Package ‘gctsc’

May 20, 2026

Title Gaussian and Student-t Copula Models for Count Time Series

Version 0.2.4

Description Provides likelihood-based inference for Gaussian and Student-t copula models for univariate count time series. Supports Poisson, negative binomial, binomial, beta-binomial, and zero-inflated marginals with ARMA dependence structures. Includes simulation, maximum-likelihood estimation, residual diagnostics, and predictive inference. Implements Time Series Minimax Exponential Tilting (TMET) <doi:10.1016/j.csda.2026.108344>, an adaptation of minimax exponential tilting of Botev (2017) <doi:10.1111/rssb.12162>. Also provides a linear-cost implementation of the Geweke–Hajivassiliou–Keane (GHK) simulator following Masarotto and Varin (2012) <doi:10.1214/12-EJS721>, and the Continuous Extension (CE) approximation of Nguyen and De Oliveira (2025) <doi:10.1080/02664763.2025.2498502>. The package follows the S3 design philosophy of 'gcmr' but is developed independently.

License MIT + file LICENSE

Imports Rcpp, Matrix, TruncatedNormal, VGAM, truncnorm, nlme, car

LinkingTo Rcpp, RcppArmadillo

URL `https://github.com/QNNHU/gctsc`

BugReports `https://github.com/QNNHU/gctsc/issues`

Encoding UTF-8

VignetteBuilder knitr

RoxygenNote 7.3.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 3.5)

LazyData false

NeedsCompilation yes

Author Quynh Nguyen [aut, cre],
Victor De Oliveira [aut]

Maintainer Quynh Nguyen <`nqnhu2209@gmail.com`>

Repository CRAN

Date/Publication 2026-05-20 20:00:02 UTC

Contents

`gctsc-package` *gctsc: Copula Count Time Series Models*

Description

Provides likelihood-based inference for Gaussian and Student-t copula models for univariate count time series. Supports Poisson, negative binomial, binomial, beta-binomial, and zero-inflated marginals with ARMA dependence structures. Includes simulation, maximum-likelihood estimation, residual diagnostics, and predictive inference. Implements Time Series Minimax Exponential Tilting (TMET) doi:10.1016/j.csda.2026.108344, an adaptation of minimax exponential tilting of Botev (2017) doi:10.1111/rssb.12162. Also provides a linear-cost implementation of the Geweke–Hajivassiliou–Keane (GHK) simulator following Masarotto and Varin (2012) doi:10.1214/12EJS721, and the Continuous Extension (CE) approximation of Nguyen and De Oliveira (2025) doi:10.1080/02664763.2025.2498502. The package follows the S3 design philosophy of 'gcmr' but is developed independently.

Author(s)

Maintainer: Quynh Nguyen <`nqnhu2209@gmail.com`>

Authors:

- Victor De Oliveira

See Also

Useful links:

- `https://github.com/QNNHU/gctsc`
- Report bugs at `https://github.com/QNNHU/gctsc/issues`

`arma.cormat` *ARMA Correlation Structure for Copula Count Time Series Models*

Description

Constructs an ARMA(p, q) correlation structure for use in Gaussian and Student–t copula count time series models.

Usage

```
arma.cormat(p = 0, q = 0, tau.lower = NULL, tau.upper = NULL)
```

Arguments

`p`	Non-negative integer specifying the autoregressive (AR) order.
`q`	Non-negative integer specifying the moving-average (MA) order. The model ARMA(0,0) is not supported.
`tau.lower`	Optional numeric vector of length `p + q` specifying lower bounds for the ARMA parameters.
`tau.upper`	Optional numeric vector of length `p + q` specifying upper bounds for the ARMA parameters.

Details

The ARMA model specifies the dependence structure of the latent copula process.

The ARMA parameters must define a stationary and invertible process. These conditions are enforced during model fitting.

Value

An object of class `"arma.gctsc"` and `"cormat.gctsc"` containing:

- `npar`: Number of ARMA parameters ($p + q$).
- `od`: Integer vector `c(p, q)`.
- `start`: Function to compute starting values from data, typically using `arima`.
- `lower`, `upper`: Parameter bounds.

See Also

`gctsc`, `poisson.marg`, `predict.gctsc`

`campyl` *Weekly Campylobacter case counts across Germany*

Description

Weekly Campylobacter case counts across Germany

Usage

```
data("campyl")
```

Format

A data frame with 1248 rows and 413 variables.

`coef.gctsc` *Extract Coefficients from a gctsc Model*

Description

Returns the estimated coefficients from a fitted `gctsc` model object.

Usage

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

Arguments

`object`	An object of class `gctsc`.
`...`	Ignored. Included for S3 method compatibility.

Value

A named numeric vector of model coefficients.

`gctsc` *Fit a Copula-Based Count Time Series Model*

Description

Fits a Gaussian or Student–t copula model to a univariate count time series with flexible discrete marginal distributions and latent ARMA dependence.

