

Package ‘gctsc’

May 8, 2026

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

Version 0.2.3

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

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 2026-03-20 06:10:31 UTC

Contents

arma.cormat	2
campyl	3
coef.gctsc	4
gctsc	4
gctsc.opts	7
KCWC	8
marginal.gctsc	8
plot.gctsc	10
pmvn_ce	11
pmvn_ghk	13
pmvn_tmet	15
predict.gctsc	18
print.gctsc	19
print.summary.gctsc	20
residuals.gctsc	20
rota	22
sim_gctsc	22
summary.gctsc	26
Index	27

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	<i>Extract Coefficients from a gctsc Model</i>
------------	--

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	<i>Fit a Copula-Based Count Time Series Model</i>
-------	---

Description

Fits a Gaussian or Student-t copula model to univariate count time series with discrete marginals (Poisson, negative binomial, binomial/beta-binomial, and zero-inflated variants) and latent dependence specified via ARMA correlation structures.

Usage

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

Arguments

formula	A formula (e.g., $y \sim x_1 + x_2$) or, for zero-inflated marginals, a named list <code>list(mu = ..., pi0 = ...)</code> .
data	A data frame containing the response and covariates referenced in the formula(s).
marginal	A marginal model object such as <code>poisson.marg</code> , <code>negbin.marg</code> , <code>zib.marg</code> , or <code>zibb.marg</code> ; must inherit class <code>"marginal.gctsc"</code> .
cormat	A correlation structure such as <code>arma.cormat</code> ; must inherit class <code>"cormat.gctsc"</code> .
method	One of <code>"TMET"</code> , <code>"GHK"</code> , or <code>"CE"</code> .
c	Smoothing constant used by CE only (ignored otherwise). Default is 0.5.
QMC	Logical; if TRUE, quasi-Monte Carlo integration is used for simulation-based methods.
pm	Integer specifying the truncated AR order used when approximating $ARMA(p, q)$ by an AR representation (TMET only; relevant when $q > 0$). Default is 30
start	Optional numeric vector of starting values (marginal parameters followed by dependence parameters). If NULL, sensible starting values and bounds are constructed from marginal and cormat.
family	Copula family. One of <code>"gaussian"</code> or <code>"t"</code> .
df	Degrees of freedom for the Student-t copula. Must be greater than 2. Required when family = <code>"t"</code> . Ignored for the Gaussian copula.
options	Optional list of tuning and optimization controls. If NULL, defaults from <code>gctsc.opts()</code> are used (e.g., $M = 1000$, randomized seed). Any supplied fields override the defaults.

Details

The high-dimensional rectangle probability defining the copula likelihood is approximated using one of:

- **TMET** (Time Series Minimax Exponential Tilting),
- **GHK** (Geweke-Hajivassiliou-Keane simulation), or
- **CE** (Continuous Extension).

The interface mirrors `glm()`. Zero-inflated marginals accept a named list of formulas, e.g., `list(mu = $y \sim x$, pi0 = $\sim z$)`. Non-zero-inflated marginals accept a single formula or `list(mu = ...)`.

Formulas. For zero-inflated marginals, if neither `mu` nor `pi0` is supplied, both default to intercept-only models ($\mu \sim 1$, $\pi_0 \sim 1$). If `mu` is supplied but `pi0` is missing, $\pi_0 \sim 1$ is used.

Dependence. The ARMA parameters are encoded in `cormat`. Models must satisfy stationarity and invertibility conditions. $ARMA(0,0)$ is not supported.

Method-specific notes. CE ignores QMC and `options$M`. GHK and TMET require `options$M` to be a positive integer. TMET additionally uses `pm` when $q > 0$.

Value

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

- coef: parameter estimates,
- maximum: approximate log-likelihood at the optimum,
- se: standard errors when available,
- terms, model, call: model metadata.

References

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

Nguyen, Q. N., & De Oliveira, V. (2026). Scalable Likelihood Inference for Student- t Copula Count Time Series. Manuscript in preparation.

