*Journal of Statistical Software*

*April 2017, Volume 77, Issue 8.* *doi: 10.18637/jss.v077.i08*

# Gaussian Copula Regression in R

**Guido Masarotto**
Università di Padova

**Cristiano Varin**
Università Ca' Foscari Venezia

**Abstract**

This article describes the R package **gcmr** for fitting Gaussian copula marginal regression models. The Gaussian copula provides a mathematically convenient framework to handle various forms of dependence in regression models arising, for example, in time series, longitudinal studies or spatial data. The package **gcmr** implements maximum likelihood inference for Gaussian copula marginal regression. The likelihood function is approximated with a sequential importance sampling algorithm in the discrete case. The package is designed to allow a flexible specification of the regression model and the dependence structure. Illustrations include negative binomial modeling of longitudinal count data, beta regression for time series of rates and logistic regression for spatially correlated binomial data.

*Keywords*: beta regression, Gaussian copula, longitudinal data, marginal regression, multivariate probit, R, spatial regression.

## 1. Introduction

Copula models (Joe 2014) are often considered to extend univariate regression models assuming independent responses to more general frameworks (e.g., Frees and Valdez 1998; Song 2000; Parsa and Klugman 2011; Kolev and Paiva 2009). The principal merit of the approach is that the specification of the regression model is separated from the dependence structure. This paper focuses on Gaussian copula regression method where dependence is conveniently expressed in the familiar form of the correlation matrix of a multivariate Gaussian distribution (Song 2000; Pitt, Chan, and Kohn 2006; Masarotto and Varin 2012). Gaussian copula regression models have been successfully employed in several complex applications arising, for example, in longitudinal data analysis (Frees and Valdez 2008; Sun, Frees, and Rosenberg 2008; Shi and Frees 2011; Song, Li, and Zhang 2013), genetics (Li, Boehnke, Abecasis, and Song 2006; He, Li, Edmondson, Raderand, and Li 2012), mixed data (Song, Li, and Yuan 2009; de Leon and Wu 2011; Wu and de Leon 2014; Jiryaie, Withanage, Wu, and de Leon

2016), spatial statistics (Kazianka and Pilz 2010; Bai, Kang, and Song 2014; Hughes 2015; Nikoloulopoulos 2016), time series (Guolo and Varin 2014).

Various authors discussed likelihood inference for Gaussian copula models (e.g., Masarotto and Varin 2012; Song *et al.* 2013; Nikoloulopoulos 2016). While likelihood computations for continuous responses are straightforward, the discrete case is considerably more difficult because the likelihood function involves multidimensional Gaussian integrals. Simulation methods are often employed to approximate the likelihood of Gaussian copula models in presence of high-dimensional discrete responses. Pitt *et al.* (2006) developed a Markov chain Monte Carlo algorithm for Bayesian inference, Masarotto and Varin (2012) adopted a sequential importance sampling algorithm, Nikoloulopoulos (2013) studied simulated maximum likelihood based on randomized quasi-Monte Carlo integration, see also Nikoloulopoulos (2016). Alternatively, composite likelihood methods (Varin, Reid, and Firth 2011) have been used to reduce the integral dimensionality and avoid inversion of high-dimensional covariance matrices (e.g., Zhao and Joe 2005; Bai *et al.* 2014; Hughes 2015).

Well-known limits of the Gaussian copula approach are the impossibility to deal with asymmetric dependence and the lack of tail dependence. These limits may impact the use of Gaussian copulas to model forms of dependence arising, for example, in extreme environmental events or in financial data. Conversely, this paper focuses on *working* Gaussian copulas used to conveniently handle dependence in regression analysis as described in Masarotto and Varin (2012). In other terms, the parameters of interest are the regression coefficients, while the dependence structure identified by the Gaussian copula is a nuisance component.

The CRAN archive contains several R (R Core Team 2017) packages devoted to copula modeling, but only a few consider copulas for regression modeling. The **weightedScores** package (Nikoloulopoulos and Joe 2015) is designed for longitudinal modeling of discrete responses. Regression parameters are estimated with optimal estimating equations, while dependence parameters are estimated by maximum composite likelihood based on a working Gaussian copula model. The principal merit of the approach is the robustness against misspecification of the copula distribution. **CopulaRegression** (Kraemer, Brechmann, Silvestrini, and Czado 2013) uses various copula models to describe the joint distribution of a pair of continuous and discrete random variables. The marginals are defined via generalized linear models and various parametric copulas are fitted to the bivariate joint distribution with the method of maximum likelihood. **copCAR** (Goren and Hughes 2017) implements a Gaussian copula regression model for areal data. Model parameters are estimated with composite likelihood and other types of likelihood approximation. The popular package **copula** (Hofert, Kojadinovic, Maechler, and Yan 2017) can also be used for multivariate regression with continuous responses, although it does not contain functions directly designed for regression, see the appendix of Yan (2007).

Package **gcmr** differs from the above packages in terms of the covered models, the fitting method in the discrete case and the functionalities for evaluation of the fitted model. Available marginal regression models include generalized linear models, negative binomial regression for overdispersed counts and beta regression for rates and proportions. Implemented Gaussian copula correlation matrices allow to handle various forms of dependence arising, for example, in longitudinal data analysis, time series and geostatistics. Advanced users may expand **gcmr** with additional marginal regression models and Gaussian copula correlation matrices as explained in the appendix. Models are fitted with the method of maximum likelihood in the continuous case and maximum simulated likelihood in the discrete case. Among the

advantages of likelihood inference there are the possibility to select models using information criteria such as AIC or BIC and the computation of the profile log-likelihood for inference on a focus parameter. Given potential concerns about the assumed copula, various types of robust sandwich standard errors are available. Moreover, **gcmr** implements residuals analysis to evaluate departures from the model assumptions.

The article is organized as follows. Section 2 briefly summarizes the theory of Gaussian copula regression with emphasis on model specification, likelihood inference, quantification of estimation uncertainty and model validation through residuals analysis. Section 3 describes the R implementation available within the **gcmr** package. Section 4 illustrates various **gcmr** functionalities using longitudinal data, time series and spatial data. The appendix provides some guidance to advanced users about how to extend the package capabilities by specification of regression models and dependence structures not yet available in **gcmr**.

## 2. Gaussian copula regression

Consider a vector of $n$ dependent variables $Y_1, \ldots, Y_n$. The marginal cumulative distribution of a single variable $Y_i$ is denoted by $F(\cdot|\boldsymbol{x}_i)$ and depends on a $p$-dimensional vector of covariates $\boldsymbol{x}_i$. We assume that $F(\cdot|\boldsymbol{x}_i)$ is parameterized in terms of a location parameter $\mu_i$, typically corresponding to the expected value $\mathsf{E}(Y_i|\boldsymbol{x}_i)$, that depends on $\boldsymbol{x}_i$ through the relationship

$$g_1(\mu_i) = \boldsymbol{x}_i^\top \boldsymbol{\beta}, \tag{1}$$

for a suitable link function $g_1(\cdot)$ and a $p$-dimensional vector of regression coefficients $\boldsymbol{\beta}$. This setting encompasses a variety of popular model classes such as, for example, generalized linear models (McCullagh and Nelder 1989) or beta regression (Cribari-Neto and Zeileis 2010). If the distribution of $Y_i$ includes a dispersion parameter, then the model can be extended to allow for variable dispersion with a second regression model (Cribari-Neto and Zeileis 2010)

$$g_2(\psi_i) = \boldsymbol{z}_i^\top \boldsymbol{\gamma}, \tag{2}$$

where $g_2(\cdot)$ is the dispersion link function, $\psi_i$ is the dispersion parameter associated to $Y_i$, $\boldsymbol{z}_i$ is the $q$-dimensional vector of dispersion covariates and $\boldsymbol{\gamma}$ is the corresponding vector of regression coefficients. For the sake of notational simplicity, thereafter the marginal cumulative univariate distribution of $Y_i$ will be denoted as $F(\cdot|\boldsymbol{x}_i)$ even in the case of variable dispersion, where indeed the model is $F(\cdot|\boldsymbol{x}_i, \boldsymbol{z}_i)$.

In Gaussian copula regression the dependence between the variables is modelled with a Gaussian copula so that the joint data cumulative distribution function is given by

$$\Pr(Y_1 \leq y_1, \ldots, Y_n \leq y_n) = \Phi_n(\epsilon_1, \ldots, \epsilon_n; \mathbf{P}),$$

where $\epsilon_i = \Phi^{-1}\{F(y_i|\boldsymbol{x}_i)\}$, with $\Phi(\cdot)$ denoting the univariate standard normal cumulative distribution function and $\Phi_n(\cdot; \mathbf{P})$ the $n$-dimensional multivariate standard normal cumulative distribution function with correlation matrix $\mathbf{P}$.