Usage

```
gctsc(
  formula = NULL,
  data,
  marginal,
  cormat,
  method = c("TMET", "GHK", "CE", "GHK_mvt"),
  c = 0.5,
  QMC = TRUE,
  pm = 30,
  start = NULL,
  family = "gaussian",
  df = 10,
  options = gctsc.opts()
)
```

Arguments

`formula`	A model formula or a named list of formulas. For non-zero-inflated marginals, this may be a formula such as `y ~ x1 + x2` or `list(mu = y ~ x1 + x2)`. For zero-inflated marginals, this must be a named list with both `mu` and `pi0` components, e.g. `list(mu = y ~ x1, pi0 = ~ z1)`. The `mu` formula must include the response variable.
`data`	A data frame containing the response and all covariates referenced in `formula`. This argument is required.
`marginal`	A marginal model object inheriting class `"marginal.gctsc"`, such as `poisson.marg`, `negbin.marg`, `binom.marg`, `bbinom.marg`, `zip.marg`, `zib.marg`, or `zibb.marg`.
`cormat`	A correlation model object inheriting class `"cormat.gctsc"`, such as `arma.cormat`.
`method`	Character string specifying the likelihood approximation method. One of `"TMET"`, `"GHK"`, `"CE"`, or `"GHK_mvt"`.
`c`	Numeric smoothing constant used by the CE method. Must be a single value between 0 and 1. Ignored by TMET and GHK methods.
`QMC`	Logical; if `TRUE`, quasi-Monte Carlo sampling is used for simulation-based methods.
`pm`	Positive integer specifying the truncated AR order used by TMET when approximating ARMA(p, q) dependence. This is mainly relevant when $q > 0$. Default is `30`.

`start`	Optional numeric vector of starting values. The vector should contain the marginal parameters followed by the dependence parameters. If `NULL`, starting values are constructed from the marginal and correlation objects.
`family`	Copula family. One of `"gaussian"` or `"t"`. Default is `"gaussian"`.
`df`	Degrees of freedom for the Student–t copula. Must be a single finite numeric value greater than 2 when `family = "t"`.
`options`	A list of computational options, usually created by `gctsc.opts`. Important components include: • `M`: positive integer or vector of two positive integers specifying the number of Monte Carlo or quasi-Monte Carlo samples used by simulation-based methods. If two values are supplied, staged optimization is performed, using the first value for an initial fit and the second value for refinement. This option is ignored for `method = "CE"`. • `seed`: optional integer seed used to make the simulated likelihood approximation reproducible; • `opt`: optimization function used for likelihood maximization. Supplying `options$seed` is strongly recommended for TMET and GHK methods because it uses common random numbers across likelihood evaluations, making the objective function more stable and improving numerical Hessian estimation.

Details

Supported marginal distributions include Poisson, negative binomial, binomial, beta-binomial, and their zero-inflated variants. The latent dependence is specified through a correlation model such as `arma.cormat`.

The copula likelihood involves a high-dimensional rectangle probability. This probability is approximated using one of the following methods:

- `"TMET"`: Time Series Minimax Exponential Tilting,
- `"GHK"`: Geweke–Hajivassiliou–Keane simulation,
- `"CE"`: Continuous Extension,
- `"GHK_mvt"`: GHK approximation for the multivariate Student–t rectangle probability. This option is experimental and is mainly intended for comparisons.

The model interface follows the usual R formula convention. An intercept is included by default and can be removed using `-1` or `0 +`. For non-zero-inflated marginals, `formula` may be a standard formula, such as `y ~ x1 + x2`, or a named list `list(mu = y ~ x1 + x2)`. For zero-inflated marginals, `formula` must be a named list with components `mu` and `pi0`, for example `list(mu = y ~ x1 + x2, pi0 = ~ z1)`.

Formula interface. For non-zero-inflated marginals, users may write either `formula = y ~ x1 + x2` or `formula = list(mu = y ~ x1 + x2)`. Internally, both are represented as a list with component `mu`.

For zero-inflated marginals, users must supply both the mean/count component and the zero-inflation component:

```
formula = list(mu  = y ~ x1 + x2,pi0 = ~ z1 + z2)
```

The response variable is taken from `formula$mu`. The `pi0` formula should be one-sided. Intercepts are handled by `model.matrix` following standard R formula rules.

Missing values. Missing values are not handled automatically. Users should remove or impute missing values before calling `gctsc`. This avoids ambiguity in the time series dependence structure.

Dependence. The dependence parameters are encoded through `cormat`. ARMA(0,0) is not supported. For ARMA dependence, admissible starting values should satisfy the usual causality and invertibility conditions.

Seed and numerical stability. Simulation-based likelihood approximations are random unless a seed is supplied. If `options$seed` is provided, the same random stream is used across likelihood evaluations, which can make optimization and standard error estimation more stable. If no seed is supplied, the model can still be fitted, but the approximate likelihood and numerical Hessian may be less stable.

Value

An object of class `"gctsc"` containing, among others:

- `coef`: parameter estimates;
- `maximum`: approximate maximized log-likelihood;
- `se`: standard errors, when available;
- `formula`: normalized model formula list;
- `terms`: model terms for each component;
- `model`: model frames for each component;
- `call`: matched function call.

References

Nguyen, Q. N. and De Oliveira, V. (2026), Approximating Gaussian Copula Models for Count Time Series: Connecting the Distributional Transform and a Continuous Extension, *Journal of Applied Statistics*, **53**: 1–22.

Nguyen, Q. N. and De Oliveira, V. (2026), Likelihood Inference in Gaussian Copula Models for Count Time Series via Minimax Exponential Tilting, *Computational Statistics & Data Analysis*, **218**: 108344.