Nguyen, Q. N., & De Oliveira, V. (2026). Approximating Gaussian copula models for count time series: Connecting the distributional transform and a continuous extension. *Journal of Applied Statistics*.

See Also

[arma.cormat](#), [poisson.marg](#), [zib.marg](#), [zibb.marg](#), [gctsc.opts](#)

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.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 and Monte Carlo settings.

Usage

```
gctsc.opts(seed = NULL, M = 1000, ...)
```

Arguments

seed	recommended to provide a seed. Integer specifying the random seed used for Monte Carlo simulation during likelihood evaluation. Fixing the seed ensures reproducible optimization results and stable maximum likelihood estimates.
M	Integer. Number of Monte Carlo samples used in the likelihood approximation (default: 1000).
...	Ignored. Included for S3 method compatibility.

Value

A list with components:

seed	Integer. The random seed used.
M	Integer. Number of Monte Carlo samples.
opt	A function used internally by gctsc() to perform optimization of 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 Models for Copula Time Series

Description

The following marginal models are currently available:

poisson.marg(link = "log") Poisson distribution.

negbin.marg(link = "log") Negative binomial distribution.

binom.marg(link = "logit", size) Binomial distribution with fixed number of trials.

bbinom.marg(link = "logit", size) Beta-binomial with overdispersion.

zip.marg(link = "log") Zero-inflated Poisson model.

zib.marg(link = "logit", size) Zero-inflated binomial.

zibb.marg(link = "logit", size) Zero-inflated beta-binomial with separate covariates for zero inflation.

Usage

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

```
zip.marg(link = "identity", 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	The link function for the mean (e.g., "log", "logit", "identity").
lambda.lower	Optional lower bounds on parameters.
lambda.upper	Optional upper bounds on parameters.
size	Number of trials (for binomial-type models).

Details

These functions define the marginal distributions used in copula-based count time series models.

Each marginal constructor returns an object of class "marginal.gctsc" which defines:

- start: a function to compute starting values.
- npar: number of parameters.
- bounds: truncation bounds on the latent Gaussian.

These marginals are designed to work with `gctsc()` and its related methods.

Value

A list of class "marginal.gctsc" representing the marginal model.

References

- Cribari-Neto, F. and Zeileis, A. (2010). Beta regression in R. *Journal of Statistical Software*, **34**(2): 1–24.
- Ferrari, S.L.P. and Cribari-Neto, F. (2004). Beta regression for modeling rates and proportions. *Journal of Applied Statistics*, **31**(7): 799–815.
- Masarotto, G. and Varin, C. (2012). Gaussian copula marginal regression. *Electronic Journal of Statistics*, **6**: 1517–1549.

See Also

[gctsc](#), [predict.gctsc](#), [arma.cormat](#)

Examples

```
poisson.marg(link = "identity")
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.

This function is called for its side effects and returns `invisible()`.

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,
  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_ce

Approximate Log-Likelihood via Continuous Extension (CE)

Description

Computes the approximate log-likelihood for a count time series model based on a Gaussian or Student-t copula using the Continuous Extension (CE) method.

Usage

```
pmvn_ce(lower, upper, tau, od, c = 0.5, ret_llk = TRUE)
```

```
pmvt_ce(lower, upper, tau, od, c = 0.5, ret_llk = TRUE, df)
```

Arguments

lower	Numeric vector of length n giving the lower bounds of the transformed latent variables.
upper	Numeric vector of length n giving the upper bounds of the transformed latent variables.
tau	Numeric vector of ARMA dependence parameters ordered as $c(\text{phi}_1, \dots, \text{phi}_p, \text{theta}_1, \dots, \text{theta}_q)$.
od	Integer vector $c(p, q)$ specifying AR and MA orders.
c	Smoothing bandwidth parameter in $(0, 1)$. Default is 0.5 .
ret_llk	Logical; if TRUE (default), returns the approximate log-likelihood.
df	Degrees of freedom for the t copula. Required only for <code>pmvt_ce()</code> .