An equivalent formulation of the Gaussian copula model that emphasizes the regression setting is described in Masarotto and Varin (2012). Consider a regression model that links each variable $Y_i$ to a vector of covariates $\boldsymbol{x}_i$ through the generic relationship

$$Y_i = h(\boldsymbol{x}_i, \epsilon_i),$$

where $\epsilon_i$ is a stochastic error. Among many possible specifications of the function $h(\cdot)$ and the error $\epsilon_i$, the Gaussian copula regression model assumes that

$$h(\boldsymbol{x}_i, \epsilon_i) = F^{-1}\{\Phi(\epsilon_i)|\boldsymbol{x}_i\},$$

and the vector of error terms $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)^\top$ has a multivariate standard normal distribution with correlation matrix $\mathbf{P}$. In other terms, the Gaussian copula identifies a regression model constructed in way to (i) preserve the marginal univariate distributions and (ii) have multivariate normal errors.

An attractive feature of the Gaussian copula approach is that various forms of dependence can be expressed through suitable parametrization of the correlation matrix $\mathbf{P}$. For example, longitudinal data can be modelled with the working correlation matrices considered in generalized estimating equations (Song 2007, § 6), serial dependence in time series with a correlation matrix corresponding to an autoregressive and moving average process (Guolo and Varin 2014), spatial dependence with a correlation matrix induced by a Gaussian random field (Bai *et al.* 2014).

## 2.1. Likelihood inference

The **gcmr** package implements maximum likelihood inference for Gaussian copula regression models. Let $\boldsymbol{\theta}$ denote the vector of model parameters consisting of the parameters of the univariate marginals and the parameters belonging to the Gaussian copula correlation matrix. The likelihood function for $\boldsymbol{\theta}$ in the continuous case has the closed-form (e.g., Song 2000)

$$L(\boldsymbol{\theta}) = \phi_n(\epsilon_1, \ldots, \epsilon_n; \mathbf{P}) \prod_{i=1}^{n} \frac{f(y_i|\boldsymbol{x}_i)}{\phi(\epsilon_i)},$$

where $\phi(\cdot)$ indicates the univariate standard normal density, $\phi_n(\cdot; \mathbf{P})$ the $n$-dimensional standard normal density with correlation matrix $\mathbf{P}$, $f(\cdot|\boldsymbol{x}_i)$ the density of $Y_i$ given $\boldsymbol{x}_i$ and dependence of the density functions on $\theta$ is kept implicit for notational simplicity. The discrete case is considerably more involved because the likelihood is given by the $n$-dimensional normal integral

$$L(\boldsymbol{\theta}) = \int_{D_1} \cdots \int_{D_n} \phi_n(\epsilon_1, \ldots, \epsilon_n; \mathbf{P}) \mathrm{d}\epsilon_1 \cdots \mathrm{d}\epsilon_n, \tag{3}$$

where the integral domain is the Cartesian product of the intervals

$$D_i = [\Phi^{-1}\{F(y_i - 1|\boldsymbol{x}_i)\}, \Phi^{-1}\{F(y_i|\boldsymbol{x}_i)\}].$$

A remarkable amount of research has been addressed to the numerical approximation of multivariate normal integrals. For example, quasi-Monte Carlo approximations are available through the popular R package **mvtnorm** (Genz and Bretz 2009; Genz *et al.* 2017). We refer to Nikoloulopoulos (2013, 2016) for numerical studies about the efficiency of simulated maximum likelihood estimation based on **mvtnorm** in Gaussian copula models. Masarotto and Varin (2012) suggest that efficient numerical approximations of Equation 3 can be obtained by suitable generalizations of numerical methods for approximate likelihood inference in multivariate probit models. The most popular of such methods is probably the Geweke-Hajivassiliou-Keane (GHK) simulator (Keane 1994). The GHK simulator is a sequential importance sampling algorithm extensively studied in the computational econometrics literature, where it is commonly considered the gold standard for likelihood computation in

multivariate probit models, see, for example, Train (2003). The **gcmr** package implements maximum simulated likelihood estimation based on a variant of the GHK algorithm described in Masarotto and Varin (2012).

There are different options to evaluate the uncertainty of the maximum likelihood estimator. First, the classical approach that evaluates the uncertainty with the inverse of the observed Fisher information. Alternatively, the asymptotic variance of the maximum likelihood estimator can be estimated with the outer product of the scores derived from the predictive decomposition of the likelihood. The above options are valid if the Gaussian copula model is correctly specified. Given the potential concerns about the Gaussian copula assumption, then it is advisable to compare model-based standard errors with robust sandwich estimators. Significant divergences between model-based and robust standard errors provide indirect indication of model misspecification.

## 2.2. Residuals

Masarotto and Varin (2012) suggest to validate Gaussian copula regression models for *continuous responses* with predictive quantile residuals

$$r_i = \Phi^{-1}\{F(y_i|y_{i-1},\ldots,y_1;\hat{\boldsymbol{\theta}})\}, \tag{4}$$

where $\hat{\boldsymbol{\theta}}$ denotes the maximum likelihood estimator of $\boldsymbol{\theta}$. Under model conditions, quantile residuals $r_i$ are, approximately, realizations of uncorrelated standard normal variables and they are unrelated to the covariates $\boldsymbol{x}_i$. The predictive quantile residuals for *continuous responses* can be expressed in the familiar form of standardized residuals

$$r_i = \frac{\hat{\epsilon}_i - \hat{m}_i}{\hat{s}_i},$$

with $\hat{\epsilon}_i = \Phi^{-1}\{F(y_i|\boldsymbol{x}_i;\hat{\boldsymbol{\theta}})\}$, $\hat{m}_i = \mathsf{E}(\epsilon_i|\epsilon_{i-1},\ldots,\epsilon_1;\hat{\boldsymbol{\theta}})$ and $\hat{s}_i^2 = \mathsf{VAR}(\epsilon_i|\epsilon_{i-1},\ldots,\epsilon_1;\hat{\boldsymbol{\theta}})$.

In the *discrete case*, quantile residuals $r_i$ are defined as any arbitrary value in the interval $[\Phi^{-1}(a_i), \Phi^{-1}(b_i)]$, with $a_i = F(y_i - 1|y_{i-1},\ldots,y_1;\hat{\boldsymbol{\theta}})$ and $b_i = F(y_i|y_{i-1},\ldots,y_1;\hat{\boldsymbol{\theta}})$. Model checking can be based on the randomized quantile residuals

$$r_i^{\text{rnd}}(u_i) = \Phi^{-1}\{a_i + u_i(b_i - a_i)\},$$

where $u_i$ is generated from a uniform random variable on the unit interval (Dunn and Smyth 1996). Randomized quantile residuals $r_i^{\text{rnd}}(u_i)$ are realizations of uncorrelated standard normal variables under model conditions and they can be used as ordinary residuals for checking model assumptions. Since $r_i^{\text{rnd}}(u_i)$ are randomized, it is opportune to examine several sets of residuals before draw conclusions about the quality of model fitting.

Zucchini and MacDonald (2009) suggest to avoid randomization with the mid-interval quantile residuals $r_i^{\text{mid}} = \Phi^{-1}\{(a_i + b_i)/2\}$, which are, however, neither normally distributed nor uncorrelated.

Quantities $r_i$, $r_i^{\text{rnd}}$ and $r_i^{\text{mid}}$ are examples of conditional quantile residuals, because they involve the conditional distribution of $Y_i$ given the "previous" observations $y_{i-1},\ldots,y_1$. It is also possible to consider "marginal versions" of these residuals based on the univariate marginal distribution of $Y_i$ obtained in the continuous case with Equation 4 replaced by

$r_i^{\text{marg}} = \Phi^{-1}\{F(y_i|\boldsymbol{x}_i;\hat{\boldsymbol{\theta}})\}$. Differently from the conditional versions, marginal quantile residuals are useful for checking the assumptions about the marginal component of the model, but they are uninformative about the correctness of the Gaussian copula assumption.

# 3. Implementation in R

The main function of the **gcmr** package is `gcmr()` which allows to fit Gaussian copula models by maximum likelihood in the continuous case and by maximum simulated likelihood in the discrete case. The arguments of `gcmr()` are the following

```
gcmr(formula, data, subset, offset, marginal, cormat, start,
  fixed, options = gcmr.options(...), model = TRUE, ...)
```

The function has standard arguments for model-frame specification (Chambers and Hastie 1993) such as a `formula`, the possibility to restrict the analysis to a `subset` of the `data`, to set an `offset`, or to fix `contrasts` for factors. The specific arguments of `gcmr()` include the two key arguments `marginal` and `cormat`, which specify the marginal part of the model and the copula correlation matrix, respectively. Finally, there are three *optional* arguments to supply starting values (`start`), fix the values of some parameters (`fixed`) and set the fitting options (`options`). The rest of this section describes the components of `gcmr()` and the related methods.