Nguyen, Q. N. and De Oliveira, V. (2026), Scalable Likelihood Inference for Student–t Copula Count Time Series, *Stats*, **9**: 1–49.

See Also

`gctsc.opts`, `arma.cormat`, `poisson.marg`, `zip.marg`, `zib.marg`, `zibb.marg`

Examples

```
## Example 1: Gaussian copula, Poisson marginal, AR(1)
set.seed(42)
n <- 500
sim_dat <- sim_poisson(mu = 10, tau = 0.3, arma_order = c(1, 0),
                       nsim = n, family = "gaussian")
```

```
dat <- data.frame(y = sim_dat$y)

fit_gauss <- gctsc(
  y ~ 1,
  data = dat,
  marginal = poisson.marg(lambda.lower = 0),
  cormat = arma.cormat(p = 1, q = 0), family = "gaussian",
  method = "CE",
  options = gctsc.opts(M = 1000, seed = 42)
)
summary(fit_gauss)

## Example 2: Student--t copula
sim_dat_t <- sim_poisson(mu = 10, tau = 0.3, arma_order = c(1, 0),
                         nsim = 500, family = "t", df = 10)

dat_t <- data.frame(y = sim_dat_t$y)

fit_t <- gctsc(
  y ~ 1,
  data = dat_t,
  marginal = poisson.marg(lambda.lower = 0),
  cormat = arma.cormat(p = 1, q = 0), family ="t",
  df= 10, method = "CE",
  options = gctsc.opts(M = 1000, seed = 42)
)
summary(fit_t)
```

`gctsc-examples` *Worked Examples Included with the Package*

Description

The **gctsc** package includes additional worked examples in the installed ‘`examples/`’ directory.

Details

Worked examples are installed with the package in the ‘`examples/`’ directory. They are organized into the subfolders ‘`gaussian/`’ and ‘`student_t/`’, corresponding to the two copula families supported by the package.

The top-level example directory can be located with `system.file("examples", package = "gctsc")`.

To list the available Gaussian examples, use

```
dir(file.path(system.file("examples", package = "gctsc"), "gaussian"))
```

To list the available Student-t examples, use

```
dir(file.path(system.file("examples", package = "gctsc"), "student_t"))
```

To inspect the contents of one of the example scripts without running it, use:

```
f <- file.path(system.file("examples", package = "gctsc"),
               "gaussian", "poisson.R")
cat(readLines(f), sep = "\n")
```

Examples

```
exdir <- system.file("examples", package = "gctsc")
dir(exdir)
```

`gctsc.opts` *Set Options for Gaussian and Student t Copula Time Series Model*

Description

Creates a control list for simulation and likelihood approximation in the Gaussian and Student t copula model, including the random seed, Monte Carlo settings, and optimization controls.

Usage

```
gctsc.opts(seed = 1, M = c(100, 1000), ...)
```

Arguments

`seed`	Integer specifying the random seed used for Monte Carlo or quasi-Monte Carlo simulation during likelihood evaluation. Setting a seed is recommended for simulation-based methods because it makes the objective function reproducible across optimization steps.
`M`	Positive integer or vector of two positive integers. Number of Monte Carlo or quasi-Monte Carlo samples used in the likelihood approximation. If a single value is supplied, that value is used throughout the optimization. If a vector of length two is supplied, staged optimization is used: the model is first fitted using the first value of `M`, and the resulting estimates are then used as starting values for a second fit using the second value of `M`. This option is used only for simulation-based methods such as `"GHK"` and `"TMET"` and is ignored for `method = "CE"`.
`...`	Additional control arguments passed to `optim`.

Value

A list with components:

`seed`	Integer. The random seed used.
`M`	Positive integer or vector of two positive integers specifying the Monte Carlo or quasi-Monte Carlo sample sizes.
`opt`	A function used internally by `gctsc()` to optimize the approximate log-likelihood.

`KCWC` *Daily aggregated weather measurements for KCWC station*

Description

Daily aggregated weather measurements for KCWC station

Usage

```
data("KCWC")
```

Format

A data frame with 2665 rows and 4 variables.

`marginal.gctsc` *Marginal Model Constructors for* **gctsc**

Description

These functions construct marginal model objects for use with `gctsc`. Each constructor returns an object of class `"marginal.gctsc"` containing the information needed to initialize marginal parameters and compute the latent truncation bounds used in the copula likelihood.

The following marginal families are currently supported:

- Poisson: `poisson.marg()`
- Negative Binomial: `negbin.marg()`
- Binomial: `binom.marg()`
- Beta–Binomial: `bbinom.marg()`
- Zero-Inflated Poisson: `zip.marg()`
- Zero-Inflated Binomial: `zib.marg()`
- Zero-Inflated Beta–Binomial: `zibb.marg()`

Supported link functions depend on the marginal family:

- `poisson.marg()`, `zip.marg()`, and `negbin.marg()` support `"identity"` and `"log"`.
- `binom.marg()`, `bbinom.marg()`, `zib.marg()`, and `zibb.marg()` currently support `"logit"` only.