Details

The CE method replaces the discrete probability mass at each observation with a smooth approximation controlled by a bandwidth parameter c . This yields a tractable approximation to the multivariate rectangle probability defining the likelihood.

Two copula families are supported:

- **Gaussian copula** via `pmvn_ce()`
- **t copula** via `pmvt_ce()`

In both cases, the latent dependence structure is specified through an $\text{ARMA}(p, q)$ model.

The CE approximation applies to discrete marginal distributions once the corresponding latent lower and upper bounds are computed. The Gaussian copula version uses the standard normal cdf, while the t copula version uses the Student t cdf with degrees of freedom df .

Value

A numeric value giving the approximate log-likelihood.

References

Nguyen, Q. N., & De Oliveira, V. (2026). Approximating Gaussian copula models for count time series: Connecting the distributional transform and a continuous extension. *Journal of Applied Statistics*.

See Also

[pmvn_ce](#), [pmvt_ce](#)

Examples

```

## Gaussian copula example
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
a <- qnorm(ppois(y - 1, lambda = mu))
b <- qnorm(ppois(y, lambda = mu))

llk_gauss <- pmvn_ce(lower = a, upper = b,
                    tau = tau, od = arma_order, c = 0.5)

## t copula example
df <- 8

sim_data_t <- sim_poisson(mu = mu, tau = tau, arma_order = arma_order,
                         nsim = 500, family = "t", df = df, seed = 1)

y_t <- sim_data_t$y
a_t <- qt(ppois(y_t - 1, lambda = mu), df = df)
b_t <- qt(ppois(y_t, lambda = mu), df = df)

llk_t <- pmvt_ce(lower = a_t, upper = b_t, tau = tau, od = arma_order,
                 c = 0.5, df = df)

```

pmvn_ghk

Approximate Log-Likelihood via GHK Simulation

Description

Computes the approximate log-likelihood for a count time series model based on a Gaussian or Student-t copula using the Geweke-Hajivassiliou-Keane (GHK) simulator.

Usage

```

pmvn_ghk(lower, upper, tau, od, M = 1000, QMC = TRUE, ret_llk = TRUE)

pmvt_ghk(
  lower,
  upper,
  tau,
  od,
  M = 1000,
  QMC = TRUE,

```

```

ret_llk = TRUE,
df,
engine = c("mvnm", "mvt")
)

```

Arguments

lower	Numeric vector of length n giving the lower bounds of the transformed latent variables.
upper	Numeric vector of length n giving the upper bounds of the transformed latent variables.
tau	Numeric vector of ARMA dependence parameters ordered as $c(\phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q)$, where ϕ_i are autoregressive (AR) coefficients and θ_j are moving-average (MA) coefficients.
od	Integer vector $c(p, q)$ specifying the AR and MA orders.
M	Positive integer specifying the number of Monte Carlo or quasi-Monte Carlo samples used in the simulation.
QMC	Logical; if TRUE (default), quasi-Monte Carlo integration is used. Otherwise, standard Monte Carlo sampling is applied.
ret_llk	Logical; if TRUE (default), returns the approximate log-likelihood. If FALSE, internal diagnostic quantities from the GHK simulator are returned. This option is primarily intended for internal use and methodological research.
df	Degrees of freedom for the t copula. Must be greater than 2. Required only for <code>pmvt_ghk()</code> .
engine	Character string specifying the conditional simulation engine used in the GHK approximation for the t copula. The default "mvnm" uses the multivariate mixture of normal (Standard implementation). The alternative "mvt" uses the direct Student-t conditional simulation scheme included for reproducibility of results reported in Nguyen and De Oliveira (2026). Ignored for the Gaussian copula.

Details

The GHK method approximates the multivariate normal or Student-t rectangle probability defining the copula likelihood by sequential simulation from truncated conditional distributions.