## 3.1. Two-part formulas

The basic formula allowed in **gcmr** is of type `y ~ x1 + x2` and it specifies the regression model for the mean response of Equation 1 with the link function $g_1(\cdot)$ defined in the argument `marginal` as explained in Section 3.2 below. Following the implementation of beta regression in package **betareg** (Cribari-Neto and Zeileis 2010), the **gcmr** package also allows to specify a second regression model for the dispersion through a "two-part" formula of type `y ~ x1 + x2 | z1 + z2` using functionalities inherited from package **Formula** (Zeileis and Croissant 2010). In the two-part formula case, the model has the same mean regression expression `y ~ x1 + x2`, while the dispersion parameter is modelled as a function of the linear predictor `~ z1 + z2`. Package **gcmr** assumes a log-linear model $g_2(\cdot) = \log(\cdot)$ for the dispersion regression model of Equation 2.

## 3.2. Specification of the marginal model

The marginal model $F(\cdot|\boldsymbol{x}_i)$ is specified through an object of class `marginal.gcmr` set in the argument `marginal` of function `gcmr()`. The marginal distributions available in **gcmr** version 1.0.0 are beta, binomial, gamma, Gaussian, negative binomial, Poisson and Weibull, see Table 1. For each of these distributions, it is possible to choose a link function that relates the mean of the response to the linear predictor as in traditional generalized linear models. All the link functions available in the class `link-glm` are allowed. Gaussian marginals are included in **gcmr** for completeness, but it is not recommended to use **gcmr** for fitting multivariate normal models trivially arising from the combination of Gaussian marginals with a Gaussian copula. In fact, the package **gcmr** is designed to work with Gaussian copula models with generic univariate marginal distributions and thus it is not numerically efficient for inference

| marginal.gcmr | Distribution | Dispersion |
|---|---|---|
| `beta.marg(link = "logit")` | beta | yes |
| `binomial.marg(link = "logit")` | binomial | no |
| `Gamma.marg(link = "inverse")` | gamma | yes |
| `gaussian.marg(link = "identity")` | Gaussian | yes |
| `negbin.marg(link = "log")` | negative binomial | yes |
| `poisson.marg(link = "log")` | Poisson | no |
| `weibull.marg(link = "log")` | Weibull | yes |

Table 1: Marginals models available in **gcmr** version 1.0.0 with the default link function. The column "Dispersion" identifies the distributions with a dispersion parameter.

| cormat.gcmr | Correlation |
|---|---|
| `arma.cormat(p, q)` | ARMA($p$, $q$) |
| `cluster.cormat(id, type)` | longitudinal/clustered data |
| `ind.cormat()` | independence |
| `matern.cormat(D, alpha)` | Matérn correlation |

Table 2: Correlation models available in **gcmr** version 1.0.0.

in multivariate linear Gaussian models, where the availability of analytic results allows for a significant speed-up of computations.

The user may also construct their own marginal model by specifying a new object of class `marginal.gcmr` as explained in Appendix A.1.

### 3.3. Specification of the correlation structure

The correlation matrix $\mathbf{P}$ of the Gaussian copula is specified through an object of class `cormat.gcmr` set in the argument `cormat` of function `gcmr()`. Version 1.0.0 of **gcmr** includes four correlation structures of wide applicability, see Table 2. The working independence correlation option is similar in spirit to that of generalized estimating equations. The other three correlation structures allow to deal with time series, clustered or longitudinal data and spatial data. Clustered and longitudinal data can be analyzed with `cluster.cormat(id, type)` constructed upon functions inherited from package **nlme** (Pinheiro, Bates, DebRoy, Sarkar, and R Core Team 2017). The inputs of `cluster.cormat()` are the vector of subject `id` and the `type` of correlation with possible options `"independence"`, `"ar1"`, `"ma1"`, `"exchangeable"` and `"unstructured"`. Subject `id` is a vector of the same length as the number of observations. Data are assumed to be sorted in such a way that observations from the same subject (or cluster) are contiguous, otherwise **gcmr** stops and returns an error message. Serial dependence in time series can be described with function `arma.cormat(p, q)`, which receives the orders $p$ and $q$ of the ARMA($p$, $q$) process as input. Spatially correlated data can be modelled by assuming a Mátern spatial correlation function set by function `matern.cormat(D, alpha)`, where `D` is the matrix of the distances between observations and `alpha` is the shape parameter (Diggle and Ribeiro 2007). Function `matern.cormat()` is constructed upon function `matern()` of the **geoR** package (Ribeiro Jr and Diggle 2016). The default value for parameter `alpha` is `0.5`, and it corresponds to an exponential correlation model.

As for the marginals, the user is allowed to construct their own correlation matrix by specifying a new object of class `cormat.gcmr`, see Appendix A.2.

### 3.4. Fitting options

The fitting options in `gcmr()` are set by argument `options` or by a direct call to function

```
gcmr.options(seed = round(runif(1, 1, 1e+05)), nrep = c(100, 1000),
  no.se = FALSE, method = c("BFGS", "Nelder-Mead", "CG"), ...)
```

Available arguments are `seed`, for fixing the pseudo-random seed used in the GHK algorithm to approximate the likelihood function with discrete responses, `nrep`, for setting the number of the Monte Carlo replications in the GHK algorithm, `no.se`, for choosing whether computing the standard errors or not, and `method`, for selection of the optimization method to be passed to `optim()`. The default optimization algorithm is the quasi-Newton BFGS algorithm. It is possible to provide a vector of Monte Carlo replications to `nrep`, so that the model is fitted with a sequence of different Monte Carlo sizes. In this case, the starting values for likelihood optimization are taken from the previous fitting. A reasonable strategy is to fit the model with a small Monte Carlo size to obtain sensible starting values and then refit with a larger Monte Carlo size. The default Monte Carlo size is 100 for the first optimization and 1,000 for the second and definitive optimization. If the responses are continuous, then the likelihood function has a closed-form expression and the values of `seed` and `nrep` are ignored.

### 3.5. Methods

The returned fitted-model object of class **gcmr** is a list that contains, among others, the maximum likelihood estimates, the maximized log-likelihood and numerical estimates of the Hessian and the Jacobian of the log-likelihood computed at the maximum likelihood estimate. A set of standard methods is available to extract information from the fitted model, see Table 3. Most of the functions and methods have standard syntax as in other R packages oriented to regression analysis, see, for example, **betareg** (Cribari-Neto and Zeileis 2010).

The `plot()` method produces various diagnostic plots of the fitted **gcmr** object that include scatterplots of the quantile residuals against the indices of the observations or against the linear predictor, the normal probability plot with confidence bands based on the implementation in the **car** package (Fox and Weisberg 2011), the scatterplot of the predicted values against the observed values, autocorrelation and partial autocorrelation plots of the residuals. The default behavior of the `plot()` method adapts to the type of correlation matrix in way that, for example, autocorrelation plots are automatically displayed for ARMA($p$, $q$) correlation specified with by function `arma.cormat(p, q)`.

The quantile residuals are computed by method

```
residuals(object, type = c("conditional", "marginal"),
  method = c("random", "mid"), ...)
```

where argument `type` allows to choose between `"conditional"` or `"marginal"` quantile residuals, see Section 2.2. Argument `method` is active only in the discrete case to select between `"random"` quantile residuals or `"mid"` interval quantile residuals.

The profile log-likelihood can be obtained with a call to method

| Function | Description |
|---|---|
| `print()` | simple printed display of coefficient estimates |
| `summary()` | standard regression output |
| `coef()` | coefficient estimates |
| `vcov()` | covariance matrix of coefficient estimates |
| `fitted()` | fitted means for observed data |
| `residuals()` | quantile residuals |
| `estfun()` | estimating functions for sandwich estimators (Zeileis 2006) |
| `bread()` | "bread" matrix for sandwich estimators (Zeileis 2006) |
| `terms()` | terms of model components |
| `model.frame()` | model frame |
| `model.matrix()` | model matrix |
| `logLik()` | maximized log-likelihood |
| `plot()` | diagnostic plots of quantile residuals |
| `profile()` | profile likelihood for focus coefficients |
| `coeftest()` | partial Wald tests of coefficients |
| `waldtest()` | Wald tests of nested models |
| `lrtest()` | likelihood ratio tests of nested models |
| `AIC()` | information criteria |

Table 3: Functions and methods available for objects of class `gcmr`.

```
profile(fitted, which, low, up, npoints = 10, display = TRUE,
  alpha = 0.05, progress.bar = TRUE, ...)
```

where argument `which` is the index of the parameter to be profiled, `low` and `up` are the lower and the upper limits used in the computation, `npoints` is the number of points used in the computation of the profile likelihood, `alpha` is the significance level, `display` controls whether the profile likelihood should be plotted or not and `progress.bar` sets a "progress bar" to visualize the progression of the time-consuming profile likelihood computation. If the values of limits `low` and `up` are not provided, then they are set equal to the estimated parameter minus and plus three times the standard error, respectively.