Usage

```
poisson.marg(link = "log", lambda.lower = NULL, lambda.upper = NULL)

binom.marg(link = "logit", size = NULL, lambda.lower = NULL, lambda.upper = NULL)

zib.marg(link = "logit", size = NULL, lambda.lower = NULL, lambda.upper = NULL)

negbin.marg(link = "log", lambda.lower = NULL, lambda.upper = NULL)

zip.marg(link = "log", lambda.lower = NULL, lambda.upper = NULL)

bbinom.marg(link = "logit", size, lambda.lower = NULL, lambda.upper = NULL)

zibb.marg(link = "logit", size,  lambda.lower = NULL, lambda.upper = NULL)
```

Arguments

`link`	Link function used for the main marginal component. Supported links depend on the marginal family; see Description.
`lambda.lower`	Optional lower bounds on the marginal parameters.
`lambda.upper`	Optional upper bounds on the marginal parameters.
`size`	Number of trials for Binomial, Zero-Inflated Binomial, Beta–Binomial, and Zero-Inflated Beta–Binomial marginals. For these marginals, `size` is currently assumed to be a single fixed positive integer. For Beta–Binomial and Zero-Inflated Beta–Binomial marginals, `size` should be greater than 1.

Details

Marginal Models for Copula Count Time Series

The marginal constructors are designed to be supplied to `gctsc`, rather than called during likelihood evaluation by the user. Each returned marginal object contains internal functions for:

- computing starting values for the marginal parameters;
- determining the number of marginal parameters;
- converting observed counts into lower and upper latent truncation bounds for the copula likelihood.

Internally, the design input `x` is represented as a named list of design matrices. For non-zero-inflated marginals, `x` contains `x$mu`. For zero-inflated marginals, `x` contains both `x$mu` and `x$pi0`. These matrices are constructed automatically by `gctsc` from the supplied formula and data.

For zero-inflated marginals, the `mu` component controls the main count distribution, while the `pi0` component controls the structural zero probability through a logit link.

The optional bounds `lambda.lower` and `lambda.upper` are attached to the starting values and used during numerical optimization.

Value

A marginal model object of class `"marginal.gctsc"`.

See Also

`gctsc`, `arma.cormat`

Examples

```
poisson.marg(link = "log")
negbin.marg(link = "log")
binom.marg(link = "logit", size = 10)
bbinom.marg(link = "logit", size = 24)
zip.marg(link = "log")
zib.marg(link = "logit", size = 10)
zibb.marg(link = "logit", size = 24)
```

`plot.gctsc` *Diagnostic Plots for Fitted Copula Count Time Series Models*

Description

Produces diagnostic plots for a fitted Gaussian or Student–t copula count time series model of class `"gctsc"`.

The diagnostics are based on randomized quantile residuals and probability integral transform (PIT) values.

Usage

```
## S3 method for class 'gctsc'
plot(
  x,
  caption = rep("", 5),
  main = rep("", 5),
  level = 0.95,
  col.lines = "gray",
  ...
)
```

Arguments

`x`	A fitted model object of class `"gctsc"`.
`caption`	Optional character vector of length 5 providing captions for the plots.
`main`	Optional main titles for the plots.
`level`	Confidence level for the Q–Q envelope (default 0.95).
`col.lines`	Color used for reference lines.
`...`	Additional graphical arguments passed to plotting functions.

Details

The following diagnostic plots are produced:

1. Time series of randomized quantile residuals.
2. Q–Q plot against the reference distribution.
3. Histogram of PIT values.
4. Autocorrelation function (ACF) of residuals.
5. Partial autocorrelation function (PACF) of residuals.

For Gaussian copulas, residuals are compared against the standard normal distribution. For Student–t copulas, residuals are compared against a Student–t distribution with degrees of freedom obtained from fitted model.

Value

Invisibly returns `NULL`.

Invisibly returns `NULL`. The function is called for its side effect of producing diagnostic plots.

See Also

`residuals.gctsc`

`residuals.gctsc` for computing the residuals used in the plots.

Examples

```
# Simulate data from a Poisson AR(1) model
set.seed(123)
n <- 2000
mu <- 5
phi <- 0.5
arma_order <- c(1, 0)
y <- sim_poisson(mu = mu, tau = phi, arma_order = arma_order, nsim = n)$y

# Fit the model using the CE method
fit <- gctsc(y~1, data = data.frame(y),
  marginal = poisson.marg(link = "identity", lambda.lower = 0),
  cormat = arma.cormat(p = 1, q = 0), family ="gaussian",
  method = "CE",
  options = gctsc.opts(seed = 1, M = 1000),
  c = 0.5
)

# Produce diagnostic plots
par(mfrow = c(2, 3))
plot(fit)
```

`pmvn` *Approximate Gaussian Copula Log-Likelihood*

Description

Computes an approximate log-likelihood for a Gaussian copula count time series model from latent lower and upper truncation bounds. The approximation method is selected through the argument `method`.