Two copula families are supported:

- **Gaussian copula** via `pmvn_ghk()`
- **Student-t copula** via `pmvt_ghk()`

In both cases, the latent dependence structure is parameterized through an $ARMA(p, q)$ process.

The GHK simulator approximates the multivariate normal or t rectangle probability by decomposing it into a sequence of one-dimensional truncated conditional distributions.

Value

By default, a numeric scalar giving the approximate log-likelihood. If `ret_llk = FALSE`, internal diagnostic quantities from the GHK simulator are returned (primarily for internal use).

References

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

Nguyen, Q. N., & De Oliveira, V. (2026). Scalable Likelihood Inference for Student- t Copula Count Time Series. Manuscript in preparation.

See Also

[pmvn_ghk](#), [pmvt_ghk](#)

Examples

```
## Gaussian copula example
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
a <- qnorm(ppois(y - 1, lambda = mu))
b <- qnorm(ppois(y, lambda = mu))

llk_gauss <- pmvn_ghk(lower = a, upper = b, tau = tau, od = arma_order,
                     M = 1000)

## Student-t copula example
df <- 8

sim_data_t <- sim_poisson(mu = mu, tau = tau, arma_order = arma_order,
                         nsim = 500, family = "t", df = df, seed = 1)

y_t <- sim_data_t$y
a_t <- qt(ppois(y_t - 1, lambda = mu), df = df)
b_t <- qt(ppois(y_t, lambda = mu), df = df)

llk_t <- pmvt_ghk(lower = a_t, upper = b_t, tau = tau, od = arma_order,
                  M = 1000, df = df)
```

pmvn_tmet

Approximate Log-Likelihood via TMET

Description

Computes the approximate log-likelihood for a count time series model based on a Gaussian and Student- t copula using the *Time Series Minimax Exponential Tilting (TMET)* method.

Usage

```
pmvn_tmet(lower, upper, tau, od, pm = 30, M = 1000, QMC = TRUE, ret_1lk = TRUE)
```

```
pmvt_tmet(
  lower,
  upper,
  tau,
  od,
  pm = 30,
  M = 1000,
  QMC = TRUE,
  ret_1lk = TRUE,
  df
)
```

Arguments

lower	Numeric vector of length n giving the lower bounds of the transformed latent variables.
upper	Numeric vector of length n giving the upper bounds of the transformed latent variables.
tau	Numeric vector of ARMA dependence parameters ordered as $c(\phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q)$, where ϕ_i are autoregressive (AR) coefficients and θ_j are moving-average (MA) coefficients.
od	Integer vector $c(p, q)$ specifying the AR and MA orders.
pm	Integer specifying the number of past lags used when approximating an $\text{ARMA}(p, q)$ process by an AR representation (required if $q > 0$).
M	Positive integer specifying the number of Monte Carlo or quasi-Monte Carlo samples used in the simulation.
QMC	Logical; if TRUE (default), quasi-Monte Carlo integration is used. Otherwise, standard Monte Carlo sampling is applied.
ret_1lk	Logical; if TRUE (default), returns the approximate log-likelihood. If FALSE, internal diagnostic quantities from the GHK simulator are returned. This option is primarily intended for internal use and methodological research.
df	Degrees of freedom for the t copula. Must be greater than 2. Required only for <code>pmvt_ghk()</code> .

Details

TMET exploits the autoregressive moving-average (ARMA) structure of the latent Gaussian or Scale mixture normal representation of Student-t process to evaluate high-dimensional multivariate normal/t rectangle probabilities via adaptive importance sampling with an optimal tilting parameter. The implementation combines the Innovations Algorithm for exact conditional mean and variance computation with exponential tilting, resulting in a scalable and variance-efficient likelihood approximation.

In this package, the latent dependence structure is parameterized through an $\text{ARMA}(p, q)$ process.

Value

A numeric scalar giving the approximate log-likelihood. If `ret_llk = FALSE`, diagnostic output from the TMET sampler is returned (primarily for research use).