# 4. Applications

The usage of **gcmr** is illustrated below with three different data sets covering various forms of dependence frequently arising in real applications.

## 4.1. Longitudinal count data

The first example considers the well-known longitudinal study on epileptic seizures described in Diggle, Heagerty, Liang, and Zeger (2002):

```
R> data("epilepsy", package = "gcmr")
```

The data comprise information about 59 individuals observed at five different occasions each. The baseline observation consists of the number of epileptic seizures in a eight-week interval,

followed by four measurements collected at subsequent visits every two weeks. Available variables are the patient identifier `id`, the patient `age`, the indicator `trt` whether the patient is treated with progabide (`trt = 1`) or not (`trt = 0`), the number of epileptic seizures `counts`, the observation period `time` in weeks, that is `time = 8` for baseline and `time = 2` for subsequent visits, and the indicator `visit` whether the observation corresponds to a visit (`visit = 1`) or the baseline (`visit = 0`). Diggle *et al.* (2002) analyzed the seizure data with the method of generalized estimating equations assuming a log-linear regression model for `counts` with the logarithm of `time` used as offset and covariates `trt`, `visit` and their interaction. Moreover, Diggle *et al.* (2002) suggested to omit an outlier patient – here corresponding to patient `id = 49` – with an extremely high seizure count at baseline (151 counts) that even double after treatment (302 counts after 8 weeks of measurement). Indeed, estimated model coefficients vary considerably if this patient is set aside.

The corresponding Gaussian copula analysis described below assumes a negative binomial marginal distribution with mean specified as in Diggle *et al.* (2002). We start the analysis assuming a working independence correlation matrix for the Gaussian copula:

```
R> mod.ind <- gcmr(counts ~ offset(log(time)) + visit + trt + visit:trt,
+    data = epilepsy, subset = (id != 49), marginal = negbin.marg,
+    cormat = cluster.cormat(id, type = "ind"))
R> summary(mod.ind)


Call:
gcmr(formula = counts ~ offset(log(time)) + visit + trt + visit:trt,
     data = epilepsy, subset = (id != 49), marginal = negbin.marg,
     cormat = cluster.cormat(id, type = "ind"))


Coefficients marginal model:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.34759    0.16649   8.094 5.77e-16 ***
visit        0.11187    0.18802   0.595    0.552
trt         -0.10685    0.23057  -0.463    0.643
visit:trt   -0.30237    0.26118  -1.158    0.247
dispersion   0.73421    0.07153  10.264  < 2e-16 ***

No coefficients in the Gaussian copula
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

log likelihood = 948.06,  AIC = 1906.1
```

The `summary()` method computes traditional standard errors derived from the inverse of the observed Fisher information. A more appropriate choice for these longitudinal data is provided by the sandwich estimator that can be computed with the **sandwich** package (Zeileis 2004, 2006) and conveniently visualized with function `coeftest()` from package **lmtest** (Zeileis and Hothorn 2002):

```
R> library("sandwich")
R> library("lmtest")
R> coeftest(mod.ind, vcov. = sandwich(mod.ind))

z test of coefficients:

             Estimate Std. Error z value  Pr(>|z|)
(Intercept)  1.347586   0.157997  8.5292 < 2.2e-16 ***
visit        0.111869   0.115634  0.9674    0.3333
trt         -0.106846   0.194159 -0.5503    0.5821
visit:trt   -0.302373   0.169183 -1.7873    0.0739 .
dispersion   0.734208   0.095039  7.7253 1.116e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Robust sandwich standard errors essentially confirm the previous results. The strong significance of the dispersion parameter provides support to the choice of the negative binomial marginal in place of the Poisson distribution.

However, a more accurate description of the data also accounts for the serial correlation of the observations from the same subject. For example, the model can be re-estimated with the AR(1) Gaussian copula correlation matrix:

```
R> mod.ar1 <- update(mod.ind, cormat = cluster.cormat(id, "ar1"),
+    seed = 12345, nrep = 100)
```

The previous command illustrates the use of the **gcmr** fitting options. The random number generator `seed` is fixed to ensure reproducibility of the results, while the number of Monte Carlo replications `nrep` is set to a number lower than the default, a possibility that it is useful during the model specification phase.

Robust sandwich standard errors confirm the presence of substantial autocorrelation between observations from the same patient. In fact, the estimated AR(1) coefficient is equal to `0.63` with standard error `0.05`:

```
R> coeftest(mod.ar1, vcov. = sandwich(mod.ar1))

z test of coefficients:

             Estimate Std. Error z value  Pr(>|z|)
(Intercept)  1.307160   0.162154  8.0612 7.553e-16 ***
visit        0.156902   0.108164  1.4506   0.14689
trt         -0.010332   0.202333 -0.0511   0.95927
visit:trt   -0.420571   0.164879 -2.5508   0.01075 *
dispersion   0.636966   0.077159  8.2552 < 2.2e-16 ***
ar1          0.628785   0.048897 12.8594 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Differently from the working independence model, the autoregressive model identifies a significant effect of the interaction between visit and treatment that was undetected with the working independence model. The result qualitatively agrees with that obtained with the generalized estimating equation analysis by Diggle *et al.* (2002) that can be reproduced with package **geepack** (Yan 2002; Højsgaard, Halekoh, and Yan 2006):

```
R> library("geepack")
R> gee.ar1 <- geeglm(counts ~ offset(log(time)) + visit + trt + visit:trt,
+    data = epilepsy, id = id, subset = (id != 49), family = poisson,
+    corstr = "ar1")
R> summary(gee.ar1)

Call:
geeglm(formula = counts ~ offset(log(time)) + visit + trt + visit:trt,
    family = poisson, data = epilepsy, subset = (id != 49), id = id,
    corstr = "ar1")

 Coefficients:
            Estimate  Std.err   Wald Pr(>|W|)
(Intercept)  1.31383  0.16159 66.103 4.44e-16 ***
visit        0.15094  0.11077  1.857   0.1730
trt         -0.07973  0.19831  0.162   0.6877
visit:trt   -0.39872  0.17454  5.218   0.0223 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    10.61    2.35

Correlation: Structure = ar1  Link = identity

Estimated Correlation Parameters:
      Estimate Std.err
alpha   0.7831 0.05192
Number of clusters:   58   Maximum cluster size: 5
```

Among the advantages of the likelihood analysis implemented in **gcmr** with respect to non-likelihood methods such as generalized estimating equations, there is the possibility to compute profile log-likelihoods. Consider, for example, the profile log-likelihood for the interaction effect of visit with treatment that can be obtained with a call to the `profile()` method:

```
R> profile(mod.ar1, which = 4)
```

where argument `which` is equal to 4 because the interaction effect corresponds to the fourth model parameter. The profile log-likelihood reported in Figure 1 illustrates the significant negative coefficient associated to the interaction of visit with treatment.

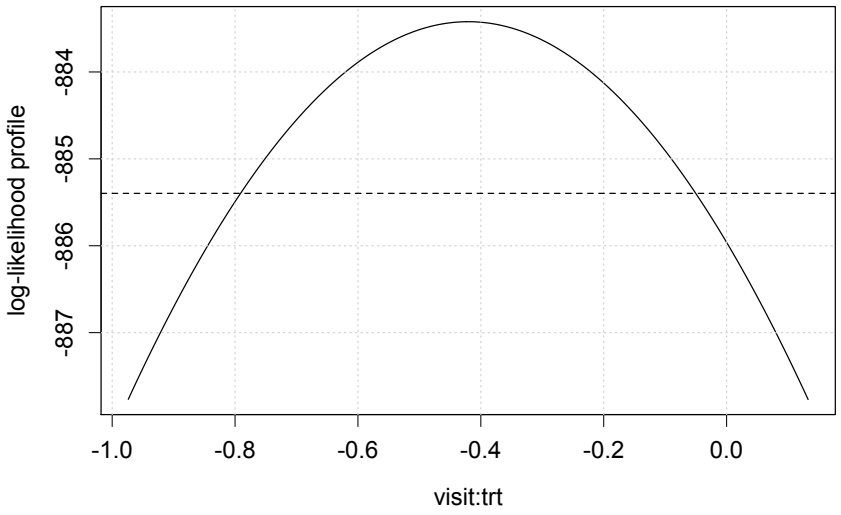


Figure 1: Seizure data. Profile log-likelihood for the interaction between visit and treatment.