Usage

```
pmvn(
  lower,
  upper,
  tau,
  od,
  method = c("CE", "GHK", "TMET"),
  c = 0.5,
  pm = 30,
  M = 1000,
  QMC = TRUE,
  ret_llk = TRUE
)
```

Arguments

`lower`	Numeric vector of length `n` giving the lower truncation bounds of the latent variables.
`upper`	Numeric vector of length `n` giving the upper truncation bounds of the latent variables.
`tau`	Numeric vector of ARMA dependence parameters ordered as `c(phi_1, ..., phi_p, theta_1, ..., theta_q)`.
`od`	Integer vector `c(p, q)` specifying the AR and MA orders of the latent ARMA process.
`method`	Character string specifying the likelihood approximation method. Must be one of `"CE"`, `"GHK"`, or `"TMET"`.
`c`	Smoothing parameter for the CE approximation. Used only when `method = "CE"`. Default is `0.5`.
`pm`	Integer specifying the number of past lags used to approximate an ARMA(p, q) process by a finite-order AR representation. Used only when `method = "TMET"`.
`M`	Positive integer specifying the number of Monte Carlo or quasi-Monte Carlo samples. Used by the simulation-based methods `"GHK"` and `"TMET"`.
`QMC`	Logical; if `TRUE` (default), quasi-Monte Carlo integration is used when applicable. Otherwise, standard Monte Carlo sampling is used.
`ret_llk`	Logical; if `TRUE` (default), returns the approximate log-likelihood. If `FALSE`, method-specific internal quantities are returned for diagnostic or research use.

Details

The package currently supports three likelihood approximations: continuous extension (CE), Geweke–Hajivassiliou–Keane simulation (GHK), and Time Series Minimax Exponential Tilting (TMET).

The function `pmvn()` provides a unified interface for Gaussian copula likelihood approximation. The argument `method` selects among:

- `"CE"`: continuous extension approximation,
- `"GHK"`: sequential importance sampling via the GHK simulator,
- `"TMET"`: minimax exponential tilting approximation.

The arguments `c`, `pm`, `M`, and `QMC` are used only by the methods to which they apply.

Value

A numeric scalar giving the approximate log-likelihood. If `ret_llk = FALSE`, method-specific diagnostic output is returned.

References

Nguyen, Q. N. and De Oliveira, V. (2026). Approximating Gaussian Copula Models for Count Time Series: Connecting the Distributional Transform and a Continuous Extension, *Journal of Applied Statistics*, **53**, 1–22.

Nguyen, Q. N., and De Oliveira, V. (2026), Likelihood Inference in Gaussian Copula Models for Count Time Series via Minimax Exponential Tilting, *Computational Statistics and Data Analysis*, **218**: 108344.

See Also

`pmvt`, `sim_poisson`, `poisson.marg`

Examples

```
mu <- 10
tau <- 0.2
arma_order <- c(1, 0)

sim_data <- sim_poisson(mu = mu, tau = tau, arma_order = arma_order,
                        nsim = 500, family = "gaussian", seed = 1)
y <- sim_data$y

lower <- qnorm(ppois(y - 1, lambda = mu))
upper <- qnorm(ppois(y, lambda = mu))

## Continuous extension
pmvn(lower, upper, tau = tau, od = arma_order, method = "CE", c = 0.5)

## GHK approximation
pmvn(lower, upper, tau = tau, od = arma_order, method = "GHK", M = 1000)

## TMET approximation
```

```
pmvn(lower, upper, tau = tau, od = arma_order, method = "TMET",
     pm = 30, M = 1000)
```

`pmvt` *Approximate Student–t Copula Log-Likelihood*

Description

Computes an approximate log-likelihood for a Student–t copula count time series model from latent lower and upper truncation bounds. The approximation method is selected through the argument `method`.

Usage

```
pmvt(
  lower,
  upper,
  tau,
  od,
  method = c("CE", "GHK", "TMET"),
  c = 0.5,
  pm = 30,
  M = 1000,
  QMC = TRUE,
  ret_llk = TRUE,
  df
)
```

Arguments

`lower`	Numeric vector of length `n` giving the lower truncation bounds of the latent variables.
`upper`	Numeric vector of length `n` giving the upper truncation bounds of the latent variables.
`tau`	Numeric vector of ARMA dependence parameters ordered as `c(phi_1, ..., phi_p, theta_1, ..., theta_q)`.
`od`	Integer vector `c(p, q)` specifying the AR and MA orders of the latent ARMA process.
`method`	Character string specifying the likelihood approximation method. Must be one of `"CE"`, `"GHK"`, or `"TMET"`.
`c`	Smoothing parameter for the CE approximation. Used only when `method = "CE"`. Default is `0.5`.
`pm`	Integer specifying the number of past lags used to approximate an ARMA(p, q) process by a finite-order AR representation. Used only when `method = "TMET"`.

`M`	Positive integer specifying the number of Monte Carlo or quasi-Monte Carlo samples. Used by the simulation-based methods `"GHK"` and `"TMET"`.
`QMC`	Logical; if `TRUE` (default), quasi-Monte Carlo integration is used when applicable. Otherwise, standard Monte Carlo sampling is used.
`ret_llk`	Logical; if `TRUE` (default), returns the approximate log-likelihood. If `FALSE`, method-specific internal quantities are returned for diagnostic or research use.
`df`	Degrees of freedom of the Student–*t* copula. Must be greater than 2.

Details

The package currently supports three likelihood approximations: continuous extension (CE), Geweke–Hajivassiliou–Keane simulation (GHK), and Time Series Minimax Exponential Tilting (TMET).

The function `pmvt()` provides a unified interface for Student–*t* copula likelihood approximation. The argument `method` selects among:

- `"CE"`: continuous extension approximation,
- `"GHK"`: sequential importance sampling via the GHK simulator,
- `"TMET"`: minimax exponential tilting approximation.

