Package ‘tsgui’

September 4, 2020

Version 0.2.1

Title A Gui for Simulating Time Series

Author Martin Schlather [aut, cre],

Maintainer Martin Schlather <`schlather@math.uni-mannheim.de`>

Depends R (>= 3.0)

Imports stats, graphics, methods, grDevices, utils, tcltk2, tcltk, tkrplot, RandomFieldsUtils

Suggests

Description This gui shows realisations of times series, currently ARMA and GARCH processes. It might be helpful for teaching and studying.

License GPL (>= 3)

URL `http://ms.math.uni-mannheim.de/de/publications/software/tsgui`

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-09-04 14:30:03 UTC

R topics documented:

ARMA.model *Definition of an ARMA model.*

Description

`ARMA.model` defines an ARMA model as input for the argument `user` of `tsgui`

Usage

```
ARMA.model(p.max=3, q.max=3)
```

Arguments

`p.max,q.max` non-negative integer. Any ARMA(p,q) model can be modelled in the gui with $p \leq$`p.max` and $q \leq$`q.max`. Note that `p.max` + `q.max` should not be larger than 6. Otherwise gui cannot manage the abundance of parameters.

Value

A list of parameters is returned.

Author(s)

Martin Schlather, `<schlather@math.uni-mannheim.de>`

See Also

`tsgui`, `GARCH.model`

Examples

```
if (interactive())
  tsgui(user=list(AR=ARMA.model(p.max=5, q.max=0)),
        included.models = "ARMA")
```

GARCH.model *Definition of an GARCH model.*

Description

`GARCH.model` defines an GARCH model as input for the argument `user` of `tsgui`

Usage

```
GARCH.model(a.max=3, b.max=3)
```

Arguments

a.max,b.max
non-negative integer. Any GARCH(p,q) model can be modelled in the gui with $p \leq$a.max and $q \leq$b.max. Note that a.max + b.max should not be larger than 6. Otherwise gui cannot manage the abundance of parameters.

Value

A list of parameters is returned.

Author(s)

Martin Schlather, <schlather@math.uni-mannheim.de>

See Also

tsgui, ARMA.model

Examples

```
if (interactive())
  tsgui(user=list(ARCH=GARCH.model(a.max=5, b.max=0)),
        included.models = "GARCH")
```

tsgui *Gui for times series*

Description

This GUI allows real-time simulations of ARMA and GARCH processes. Further models can be added by the user.

Usage

```
tsgui(user = NULL, wait = 1000, included.models =c("ARMA", "GARCH"))
```

Arguments

user
a further class of model (or a list of such models), additional to ARCH(3,3) and GARCH(3,3)

wait
non-negative integer. Users will usually not change this value. if the value is greater than 0, the control over the command line of R is returned to the user only after the tcltk window has been left. tsgui checks every wait / 1000 seconds whether the tcltk window has been left.

included.models
NULL or string vector giving the internally defined models that shall be available.

Details

Sliders and Buttons of the GUI are:

'New innovations' The random seed is fixed when model parameters are changed. 'New Innovations' draws a new random seed

'Return' leaves the program

'MODELS' choose one of the given models to be displayed

'time' length of the time series

'burn.in' the length of the burn in period

'repetitions' number of simulations (with different seeds) shown in the graphics

'INNOVATIONS' the distribution families of the independent errors. If more than one family is chosen the graphs are put on top of each other.

'green button cpy' allows to freeze the current choice of the parameter values. If it is pressed a new column appears with the frozen values. The effect becomes apparent when now the sliders are moved. The new column has a `d` for deleting that column and a `1` to switch this column with the current columns of the sliders.

'MODEL PARAMETERS' The ARMA model is given by

$$Z_t - \phi_1 Z_{t-1} - \phi_2 Z_{t-2} - \phi_3 Z_{t-3} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta 2 \varepsilon_{t-2} + \theta 3 \varepsilon_{t-3}$$

The GARCH model is given by $Z_t = \sigma_t \varepsilon_t$ and

$$\sigma_t^2 - \beta_1 \sigma_{t-1}^2 - \beta_2 \sigma_{t-2}^2 - \beta_3 \sigma_{t-3}^2 = \alpha_0 + \alpha_1 Z_{t-1}^2 + \alpha_2 Z_{t-2}^2$$

'INNOVATION PARAMETERS' the parameters corresponding to the distributions of the INNOVATIONS

'STARTING VALUES' starting values for the burn in period

The argument `user` allows to define ARCH and GARCH models with different numbers of parameters, see the examples in `ARMA.model` and `GARCH.model`.

The argument `user` also allows to define one's own model. The example of such a definition is returned by `ARMA.model`. While `ARMA.model()` returns an overview and `str(ARMA.model(),give.attr=TRUE)` gives some more details, the code `print(ARMA.model(),details=TRUE)` gives the full definition of a time series:

`time`, `burn.in`, `repetitions` the default values of the respective sliders, see above

`*.min`, `*.max` The lower and upper bound of the slider values.

`phi` named vector; the names will be the titles of the sliders; phi contains all model parameters

`distr` note that all the three distribution functions have access to the values of all three INNOVATION PARAMETERS in the gui, since `param` is a vector of three elements

`distr.param` named vector of parameters. It could be also a shorter or longer vector than the standard vector of length 3

`distr.show` at least one of the components must be `TRUE`, indicating the default distribution

`titles` a list of 2 functions

- `titles[[1]]` returns the title of the model in dependence of the current values of the model parameters
- `titles[[2]]` returns a string of additional information on the model

`update.function` for given `innovations`, i.e. realisations of the errors ε_i including burn in, the parameter values ϕ of the model and for given starting values `start.ts` the function calculates the time series, including burn in. Note that `start.ts` is a matrix of `length{param}` rows and nr columns, where n is the number of chosen distribution families and r is the number of repetitions. Similar, `innovations` is also

`starting.ts` the starting values of the process in the burn-in period of the simulation

`starting.matrix.ts` This function turns the starting values `starting.ts` into a matrix where is essentially the starting values `starting.ts` are written in each column. The number of columns are the number of chosen distribution family time the number of repetitions. Note that in the matrix the repetitions are grouped to gether, i.e. if r is the number of repetitions the first r rows refer to the r repetitions of the first chosen distribution.

The function is introduced to allow also for transformations of `starting.ts` for different distributions.

`starting.innovations` the starting values of the innovations in the burn-in period of the simulation

`starting.matrix.innovations` function analogue to `starting.matrix.ts`

Value

A list with the latest chosen paremeters is returned.

Author(s)

Martin Schlather, <schlather@math.uni-mannheim.de>

See Also

`ARMA.model`, `GARCH.model`

Examples

```
if (interactive()) tsgui()
```

Index