### 4.2. Time series of rates

The second example regards the time series of the hidden unemployment rate (HUR) in São Paulo, Brazil, obtained from the database of the Applied Economic Research Institute (IPEA) of the Brazilian Federal Government (http://www.ipea.gov.br/):

```
R> data("HUR", package = "gcmr")
R> plot(HUR, ylab = "rate", xlab = "time")
```

The data, displayed in Figure 2, were analyzed by Roca and Cribari-Neto (2009) with an observation-driven beta autoregressive and moving average model. As an alternative to the analysis made by Roca and Cribari-Neto (2009), we consider a Gaussian copula model with marginal beta distribution and ARMA($p$, $q$) copula correlation. The mean and precision of the beta marginals are assumed both to depend on a linear trend. In order to avoid numerical instabilities, the trend is centered and scaled:

```
R> trend <- scale(time(HUR))
```

Below we illustrate the model with ARMA(1, 3) errors. This model was selected because it has the minimum AIC value among the sixteen ARMA($p$, $q$) models obtained with orders $p$ and $q$ ranging from 0 to 3:

```
R> mod <- gcmr(HUR ~ trend | trend, marginal = beta.marg,
+    cormat = arma.cormat(1, 3))
```

The previous command illustrates the use of the extended formula `HUR ~ trend | trend` to specify that both the mean and the dispersion depend on the (scaled) trend. The summary of the fitted model confirms the presence of a statistically significant trend:

```
R> summary(mod)
```

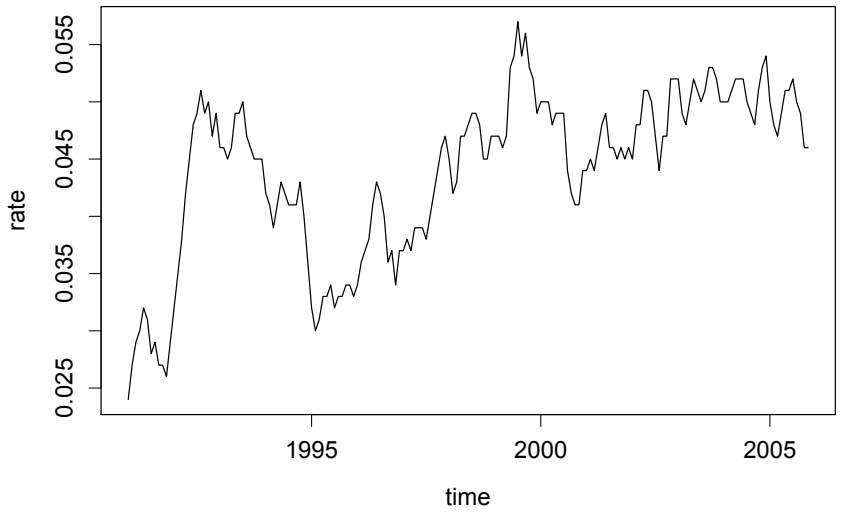


Figure 2: Hidden unemployment rate data in São Paulo, Brazil.

```
Call:
gcmr(formula = HUR ~ trend | trend, marginal = beta.marg,
     cormat = arma.cormat(1, 3))


Coefficients marginal model:
                     Estimate Std. Error z value Pr(>|z|)
mean.(Intercept)     -3.10775    0.04612 -67.385  < 2e-16 ***
mean.trend            0.11509    0.03944   2.918  0.00352 **
precision.(Intercept) 7.24869    0.37195  19.488  < 2e-16 ***
precision.trend       0.36108    0.11366   3.177  0.00149 **

Coefficients Gaussian copula:
    Estimate Std. Error z value Pr(>|z|)
ar1  0.91032    0.04589  19.836  < 2e-16 ***
ma1  0.34152    0.09467   3.608 0.000309 ***
ma2  0.47147    0.08574   5.499 3.83e-08 ***
ma3 -0.42904    0.10300  -4.165 3.11e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

log likelihood = -895.06,  AIC = -1774.1
```

Evidence that the assumptions of the above model are met is provided by graphical inspection of quantile residuals reported in Figure 3:

```
R> par(mfrow = c(2, 2))
R> plot(mod)
```

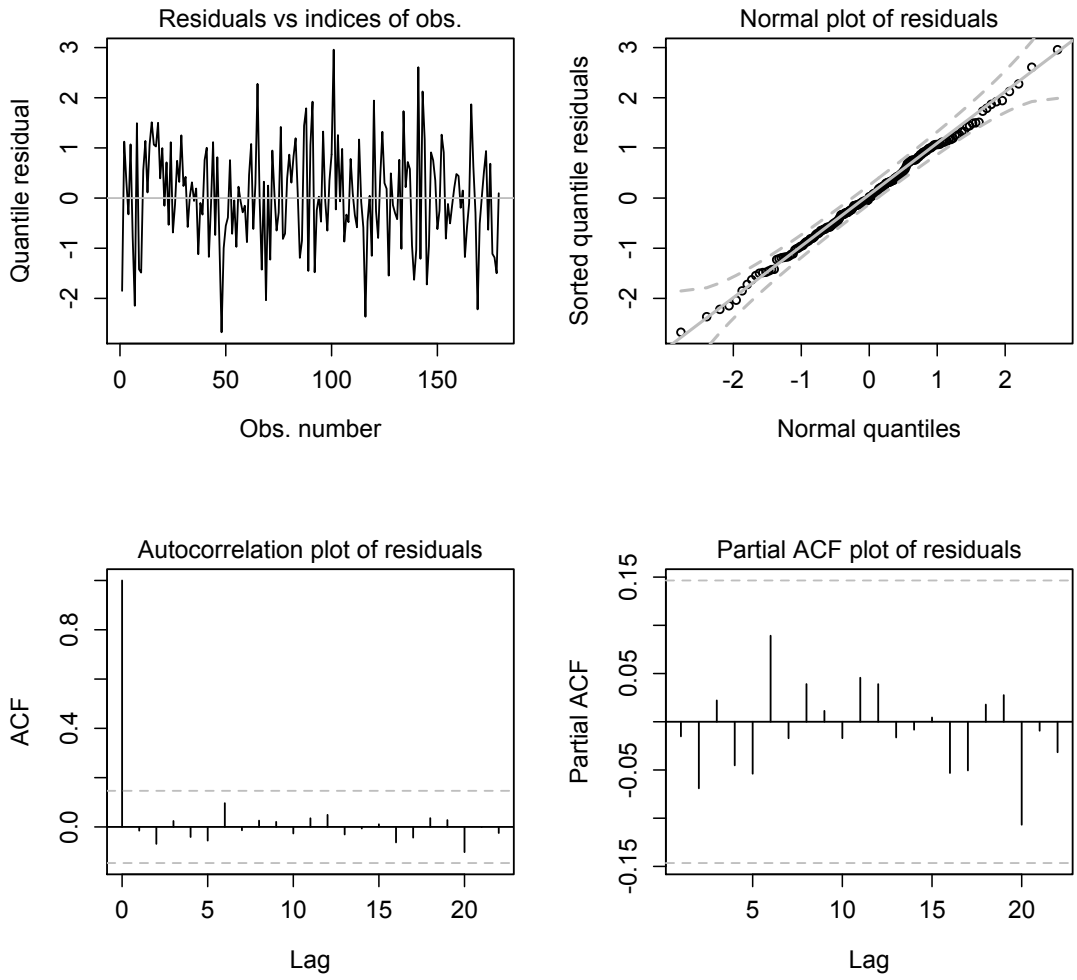


Figure 3: Hidden unemployment rate data. Standard diagnostic plots for time series data produced by the `plot` method.

### 4.3. Spatially correlated binomial data

The last example regards the malaria prevalence in children recorded at 65 villages in Gambia. Differently from the original data presented in Thomson, Connor, D'Alessandro, Rowlingson, Diggle, and Cresswell (1999) and available in the **geoR** package (Ribeiro Jr and Diggle 2016), here we consider aggregated data at village level available through **gcmr** with the data frame `malaria`:

```
R> data("malaria", package = "gcmr")
```

The data contain information about the village coordinates (`x`, `y`), the number of sampled children (`size`) with malaria (`cases`) in each village, the mean age of the sampled children in each village (`age`), the frequency of sampled children who regularly sleep under a bed-net in each village (`netuse`), the frequency of sampled children whose bed-net is treated (`treated`), a satellite-derived measure of the greenness of vegetation in the immediate proximity of the village (`green`), the indicator variable denoting the presence (`1`) or absence (`0`) of a health

center in the village (`pch`) and an indicator of geographical regions characterized by potentially different malaria risk (`area`). We refer to Diggle and Ribeiro (2007) for more details. The aim is to model the relationship between the number of cases and the various covariates, while accounting for the potential presence of spatial dependence of malaria spread between the villages.

