that a positive value indicates one process tends to increase another. This can simplify the interpretation of the parameters. The default is to reverse the sign of off-diagonal elements of Theta. For print.summary.monocar, whether these elements are reversed is set by summary.monocar and stored in the summary.monocar object.
digits	The minimal number of significant digits, see print.default.
signif.stars	A logical value indicating whether stars should be printed based on p-values, see printCoefmat.
...	Additional arguments to pass to print.default or printCoefmat.

Value

summary.monocar returns an object of class summary.monocar containing a summary information about the model.

See Also

monocar.estimate and monocar.hist.

Examples

```
## Run a simple MONOCAR model
car.model <- monocar.estimate(data = PresUnemploy)
## Print model estimates
car.model
## Print model estimates with diagonal not reversed
print(car.model, reverse.offdiag=FALSE)
## Print summary information about the model
summary(car.model)
```

Index

∗Topic **datagen**
    simulate.monocar, 9
∗Topic **datasets**
    PresUnemploy, 9
∗Topic **models**
    monocar.estimate, 4
∗Topic **monocar**
    create.ctdata, 2
    monocar.estimate, 4
    monocar.hist, 7
    simulate.monocar, 9
    summary.monocar, 11
∗Topic **ts**
    create.ctdata, 2
    monocar.estimate, 4
    monocar.hist, 7
    simulate.monocar, 9
    summary.monocar, 11

create.ctdata, 2, *4*, *6–8*, *10*

join.ctdata *(*create.ctdata*)*, 2

monocar.cal, *6*, *10*
monocar.data, *10*
monocar.estimate, *3*, 4, *7, 8*, *10, 11*
monocar.hist, *3*, 7, *10, 11*

PresUnemploy, 9
print.monocar *(*summary.monocar*)*, 11
print.summary.monocar
        *(*summary.monocar*)*, 11

simulate.monocar, 9
summary.monocar, *7*, 11