

Package ‘garma’

August 19, 2023

Type Package

Title Fitting and Forecasting Gegenbauer ARMA Time Series Models

Version 0.9.13

Date 2023-08-19

Maintainer Richard Hunt <maint@huntemail.id.au>

Description Methods for estimating univariate long memory-seasonal/cyclical Gegenbauer time series processes. See for example (2022) <[doi:10.1007/s00362-022-01290-3](https://doi.org/10.1007/s00362-022-01290-3)>. Refer to the vignette for details of fitting these processes.

License GPL-3

URL <https://github.com/r1ph50/garma>

Encoding UTF-8

Depends forecast, ggplot2

Imports Rsolnp, pracma, signal, zoo, lubridate, crayon, utils, nloptr, BB, GA, dfoptim, pso, FKF, tswge, hypergeo, ltsa

Suggests longmemo, yardstick, testthat, knitr, rmarkdown

RoxygenNote 7.2.3

VignetteBuilder knitr

NeedsCompilation no

Author Richard Hunt [aut, cre]

Repository CRAN

Date/Publication 2023-08-19 12:52:32 UTC

R topics documented:

AIC.garma_model	2
autoplot.garma_model	3
coef.garma_model	3
extract_arma	4
fitted.garma_model	5
forecast.garma_model	5

garma_ggtsdisplay	6
ggbr_semipara	7
gg_raw_pgram	8
gof	8
logLik.garma_model	9
plot.garma_model	9
predict.garma_model	10
predict2	11
print.garma_model	11
print.garma_semipara	12
print.ggbr_factors	12
residuals.garma_model	13
summary.garma_model	13
tsdiag.garma_model	14
vcov.garma_model	15
version	15

Index	16
--------------	-----------

AIC.garma_model	<i>AIC for model</i>
-----------------	----------------------

Description

AIC for model if available.

Usage

```
## S3 method for class 'garma_model'
AIC(object, ...)
```

Arguments

object	The garma_model object
...	Other parameters. Ignored.

Value

(double) Approximate AIC - uses approximation of whichever method is used to find model params.

autoplot.garma_model *ggplot of the Forecasts of the model.*

Description

The ggplot function generates a ggplot of actuals and predicted values for a "garma_model" object. This adds in sensible titles etc as best it can determine.

Usage

```
## S3 method for class 'garma_model'
autoplot(object, h = 24, include_fitted = FALSE, ...)
```

Arguments

object (garma_model) The garma_model from which to ggplot the values.
 h (int) The number of time periods to predict ahead. Default: 24
 include_fitted (bool) whether to include the 1-step ahead 'fitted' values in the plot. Default: FALSE
 ... other parameters passed to ggplot.

Value

A ggplot2 "ggplot" object. Note that the standard ggplot2 "+" notation can be used to enhance the default output.

Examples

```
library(ggplot2)

data(AirPassengers)
ap <- as.numeric(diff(AirPassengers,12))
mdl <- garma(ap,order=c(9,1,0),k=0,method='CSS',include.mean=FALSE)
autoplot(mdl)
```

coef.garma_model *Model Coefficients*

Description

Model Coefficients/parameters.

Usage

```
## S3 method for class 'garma_model'
coef(object, ...)
```

Arguments

object The gamma_model object
 ... Other parameters. Ignored.

Value

(double) array of parameter value estimates from the fitted model.

extract_arma *Extract underlying ARMA process.*

Description

For a Gegenbauer process, transform to remove Gegenbauer long memory component to get a short memory (ARMA) process.

Usage

```
extract_arma(x, ggbr_factors)
```

Arguments

x (num) This should be a numeric vector representing the Gegenbauer process.
 ggbr_factors (class) Each element of the list represents a Gegenbauer factor and includes f, u and fd elements.

Value

An object of same class as x.

Examples

```
data(AirPassengers)
ap <- as.numeric(diff(AirPassengers,12))
# find semiparametric estimates of the Gegenbauer parameters.
sp <- ggbr_semipara(ap)
# extract the underlying short-memory ARMA process
ap_arma <- extract_arma(ap,sp$ggbr_factors)
summary(arima(ap_arma,order=c(1,0,0)))
```

fitted.garma_model *Fitted values*

Description

Fitted values are 1-step ahead predictions.

Usage

```
## S3 method for class 'garma_model'
fitted(object, ...)
```

Arguments

object The garma_model object
 ... Other parameters. Ignored.

Value

(double) array of 1-step ahead fitted values for the model.

forecast.garma_model *Forecast future values.*

Description

The forecast function predicts future values of a "garma_model" object, and is exactly the same as the "predict" function with slightly different parameter values.

Usage

```
## S3 method for class 'garma_model'
forecast(object, h = 1, ...)
```

Arguments

object (garma_model) The garma_model from which to forecast the values.
 h (int) The number of time periods to predict ahead. Default: 1
 ... Other parameters passed to the forecast function. For "garma_model" objects, these are ignored.