The arguments `c`, `pm`, `M`, and `QMC` are used only by the methods to which they apply.

Value

A numeric scalar giving the approximate log-likelihood. If `ret_llk = FALSE`, method-specific diagnostic output is returned.

References

Nguyen, Q. N. and De Oliveira, V. (2026). Scalable Likelihood Inference for Student–t Copula Count Time Series, *Stats*, **9**: 1–49.

See Also

`pmvn`, `gctsc`, `sim_poisson`

Examples

```
mu <- 10
tau <- 0.2
arma_order <- c(1, 0)
df <- 8

sim_data <- sim_poisson(mu = mu, tau = tau, arma_order = arma_order,
                        nsim = 200, family = "t", df = df, seed = 1)
y <- sim_data$y

lower <- qt(ppois(y - 1, lambda = mu), df = df)
upper <- qt(ppois(y, lambda = mu), df = df)
```

```
## Continuous extension
pmvt(lower, upper, tau = tau, od = arma_order, method = "CE", c = 0.5, df = df)

## GHK approximation
pmvt(lower, upper, tau = tau, od = arma_order, method = "GHK", M = 200, df = df)

## TMET approximation
pmvt(lower, upper, tau = tau, od = arma_order, method = "TMET", pm = 30, M = 200, df = df)
```

`predict.gctsc` *One-Step-Ahead Predictive Distribution for Copula Count Time Series Models*

Description

Computes the one-step-ahead predictive distribution for a fitted Gaussian or Student–t copula count time series model.

The predictive probability mass function is evaluated over the grid `0:y_max`. If `y_max` is not supplied, it is chosen automatically from the fitted response values as `ceiling(max(y) + k * sd(y))`. Summary statistics of the predictive distribution are returned. If the observed response is included in `newdata`, the Continuous Ranked Probability Score (CRPS) and Logarithmic Score (LOGS) are also computed.

Usage

```
## S3 method for class 'gctsc'
predict(object, newdata = NULL, y_max = NULL, k = 3, ...)
```

Arguments

`object`	A fitted model object of class `"gctsc"`, as returned by `gctsc`.
`newdata`	Optional one-row `data.frame` containing the covariate values at the prediction time point. The variables in `newdata` should match the variables used in the fitted model formula. If the fitted model is intercept-only, `newdata` may be omitted. If the observed response at the prediction time is available, it may also be included in `newdata` using the same response variable name as in the fitted model. In that case, CRPS and LOGS are computed and returned.
`y_max`	Optional nonnegative integer specifying the largest count value included in the predictive grid `0:y_max`. If `NULL`, `y_max` is selected automatically using `ceiling(max(y) + k * sd(y))`, where `y` is the fitted response series.
`k`	Nonnegative numeric multiplier used to choose `y_max` when `y_max = NULL`. The default is `k = 3`.
`...`	Ignored. Included for S3 method compatibility.

Details

The function constructs prediction design matrices from the stored model terms using `model.matrix`. Thus the same formula convention used in model fitting is used for prediction, including automatic intercept handling and factor-variable expansion.

For zero-inflated marginals, `newdata` is used to construct both the mean-component design matrix and the zero-inflation design matrix. The column names of the new design matrices must match those from the fitted model.

The predictive distribution is computed using the copula family and approximation method stored in the fitted object. For Gaussian copulas, multivariate normal rectangle probabilities are used. For Student–t copulas, multivariate t rectangle probabilities are used.

If the observed response is included in `newdata`, it must be a single nonnegative integer count and should not exceed `y_max`.

Value

A list containing:

- `mean`: Predictive mean.
- `median`: Predictive median.
- `mode`: Predictive mode.
- `variance`: Predictive variance.
- `p_y`: Predictive probability mass function evaluated on `y_grid`.
- `y_grid`: Count grid `0:y_max` over which the predictive distribution is evaluated.
- `lower`, `upper`: Lower and upper endpoints of the 95\ predictive interval.
- `CRPS`: Continuous Ranked Probability Score, returned only if the observed response is supplied in `newdata`.
- `LOGS`: Logarithmic Score, returned only if the observed response is supplied in `newdata`.
- `y_true`: Observed response value, returned only if supplied in `newdata`.

References

Nguyen, Q. N. and De Oliveira, V. (2026), Likelihood Inference in Gaussian Copula Models for Count Time Series via Minimax Exponential Tilting, *Computational Statistics & Data Analysis*, **218**: 108344.

Nguyen, Q. N. and De Oliveira, V. (2026), Scalable Likelihood Inference for Student–*t* Copula Count Time Series, *Stats*, **9**: 1–49.

See Also

`gctsc`, `arma.cormat`, `gctsc.opts`

Examples

```
# Simulate Poisson AR(1) data
set.seed(1)
y_sim <- sim_poisson(
  mu = 10,
  tau = 0.2,
  arma_order = c(1, 0),
  nsim = 200,
  family = "gaussian"
)$y

dat <- data.frame(y = y_sim)

# Fit Gaussian copula model
fit <- gctsc(
  formula = y ~ 1,
  data = dat,
  marginal = poisson.marg(link = "log"),
  cormat = arma.cormat(p = 1, q = 0),
  method = "GHK",
  family = "gaussian",
  options = gctsc.opts(M = 1000, seed = 42)
)

# One-step-ahead prediction for an intercept-only model
pred <- predict(fit, y_max = 30)

# If the future observed value is available, include it in newdata
pred_score <- predict(fit, newdata = data.frame(y = 8), y_max = 30)
```

`print.gctsc` *Print a gctsc Model Object*

Description

Displays the call, estimation method, parameter estimates, and likelihood information.