The first step of the data analysis is the construction of the matrix of the distances between the villages using, for example, the function `spDists()` from package **sp** (Pebesma and Bivand 2005; Bivand, Pebesma, and Gomez-Rubio 2013):

```
R> library("sp")
R> D <- spDists(cbind(malaria$x, malaria$y)) / 1000
```

The distances are expressed in kilometers through scaling by factor 1,000. Scaling is helpful for avoiding potential numerical instabilities in the estimation of the spatial dependence parameter.

The first model describes the cases of malaria with a spatial Gaussian copula logistic regression model. The covariates are `netuse`, `pch` and `green` scaled by factor 100. Spatial dependence is modelled with an exponential correlation matrix corresponding to the default value of the shape parameter (`alpha = 0.5`) in `matern.cormat(D, alpha)`:

```
R> mod <- gcmr(cbind(cases, size-cases) ~ netuse + I(green / 100) + phc,
+    data = malaria, marginal = binomial.marg, cormat = matern.cormat(D),
+    seed = 12345)
R> summary(mod)

Call:
gcmr(formula = cbind(cases, size - cases) ~ netuse + I(green/100) +
     phc, data = malaria, marginal = binomial.marg, cormat =
     matern.cormat(D), seed = 12345)


Coefficients marginal model:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.8276     0.4065  -2.036   0.0418 *
netuse        -1.1758     0.1605  -7.325 2.40e-13 ***
I(green/100)   2.9487     0.7498   3.933 8.39e-05 ***
phc           -0.4052     0.1019  -3.978 6.95e-05 ***

Coefficients Gaussian copula:
    Estimate Std. Error z value Pr(>|z|)
tau   1.5086     0.3771   4 6.33e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

log likelihood = 252.68,  AIC = 515.36
```

Covariates `netuse` and `phc` are associated to a significant reduction of the malaria cases while `green` is associated to a higher risk of disease. The maximum simulated likelihood estimate

of the dependence parameter `tau` is 1.51 km, a value that indicates the presence of significant but weak spatial dependence.

The second model includes an additional effect due to covariate `area`:

```
R> mod.area <- update(mod, . ~ . + area)
R> summary(mod.area)

Call:
gcmr(formula = cbind(cases, size - cases) ~ netuse + I(green/100) +
     phc + area, data = malaria, marginal = binomial.marg,
     cormat = matern.cormat(D), seed = 12345)


Coefficients marginal model:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    0.2046     0.6398   0.320 0.749072
netuse        -0.6387     0.1789  -3.570 0.000357 ***
I(green/100)  -0.0611     1.4057  -0.043 0.965327
phc           -0.4081     0.1073  -3.802 0.000143 ***
area2         -0.6133     0.1792  -3.422 0.000621 ***
area3         -0.7515     0.1945  -3.864 0.000112 ***
area4          0.3441     0.2432   1.415 0.157121
area5          0.6840     0.2316   2.953 0.003142 **

Coefficients Gaussian copula:
    Estimate Std. Error z value Pr(>|z|)
tau   0.6816     0.3621   1.882   0.0598 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

log likelihood = 222.29,  AIC = 462.57
```

The inclusion of `area` in the model yields a large drop in the AIC statistics:

```
R> AIC(mod, mod.area)

         df      AIC
mod       5 515.4
mod.area  9 462.3
```

The summary confirms that covariate `area` contains relevant information about the geographic variation of malaria risk in the study region. Indeed, the estimate of the spatial dependence parameter `tau` in model `mod.area` shows that the residual spatial dependence is essentially negligible.

Finally, graphical diagnostics reported in Figure 4 suggest that the model conditions are met:

```
R> par(mfrow = c(2, 2))
R> plot(mod.area)
```

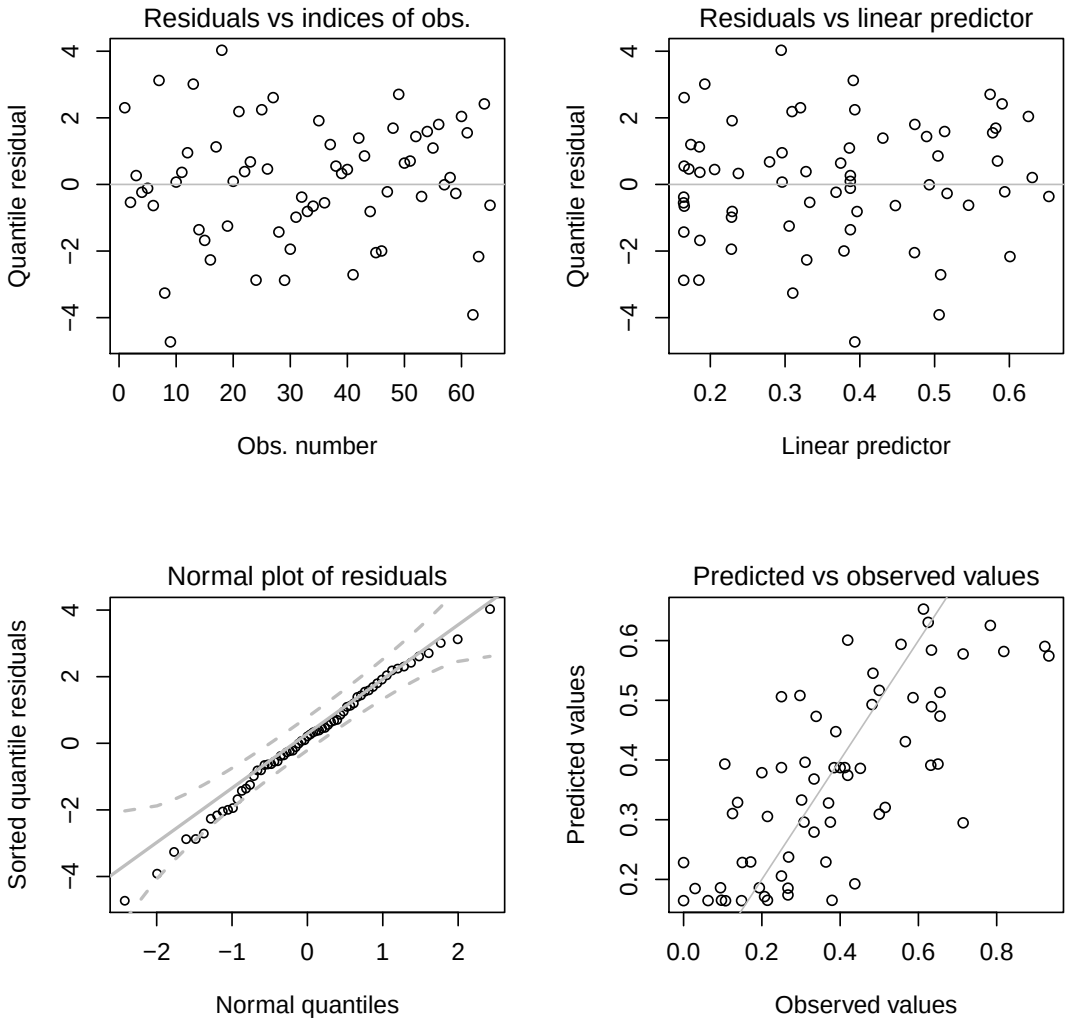


Figure 4: Malaria data. Standard diagnostic plots produced by the `plot` method.

## 5. Conclusions

This article presented the R implementation of Gaussian copula marginal regression available in the **gcmr** package. The discussed examples illustrate the capability of the package to handle various types of data and dependence structures. Models are fitted with the method of maximum (simulated) likelihood that requires repeated Cholesky factorization of the Gaussian copula correlation matrix. In the current version of **gcmr**, the order of computations needed for likelihood evaluation is $O(n^3)$, with $n$ denoting the number of observations. In case of large data sets, consisting, for example, of several thousands of observations, the computational cost may prevent *routine* use of **gcmr**. However, the Cholesky factorization can be implemented more efficiently for some specific dependence structures. For example, autoregressive and moving average correlation matrices can be factorized in a linear number of computations exploiting the Kalman filter through the state space representation.