Value

- a "ts" object containing the requested forecasts.

Examples

```
library(forecast)

data(AirPassengers)
ap <- as.numeric(diff(AirPassengers,12))
mdl <- garma(ap,order=c(9,1,0),k=0,method='CSS',include.mean=FALSE)
forecast(mdl, h=12)
```

garma_ggtsdisplay *ggtsdisplay of underlying ARMA process.*

Description

For a Gegenbauer process, use semi-parametric methods to obtain short memory version of the process, then run a ggtsdisplay().

Usage

```
garma_ggtsdisplay(x, k = 1, ...)
```

Arguments

x	(num) This should be a numeric vector representing the process to estimate.
k	(int) The number of Gegenbauer factors
...	additional parameters to pass to ggtsdisplay

Details

The purpose of this function is to ease the process of identifying the underlying short memory process.

Value

A ggplot object.

Examples

```
data(AirPassengers)
ap <- as.numeric(diff(AirPassengers,12))
garma_ggtsdisplay(ap)
```

`ggbr_semipara`*Extract semiparametric estimates of the Gegenbauer factors.*

Description

For a Gegenbauer process, use semi-parametric methods to estimate the Gegenbauer frequency and fractional differencing.

Usage

```
ggbr_semipara(  
  x,  
  k = 1,  
  alpha = 0.8,  
  method = "lpr",  
  min_freq = 0,  
  max_freq = 0.5  
)
```

Arguments

<code>x</code>	(num) This should be a numeric vector representing the process to estimate.
<code>k</code>	(int) The number of Gegenbauer frequencies
<code>alpha</code>	(num)
<code>method</code>	(char) One of "gsp" or "lpr" - lpr is the log-periodogram-regression technique, "gsp" is the Gaussian semi-parametric technique. "lpr" is the default. Refer Arteche (1998).
<code>min_freq</code>	(num) The minimum frequency to search through for peaks - default 0.0.
<code>max_freq</code>	(num) The maximum frequency to search through for peaks - default 0.5.

Value

An object of class "gamma_semipara".

Examples

```
data(AirPassengers)  
ap <- as.numeric(diff(AirPassengers,12))  
sp <- ggbr_semipara(ap)  
print(sp)
```

`gg_raw_pgram`*Display raw periodogram*

Description

Display the raw periodogram for a time series, and not on a log scale.

Usage

```
gg_raw_pgram(x, k = 1)
```

Arguments

`x` (num) This should be a numeric vector representing the process to estimate.
`k` (int) The number of Gegenbauer factors

Details

The standard "R" functions display periodograms on a log scale which can make it more difficult to locate high peaks in the spectrum at differing frequencies. This routine will display the peaks on a raw scale.

Value

A ggplot object representing the raw periodogram

Examples

```
data(AirPassengers)
ap <- as.numeric(diff(AirPassengers,12))
sp <- ggbr_semipara(ap)
print(sp)
```

`gof`*Goodness-of-Fit test for a gamma_model.*

Description

Provides a goodness-of-fit test for a GARMA Model, using Bartlett's T_p test. This has been justified for long memory and for GARMA models by Delgado, Hidalgo and Velasco (2005).

Usage

```
gof(object)
```


Arguments

object (garma_model) The garma_model to test.

Value

Invisibly returns the array of p-values from the test.

logLik.garma_model *Log Likelihood*

Description

Log Likelihood, or approximate likelihood or part likelihood, depending on the method.

Usage

```
## S3 method for class 'garma_model'
logLik(object, ...)
```

Arguments

object The garma_model object
 ... Other parameters. Ignored.

Value

Object of class "logLik" with values for the (approx) log-likelihood for the model

plot.garma_model *Plot Forecasts from model.*

Description

The plot function generates a plot of actuals and predicted values for a "garma_model" object.

Usage

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

Arguments

x (garma_model) The garma_model from which to plot the values.
 ... other arguments to be passed to the "plot" function, including h (int) - the number of periods ahead to forecast.

Value

An R "plot" object.

Examples

```
data(AirPassengers)
ap <- as.numeric(diff(AirPassengers,12))
mdl <- garma(ap,order=c(9,1,0),k=0,method='CSS',include.mean=FALSE)
plot(mdl)
```

predict.garma_model *Predict future values.*

Description

Predict ahead using algorithm of (2009) Godet, F "Linear prediction of long-range dependent time series", ESAIM: PS 13 115-134. DOI: 10.1051/ps:2008015

Usage

```
## S3 method for class 'garma_model'
predict(
  object,
  n.ahead = 1,
  max_wgts = length(object$diff_y),
  ggbr_scale = FALSE,
  ...
)
```

Arguments

object	(garma_model) The garma_model from which to predict the values.
n.ahead	(int) The number of time periods to predict ahead. Default: 1
max_wgts	(int) The number of past values to use when forecasting ahead. By default, all available data is used.
ggbr_scale	(logical) - whether or not to scale the Gegenbauer weights to add up to 1. By default this is FALSE.
...	Other parameters. Ignored.