Usage

```
## S3 method for class 'gctsc'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`x`	An object of class `gctsc`.
`digits`	Number of significant digits to display.
`...`	Ignored. Included for S3 method compatibility.

`print.summary.gctsc` *Print Summary of a gctsc Model*

Description

Displays summary statistics and model fit information for a fitted `gctsc` model.

Usage

```
## S3 method for class 'summary.gctsc'
print(x, digits = 4, ...)
```

Arguments

`x`	An object of class `summary.gctsc`.
`digits`	Number of significant digits to display.
`...`	Ignored. Included for S3 method compatibility.

`residuals.gctsc` *Randomized Quantile Residuals for Copula Count Time Series Models*

Description

Computes randomized quantile residuals for a fitted Gaussian or Student–t copula count time series model.

For discrete responses, residuals are constructed using the randomized probability integral transform (PIT) of Dunn and Smyth (1996). The conditional probabilities required for the PIT are approximated according to the fitted copula family and likelihood method. For Gaussian copula models fitted by TMET or GHK, the same simulation-based method is used in the residual computation. For Gaussian copula models fitted by CE, the required conditional probabilities are approximated by GHK. For Student–t copula models, the required conditional probabilities are approximated by GHK.

Usage

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

Arguments

`object`	A fitted model object of class `"gctsc"`, as returned by `gctsc`.
`...`	Ignored. Included for S3 method compatibility.

Details

For observation y_t, let $F_t(y_t^-|y)$ and $F_t(y_t|y)$ denote the conditional CDF evaluated at $y_t - 1$ and y_t given past observations, respectively. The PIT value is computed as

$$e_t = F_t(y_t^-|y) + u_t\{F_t(y_t|y) - F_t(y_t^-|y)\},$$

where $u_t \sim \mathrm{Uniform}(0, 1)$.

For Gaussian copulas, residuals are obtained as $r_t = \Phi^{-1}(e_t)$.

For Student–t copulas with degrees of freedom `df`, the residuals are defined as $r_t = t_\nu^{-1}(e_t)$, where t_ν^{-1} denotes the quantile function of the Student–t distribution.

Value

A list of class `"gctsc.residuals"` containing:

- `residuals`: Numeric vector of randomized quantile residuals.
- `pit`: Numeric vector of probability integral transform values.

References

Nguyen, Q. N. and De Oliveira, V. (2026), Approximating Gaussian Copula Models for Count Time Series: Connecting the Distributional Transform and a Continuous Extension, *Journal of Applied Statistics*, **53**: 1–22.

Nguyen, Q. N. and De Oliveira, V. (2026), Likelihood Inference in Gaussian Copula Models for Count Time Series via Minimax Exponential Tilting, *Computational Statistics & Data Analysis*, **218**: 108344.

Nguyen, Q. N. and De Oliveira, V. (2026), Scalable Likelihood Inference for Student–t Copula Count Time Series, *Stats*, **9**: 1–49.

See Also

`gctsc`, `sim_gctsc`, `pmvn`, `pmvt`, `predict.gctsc`

Examples

```
# Simulate Poisson AR(1) data under a Gaussian copula
set.seed(1)
y <- sim_poisson(mu = 5, tau = 0.7,
                 arma_order = c(1, 0),
                 nsim = 500,
                 family = "gaussian")$y

fit <- gctsc(
  y ~ 1,
  data = data.frame(y),
  marginal = poisson.marg(),
  cormat = arma.cormat(1, 0),
  family = "gaussian",
  method = "CE",
```

```
  options = gctsc.opts(seed = 1, M = 1000)
)

res <- residuals(fit)
hist(res$residuals, main = "Randomized Quantile Residuals")
hist(res$pit, main = "PIT Histogram")
```

`rota` *Weekly Rotavirus case counts across Germany*

Description

Weekly Rotavirus case counts across Germany

Usage

```
data("rota")
```

Format

A data frame with 1248 rows and 413 variables.

`sim_gctsc` *Simulate from Gaussian and t Copula Time Series Models*

Description

These functions simulate time series data from Gaussian and t copula models with various discrete marginals and an ARMA dependence structure.

Usage

```
sim_poisson(
  mu,
  tau,
  arma_order,
  nsim,
  family = c("gaussian", "t"),
  df = NULL,
  seed = NULL
)

sim_negbin(
  mu,
  dispersion,
  tau,
```

```
  arma_order,
  nsim = 100,
  family = c("gaussian", "t"),
  df = NULL,
  seed = NULL
)

sim_zip(
  mu,
  pi0,
  tau,
  arma_order,
  nsim = 100,
  family = c("gaussian", "t"),
  df = NULL,
  seed = NULL
)

sim_binom(
  prob,
  size,
  tau,
  arma_order,
  nsim = 100,
  family = c("gaussian", "t"),
  df = NULL,
  seed = NULL
)

sim_bbinom(
  prob,
  rho,
  size,
  tau,
  arma_order,
  nsim = 100,
  family = c("gaussian", "t"),
  df = NULL,
  seed = NULL
)