Future research will focus on implementation of computationally convenient methods to handle specific dependence forms within the general framework of Gaussian copula regres-

sion. Promising approaches include composite likelihoods to reduce the computational effort through convenient likelihood factorizations (Varin *et al.* 2011) and sparse methods designed to approximate the Gaussian copula correlation matrix with a more manageable block-diagonal matrix.

Several authors exploited Gaussian and t copulas to construct joint regression models for multiple responses, also of mixed type (e.g., Frees and Valdez 2008; Song *et al.* 2009; Wu and de Leon 2014; Jiryaie *et al.* 2016). Methods for handling multiple responses are planned to be included in future versions of **gcmr**.

## Acknowledgments

The Authors are grateful to Achim Zeileis for worthwhile suggestions about both the design of the package methods and this paper. The Authors also thank the Associate Editor and the referee for helpful comments.

## References

Bai Y, Kang J, Song PXK (2014). "Efficient Pairwise Composite Likelihood Estimation for Spatial-Clustered Data." *Biometrics*, **70**(3), 661–670. doi:10.1111/biom.12199.

Bivand RS, Pebesma E, Gomez-Rubio V (2013). *Applied Spatial Data Analysis with R*. Springer-Verlag, New York. doi:10.1007/978-1-4614-7618-4.

Chambers JM, Hastie TJ (1993). *Statistical Models in S*. Chapman & Hall, London.

Cribari-Neto F, Zeileis A (2010). "Beta Regression in R." *Journal of Statistical Software*, **34**(2), 1–24. doi:10.18637/jss.v034.i02.

de Leon AR, Wu B (2011). "Copula-Based Regression Models for a Bivariate Mixed Discrete and Continuous Outcome." *Statistics in Medicine*, **30**(2), 175–185. doi:10.1002/sim.4087.

Diggle PJ, Heagerty P, Liang KY, Zeger SL (2002). *Analysis of Longitudinal Data*. 2nd edition. Oxford University Press, Oxford.

Diggle PJ, Ribeiro PJJ (2007). *Model-Based Geostatistics*. Springer-Verlag, New York.

Dunn PK, Smyth GK (1996). "Randomized Quantile Residuals." *Journal of Computational and Graphical Statistics*, **5**, 236–244. doi:10.2307/1390802.

Fox J, Weisberg S (2011). *An R Companion to Applied Regression*. 2nd edition. Sage, Thousand Oaks.

Frees EW, Valdez EA (1998). "Understanding Relationships Using Copulas." *North American Actuarial Journal*, **2**(1), 1–25. doi:10.1080/10920277.1998.10595667.

Frees EW, Valdez EA (2008). "Hierarchical Insurance Claims Modeling." *Journal of the American Statistical Association*, **103**(484), 1457–1469. doi:10.1198/016214508000000823.

Genz A, Bret F, Miwa T, Mi X, Leisch F, Scheipl F, Hothorn T (2017). ***mvtnorm**: Multivariate Normal and t Distributions*. R package version 1.0-6, URL https://CRAN.R-project.org/package=mvtnorm.

Genz A, Bretz F (2009). *Computation of Multivariate Normal and t Probabilities*. Lecture Notes in Statistics. Springer-Verlag, Heidelberg. doi:10.1007/978-3-642-01689-9.

Goren E, Hughes J (2017). ***copCAR**: Fitting the copCAR Regression Model for Discrete Areal Data*. R package version 2.0-2, URL https://CRAN.r-project.org/package=copCAR.

Guolo A, Varin C (2014). "Beta Regression for Time Series Analysis of Bounded Data, with Application to Canada Google Flu Trends." *The Annals of Applied Statistics*, **8**(1), 74–88. doi:10.1214/13-aoas684.

He J, Li H, Edmondson AC, Raderand DJ, Li M (2012). "A Gaussian Copula Approach for the Analysis of Secondary Phenotypes in Case-Control Genetic Association Studies." *Biostatistics*, **13**(3), 497–508. doi:10.1093/biostatistics/kxr025.

Hofert M, Kojadinovic I, Maechler M, Yan J (2017). ***copula**: Multivariate Dependence with Copulas*. R package version 0.999-16, URL https://CRAN.R-project.org/package=copula.

Højsgaard S, Halekoh U, Yan J (2006). "The R Package **geepack** for Generalized Estimating Equations." *Journal of Statistical Software*, **15**(2), 1–11. doi:10.18637/jss.v015.i02.

Hughes J (2015). "copCAR: A Flexible Regression Model for Areal Data." *Journal of Computational and Graphical Statistics*, **24**(3), 733–755. doi:10.1080/10618600.2014.948178.

Jiryaie F, Withanage N, Wu B, de Leon AR (2016). "Gaussian Copula Distributions for Mixed Data, with Application in Discrimination." *Journal of Statistical Computation and Simulation*, **86**(9), 1643–1659. doi:10.1080/00949655.2015.1077386.

Joe H (2014). *Dependence Modelling with Copulas*. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, London.

Kazianka H, Pilz J (2010). "Copula-Based Geostatistical Modeling of Continuous and Discrete Data Including Covariates." *Stochastic Environmental Research and Risk Assessment*, **24**(5), 661–673. doi:10.1007/s00477-009-0353-8.

Keane M (1994). "A Computationally Practical Simulation Estimator for Panel Data." *Econometrica*, **62**, 95–116. doi:10.2307/2951477.

Kolev N, Paiva D (2009). "Copula-Based Regression Models: A Survey." *Journal of Statistical Planning and Inference*, **139**(11), 3847–3856. doi:10.1016/j.jspi.2009.05.023.

Kraemer N, Brechmann EC, Silvestrini D, Czado C (2013). "Total Loss Estimation Using Copula-Based Regression Models." *Insurance: Mathematics and Economics*, **53**(3), 829–839. doi:10.1016/j.insmatheco.2013.09.003.

Li M, Boehnke M, Abecasis GR, Song PXK (2006). "Quantitative Trait Linkage Analysis Using Gaussian Copulas." *Genetics*, **173**(4), 2317–2327. doi:10.1534/genetics.105.054650.

Masarotto G, Varin C (2012). "Gaussian Copula Marginal Regression." *Electronic Journal of Statistics*, **6**, 1517–1549. doi:10.1214/12-ejs721.

McCullagh P, Nelder J (1989). *Generalized Linear Models*. 2nd edition. Chapman and Hall/CRC, Boca Raton.

Nikoloulopoulos AK (2013). "On the Estimation of Normal Copula Discrete Regression Models Using the Continuous Extension and Simulated Likelihood." *Journal of Statistical Planning and Inference*, **143**(11), 1923–1937. doi:10.1016/j.jspi.2013.06.015.

Nikoloulopoulos AK (2016). "Efficient Estimation of High-Dimensional Multivariate Normal Copula Models with Discrete Spatial Responses." *Stochastic Environmental Research and Risk Assessment*, **30**(2), 493–505. doi:10.1007/s00477-015-1060-2.

Nikoloulopoulos AK, Joe H (2015). ***weightedScores**: Weighted Scores Method for Regression Models with Dependent Data*. R package version 0.9.5.1, URL https://CRAN.R-project.org/package=weightedScores.

Parsa RA, Klugman SA (2011). "Copula Regression." *Variance*, **5**(1), 45–54. doi:10.1177/0267659111416877.

Pebesma EJ, Bivand RS (2005). "Classes and Methods for Spatial Data in R." *R News*, **5**(2), 9–13.

Pinheiro J, Bates D, DebRoy S, Sarkar D, R Core Team (2017). ***nlme**: Linear and Nonlinear Mixed Effects Models*. R Package Version 3.1-131, URL https://CRAN.R-project.org/package=nlme.

Pitt M, Chan D, Kohn R (2006). "Efficient Bayesian Inference for Gaussian Copula Regression Models." *Biometrika*, **93**, 537–554. doi:10.1093/biomet/93.3.537.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Ribeiro Jr PJ, Diggle PJ (2016). ***geoR**: Analysis of Geostatistical Data*. R package version 1.7-5.2, URL https://CRAN.R-project.org/package=geoR.

Roca VA, Cribari-Neto F (2009). "Beta Autoregressive Moving Average Models." *Test*, **18**, 529–545. doi:10.1007/s11749-008-0112-z.

Shi P, Frees EW (2011). "Dependent Loss Reserving Using Copulas." *ASTIN Bulletin: Journal of the International Actuarial Association*, **41**(2), 449–486. doi:10.2143/ast.39.1.2038061.

Song PXK (2000). "Multivariate Dispersion Models Generated from Gaussian Copula." *Scandinavian Journal of Statistics*, **27**, 305–320. doi:10.1111/1467-9469.00191.

Song PXK (2007). *Correlated Data Analysis: Modeling, Analytics and Applications*. Springer-Verlag, New York.