References

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

Nguyen, Q. N., & De Oliveira, V. (2026). Scalable Likelihood Inference for Student- t Copula Count Time Series. Manuscript in preparation.

See Also

[pmvn_ghk](#), [pmvt_ghk](#)

Examples

```
## Gaussian copula example
mu <- 10
tau <- 0.2
arma_order <- c(1, 0)

sim_data <- sim_poisson(mu = mu, tau = tau, arma_order = arma_order,
                       nsim = 1000, seed = 1)

y <- sim_data$y
a <- qnorm(ppois(y - 1, lambda = mu))
b <- qnorm(ppois(y, lambda = mu))

# Approximate log-likelihood using TMET
llk_tmet <- pmvn_tmet(lower = a, upper = b,
                    tau = tau, od = arma_order)

llk_tmet

## Student-t copula example
df <- 8

sim_data_t <- sim_poisson(mu = mu, tau = tau, arma_order = arma_order,
                        nsim = 500, family = "t", df = df, seed = 1)

y_t <- sim_data_t$y
a_t <- qt(ppois(y_t - 1, lambda = mu), df = df)
b_t <- qt(ppois(y_t, lambda = mu), df = df)

llk_t <- pmvt_tmet(lower = a_t, upper = b_t, tau = tau, od = arma_order,
                  M = 1000, df = df)
```

predict.gctsc	<i>One-Step-Ahead Predictive Distribution for Copula Count Time Series Models</i>
---------------	---

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 using the estimation method stored in the fitted object (TMET or GHK). Summary statistics of the predictive distribution are returned, and optional scoring rules are computed if the observed value is supplied.

Usage

```
## S3 method for class 'gctsc'
predict(object, y_obs = NULL, X_test = NULL, ...)
```

Arguments

object	A fitted model object of class "gctsc", as returned by gctsc .
y_obs	Optional non-negative integer giving the observed value at the prediction time. If supplied, the Continuous Ranked Probability Score (CRPS) and Logarithmic Score (LOGS) are computed.
X_test	Covariate values at the prediction time point. Can be provided as: <ul style="list-style-type: none"> • a named numeric vector whose names match the covariates used during model fitting, or • a one-row data.frame or matrix with matching column names. <p>If the fitted model is intercept-only, X_test may be omitted, in which case the intercept is set to 1 automatically. An error is raised if more than one row is supplied.</p>
...	Ignored. Included for S3 method compatibility.

Details

The predictive distribution integrates over the latent copula dependence structure specified in the fitted model. For Gaussian copulas, multivariate normal rectangle probabilities are evaluated; for Student–t copulas, multivariate t rectangle probabilities are used.

Value

A list containing:

- mean: Predictive mean.
- median: Predictive median.
- mode: Predictive mode.

- variance: Predictive variance.
- p_y: Predictive probability mass function over $\theta:y_{\max}$.
- lower, upper: Bounds of the 95\
- CRPS: Continuous Ranked Probability Score (if y_obs is provided).
- LOGS: Logarithmic Score (if y_obs is provided).

References

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

Nguyen, Q. N., & De Oliveira, V. (2026). Scalable Likelihood Inference for Student- t Copula Count Time Series. Manuscript in preparation.

See Also

[gctsc](#), [arma.cormat](#)