sim_zib(
  prob,
  pi0,
  size,
  tau,
  arma_order,
  nsim = 100,
```

```
  family = c("gaussian", "t"),
  df = NULL,
  seed = NULL
)

sim_zibb(
  prob,
  rho,
  pi0,
  size,
  tau,
  arma_order,
  nsim = 100,
  family = c("gaussian", "t"),
  df = NULL,
  seed = NULL
)
```

Arguments

`mu`	Mean parameter(s) for Poisson-, ZIP-, and negative binomial-type marginals. Must satisfy $\mu > 0$. May be specified as a scalar or as a numeric vector of length `nsim` to allow time-varying means.
`tau`	Numeric vector of ARMA dependence coefficients, ordered as `c(phi_1, ..., phi_p, theta_1, ..., theta_q)`, where ϕ_i are autoregressive (AR) coefficients and θ_j are moving-average (MA) coefficients. The model `ARMA(0, 0)` is not supported.
`arma_order`	Integer vector `c(p, q)` specifying the AR and MA orders.
`nsim`	Positive integer giving the number of time points to simulate.
`family`	Character string specifying the copula family: `"gaussian"` or `"t"`.
`df`	Degrees of freedom for the t copula. Must be a single numeric value greater than 2. Required only when `family = "t"`.
`seed`	Optional integer used to set the random seed.
`dispersion`	Overdispersion parameter for negative binomial marginals. Must satisfy $\kappa > 0$, where $\mathrm{Var}(Y) = \mu + \kappa\mu^2$. May be a scalar or a numeric vector of length `nsim`.
`pi0`	Zero-inflation probability for ZIP, ZIB, and ZIBB marginals. Must satisfy $0 \leq \pi_0 < 1$. May be a scalar or a numeric vector of length `nsim`.
`prob`	Success probability parameter(s) for binomial-type marginals. Must satisfy $0 < p < 1$. May be a scalar or a numeric vector of length `nsim`.
`size`	Number of trials for binomial-type marginals; a positive integer scalar.
`rho`	Intra-class correlation parameter for beta-binomial and ZIBB marginals. Must satisfy $0 < \rho < 1$, where $\mathrm{Var}(Y) = np(1-p)\{1 + (n-1)\rho\}$ and n is the number of trials. May be a scalar or a numeric vector of length `nsim`.

Details

Marginal types:

- Poisson: Counts with mean μ.
- Negative binomial (NB): Overdispersed counts with mean μ and dispersion parameter κ.
- Binomial: Number of successes in n trials with success probability p.
- Beta—binomial (BB): Binomial with success probability p following a beta distribution, allowing intra-cluster correlation ρ.
- Zero–inflated Poisson (ZIP): Poisson with extra probability π_0 of an excess zero.
- Zero–inflated binomial (ZIB): Binomial with extra probability π_0 of an excess zero.
- Zero–inflated beta–binomial (ZIBB): Beta–binomial with extra probability π_0 of an excess zero.

Parameterization notes:

- Negative binomial uses `dispersion` (κ) to model overdispersion: larger `dispersion` increases variance for a fixed mean.
- Beta–binomial and ZIBB use `rho` as the overdispersion parameter: ρ is the intra-class correlation, with $\rho \to 0$ giving the binomial model.
- Zero–inflated marginals include a separate `pi0` parameter that controls the extra probability mass at zero.

Worked examples. Additional worked examples, including Gaussian and Student–t copula models with zero-inflated marginals, are provided in the installed example scripts; see `gctsc-examples`.

Value

A list with components:

- `y`: Simulated time series data.
- `z`: Latent Gaussian process values.
- `marginal`: Marginal distribution name.
- `parameters`: List of parameters used.
- `cormat`: ARMA structure.

See Also

`gctsc`, `sim_gctsc`, `marginal.gctsc`, `pmvn`, `pmvt`, `predict.gctsc`

Examples

```
# Poisson example
sim_poisson(mu = 10, tau = c(0.2, 0.2),
  arma_order = c(1, 1), nsim = 100,
  family = "gaussian", seed = 42)

# Negative Binomial example
```

```
sim_negbin(mu = 10, dispersion = 2, tau = c(0.5, 0.5),
  arma_order = c(1, 1),family = "gaussian",
  nsim = 100, seed =1)

# Zero Inflated Beta-Binomial example with seasonal covariates
n <- 100
xi <- numeric(n)
zeta <- rnorm(n)
for (j in 3:n) {
  xi[j] <- 0.6 * xi[j - 1] - 0.4 * xi[j - 2] + zeta[j]
}
prob <- plogis(0.2 + 0.3 * sin(2 * pi * (1:n) / 12) +
             0.5 * cos(2 * pi * (1:n) / 12) + 0.3 * xi)
sim_zibb(prob, rho = 1/6, pi0 = 0.2, size = 24, tau = 0.5,
 arma_order = c(1, 0),family = "t", df = 10, nsim = 100)
```

`summary.gctsc` *Summarize a gctsc Model Fit*

Description

Computes standard errors, z-values, and p-values for the estimated parameters in a fitted `gctsc` object.

Usage

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

Arguments

`object`	An object of class `gctsc`.
`...`	Ignored. Included for S3 method compatibility.

Value

A list of class `summary.gctsc` containing model summary statistics.

Index