Song PXK, Li M, Yuan Y (2009). "Joint Regression Analysis of Correlated Data Using Gaussian Copulas." *Biometrics*, **65**, 60–68. doi:10.1111/j.1541-0420.2008.01058.x.

Song PXK, Li M, Zhang P (2013). "Vector Generalized Linear Models: A Gaussian Copula Approach." In P Jaworski, F Durante, W Härdle (eds.), *Copulae in Mathematical and Quantitative Finance*, pp. 251–276. Springer-Verlag, Berlin.

Sun J, Frees EW, Rosenberg MJ (2008). "Heavy-Tailed Longitudinal Data Modeling Using Copulas." *Insurance: Mathematics and Economics*, **42**(2), 817–830. doi:10.1016/j.insmatheco.2007.09.009.

Thomson M, Connor S, D'Alessandro U, Rowlingson B, Diggle PJ, Cresswell BGM (1999). "Predicting Malaria Infection in Gambian Children from Satellite Data and Bednet Use Surveys: The Importance of Spatial Correlation in the Interpretation of Results." *American Journal of Tropical Medicine and Hygiene*, **61**, 2–8. doi:10.1016/s0035-9203(00)90257-8.

Train KE (2003). *Discrete Choice Methods with Simulation*. Cambridge University Press, Cambridge.

Varin C, Reid N, Firth D (2011). "An Overview of Composite Likelihood Methods." *Statistica Sinica*, **21**, 5–42. doi:10.5705/ss.2013.084t.

Wu B, de Leon AR (2014). "Gaussian Copula Mixed Models for Clustered Mixed Outcomes, with Application in Developmental Toxicology." *Journal of Agricultural, Biological, and Environmental Statistics*, **19**(1), 39–56. doi:10.1007/s13253-013-0155-9.

Yan J (2002). "**geepack**: Yet Another Package for Generalized Estimating Equations." *R News*, **2/3**, 12–14.

Yan J (2007). "Enjoy the Joy of Copulas: With a Package **copula**." *Journal of Statistical Software*, **21**(2), 1–21. doi:10.18637/jss.v021.i04.

Zeileis A (2004). "Econometric Computing with HC and HAC Covariance Matrix Estimators." *Journal of Statistical Software*, **11**(10), 1–17. doi:10.18637/jss.v011.i10.

Zeileis A (2006). "Object-Oriented Computation of Sandwich Estimators." *Journal of Statistical Software*, **16**(9), 1–16. doi:10.18637/jss.v016.i09.

Zeileis A, Croissant Y (2010). "Extended Model Formulas in R: Multiple Parts and Multiple Responses." *Journal of Statistical Software*, **34**(1), 1–13. doi:10.18637/jss.v034.i01.

Zeileis A, Hothorn T (2002). "Diagnostic Checking in Regression Relationships." *R News*, **2**(3), 7–10.

Zhao Y, Joe H (2005). "Composite Likelihood Estimation in Multivariate Data Analysis." *The Canadian Journal of Statistics*, **33**, 335–356. doi:10.1002/cjs.5540330303.

Zucchini W, MacDonald IL (2009). *Hidden Markov Models for Time Series*. Chapman & Hall/CRC, Boca Raton. doi:10.1201/9781420010893.

# A. Specifying new models and correlations

This appendix is addressed to users interested in the possibility of specifying marginals and Gaussian copula correlation matrices not yet available in package **gcmr**.

## A.1. Specify a new marginal model

The simpler way to specify a new object of class `marginal.gcmr` is to use one of the available marginal distributions in **gcmr** as prototype, such as, for example, the Poisson marginal:

```
R> poisson.marg

function (link = "log")
{
    fm <- poisson(substitute(link))
    ans <- list()
    ans$start <- function(y, x, z, offset) {
        lambda <- coef(glm.fit(x, y, offset = offset$mean, family = fm))
        names(lambda) <- dimnames(as.matrix(x))[[2L]]
        lambda
    }
    ans$npar <- function(x, z) NCOL(x)
    ans$dp <- function(y, x, z, offset, lambda) {
        mu <- fm$linkinv(x %*% lambda + offset$mean)
        cbind(dpois(y, mu), ppois(y, mu))
    }
    ans$q <- function(p, x, z, offset, lambda) {
        mu <- fm$linkinv(x %*% lambda + offset$mean)
        qpois(p, mu)
    }
    ans$fitted.val <- function(x, z, offset, lambda) {
        fm$linkinv(x %*% lambda + offset$mean)
    }
    ans$type <- "integer"
    class(ans) <- c("marginal.gcmr")
    ans
}
<environment: namespace:gcmr>
```

Function `poisson.marg()` receives as input the link function as in `glm()` and produces as output a list with several components described below.

`start()` is a function of the vector of responses `y`, the design matrix `x` and the `offset`. Among the inputs, there is also the design matrix `z` for the dispersion, although this argument is superfluous for the Poisson model because it assumes a constant dispersion. The output of `start()` is the vector of starting values for the marginal parameters `lambda`. The starting values are typically computed as if the observations were independent. In

the specific case of Poisson marginals, the starting values are obtained with a call to `glm.fit(x, y, family = Poisson)`;

`npar()` is a function of the design matrix that returns the number of marginal parameters `lambda`. In the special case of the Poisson model, the number of parameters corresponds to the number of mean regression coefficients. If the model includes also a dispersion component, as, for example, in the case of the negative binomial distribution, then the number of dispersion parameters have to be added to the number of mean regression coefficients;

`dp()` is a function of the vector of responses `y`, the design matrix `x`, the `offset` and the vector of marginal parameters `lambda`. The output is a $n \times 2$ matrix whose two columns correspond to the marginal density (`d`) and the marginal cumulative distribution function (`p`) of the $n$ observations;

`q()` is a function of the vector of probability values `p`, the design matrix `x`, the `offset` and the vector of marginal parameters `lambda`. The output is the vector of the quantiles corresponding to `p`;

`fitted.val()` is a function of the design matrix `x`, the `offset` and the vector of marginal parameters `lambda` that computes the vector of fitted values;

`type` is a string that indicates whether the response is continuous (`"numeric"`) or discrete (`"integer"`), as in the Poisson case.

Marginal models that allow for variable dispersion are similarly specified with the complication to supply the above listed components also for the dispersion part. See, for example, the negative binomial model specified by function `negbin.marg()`.

## A.2. Specify a new correlation structure

The Matérn spatial correlation is considered below as a prototype of the Gaussian copula correlation:

```
R> matern.cormat

function (D, alpha = 0.5)
{
    ans <- list()
    ans$npar <- 1
    ans$start <- function() {
        tau <- median(D)
        names(tau) <- c("tau")
        attr(tau, "lower") <- sqrt(.Machine$double.eps)
        tau
    }
    ans$chol <- function(tau, not.na) {
        S <- geoR::matern(D, tau, alpha)
        q <- try(chol(S[not.na, not.na]), silent = TRUE)
```

```
        if (inherits(q, "try-error"))
            NULL
        else q
    }
    class(ans) <- "cormat.gcmr"
    ans
}
<environment: namespace:gcmr>
```

Function `matern.cormat(D, alpha)` receives as input the matrix `D` of the distances between the observations and the shape parameter `alpha` of the Matérn correlation model. The output is a list with three components:

`npar` returns the number of dependence parameters in the Gaussian copula correlation matrix. In the specific case of the Matérn correlation model, there is a single dependence parameter `tau` that describes the degree of spatial dependence;

`start()` is a function that returns the vector of starting values for the dependence parameters `tau`. In the specific case of the Matérn correlation model with given shape parameter `alpha`, the spatial dependence parameter `tau` is set, arbitrarily, equal to the median distance observed in the data;

`chol()` is a function of the vector of dependence parameters `tau` in the Gaussian copula correlation matrix and the vector of indices of the observed data `not.na`. The output is the Cholesky factor of the Gaussian copula correlation matrix. If the Cholesky factorization fails, then `chol` returns `NULL`.

**Affiliation:**

Guido Masarotto
Department of Statistical Sciences
University of Padua
Via Cesare Battisti, 241
35121 Padova, Italy
E-mail: guido.masarotto@unipd.it
URL: http://sirio.stat.unipd.it

Cristiano Varin
Department of Environmental Sciences, Informatics and Statistics
Ca' Foscari University of Venice
Via Torino, 150
30170 Venezia Mestre, Italy
E-mail: cristiano.varin@unive.it
URL: http://cristianovarin.weebly.com

*Journal of Statistical Software* http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
April 2017, Volume 77, Issue 8 *Submitted:* 2015-06-16
doi:10.18637/jss.v077.i08 *Accepted:* 2016-01-18