Value

A "ts" object containing the requested forecasts.

Examples

```
data(AirPassengers)
ap <- as.numeric(diff(AirPassengers,12))
mdl <- garma(ap,order=c(9,1,0),k=0,method='CSS',include.mean=FALSE)
predict(mdl, n.ahead=12)
```

predict2	<i>Predict2 future values.</i>
----------	--------------------------------

Description

Predict ahead using algorithm of (2009) Godet, F "Linear prediction of long-range dependent time series", ESAIM: PS 13 115-134. DOI: 10.1051/ps:2008015

Usage

```
predict2(object, n.ahead = 1)
```

Arguments

object	(garma_model) The garma_model from which to predict the values.
n.ahead	(int) The number of time periods to predict ahead. Default: 1

Value

A "ts" object containing the requested forecasts.

print.garma_model	<i>print a garma_model object.</i>
-------------------	------------------------------------

Description

The print function prints a summary of a "garma_model" object, printed to the output.

Usage

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

Arguments

x	(garma_model) The garma_model from which to print the values.
...	Other arguments. Ignored.

Value

(null)

Examples

```
data(AirPassengers)
ap <- as.numeric(diff(AirPassengers,12))
mdl <- gamma(ap,order=c(9,1,0),k=0,method='CSS',include.mean=FALSE)
print(mdl)
```

print.garma_semipara *Print Semiparametric Estimates*

Description

Print a semiparameteric Gegenbauer estimation object.

Usage

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

Arguments

x An object of class gamma_semipara.
... further parameters for print function

Value

null

print.ggbr_factors *Print a 'ggbr_factors' object.*

Description

Print a 'ggbr_factors' object.

Usage

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

Arguments

x An object of class ggbr_factors
 ... further parameters for print function

Value

null

residuals.garma_model *Residuals*

Description

Response Residuals from the model.

Usage

```
## S3 method for class 'garma_model'
residuals(object, type = "response", h = 1, ...)
```

Arguments

object The garma_model object
 type (chr) The type of residuals. Must be 'response'.
 h (int) The number of periods ahead for the residuals. Must be 1.
 ... Other parameters. Ignored.

Value

(double) array of residuals from the model.

summary.garma_model *summarise a garma_model object.*

Description

The summary function provides a summary of a "garma_model" object, printed to the output.

Usage

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

Arguments

object (garma_model) The garma_model from which to print the values.
 ... Other arguments. Ignored.

Value

(null)

Examples

```
data(AirPassengers)
ap <- as.numeric(diff(AirPassengers,12))
mdl <- garma(ap,order=c(9,1,0),k=0,method='CSS',include.mean=FALSE)
summary(mdl)
```

tsdiag.garma_model *Diagnostic fit of a garma_model.*

Description

Produces diagnostic plots of the model fit. This function is copied from stats::tsdiag but modifies the fit_df for the Ljung-Box test for use with garma models.

Usage

```
## S3 method for class 'garma_model'
tsdiag(object, gof.lag = 10, ...)
```

Arguments

object (garma_model) The garma_model to produce the diagnostic plots for.
 gof.lag (int) The number of lags to examine for the Ljung-Box white noise test.
 ... further arguments to be passed to particular methods.

Value

None. Diagnostics are generated.

See Also

The stats package tsdiag function: <https://stat.ethz.ch/R-manual/R-patched/library/stats/html/tsdiag.html>.

Examples

```
data(AirPassengers)
ap <- as.numeric(diff(AirPassengers,12))
mdl <- garma(ap,order=c(9,1,0),k=0,method='CSS',include.mean=FALSE)
tsdiag(mdl)
```

vcov.garma_model	<i>Covariance matrix</i>
------------------	--------------------------

Description

Covariance matrix of parameters if available

Usage

```
## S3 method for class 'garma_model'  
vcov(object, ...)
```

Arguments

object	The garma_model object
...	Other parameters. Ignored.

Value

(double) estimated variance-covariance matrix of the parameter estimates

version	<i>garma package version</i>
---------	------------------------------

Description

The version function returns the garma package version.

Usage

```
version()
```

Value

The package version.

Examples

```
library(garma)  
garma::version()
```

Index

AIC.garma_model, 2
autoplot.garma_model, 3

coef.garma_model, 3

extract_arma, 4

fitted.garma_model, 5
forecast.garma_model, 5

garma_ggtsdisplay, 6
gg_raw_pgram, 8
ggbr_semipara, 7
gof, 8

logLik.garma_model, 9

plot.garma_model, 9
predict.garma_model, 10
predict2, 11
print.garma_model, 11
print.garma_semipara, 12
print.ggbr_factors, 12

residuals.garma_model, 13

summary.garma_model, 13

tsdiag.garma_model, 14

vcov.garma_model, 15
version, 15