Examples

```
# Simulate Poisson AR(1) data
set.seed(1)
y_sim <- sim_poisson(mu = 10, tau = 0.2,
                    arma_order = c(1, 0),
                    nsim = 1000,
                    family = "gaussian")$y

# Fit Gaussian copula model
fit <- gctsc(
  y ~ 1,
  data = data.frame(y = y_sim),
  marginal = poisson.marg(lambda.lower = 0),
  cormat = arma.cormat(p = 1, q = 0),
  method = "CE",
  family = "gaussian",
  options = gctsc.opts(M = 1000, seed = 42)
)

# One-step-ahead prediction
predict(fit)
```

```
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) as proposed by Dunn and Smyth (1996). When the likelihood is evaluated via simulation (TMET or GHK), the same engine is used to approximate the required conditional probabilities.

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 $F_t(y_t^-)$ and $F_t(y_t)$ denote the conditional CDF evaluated at $y_t - 1$ and y_t , respectively. The PIT value is computed as

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

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

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

Dunn, P. K. and Smyth, G. K. (1996), Randomized quantile residuals, *Journal of Computational and Graphical Statistics*, **5**(3): 236–244.

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*.

Nguyen, Q. N., and De Oliveira, V. (2026), Scalable Likelihood Inference for Student- t Copula Count Time Series, Manuscript in preparation.

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 = 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	<i>Weekly Rotavirus case counts across Germany</i>
------	--

Description

Weekly Rotavirus case counts across Germany

Usage

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

Format

A data frame with 1248 rows and 413 variables.

sim_gctsc	<i>Simulate from Gaussian and t Copula Time Series Models</i>
-----------	---

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 <code>nsim</code> to allow time-varying means.
tau	Numeric vector of ARMA dependence coefficients, ordered as <code>c(phi_1, ..., phi_p, theta_1, ..., theta_q)</code> , where ϕ_i are autoregressive (AR) coefficients and θ_j are moving-average (MA) coefficients. The model $\text{ARMA}(0, 0)$ is not supported.
arma_order	Integer vector <code>c(p, q)</code> 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 <code>family = "t"</code> .
seed	Optional integer used to set the random seed.
dispersion	Overdispersion parameter for negative binomial marginals. Must satisfy $\kappa > 0$, where $\text{Var}(Y) = \mu + \kappa\mu^2$. May be a scalar or a numeric vector of length <code>nsim</code> .
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 <code>nsim</code> .
prob	Success probability parameter(s) for binomial-type marginals. Must satisfy $0 < p < 1$. May be a scalar or a numeric vector of length <code>nsim</code> .
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 $\text{Var}(Y) = np(1-p)\{1 + (n-1)\rho\}$ and n is the number of trials. May be a scalar or a numeric vector of length <code>nsim</code> .

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 $\rho \rightarrow 0$ giving the binomial model.
- Zero–inflated marginals include a separate π_0 parameter that controls the extra probability mass at zero.

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.

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

* datasets

- campyl, 3
- KCWC, 8
- rota, 22

arima, 3

arma.cormat, 2, 5, 6, 9, 19

bbinom.marg (marginal.gctsc), 8

binom.marg (marginal.gctsc), 8

campyl, 3

coef.gctsc, 4

gctsc, 3, 4, 9, 18, 19, 21

gctsc.opts, 6, 7

KCWC, 8

marginal.gctsc, 8

negbin.marg, 5

negbin.marg (marginal.gctsc), 8

plot.gctsc, 10

pmvn_ce, 11, 12

pmvn_ghk, 13, 15, 17

pmvn_tmet, 15

pmvt_ce, 12

pmvt_ce (pmvn_ce), 11

pmvt_ghk, 15, 17

pmvt_ghk (pmvn_ghk), 13

pmvt_tmet (pmvn_tmet), 15

poisson.marg, 3, 5, 6

poisson.marg (marginal.gctsc), 8

predict.gctsc, 3, 9, 18

print.gctsc, 19

print.summary.gctsc, 20

residuals.gctsc, 11, 20

rota, 22

sim_bbinom (sim_gctsc), 22

sim_binom (sim_gctsc), 22

sim_gctsc, 22

sim_negbin (sim_gctsc), 22

sim_poisson (sim_gctsc), 22

sim_zib (sim_gctsc), 22

sim_zibb (sim_gctsc), 22

sim_zip (sim_gctsc), 22

summary.gctsc, 26

zib.marg, 5, 6

zib.marg (marginal.gctsc), 8

zibb.marg, 5, 6

zibb.marg (marginal.gctsc), 8

zip.marg (marginal.gctsc), 8